

Учреждение образования
«Белорусский государственный университет
информатики и радиоэлектроники»

Государственное учреждение
«Администрация Парка высоких технологий»

Открытые семантические технологии проектирования интеллектуальных систем

OSTIS-2015

Open Semantic Technologies for Intelligent Systems

МАТЕРИАЛЫ
V МЕЖДУНАРОДНОЙ
НАУЧНО-ТЕХНИЧЕСКОЙ КОНФЕРЕНЦИИ

(Минск, 19–21 февраля 2015 года)

УДК 004.822+004.89-027.31
ББК 32.813-02+73
О-83

Редакционная коллегия :

*В. В. Голенков (отв. ред.), Л. С. Глоба, Н. А. Гулякина, И. В. Ефименко, О. П. Кузнецов, Б. М. Лобанов,
Д. Ш. Сулейманов, А. А. Харламов, В. Ф. Хорошевский*

Организаторы конференции:

Учреждение образования «Белорусский
государственный университет информатики и
радиоэлектроники»
Российская ассоциация искусственного интеллекта
Государственное учреждение «Администрация Парка
высоких технологий» (Республика Беларусь)
Научно-технологическая ассоциация «Инфопарк»
Объединённый институт проблем информатики
Национальной академии наук Беларуси
Научно-исследовательский институт «Прикладная
семиотика» АН РТ
Институт информатизации образования Российской
академии образования

Международная ИТ-компания «Itransition»
Компания «Melesta»
Компания «Qulix Systems»
Digital-агентство «ARTOX media»
Компания «ВирусБлокАда»
Компания «LogicNow»
ЧУП «Андерсан-софт»
ООО «Прикладные системы»
ООО «АйтиРекс Групп»
ЧУП «Айтими́нт»
ЧП «Нейролаб»
ИООО «Седон БЛР»

Техническая и информационная поддержка:

Международный журнал «Программные продукты и
системы»
Научный журнал «Информатика»

Научно-практический журнал для специалистов
«Электроника ИНФО»
Научный журнал «Онтология проектирования»

*Издание осуществлено по заказу государственного учреждения
«Администрация Парка высоких технологий»*

О-83 **Открытые семантические технологии проектирования интеллектуальных систем = Open Semantic Technologies for Intelligent Systems (OSTIS-2015) :** материалы V междунар. науч.-техн. конф. (Минск, 19–21 февраля 2015 года)/ редкол. : В. В. Голенков (отв. ред.) [и др.]. – Минск : БГУИР, 2015. – 620 с.
ISBN 978-985-543-034-7.

Сборник включает прошедшие рецензирование статьи V международной научно-технической конференции «Открытые семантические технологии проектирования интеллектуальных систем».

Сборник предназначен для преподавателей высших учебных заведений, научных сотрудников, студентов, аспирантов, магистрантов, а также для специалистов предприятий в сфере проектирования интеллектуальных систем.

Материалы сборника одобрены Программным комитетом OSTIS-2015 и печатаются в виде, представленном авторами.

УДК 004.822+004.89-027.31
ББК 32.813-02+73

ISBN 978-985-543-034-7

© УО «Белорусский государственный
университет информатики
и радиоэлектроники», 2015

ОРГАНИЗАЦИОННЫЙ КОМИТЕТ КОНФЕРЕНЦИИ

Батура Михаил Павлович	председатель , ректор Белорусского государственного университета информатики и радиоэлектроники
Живицкая Елена Николаевна	проректор по учебной работе и менеджменту качества Белорусского государственного университета информатики и радиоэлектроники
Кузнецов Александр Петрович	проректор по научной работе Белорусского государственного университета информатики и радиоэлектроники
Никольшин Борис Викторович	проректор по учебной работе и информатизации Белорусского государственного университета информатики и радиоэлектроники
Шилин Леонид Юрьевич	декан факультета информационных технологий и управления Белорусского государственного университета информатики и радиоэлектроники
Голенков Владимир Васильевич	заведующий кафедрой интеллектуальных информационных технологий Белорусского государственного университета информатики и радиоэлектроники
Титович Анна Францевна	директор центра международного сотрудничества Белорусского государственного университета информатики и радиоэлектроники
Лихачевский Дмитрий Викторович	декан факультета компьютерного проектирования Белорусского государственного университета информатики и радиоэлектроники
Боярко Алла Викторовна	руководитель пресс-службы Белорусского государственного университета информатики и радиоэлектроники

РАБОЧАЯ ГРУППА ОРГКОМИТЕТА КОНФЕРЕНЦИИ

Гулякина Наталья Анатольевна	руководитель группы , заместитель заведующего кафедрой ИИТ Белорусского государственного университета информатики и радиоэлектроники
Гракова Наталья Викторовна	заместитель заведующего кафедрой ИИТ по воспитательным вопросам Белорусского государственного университета информатики и радиоэлектроники
Губаревич Анастасия Владимировна	лаборант кафедры ИИТ Белорусского государственного университета информатики и радиоэлектроники
Лучина Татьяна Леонидовна	инженер кафедры ИИТ Белорусского государственного университета информатики и радиоэлектроники
Русецкий Кирилл Валерьевич	инженер-программист кафедры ИИТ Белорусского государственного университета информатики и радиоэлектроники
Самодумкин Сергей Александрович	старший преподаватель кафедры ИИТ Белорусского государственного университета информатики и радиоэлектроники

ORGANIZATIONAL COMMITTEE

Mikhail Batura	Rector, Belarusian State University of Informatics and Radio Electronics
Elena Zhivitskaya	Vice-Rector for Academic Work and Quality Management, Belarusian State University of Informatics and Radio Electronics
Alexander Kuznetsov	Vice-Rector for Research and Development, Belarusian State University of Informatics and Radio Electronics
Boris Nikulshin	Vice Rector for Education and Information, Belarusian State University of Informatics and Radio Electronics
Leonid Shilin	Dean of the Faculty of Information Technologies and Control, Belarusian State University of Informatics and Radio Electronics
Vladimir Golenkov	Head of Informational Intelligent Technologies Department, Belarusian State University of Informatics and Radio Electronics
Anna Titovich	Director of the Center for International Cooperation, Belarusian State University of Informatics and Radio Electronics
Dmitry Likhachevsky	Dean of the Faculty of Computer-Aided Design, Belarusian State University of Informatics and Radio Electronics
Alla Boyarko	Head of Press Service, Belarusian State University of Informatics and Radio Electronics

CONFERENCE ORGANIZATIONAL COMMITTEE WORKGROUP

Natalia Guliakina	Group leader assistant head of IIT Department at Belarusian State University of Informatics and Radioelectronics
Natalia Grakova	assistant head of IIT department at Belarusian State University of Informatics and Radioelectronics
Anastasia Hubarevich	lab assistant at IIT department of Belarusian State University of Informatics and Radioelectronics
Tatiana Lutchina	engineer at IIT department of Belarusian State University of Informatics and Radioelectronics
Kirill Rusetski	software engineer at IIT department of Belarusian State University of Informatics and Radioelectronics
Sergei Samodumkin	senior teacher at IIT department of Belarusian State University of Informatics and Radioelectronics

ПРОГРАММНЫЙ КОМИТЕТ КОНФЕРЕНЦИИ

д.т.н., проф.	Кузнецов Олег Петрович	Председатель , заведующий лабораторией, Институт проблем управления РАН, г. Москва, Российская Федерация
к.т.н., доц.	Боргест Николай Михайлович	Профессор, Самарский государственный аэрокосмический университет имени Академика С.П.Королева, г. Самара, Российская Федерация
д.т.н., проф.	Борисов Аркадий Николаевич	Профессор, Рижский технический университет, Институт информационных технологий, г. Рига, Латвия
д.т.н., доц.	Валькман Юрий Роландович	Заведующий отделом распределенных интеллектуальных систем Международного научно-учебного центра информационных технологий и систем НАНУ и МОНУ, профессор кафедры математических методов системного анализа УНК «ИПСА» НТУУ «КПИ», г. Киев, Украина
д.ф.-м.н. академик РАН	Васильев Станислав Николаевич	Директор Института проблем управления им. В. А. Трапезникова РАН, председатель Ученого Совета, г. Москва, Российская Федерация
д.т.н., проф.	Гаврилова Татьяна Альбертовна	Заведующая кафедрой информационных технологий в менеджменте, Высшая Школа менеджмента СПбГУ, г. Санкт-Петербург, Российская Федерация
д.т.н., проф.	Глоба Лариса Сергеевна	Заведующая кафедрой, Национальный технический университет Украины «Киевский политехнический институт», г. Киев, Украина
д.т.н., проф.	Голенков Владимир Васильевич	Заведующий кафедрой интеллектуальных информационных технологий, Белорусский государственный университет информатики и радиоэлектроники, г. Минск, Беларусь
д.т.н., проф.	Головко Владимир Адамович	Заведующий кафедрой интеллектуальных информационных технологий, Брестский государственный технический университет, г. Брест, Беларусь
д.фил.н., проф.	Гордей Александр Николаевич	Директор, Республиканский институт китаеведения имени Конфуция БГУ, г. Минск, Беларусь
д.т.н., с.н.с.	Грибова Валерия Викторовна	Заведующая лабораторией интеллектуальных систем, Федеральное государственное бюджетное учреждение науки Институт автоматизации и процессов управления Дальневосточного отделения Российской академии наук, г. Владивосток, Российская Федерация
к.ф.-м.н., доц.	Гулякина Наталья Анатольевна	Заместитель заведующего кафедрой интеллектуальных информационных технологий, Белорусский государственный университет информатики и радиоэлектроники г. Минск, Беларусь
д.т.н., проф.	Еремеев Александр Павлович	Заведующий кафедрой прикладной математики, Национальный исследовательский университет «МЭИ», г. Москва, Российская Федерация
к.фил.н.	Ефименко Ирина Владимировна	Заместитель декана факультета филологии, НИУ ВШЭ; Ведущий научный сотрудник, ЦИАС ИСИЭЗ НИУ ВШЭ г. Москва, Российская Федерация
д.т.н., проф.	Заболеева-Зотова Алла Викторовна	Начальник Управления региональных и межгосударственных программ РФФИ, г. Москва, Российская Федерация
к.т.н., с.н.с.	Загорюлько Юрий Алексеевич	Заведующий лабораторией, Институт систем информатики им. А.П. Ершова СО РАН, г. Новосибирск, Российская Федерация

д.ф.-м.н., проф.	Клещев Александр Сергеевич	Главный научный сотрудник, Федеральное государственное бюджетное учреждение науки Институт автоматизации и процессов управления Дальневосточного отделения Российской академии наук, г. Владивосток, Российская Федерация
д.мед.н., проф.	Кобринский Борис Аркадьевич	Руководитель, Медицинский центр новых информационных технологий, г. Москва, Российская Федерация
д.п.н., проф.	Козлов Олег Александрович	Заместитель директора института, Учреждение Российской Академии Образования «Институт Информатизации Образования», г. Москва, Российская Федерация
д.т.н., проф.	Комарцова Людмила Георгиевна	Профессор кафедры «Компьютерные системы и сети», Московский Государственный технический университет им. Н.Э. Баумана (Калужский филиал), г. Калуга, Российская Федерация
д.т.н., проф.	Курейчик Виктор Михайлович	Заместитель руководителя по научной и инновационной деятельности, Таганрогский кампус ЮФУ, г. Таганрог, Российская Федерация
д.т.н., с.н.с	Ландэ Дмитрий Владимирович	Заведующий отделом, Институт проблем регистрации информации НАН Украины, г. Киев, Украина
д.т.н., с.н.с.	Лобанов Борис Мефодьевич	Главный научный сотрудник, Объединённый институт проблем информатики НАН Беларуси, г. Минск, Беларусь
к.ф.-м.н.	Лукашевич Наталья Валентиновна	Ведущий научный сотрудник, Московский государственный университет имени М.В.Ломоносова, г. Москва, Российская Федерация
д.т.н., проф.	Массель Людмила Васильевна	Главный научный сотрудник, заведующая лабораторией, Институт систем энергетики им. Л.А. Мелентьева СО РАН, г. Иркутск, Российская Федерация
к.т.н.	Найденова Ксения Александровна	Старший научный сотрудник, Военно-медицинская академия им. С. М. Кирова, г. Санкт-Петербург, Российская Федерация
к.т.н., доц.	Невзорова Ольга Авенировна	Заместитель директора по научной работе, Научно-исследовательский институт «Прикладная семиотика» АН РТ, г. Казань, Российская Федерация
д.ф.-м.н., проф.	Осипов Геннадий Семенович	Заместитель заведующего кафедрой «Математические методы системного анализа», Национальный исследовательский университет «Высшая школа экономики», г. Москва, Российская Федерация
д.т.н., проф.	Палюх Борис Васильевич	Заведующий кафедрой Информационных систем, Тверской государственный технический университет, г. Тверь, Российская Федерация
д.т.н., проф.	Петровский Александр Александрович	Заведующий кафедрой ЭВС, Белорусский государственный университет информатики и радиоэлектроники, г. Минск, Беларусь
д.т.н., проф.	Петровский Алексей Борисович	Заведующий лабораторией, ФГБУН Институт системного анализа Российской академии наук, г. Москва, Российская Федерация
к.ф.-м.н., проф.	Плесневич Геральд Станиславович	Профессор кафедры «Информационные технологии», Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования «МАТИ – Российский государственный технологический университет имени К.Э. Циолковского», г. Москва, Российская Федерация

академик РАО, д.пед.н., проф.	Роберт Ирэна Веньяминовна	Директор ФГБНУ «Институт информатизации образования Российской академии образования», г. Москва, Российская Федерация
к.т.н., доц.	Родченко Вадим Григорьевич	Доцент кафедры программного обеспечения интеллектуальных и компьютерных систем, Учреждение образования «Гродненский государственный университет имени Янки Купалы», г. Гродно, Беларусь
д.т.н., проф.	Сидоркина Ирина Геннадьевна	Декан факультета информатики и вычислительной техники, Поволжский государственный технологический университет, г. Йошкар-Ола, Республика Марий Эл, Российская Федерация
д.т.н.	Смирнов Сергей Викторович	Директор, ведущий научный сотрудник, заведующий лабораторией анализа и моделирования сложных систем, Институт проблем управления сложными системами Российской академии наук, г. Самара, Российская Федерация
д.ф.-м.н., проф.	Соловьёв Сергей Юрьевич	Профессор, Московский государственный университет имени М.В.Ломоносова, г. Москва, Российская Федерация
д.т.н., проф.	Соснин Петр Иванович	Заведующий кафедрой «Вычислительная техника», Ульяновский государственный технический университет, г. Ульяновск, Российская Федерация
д.т.н., проф.	Стефанюк Вадим Львович	Главный научный сотрудник, Институт проблем передачи информации РАН, г. Москва, Российская Федерация
д.т.н., Академик АН РТ, проф.	Сулейманов Джавдет Шевкетович	Заведующий кафедрой, Казанский (Приволжский) федеральный университет, г. Казань, Российская Федерация
к.т.н., доц.	Тарасов Валерий Борисович	Доцент кафедры компьютерных систем автоматизации производства, Московский государственный технический университет им. Н.Э.Баумана, г. Москва, Российская Федерация
д.э.н., проф.	Тельнов Юрий Филиппович	Заведующий кафедрой, Московский государственный университет экономики, статистики и информатики, г. Москва, Российская Федерация
д.ф.-м.н., проф.	Тузиков Александр Васильевич	Генеральный директор, Объединенный институт проблем информатики НАН Беларуси, г. Минск, Беларусь
д.т.н.	Харламов Александр Александрович	Старший научный сотрудник, Институт высшей нервной деятельности и нейрофизиологии РАН, г. Москва, Российская Федерация
д.т.н., проф.	Хорошевский Владимир Федорович	Заведующий сектором, ВЦ им. А.А. Дородницына РАН Главный научный сотрудник, ЦИАС ИСИЭЗ НИУ ВШЭ, г. Москва, Российская Федерация
Академик НАН Беларуси, д.т.н., проф.	Чернявский Александр Фёдорович	Заведующий кафедрой интеллектуальных систем, Белорусский государственный университет, г. Минск, Беларусь
д.т.н., проф.	Шарипбай Алтынбек Амирович	Директор научного центра «Искусственный интеллект», Евразийский национальный университет имени Л.Н.Гумилева, г. Астана, Казахстан
к.т.н., доц.	Щербак Сергей Сергеевич	Доцент, Кременчугский национальный университет имени Михаила Остроградского, г. Кременчуг, Украина

PROGRAM COMMITTEE

Borgest N.	c. of t.s., RF ass. proff.	Palyukh B.	d. of t.s., RF proff.
Borisov A.	d. of t.s., Latvia proff.	Petrovsky A.	d. of t.s., RB proff.
Chernyavsky A.	NAS Belarus academician	Petrovsky A.	c. of t.s., d. of t.s., RF proff.
Efimenko I.	c. of phyl.s., RF	Plesniewicz G.	c. of ph.-m.s., RF
Eremeev A.	d. of t.s., RF proff.	Robert I.	Academician of REA, d. of teach.s., RF proff.
Gavrilova T.	d. of t.s., RF proff.	Rodchenko V.	c. of t.s., RB ass. proff.
Globa L.	d. of t.s., Ukraine proff.	Sharipbay A.	d. of t.s., Kazakhstan proff.
Golenkov V.	d. of t.s., RB proff.	Shcherbak S.	c. of t.s., Ukraine ass. proff.
Golovko V.	d. of t.s., RB proff.	Sidorkina I.	d. of t.s., RF proff.
Gribova V.	d. of t.s., RF	Smirnov S.	d. of t.s., RF proff.
Guliakina N.	c. of t.s., RB ass. proff.	Soloviev S.	d. of ph.-m.s., RF proff.
Hardzei A.	d. of phyl.s., RB proff.	Sosnin P.	d. of t.s., RF proff.
Kharlamov A.	d. of t.s., RF	Stefanuk V.	d. of t.s., RF proff.
Khoroshevsky V.	d. of t.s., RF proff.	Suleymanov D.	AS Tatarstan academician, KF
Kleshev A.	d. of t.s., RF proff.	Tarasov V.	c. of t.s., RF ass. proff.
Kobrinskiy B.	d. of med.s., RF	Telnov Yu.	d. of e.s., RF proff.
Komartsova L.	d. of t.s., RF	Tuzikov A.	d. of ph.-m.s., RB proff.
Kozlov O.	d. of teach.s., RF proff.	Vasilyev S.	academician of RAS, d. of ph.-m.s., RB proff.
Kureychik V.	d. of t.s., RF proff.	Walkman Yu.R.	d. of t.s., Ukraine proff.
Kuznetsov O.	d. of t.s., RF proff.	Zaboleeva-Zotova A.	d. of t.s., RF
Lande D.	d. of t.s., Ukraine	Zagorulko Yu.	c. of t.s., RF ass. proff.
Lobanov B.	d. of t.s., RB proff.		
Loukachevitch N.	c. of t.s., RF		
Massel L.	d. of t.s., RF proff.		
Naidenova K.	c. of t.s., RF		
Nevzorova O.	c. of t.s., RF ass. proff.		
Osipov G.	d. of ph.-m.s., RF proff.		

СОДЕРЖАНИЕ

ПРЕДИСЛОВИЕ	23
ПРОБЛЕМА ПОНИМАНИЯ: НАСТОЯЩЕЕ И БУДУЩЕЕ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА Тарасов В.Б.	25
СЕМАНТИЧЕСКАЯ ТЕХНОЛОГИЯ КАРТИРОВАНИЯ СЕМАНТИЧЕСКИХ ТЕХНОЛОГИЙ (Наукометрический анализ конференций OSTIS) Хорошевский В.Ф., Ефименко И.В.	43
СЕМАНТИЧЕСКАЯ ТЕХНОЛОГИЯ КОМПОНЕНТНОГО ПРОЕКТИРОВАНИЯ СИСТЕМ, УПРАВЛЯЕМЫХ ЗНАНИЯМИ Голенков В.В., Гулякина Н.А.	57
СРЕДСТВА ПОДДЕРЖКИ КОМПОНЕНТНОГО ПРОЕКТИРОВАНИЯ СИСТЕМ, УПРАВЛЯЕМЫХ ЗНАНИЯМИ Шункевич Д.В., Давыденко И.Т., Корончик Д.Н., Жуков И.И., Паркалов А.В.	79
РЕАЛИЗАЦИЯ ПЛАТФОРМЫ ДЛЯ WEB-ОРИЕНТИРОВАННЫХ СИСТЕМ, УПРАВЛЯЕМЫХ ЗНАНИЯМИ Корончик Д. Н.	89
МЕТОДИКА КОМПОНЕНТНОГО ПРОЕКТИРОВАНИЯ СИСТЕМ, УПРАВЛЯЕМЫХ ЗНАНИЯМИ Шункевич Д.В., Давыденко И.Т., Корончик Д.Н., Губаревич А.В., Борискин А.С.	93
МОДЕЛИ И АЛГОРИТМЫ ИНТЕГРАЦИИ ЗНАНИЙ НА ОСНОВЕ ОДНОРОДНЫХ СЕМАНТИЧЕСКИХ СЕТЕЙ Ивашенко В.П.	111
ПРЕДСТАВЛЕНИЕ СЕМАНТИЧЕСКИХ СЕТЕЙ И АЛГОРИТМЫ ИХ ОРГАНИЗАЦИИ И СЕМАНТИЧЕСКОЙ ОБРАБОТКИ НА ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМАХ С МАССОВЫМ ПАРАЛЛЕЛИЗМОМ Ивашенко В.П., Вереник Н.Л., Гирель А.И., Сейткулов Е.Н., Татур М.М.	133
КОМПОНЕНТНАЯ АРХИТЕКТУРА ИНТЕЛЛЕКТУАЛЬНОЙ СИСТЕМЫ КОНСУЛЬТАЦИОННОГО ОБСЛУЖИВАНИЯ И ОБУЧЕНИЯ РАЗРАБОТЧИКОВ ПРОГРАММ Пивоварчик О.В.	141
ПРИНЦИПЫ ПОСТРОЕНИЯ КЛИНИЧЕСКОЙ СИСТЕМЫ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ НА ОСНОВЕ ТЕХНОЛОГИИ OSTIS Каешко А.И., Маргунов Е.А.	149
ПРОЕКТИРОВАНИЕ БАЗЫ ЗНАНИЙ ИНТЕЛЛЕКТУАЛЬНОЙ СПРАВОЧНОЙ СИСТЕМЫ ПО АЛГЕБРЕ Шарипбай А.А., Омарбекова А.С., Нургазинова Г.Ш.	157
ПУТИ РАЗВИТИЯ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА В КАЗАХСТАНЕ Шарипбай А.А.	161
СОЗДАНИЕ ОНТОЛОГИЧЕСКОЙ МОДЕЛИ ЛОКАЛЬНОЙ КОМПЬЮТЕРНОЙ СЕТИ Шарипбай А.А., Аскарова С.А., Муканова А.С.	165

ОНТОЛОГИЧЕСКАЯ МОДЕЛЬ ПРОЦЕССА ОБЕСПЕЧЕНИЯ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ	169
Ниязова Р.С., Буданова Н.	
КОНЦЕПЦИЯ ИНСТРУМЕНТАЛЬНОЙ ПЛАТФОРМЫ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ В СРЕДЕ ОБЛАЧНЫХ ВЫЧИСЛЕНИЙ С ИСПОЛЬЗОВАНИЕМ ИНТЕЛЛЕКТУАЛЬНЫХ ТЕХНОЛОГИЙ	173
Вишняков В.А., Гондаз Саз М.М., Моздуоани Шираз М.Г.	
ОНТОЛОГИЧЕСКИЙ ПОДХОД К СОЗДАНИЮ НАУЧНЫХ ИНТЕРНЕТ-РЕСУРСОВ	177
Загорулько Ю.А., Загорулько Г.Б.	
РАЗРАБОТКА И ГЕНЕРАЦИЯ ВИРТУАЛЬНЫХ СРЕД НА ОСНОВЕ СЕМАНТИЧЕСКИХ ПРЕДСТАВЛЕНИЙ	183
Грибова В.В., Федорищев Л.А.	
КЛАССИФИКАЦИЯ ДЛЯ ЗАДАЧ, ВЫЯВЛЯЕМЫХ ПРИ СИСТЕМНОМ АНАЛИЗЕ ИНТЕЛЛЕКТУАЛЬНОЙ ДЕЯТЕЛЬНОСТИ	187
Шалфеева Е.А.	
ТЕХНОЛОГИЯ РАЗРАБОТКИ РЕШАТЕЛЕЙ ЗАДАЧ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ С ИСПОЛЬЗОВАНИЕМ ИНСТРУМЕНТАЛЬНЫХ СЕРВИСОВ ОБЛАЧНОЙ ПЛАТФОРМЫ IASRAAS	193
Крылов Д.А., Москаленко Ф.М., Тимченко В.А.	
МЕТОДЫ И СРЕДСТВА СИТУАЦИОННОГО УПРАВЛЕНИЯ В ЭНЕРГЕТИКЕ НА ОСНОВЕ СЕМАНТИЧЕСКОГО МОДЕЛИРОВАНИЯ	199
Массель Л.В., Массель А.Г.	
ОТ ИНТЕЛЛЕКТУАЛЬНЫХ ПОРТАЛОВ К ПОРТАЛАМ ЗНАНИЙ: О РОЛИ ИНТЕЛЛЕКТУАЛЬНЫХ СЕРВИСОВ	205
Страхович Э.В., Власов С.А., Гаврилова Т.А.	
ОБЗОР СИСТЕМ КОЛЛАБОРАТИВНОЙ ФИЛЬТРАЦИИ	209
Шереметова Е.И.	
МЕТОДЫ И ИНСТРУМЕНТЫ РАБОТЫ С БИЗНЕС-ПРАВИЛАМИ	215
Савин А.М.	
СИСТЕМЫ УПРАВЛЕНИЯ ЗНАНИЯМИ В АТОМНОЙ ПРОМЫШЛЕННОСТИ	221
Костебелова В.К.	
СЦЕНАРНЫЙ ПОДХОД ПРИ ИССЛЕДОВАНИИ ДИНАМИКИ ИНФОРМАЦИОННЫХ ПОТОКОВ В СЕТИ ИНТЕРНЕТ	225
Додонов А.Г., Ландэ Д.В., Бойченко А.В.	
ПОДХОД К ФОРМИРОВАНИЮ СЛОЖНОГО ИНЖЕНЕРНОГО РАСЧЕТА НА ПОРТАЛАХ	231
Новогрудская Р.Л., Глоба Л.С.	
МЕТАГРАФЫ КАК ОСНОВА ДЛЯ ПРЕДСТАВЛЕНИЯ И ИСПОЛЬЗОВАНИЯ БАЗ НЕЧЕТКИХ ЗНАНИЙ	237
Глоба Л. С., Терновой М. Ю., Штогрин Е. С.	

ДЕТЕКТИРОВАНИЕ ИЗМЕНЕНИЙ В XML-ПОДОБНЫХ СТРУКТУРАХ ДОКУМЕНТОВ	241
Глоба Л.С., Молчанов Ю.Н.	
ИНТЕГРАЦИЯ КОРПОРАТИВНЫХ ДАННЫХ НА ОСНОВЕ WIKI-СИСТЕМ В УСЛОВИЯХ СЛАБОЙ СВЯЗАННОСТИ ИСТОЧНИКОВ	249
Галушка И.Н., Оксанич И.Г., Щербак С.С.	
СЕМАНТИЧЕСКИЙ ПОДХОД К ИНТЕГРАЦИИ ОБЪЕКТОВ WEB OF THINGS	255
Рогушина Ю.В., Гладун А.Я.	
РАЗРАБОТКА СРЕДСТВ ПЕРСОНИФИКАЦИИ ИНТЕЛЛЕКТУАЛЬНЫХ WEB- ПРИЛОЖЕНИЙ	265
Рогушина Ю.В.	
ОБЗОР И АНАЛИЗ МЕТОДОЛОГИЙ И МЕТОДОВ ПОСТРОЕНИЯ ОНТОЛОГИЙ	271
Хала Е.А.	
СПЕЦИФИКАЦИЯ ТЕМПОРАЛЬНЫХ ОТНОШЕНИЙ В СИСТЕМЕ «БИНАРНАЯ МОДЕЛЬ ЗНАНИЙ»	275
Плесневич Г.С., Нгуен Тхи Минь Ву	
ОТ ТЕРМИНОЛОГИЧЕСКИХ СЕТЕЙ К ТОЛКОВЫМ СЛОВАРЯМ	281
Мальковский М.Г., Соловьев С.Ю.	
ОБ ОДНОМ ПОДХОДЕ ТЕМАТИЧЕСКОГО МОДЕЛИРОВАНИЯ ТЕКСТОВ НА ЕСТЕСТВЕННОМ ЯЗЫКЕ	285
Чан Ван Ан	
ОБОСНОВАНИЕ МОДЕЛИРОВАНИЯ ПРОЦЕССОВ ПРАВДОПОДОБНЫХ РАССУЖДЕНИЙ	291
Найденова К. А.	
ЭВРИСТИЧЕСКИЙ МЕТОД УДОВЛЕТВОРЕНИЯ ОГРАНИЧЕНИЙ НА ОСНОВЕ ИХ МАТРИЧНОГО ПРЕДСТАВЛЕНИЯ	297
Зуенко А.А., Очинская А.А.	
СЕТЕВАЯ МОДЕЛЬ РАСПРОСТРАНЕНИЯ АКТИВНОСТИ В СРЕДЕ ГЕТЕРОГЕННЫХ АВТОМАТОВ	303
Жилякова Л.Ю.	
СИСТЕМА ПРИОРИТЕТНОГО ДВИЖЕНИЯ НА ПЕРЕКРЕСТКАХ «ЗЕЛЕНАЯ ВОЛНА» ДЛЯ ТРАНСПОРТНЫХ СРЕДСТВ ОПЕРАТИВНОГО НАЗНАЧЕНИЯ	309
Согоян А.Л., Шуть В.Н.	
ПРОГНОЗ РАЗВИТИЯ ИНФОРМАТИЗАЦИИ ОТЕЧЕСТВЕННОГО ОБРАЗОВАНИЯ	315
Роберт И.В.	
ИНТЕЛЛЕКТУАЛЬНЫЕ ТЕХНОЛОГИИ ОБУЧЕНИЯ В ИНФОРМАЦИОННО- ОБРАЗОВАТЕЛЬНОМ ПРОСТРАНСТВЕ	325
Тельнов Ю.Ф., Трембач В.М.	
ОПРЕДЕЛЕНИЕ УРОВНЯ УСВОЕННЫХ ЗНАНИЙ ПО ОБУЧАЮЩЕМУ КУРСУ, ПРЕДСТАВЛЕННОМУ СЕМАНТИЧЕСКОЙ СЕТЬЮ	331
Янковская А.Е., Шурыгин Ю.А., Ямшанов А.В., Кривдюк Н.М.	

WOLFRAM МАТЕМАТИКА СРЕДСТВА И ТЕХНОЛОГИИ РАЗРАБОТКИ ИНТЕЛЛЕКТУАЛЬНЫХ ОБУЧАЮЩИХ СИСТЕМ Таранчук В.Б.	339
РАЗРАБОТКА КОМПЛЕКСНОЙ СЕМАНТИЧЕСКОЙ МЕТОДИКИ К КОМПОНЕНТНОМУ ПРОЕКТИРОВАНИЮ СЛОЖНЫХ ПРОЕКТНО- ПРОМЫШЛЕННЫХ СРЕД ОБУЧЕНИЯ (ППСО) Афанасьев А.Н., Войт Н.Н.	347
ОНТОЛОГИЧЕСКИЙ ПОДХОД К АНАЛИЗУ КОМПЕТЕНЦИЙ ВЫПУСКНИКОВ ВУЗОВ Шеркунов В.В.	351
НЕЙРОСЕТЕВАЯ МОДЕЛЬ ПРОЦЕССА ПРОФЕССИОНАЛЬНОГО ОБУЧЕНИЯ МОЛОДЫХ СПЕЦИАЛИСТОВ Федяев О.И.	357
МЕТОДЫ НЕЧЕТКОГО РЕГУЛИРОВАНИЯ В ВОПРОСАХ УПРАВЛЕНИЯ ОБРАЗОВАТЕЛЬНЫМ ПРОЦЕССОМ Нишанов А.Х., Дощанова М.Ю., Мирзаев Д.А.	365
ИНТЕЛЛЕКТУАЛЬНЫЕ СРЕДСТВА АВТОМАТИЗАЦИИ УПРАВЛЕНИЕМ ОБУЧЕНИЯ Бабамухамедова М.З., Дощанова М.Ю., Джангазова К. А.	369
КОНЦЕПТУАЛЬНОЕ ЭКСПЕРИМЕНТИРОВАНИЕ В ПРОЕКТИРОВАНИИ КОНФИГУРИРУЕМЫХ ШАБЛОНОВ АВИАЦИОННЫХ ДЕТАЛЕЙ Соснин П.И., Чоракаев О.Э.	373
ОНТОЛОГИЯ ПРОЕКТИРОВАНИЯ ШАБЛОНОВ АВИАЦИОННЫХ ДЕТАЛЕЙ Гришин М.В., Ларин С.Н., Соснин П.И.	381
СЕМАНТИЧЕСКИЙ АНАЛИЗ ДИАГРАММАТИЧЕСКИХ МОДЕЛЕЙ В ПРОЕКТИРОВАНИИ СЛОЖНЫХ АВТОМАТИЗИРОВАННЫХ СИСТЕМ Афанасьев А.Н., Гайнуллин Р.Ф., Афанасьева Т.В.	385
ПРИМЕНЕНИЕ НАВИГАЦИОННОЙ СТРУКТУРЫ ЭЛЕКТРОННОГО АРХИВА ПРОЕКТНОЙ ОРГАНИЗАЦИИ В ЗАДАЧАХ ИНТЕЛЛЕКТУАЛЬНОГО АНАЛИЗА ТЕХНИЧЕСКИХ ДОКУМЕНТОВ Субхангулов Р.А., Филиппов А.А.	389
ОЦЕНКА ТЕРМИНОЛОГИЧНОСТИ ЛЕКСИЧЕСКИХ ЕДИНИЦ НА ОСНОВЕ ОНТОЛОГИИ ПРЕДМЕТНОЙ ОБЛАСТИ Андреев И.А., Башаев В.А., Клейн В.В., Мошкин В.С., Ярушкина Н.Г.	395
МЕТОДИКИ ПОСТРОЕНИЯ НЕЧЕТКИХ ОНТОЛОГИЙ СЛОЖНЫХ ПРЕДМЕТНЫХ ОБЛАСТЕЙ Мошкин В.С., Ярушкина Н.Г.	401
ОНТОЛОГИЧЕСКИЙ ПОДХОД К ФОРМИРОВАНИЮ ПРОЕКТНЫХ ЗАПРОСОВ ИНТЕЛЛЕКТУАЛЬНОГО АГЕНТА Наместников А.М., Субхангулов Р.А.	407

ОПЫТ СОЗДАНИЯ СРЕДСТВ СЕМАНТИЧЕСКОГО МОДЕЛИРОВАНИЯ И ПРОЕКТИРОВАНИЯ НА МАССОВОЙ ПРОГРАММНОЙ ПЛАТФОРМЕ Смирнов С.В.	413
РЕАЛИЗАЦИЯ УДАЛЕННОГО УПРАВЛЕНИЯ ПАРАМЕТРИЗОВАННОЙ ТРЕХМЕРНОЙ МОДЕЛЬЮ САМОЛЕТА С ПОМОЩЬЮ КЛИЕНТ-СЕРВЕРНОГО ПРИЛОЖЕНИЯ Боргест Н.М., Власов С.А., Коровин М.Д.	417
ПОДХОД К ПАРАМЕТРИЗАЦИИ ТРЕХМЕРНЫХ МОДЕЛЕЙ ДЛЯ РЕШЕНИЯ ЗАДАЧ АВТОМАТИЗАЦИИ ПРЕДВАРИТЕЛЬНОГО ПРОЕКТИРОВАНИЯ САМОЛЕТА Боргест Н.М., Коровин М.Д., Спирина М.О.	421
МЕТОДЫ ПРЕДСТАВЛЕНИЯ ЗНАНИЙ ДЛЯ РЕШЕНИЯ ЗАДАЧ МОДЕЛИРОВАНИЯ Коршиков Д.Н., Лахин О.И., Носкова А.И., Юрыгина Ю.С.	425
ОНТОЛОГИЧЕСКОЕ МОДЕЛИРОВАНИЕ ТЕХНИЧЕСКОГО ОБСЛУЖИВАНИЯ Федотова А.В., Давыденко И.Т.	429
СЕМАНТИЧЕСКИЕ АСПЕКТЫ ПРЕДСТАВЛЕНИЯ И ОБРАБОТКИ ПОИСКОВЫХ ЗАПРОСОВ В СИСТЕМЕ КОРПУС-МЕНЕДЖЕР Невзорова О.А., Мухамедшин Д.Р., Билалов Р.Р.	439
ТЕХНОЛОГИЯ АВТОМАТИЗИРОВАННОГО СИНТЕЗА ИНФОРМАЦИОННЫХ СИСТЕМ С ПОМОЩЬЮ СЕМАНТИЧЕСКИХ МОДЕЛЕЙ ПРЕДМЕТНОЙ ОБЛАСТИ Бикмуллина И.И.	445
ВЕБ-ИНТЕРФЕЙС ДЛЯ СНЯТИЯ МОРФОЛОГИЧЕСКОЙ МНОГОЗНАЧНОСТИ В КОРПУСЕ ТАТАРСКОГО ЯЗЫКА Гильмуллин Р.А. Гатауллин Р.Р.,	451
МНОГОКРИТЕРИАЛЬНЫЙ ПОДХОД К ФОРМИРОВАНИЮ ПОЛИТИКИ НАУЧНОГО ФОНДА И ЭКСПЕРТНОГО СООБЩЕСТВА Бойченко В.С., Заболеева-Зотова А.В., Петровский А.Б.	455
ОНТОЛОГИЧЕСКИЙ ПОДХОД ПРИ ПРОЕКТИРОВАНИИ ТЕХНОЛОГИЧЕСКИХ ПРОЦЕССОВ Бурдо Г.Б., Воробьева Е.В.	461
ИДЕНТИФИКАЦИЯ МОДЕЛИ ПРЕДСТАВЛЕНИЙ АГЕНТА О ПРЕДМЕТНОЙ ОБЛАСТИ В СИТУАЦИИ ВЫБОРА Виноградов Г.П., Борзов Д.А.	465
ИССЛЕДОВАНИЕ ЭФФЕКТИВНОСТИ ГЕНЕТИЧЕСКОГО АЛГОРИТМА ДЛЯ ТЕМАТИЧЕСКОГО ДОКУМЕНТАЛЬНОГО ПОИСКА Иванов В.К., Палюх Б.В.	471
МОДЕЛИРОВАНИЕ РАБОТЫ АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ УПРАВЛЕНИЯ КАЧЕСТВОМ СЕТЯМИ ПЕТРИ Бурдо Г. Б., Виноградов Г. П., Сорокин А. Ю.	477
ПРИМЕНЕНИЕ НЕЙРОННЫХ СЕТЕЙ ГЛУБОКОГО ДОВЕРИЯ ДЛЯ ВЫДЕЛЕНИЯ СЕМАНТИЧЕСКИ ЗНАЧИМЫХ ПРИЗНАКОВ Головки В.А., Крощенко А.А.	481

РАЗРАБОТКА ОТКРЫТЫХ ТЕХНОЛОГИЙ ПРОЕКТИРОВАНИЯ ИНТЕЛЛЕКТУАЛЬНЫХ РОБОТОТЕХНИЧЕСКИХ СИСТЕМ	487
Прокопович Г.А., Сычёв В.А., Герасюто С.Л.	
ПОИСК В СТАТИЧЕСКИХ РОЯХ	489
Воробьев В.В.	
ПОСТРОЕНИЕ ИНТЕЛЛЕКТУАЛЬНЫХ МУЛЬТИАГЕНТНЫХ КОБОРГ-СИСТЕМ	495
Соловьев В.И.	
АДКРЫТЫЯ КАМПАНАНТЫ WWW.CORPUS.BY ДЛЯ НАТУРАЛЬНА- МАЎЛЕНЧАГА ІНТЭРФЕЙСУ	499
Гецэвіч Ю.С., Лабанаў Б.М., Лысы С.І., Гюнтар А.В., Дзенісюк Д.А., Захар’еў В.А.	
РАСПРАЦОЎКА КАМПАНАНТА РАСПАЗНАВАННЯ МАЎЛЕННЯ ДЛЯ НАТУРАЛЬНА МАЎЛЕНЧАГА ІНТЭРФЕЙСУ	507
Нікалаенка К.А., Кайгародава Л.І., Гецэвіч Ю.С.	
РАЗДЕЛЬНОЕ МОДЕЛИРОВАНИЕ РЕЧЕВОГО СООБЩЕНИЯ В ВИДЕ ГОЛОСОВЫХ, ФОНЕТИЧЕСКИХ И ПРОСОДИЧЕСКИХ ПАРАМЕТРОВ	513
Азаров И.С., Петровский А.А.	
ПРИМЕНЕНИЕ МЕТОДОВ ИНТЕЛЛЕКТУАЛЬНОГО АНАЛИЗА ВРЕМЕННЫХ РЯДОВ ДЛЯ ЗАДАЧИ КЛАСТЕРИЗАЦИИ ПОЛЬЗОВАТЕЛЕЙ ПО ГОЛОСУ	519
Вагин В.Н., Ганишев В.А.	
ГОЛОСОВАЯ ИДЕНТИФИКАЦИЯ ПОЛЬЗОВАТЕЛЯ В СИСТЕМАХ КОНТРОЛЯ ДОСТУПА	525
Меньшаков П.А., Мурашко И.А.	
ИНФОРМАЦИОННАЯ МОДЕЛЬ ЛИЧНОСТИ	529
Харламов А.А.	
ОБ ОДНОМ МЕТОДЕ ПРИМЕНЕНИЯ ОБОБЩЕННОЙ ОНТОЛОГИИ ДЛЯ АНАЛИЗА ЕСТЕСТВЕННО-ЯЗЫКОВЫХ СООБЩЕНИЙ ПОЛЬЗОВАТЕЛЕЙ СЕТИ ИНТЕРНЕТ	533
Деева Н.В., Вишневский С.Я.	
ЧАСТОТНЫЙ МЕТОД КЛАССИФИКАЦИИ ТЕКСТОВ С ЛЕКСИЧЕСКИМ РАЗБОРОМ СЛОВА	537
Третьяков Ф.И., Серебряная Л.В.	
ПРОВЕРКА ИНФОРМАТИВНОСТИ КЛАССИФИКАЦИОННЫХ ПРИЗНАКОВ В ЗАДАЧЕ АВТОМАТИЧЕСКОЙ КЛАССИФИКАЦИИ ТЕКСТОВ НА ЕСТЕСТВЕННОМ ЯЗЫКЕ	541
Глазкова А.В.	
ВЕРИФИКАЦИЯ МОДЕЛЕЙ ПРОЦЕССОВ В ДИНАМИЧЕСКИХ СИСТЕМАХ ПО МЕТОДУ MODEL CHECKING	545
Королев Ю.И.	
ИССЛЕДОВАНИЕ И РАЗРАБОТКА СИСТЕМЫ АБСТРАКТНОЙ АРГУМЕНТАЦИИ С ВЕРОЯТНОСТНЫМИ СТЕПЕНЯМИ ОБОСНОВАНИЯ	549
Дервянко А. В., Моросин О. Л.	

МЕТОДЫ ТЕМАТИЧЕСКОЙ КЛАСТЕРИЗАЦИИ ПРИМЕНИТЕЛЬНО К АНАЛИЗУ НОВОСТНЫХ СТАТЕЙ	555
Солошенко А.Н., Орлова Ю.А., Заболеева-Зотова А.В.	
РАЗРАБОТКА ИНТЕЛЛЕКТУАЛЬНОГО ОБУЧАЮЩЕГО ИНТЕРФЕЙСА ДЛЯ ДЕТЕЙ С ИСПОЛЬЗОВАНИЕМ СЕНСОРА LEAP MOTION	561
Розалиев В.Л., Вяхирев А.А., Заболеева-Зотова А.В.	
АВТОМАТИЗИРОВАННОЕ ВЫДЕЛЕНИЕ КИСТЕЙ РУК ЧЕЛОВЕКА ДЛЯ РАСПОЗНАВАНИЯ ЖЕСТОВОЙ РЕЧИ	565
Розалиев В.Л., Агафонов Г.В., Кириченко М.И.	
АВТОМАТИЗАЦИЯ РАСПОЗНАВАНИЯ ВНЕШНЕГО ВИДА ЧЕЛОВЕКА ПО ТЕКСТУ НА ЕСТЕСТВЕННОМ ЯЗЫКЕ И СИНТЕЗ ПОРТРЕТНОГО ИЗОБРАЖЕНИЯ	571
Орлова Ю.А., Долбин А.В., Кипаева Е.В.	
КОНТЕКСТНО-ЗАВИСИМЫЙ АНАЛИЗ ПОРТРЕТНЫХ ФОТОГРАФИЙ	577
Алексеев А.В., Орлова Ю.А.	
МЕТОД ХРАНЕНИЯ ИЕРАРХИЧЕСКИХ СЕМАНТИЧЕСКИХ СЕТЕЙ В РЕЛЯЦИОННОЙ БАЗЕ ДАННЫХ	581
Ванясин Н.В., Сидоркина И.Г.	
НР-ЗАДАЧА БАЛАНСИРОВКИ НАГРУЗОЧНОЙ СПОСОБНОСТИ СЕТИ В САПР	585
Сидоркина И.Г. Сорокин О.Л.	
АППАРАТНЫЙ ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ ДЛЯ СЕТЕВОЙ ИНФРАСТРУКТУРЫ САПР И СИСТЕМ УПРАВЛЕНИЯ	589
Шелеметьев А.М., Шелеметьева Я.В., Сидоркина И.Г.	
РЕШЕНИЕ ЗАДАЧИ ПРОГНОЗИРОВАНИЯ ВОЗРАСТАНИЯ НАГРУЗКИ В СИСТЕМЕ ДИСТАНЦИОННОГО ОБУЧЕНИЯ «ISPRING LEARN»	593
Сокольников А.М., Сидоркина И.Г.	
ОПЫТ РАЗРАБОТКИ СИСТЕМЫ ИМИТАЦИИ, ОСНОВАННОЙ НА АГЕНТАХ	597
Замятина Е.Б., Каримов Д.Ф. , Митраков А.А.	
РАЗРАБОТКА УЧЕБНОЙ ПРОГРАММНОЙ ПЛАТФОРМЫ ДЛЯ РОБОТА iROBOT CREATE	605
Бармина Е.И., Ланин В.В., Плетнёв А.О.	
АВТОРСКИЙ УКАЗАТЕЛЬ	609

TABLE OF CONTENTS

FOREWORD	23
PROBLEM OF UNDERSTANDING: PRESENT AND FUTURE OF ARTIFICIAL INTELLIGENCE Tarassov V.B.	25
SCIENTOMETRICS OF THE DOMAIN: OSTIS-CONFERENCES CASE STUDIES Khoroshevsky V.F., Efimenko I.V.	43
EMANTIC TECHNOLOGY OF COMPONENT DESIGN OF SYSTEMS, MANAGED BY KNOWLEDGES Golenkov V.V., Guliakina N.A.	57
SUPPORT TOOLS KNOWLEDGE-BASED SYSTEMS COMPONENT DESIGN Shunkevich D.V., Davydenko I.T., Koronchik D.N., Zukov I.I., Parkalov A.V.	79
IMPLEMENTATION OF WEB-PLATFORM FOR SYSTEMS BASED ON KNOWLEDGES Koronchik D. N	89
THE METHODOLOGY OF KNOWLEDGE BASED SYSTEM COMPONENT DESIGN Shunkevich D.V., Davydenko I.T., Koronchik D.N., Hubarevich N.U., Boriskin A.S.	93
MODELS AND ALGORITHMS OF INTEGRATION OF KNOWLEDGE BASED ON HOMOGENEOUS SEMANTIC NETWORKS Ivashenko V.P.	111
SEMANTIC NETWORKS REPRESENTATION AND ALGORITHMS FOR THEIR ORGANIZATION AND SEMANTIC PROCESSING ON MASSIVELY PARALLEL COMPUTERS Ivashenko V.P., Verenik N.L., Girel A.I., Seitkulov Y.N., Tatur M.M.	133
A COMPONENT-BASED ARCHITECTURE OF INTELLIGENT SYSTEMS OF CONSULTATION PROVIDING AND TRAINING OF SOFTWARE DEVELOPERS Pivovarchyk O.	141
PRINCIPLES OF CONSTRUCTION CLINICAL DECISION SUPPORT SYSTEM BASED ON OSTIS TECHNOLOGY Kayeshko A.I., Marhunou Y.A.	149
KNOWLEDGE BASE DESIGN INTELLECTUAL REFERENCE SYSTEMS IN ALGEBRA Sharipbay A.A., Omarbekova A.S., Nurgazinova G. Sh.	157
THE WAYS OF ARTIFICIAL INTELLIGENCE EVOLUTION IN KAZAKHSTAN Sharipbay A.A.	161
CREATING AN ONTOLOGICAL MODEL FOR THE LOCAL COMPUTER NETWORK Sharipbay A.A., Askarova S.A., Mukanova A.S.	165
ONTOLOGIC MODEL OF PROCESS OF ENSURING INFORMATION SECURITY Niyazova R.S., Budanova N.	169

CONCEPTION INSTRUMENTAL PLATFORM INFORMATION SECURITY IN CLOUD COMPUTING WITH INTELLIGENCE TECHNOLOGIES	173
Vishniakou U.A., Gongas Sas M.M., Mosdurani Shiras M.G.	
ONTOLOGICAL APPROACH TO DEVELOPMENT OF SCIENTIFIC INTERNET RESOURCE	177
Zagorulko Yu.A., Zagorulko G.B.	
DEVELOPMENT AND GENERATION OF VIRTUAL ENVIRONMENTS ON BASE OF SEMANTYC FORMS	183
Gribova V.V., Fedorischev L.A.	
A CLASSIFICATION FOR THE TASKS REVEALED DURING INTELIGENT ACTIVITY SYSTEM ANALYSIS	187
Shalfeeva E.	
A TECHNOLOGY FOR DEVELOPMENT OF PROBLEM SOLVERS OF INTELLIGENT SYSTEMS WITH THE USE OF IACPAAS CLOUD PLATFORM TOOLS	193
Krylov D.A., Moskalenko Ph.M., Timchenko V.A.	
METHODS AND TOOLS OF CONTINGENCY MANAGEMENT IN ENERGY SECTOR BASED ON SEMANTIC MODELING	199
Massel L.V., Massel A.G.	
FROM ENTERPRISE INFORMATION PORTALS TO KNOWLEDGE MANAGEMENT PORTALS: THE ROLE OF INTELLIGENT SERVICES	205
Strakhovich E.V., Vlasov S.A., Gavrilova T.A.	
COLLABORATIVE RECOMMENDER SYSTEMS	209
SHEREMETOVA E.I.	
TOOLS AND METHODS IN BUSINESS RULES APPROACH	215
SAVIN A.M.	
NUCLEAR KNOWLEDGE MANAGEMENT	221
Kostebelova V.K.	
A SCENARIO APPROACH IN THE RESEARCH OF DYNAMICS OF INFORMATION STREAMS IN THE INTERNET	225
Dodonov A.G., Lande D.V., Boychenko A.V.	
APPROACHES OF THE FORMATION OF COMPLEX ENGINEERING CALCULATIONS ON THE PORTAL	231
Novogradskaya R.L., Globa L.S.	
METAGRAPH BASED REPRESENTATION AND PROCESSING OF FUZZY KNOWLEDGBASES	237
Globa L.S., Ternovoy M.Y., Shtogrina O.S.	
CHANGES DETECTION OF XML DOCUMENTS	241
Globa L.S., Molchanov Y.N.	
ENTERPRISE DATA INTEGRATION METHODS UNDER CONDITIONS OF LOW SOURCES RELATEDNESS	249
Galushka I.M, Oksanich I.G., Shcherbak S.S.	

SEMANTIC APPROACH TO THE WEB OF THINGS OBJECTS INTEGRATION Rogushina J., Gladun A.	255
KNOWLEDGE-ORIENTED MEANS OF SEMANTIC SEARCH INTO THE WEB Rogushina J.	265
REVIEW AND ANALYSIS METHODOLOGIES AND TECHNIQUES OF ONTOLOGY BUILDING Khala C.A.	271
SPECIFICATION OF TEMPORAL RELATIONS IN THE SYSTEM "BINARY MODEL OF KNOWLEDGE" Plesniewicz G.S., Nguyen Thi Minh Vu	275
FROM TERMINOLOGICAL NETWORKS TO THE EXPLANATORY DICTIONARIES Malkovsky M.G., Soloviev S.Y.	281
ABOUT AN APPROACH OF TOPIC MODELING FOR TEXT IN NATURAL LANGUAGE Tran Van An	285
MOTIVATION OF MODELING COMMONSENSE REASONING PROCESS Naidenova X. A.	291
HEURISTIC METHOD OF CONSTRAINT SATISFACTION BASED ON MATRIX REPRESENTATION OF CONSTRAINTS Zuenko A.A., Ochinskaya A.A.	297
NETWORK MODEL OF ACTIVITY PROPAGATION AMONG HETEROGENEOUS AUTOMATA Zhilyakova L.Yu.	303
THE SYSTEM PRIORITY TRAFFIC AT INTERSECTIONS "GREEN WAVE" FOR EMERGENCY VEHICLES Sogoyan A.L., Shuts V.N.	309
NATIONAL EDUCATION INFORMATIZATION DEVELOPMENT PROGNOSIS Robert I.V.	315
INTELLIGENT TECHNOLOGIES OF TUTORING IN INFORMATION-EDUCATIONAL SPACE Telnov Yu.F., Trembach V.M.,	325
APPLICATION OF COGNITIVE GRAPHICS TOOLS BASED ON THE 3-SIMPLEX IN INTELLIGENT TRAINING-TESTING SYSTEMS Yankovskaya A.E., Shurygin Y.A., Yamshanov A.V., Krivdyuk N.M.	331
WOLFRAM MATHEMATICA TOOLS AND TECHNOLOGIES OF DEVELOPMENT OF INTELLIGENT TUTORING SYSTEMS Taranchuk VB	339
INTELLECTUAL TECHNOLOGY DEVELOPMENT OF COMPONENT DESIGN OF DESIGN AND INDUSTRIAL LEARNING ENVIRONMENTS Afnas'ev A.N., Voit N.N.	347

ONTOLOGICAL APPROACH TO THE ANALYSIS OF COMPETENCIES OF UNIVERSITY GRADUATES Sherkunov V.V.	351
NEURAL NETWORK MODEL OF THE PROCESS OF PROFESSIONAL EDUCATION OF YOUNG SPECIALISTS Fedyayev O.I.	357
METHODS OF INDISTINCT REGULATION IN MANAGEMENT PROBLEMS EDUCATIONAL PROCESS Nishanov A.H., Doshchanova M. Yu., Mirzaev D.A.	365
INTELLECTUAL MEANS OF AUTOMATION OF MANAGEMENT OF TRAINING Babamukhamedova M.Z., Doshchanova M. Yu., Djangazova ., K.A.	369
CONCEPTUAL EXPERIMENTING IN DESIGNING THE CONFIGURED PATTERNS OF AVIATION PARTS Sosnin P.I., Chorakaev O.E.	373
ONTOLOGY OF DESIGNING FOR TEMPLATES OF AIRCRAFT PARTS Grishin M.V., Larin S.N., Sosnin P.I.	381
SEMANTIC ANALYSIS DIAGRAMMATICAL MODELS IN THE DESIGN OF COMPLEX AUTOMATED SYSTEMS Afanasjev A.N., Gainullin R.F., Afanasjeva T.V.	385
APPLICATION OF NAVIGATION STRUCTURE OF DIGITAL ARCHIVE OF PROJECT ORGANIZATION IN TASKS OF THE INTELLECTUAL ANALYSIS OF CAD DOCUMENTS Subkhangulov R.A., Filippov A.A.	389
ALGORITHMS FOR EVALUATION OF WORD COMBINATIONS OR WORDS MEMBERSHIP DEGREE TO TERM LIST BASED ON SUBJECT AREA ONTOLOGY Andreev I.A., Bashaev V.A., Klein V.V., Moshkin V.S., Yarushkina N.G.	395
METHODS OF CONSTRUCTION OF FUZZY ONTOLOGIES OF COMPLEX SUBJECT AREAS Moshkin V.S., Yarushkina N.G.	401
ONTOLOGICAL APPROACH TO THE FORMATION OF PROJECT REQUESTS OF INTELLIGENT AGENT Namestnikov A.M., Subkhangulov R.A.	407
THE EXPERIENCE OF SEMANTIC MODELING AND DESIGNING TOOLS ON WIDELY USED PLATFORM Smirnov S.V.	413
IMPLEMENTATION OF REMOTE CONTROL OVER A PARAMETERIZED THREE-DIMENSIONAL MODEL OF THE AIRPLANE BY A CLIENT-SERVER APPLICATION Borgest N.M., Vlasov S.A., Korovin M.D.	417
APPROACH TO THE PARAMETRIZATION OF THREE-DIMENTIONAL MODELS FOR SLOVING THE TASK OF THE AIRCRAFT PRELIMINARY DESIGN AUTOMATION Borgest N.M., Korovin M.D., Spirina M.O.	421

METHODS OF KNOWLEDGE REPRESENTATION TECHNIQUES FOR USE IN MODELING Korshikov D, Lakhin O, Noskova A, Yurygina Yu	425
ONTOLOGICAL MODELING OF MAINTENANCE Fedotova A.V., Davydenko I.T.	429
SEMANTIC ASPECTS OF SEARCH REQUEST REPRESENTATION AND PROCESSING IN CORPUS-MANAGER SYSTEM Nevzorova O.A., Mukhamedshin D.R., Bilalov R.R.	439
TECHNOLOGY OF AUTOMATED SYNTHESIS THE INFORMATION SYSTEMS USING SEMANTIC MODELS OF SUBJECT AREA Bikmullina I.I.	445
WEB-SITE FOR HANDY MORPHOLOGICAL DISAMBIGUATION IN TATAR LANGUAGE CORPUS Gilmullin R.R., Gataullin R.R.	451
MULTICRITERIA APPROACH TO FORMING POLICY OF SCIENTIFIC FOUNDATION AND EXPERT COMMUNITY Boychenko V.S., Zaboлева-Zotova A.V., Petrovsky A.B.	455
ONTOLOGICAL APPROACH THE DESIGN PROCESS Burdo G.B., Vorobyeva E.V.	461
IDENTIFICATION MODEL REPRESENTATIONS OF THE AGENT ABOUT THE SUBJECT IN A SITUATION OF CHOICE Vinogradov, P., Borzov D. A.	465
STUDY THE EFFECTIVENESS OF GENETIC ALGORITHM FOR DOCUMENTARY SUBJECT SEARCH Ivanov V.K., Palyukh B.V.	471
MODELING OF AUTOMATED QUALITY CONTROL BY PETRI NETS Burdo G.B., Vinogradov G. P., Sorokin A. Y.	477
APPLYING DEEP BELIEF NEURAL NETWORKS TO EXTRACTION VALUEBLE SEMANTIC FEATURES Golovko V.A., Kroshchanka A.A.	481
DEVELOPMENT OF OPEN TECHNOLOGY DESIGNING OF INTELLIGENT ROBOTIC SYSTEMS Prakapovich R.A., Sychyou U.A., Gerasuto S.L.	487
THE SEARCHING TASK IN THE STATIC SWARM Vorobiev V.V.	489
CREATION OF INTELLEAGENT MULTIAGENT COBOR–SYSTEMS Soloviev V. I.	495
WWW.CORPUS.BY: OPEN-SOURCE COMPONENTS FOR NATURAL LANGUAGE INTERFACES Hetsevich Y.S., Lobanov B.M., Lysy S.I., Hiuntar E.V., Denisjuk D.A., Zakharyeu V.A.	499

COMPONENT DESIGN FOR SPEECH RECOGNITION OF NATURAL LANGUAGE INTERFACE	507
Nikolaenko K.A., Kaigorodova L.I. , Hetsevich Y.S.	
SEPARATE MODELING OF SPEECH USING VOICE, PHONETICAL AND PROSODIC CHARACTERISTICS	513
Azarov E., Petrovsky A.	
APPLICATION OF TIME SERIES ANALYSIS FOR SPEAKER CLUSTERING	519
Vagin V.N., Ganishev V.A.	
VOICE USER IDENTIFICATION IN ACCESS CONTROL SYSTEMS	525
Menshakov P.A., Murashko I.A.	
PERSON PHYSIOGNOMY INFORMATIONAL MODEL	529
Kharlamov A.A.	
ABOUT ONE METHOD OF USE OF GENERALIZED ONTOLOGY FOR THE ANALYSIS OF THE NATURAL LANGUAGE MESSAGES OF INTERNET USERS	533
Deeva N.V. , Vishneuski S.Y.	
TEXT CLASSIFICATION FREQUENCY METHOD WITH WORD LEXICAL ANALYSIS	537
Tretyakov F.I., Serebryanaya L.V.	
CLASSIFICATION FEATURES INFORMATIONAL CONTENT TESTING FOR AUTOMATIC NATURAL TEXTS CLASSIFICATION TASK	541
Glazkova A.V.	
VERIFICATION OF MODELS OF PROCESSES IN DYNAMIC SYSTEMS USING MODEL CHECKING METHOD	545
Eremeev A.P., Korolev Y.I.	
RESEARCH AND DEVELOPMENT OF THE ABSTRACT ARGUMENTATION SYSTEM WITH PROBABILISTIC DEGREE OF JUSTIFICATION	549
Derevyanko A., Morosin O.	
THEMATIC CLUSTERING METHODS APPLIED TO NEWS ARTICLES ANALYSIS	555
Soloshenko A.N., Orlova Yu.A., Zaboлева-Zotova A.V.	
THE DEVELOPMENT OF INTELLIGENT LEARNING INTERFACE FOR CHILDREN USING SENSOR LEAP MOTION	561
Rozaliev V.L., Vyakhirev A.A., Zaboлева-Zotova A.V.	
AUTOMATED ALLOCATION OF THE HANDS OF THE PERSON TO RECOGNIZE SIGN LANGUAGE	565
Rozaliev V.L., Agafonov G.V., Kirichenko M.I.	
AUTOMATED RECOGNITION OF THE APPEARANCE OF MAN ON THE NATURAL LANGUAGE AND SYNTHESIS OF PORTRAIT IMAGES	571
Orlova Yu.A., Doldin A.V., Kipaeva E.V.	
CONTEXT-SENSITIVE ANALYSIS OF PORTRAIT IMAGES	577
Alekseev A.V., Orlova Yu.A.	
HIERARCHICAL SEMANTIC NETWORK STORAGE METHOD FOR RELATIONAL DATABASES	581
Vanyasin N.V., Sidorkina I.G.	

NP-BALANCING TASK RATED ABILITY OF NETWORK IN CAD Sidorkina I.G. Sorokin O.L	585
HARDWARE ARTIFICIAL INTELLIGENCE FOR CADS NETWORK INFRASTRUCTURE AND FOR CONTROL SYSTEMS Shelemeteva Y.V., Shelemetev.A.M., Sidororkina I.G.	589
SOLUTION TO THE PROBLEM OF PREDICTING LOAD INCREASES IN LEARNING MANAGEMENT SYSTEM "ISPRING LEARN" Sokolnikov A.M., Sidorkina I.G.	593
THE EXPIERANCE OF AGENT-BASED SIMULATION SYSTEM IMPLEMENTATION Zamyatina E.B., Karimov D.F. , Mitrakov A.A.	597
EDUCATIONAL SOFTWARE PLATFORM DEVELOPMENT FOR iROBOT CREATE Barmina E., Lanin V., Pletnev A.	605
AUTHOR INDEX	609

ПРЕДИСЛОВИЕ

Основным практическим результатом исследований в области искусственного интеллекта является разработка не только интеллектуальных систем, но и технологий, обеспечивающих быстрое и качественное построение таких систем. Создание указанных технологий требует решения следующих задач:

- чёткого выделения логико-семантического уровня интеллектуальных систем, который абстрагируется от всевозможных вариантов технической реализации этих систем (в том числе и от использования принципиально новых компьютеров, ориентированных на аппаратную поддержку интеллектуальных систем);
- разработки онтологии проектирования интеллектуальных систем и унификации описания логико-семантических моделей интеллектуальных систем;
- обеспечения платформенно независимого характера логического проектирования интеллектуальных систем, результатом которого является унифицированное описание логико-семантических моделей проектируемых интеллектуальных систем;
- использования методики компонентного проектирования интеллектуальных систем, в основе которой лежит постоянно пополняемая библиотека многократно используемых компонентов интеллектуальных систем (многократно используемых подсистем, компонентов баз знаний, агентов обработки знаний, компонентов пользовательских интерфейсов);
- обеспечения семантической совместимости многократно используемых компонентов интеллектуальных систем и семантической совместимости самих интеллектуальных систем и технологий их проектирования.

Основной целью ежегодных конференций OSTIS (Open Semantic Technology for Intelligent Systems) является создание условий для расширения сотрудничества различных научных школ, вузов и коммерческих организаций, направленного на разработку и применение массовых и постоянно совершенствуемых технологий компонентного проектирования интеллектуальных систем.

V международная научно-техническая конференция «Открытые семантические технологии проектирования интеллектуальных систем» (OSTIS-2015) посвящается 20-летию открытия специальности «Искусственный интеллект» в Республике Беларусь, а также 20-летию начала обучения студентов по этой специальности на созданной для этого кафедре Интеллектуальных информационных технологий Белорусского государственного университета информатики и радиоэлектроники.

Председатель Программного комитета конференции OSTIS-2015
Председатель Совета Российской ассоциации искусственного интеллекта
Кузнецов Олег Петрович

FOREWORD

Development not only intelligent systems but also technologies that ensure fast and efficient construction of intelligent systems are the main practical result of research in artificial intelligence. Development of these technologies requires the following tasks:

- precise separation of logical and semantic level of intelligent systems, which is abstracted from various versions of the technical implementation of these systems (including the use of innovative computer-based hardware support for Intelligent Systems);
- developing an ontology design of intelligent systems and unifying description of logical and semantic models of intelligent systems;
- provide a platform independent nature of the logical design of intelligent systems, which result is a unified description of logical and semantic models of an intelligent systems;
- use of a component design methodology of intelligent systems, which is based on permanently increasing library of reusable components of intelligent systems (reusable subsystems, knowledge bases components, knowledge processing agents, user interfaces components);
- ensuring of semantic compatibility of reusable components of intelligent systems and semantic compatibility of these intelligent systems and technologies of such systems design.

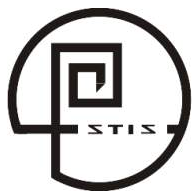
Creating the conditions for the expansion of cooperation between different scientific schools, universities and business organizations, aimed on development and using of mass and continuously improved component design technologies for intelligent systems is the main purpose of annual conferences OSTIS (Open Semantic Technology for Intelligent Systems).

V International scientific and technical conference “Open Semantic Technology for Intelligent Systems” (OSTIS-2015) is dedicated to 20-th anniversary of the “Artificial intelligence” speciality in The Republic of Belarus, and to 20-th anniversary of the students education starting on that speciality on created for that purpose department of Intelligent information technologies of Belarusian State University of Informatics and Radioelectronics.

Programme Committee Chair.

Chairman of the Council of the Russian Association for Artificial Intelligence

Kuznetsov Oleg Petrovich



OSTIS-2015

(Open Semantic Technologies for Intelligent Systems)

УДК 004.8:165.1

ПРОБЛЕМА ПОНИМАНИЯ: НАСТОЯЩЕЕ И БУДУЩЕЕ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

Тарасов В.Б.

*ФГБОУ ВПО «Московский государственный технический университет им. Н.Э.Баумана
г. Москва, Россия*

Vbulbov@yahoo.com

Описаны результаты предварительного исследования междисциплинарной проблемы понимания в контексте развития проекта OSTIS, связанного с разработкой искусственных понимающих агентов и других классов когнитивных интеллектуальных систем новых поколений. В этом плане особый интерес представляют сравнительно новые объекты понимания – поведение, действия, ситуации, измерения, и пр. В статье даны некоторые ответы на вопрос, «что такое понимание»?; приведены и проанализированы основные определения понимания. Проведено обсуждение классических концепций понимания, выработанных в герменевтике, семиотике, дидактике, психологии. Систематизированы ключевые категории, относящиеся к пониманию, такие как «значение», «смысл», «ценность», «истина», «интерпретация», «объяснение», «оправдание», прослежены связи между ними. Благодаря рассмотрению ряда характеристик понимания (полнота, отчетливость, глубина, и др.), обоснованы варианты классификации уровней понимания, введена иерархия уровней понимания для автономного робота. С целью раскрытия базовых механизмов понимания сделан краткий обзор интеллектуальных (мыслительных и перцептивных) операций. Главное внимание в работе уделено построению системы гибридных когнитивных микронаук (отличной от канонической системы когнитивистики по Гарднеру), ориентированной на изучение проблемы понимания. Предложены формальные подходы к моделированию понимания на базе абстрактных алгебр и прагматических логик.

Посвящается 20-летию юбилею Кафедры интеллектуальных информационных технологий Белорусского государственного университета информатики и радиоэлектроники

Ключевые слова: искусственный интеллект; знания; понимание; интерпретация; прагматика; значение; смысл; герменевтика; когнитивная семантика; когнитивная семиотика; когнитивная информатика.

Введение

*Знать – хорошо, а понимать – еще лучше
(П.Ланжевен)*

Кафедра интеллектуальных информационных технологий (ИИТ) БГУИР, основанная 10 января 1995 года, стала первой кафедрой не только в Республике Беларусь, но и вообще на территории СНГ, начавшей подготовку кадров по специальности «искусственный интеллект». За 20 лет работы кафедры ИИТ состоялось уже 15 выпусков специалистов по информатике и искусственному интеллекту, а общее число выпускников приближается к одной тысяче.

Важной исторической вехой международной учебно-организационной деятельности кафедры ИИТ стала организация под эгидой БГУИР и с участием Российской ассоциации искусственного интеллекта (РАИИ) так называемых Браславских

школ. Эти летние международные школы-семинары по искусственному интеллекту (ИИ) для студентов, аспирантов и молодых ученых, проходившие ежегодно с 1997 по 2001 г. на Браславских озерах, пользовались большой популярностью не только у научной молодежи, но и у маститых ученых-лекторов школ (см., например, [НИИ,1998]). С одной стороны, представители научной молодежи из разных стран СНГ могли прослушать лекции по актуальным и перспективным проблемам ИИ, пообщаться в неформальной обстановке с ведущими учеными в этой области, задать любые вопросы, обсудить новые идеи. Для некоторых из них это общение дало важный толчок для выбора профессии, связанной с разработкой и применением новых информационных технологий на базе ИИ, перехода к активной преподавательской деятельности. С другой стороны, выступления молодых ученых и совместные обсуждения позволили выявить наиболее способных студентов и аспирантов, привлечь их к научным

исследованиям и разработках, в том числе, в рамках российско-белорусских проектов. Отметим, что большинство нынешних преподавателей кафедры ИИТ – представители ее среднего поколения – были молодыми участниками Браславских школ.

С первых дней существования кафедры ИИТ на ней были развернуты научно-исследовательские работы в области технологий и инструментальных средств для быстрой разработки интеллектуальных систем различного назначения. «Визитной карточкой» кафедры стали теоретические исследования и практические разработки по графодинамическим моделям обработки информации (см. [Голенков, 2012]) и их приложениям. Так по результатам выполнения совместных проектов по линии Российского фонда фундаментальных исследований – Белорусского республиканского фонда фундаментальных исследований была опубликована монография «Интеллектуальные обучающие системы и виртуальные учебные организации» [ИОС и ВУО, 2001], причем в состав ее авторского коллектива входили представители ряда ведущих вузов РФ и РБ. По сути, работа над этой книгой стала одной из предпосылок возникновения проекта OSTIS, стартовавшего спустя 10 лет.

Проект OSTIS (Open Semantic Technologies for Intelligent Systems) направлен на создание открытых семантических технологий моделирования и проектирования интеллектуальных систем разных классов. Его цели заключаются в создании массовой, комплексной и легко расширяемой технологии разработки интеллектуальных систем, включающую в себя теоретические и практические, программные и аппаратные аспекты, а также в формировании общей инфраструктуры, обеспечивающей сочетание научной и учебной, инженерной и коммерческой деятельности в области искусственного интеллекта [Голенков, 2013].

Проект OSTIS – это открытый, комплексный проект, включающий в себя: а) создание технологии OSTIS; б) разработку интеллектуальной метасистемы для поддержки проектирования прикладных интеллектуальных систем; в) разработку широкого семейства прикладных интеллектуальных систем по технологии OSTIS. Сегодня не только научная, но и учебная деятельность кафедры ИИТ, в том числе, выбор тем и подготовка диссертаций аспирантами и магистрантами, самостоятельная работа студентов, курсовое проектирование, осуществляются в русле проекта OSTIS.

Один из важных подпроектов общего проекта OSTIS связан с построением семантических интеллектуальных систем (одним из первых этот термин использовал В.В.Мартынов в [Мартынов, 1978], [Мартынов, 2009]), что означает явное включение смысла в формализмы представления информации и знаний. Соответственно, в рамках конференции OSTIS-2014, посвященной 90-летию

со дня рождения профессора В.В.Мартынова, эта проблема была затронута в докладах [Тарасов, 2014], [Ефименко и др., 2014], [Голенков и др., 2014] и пр. Например, в докладе В.В. Голенкова и Н.А.Гулякиной «Структуризация смыслового пространства» были вынесены на обсуждение такие вопросы, как: Что такое смысл? В чем заключаются теоретические проблемы формализации семантики? Каким требованиям должно удовлетворять смысловое представление знаний? Что такое семантическое пространство, и какова его общая структура?

Переход от знаний и смысла к пониманию выступает как естественное расширение подходов классических и семантических интеллектуальных систем в сторону понимающих интеллектуальных систем.

Настоящая работа посвящена системному анализу междисциплинарной проблемы понимания в интересах продвижения по пути к машинному пониманию, т.е. созданию искусственных понимающих агентов, например, «понимающего компьютера» [Winograd et al., 1986], «понимающего робота» (см. раздел 2.4), «понимающей умной пыли» (Comprehensive Smart Dust) или «понимающей интеллектуальной среды» (Comprehensive Ambient Intelligence). С одной стороны, понимание характеризуется как главная цель любого обучения и составляет предмет дидактики. С другой стороны, понимание есть одна из центральных когнитивных способностей человека, поэтому развитие когнитивистского подхода в ИИ предполагает исследование и моделирование различных сторон и операций понимания. Рассмотрение проблемы понимания неотделимо от становления таких новых областей компьютерных наук как когнитивная информатика и когнитивные компьютеры, когнитивные агенты и когнитивная робототехника, когнитивные, гранулярные и мягкие вычисления.

Взаимосвязи между познанием и пониманием являются двусторонними: сама возможность познания и результативность действий человека во многом определяется достигнутым уровнем понимания (предпониманием) [СФС, 2004], но понимание, в свою очередь, сильно зависит от исходных знаний, когнитивных возможностей и структуры языка. В частности, понимание фраз на естественном языке обеспечивается не столько языковыми, сколько внеязыковыми факторами, а прежде всего, предположением, т.е. предшествующим опытом.

Понимание выступает в качестве предпосылки и необходимого условия общения когнитивных агентов, в частности, коммуникативно-аффективных и коммуникативно-информационных процессов.

В то же время термин «понимание» часто раскрывается с помощью категорий «смысл», «значение» «ценность», «истина» как способность постичь смысл текста или мотивы поведения, определить значение события, установить ценность

сведений, найти истину и т.п. Соответственно, проводят различия между объективными теориями понимания, связанными с установлением значений знаков, и субъективными теориями, в которых ключевую роль играет оценка результатов понимания [Васильев,1999]. С механизмами понимания тесно связаны понятия «интерпретация» и «объяснение». В качестве атрибутов понимания используются такие слова как «уровень», «глубина», «полнота», «отчетливость», «операция» и пр. [Чудова, 2012].

Все это требует систематизации и критического анализа методов и подходов к изучению понимания, используемых в герменевтике, логике, семантике, семиотике, дидактике, информатике и других научных дисциплинах.

1. Что такое понимание? Краткая подборка определений

Проблема понимания, представляющая собой одну из центральных и наиболее сложных проблем науки в целом и искусственного интеллекта в частности, носит междисциплинарный характер. Объекты понимания многообразны: ими могут быть тексты, сообщения, люди, поведение, события, ситуации, и пр.

Спектр подходов к изучению данной проблемы характеризуется большим разнообразием: так в [Знаков,1986] выделены методологические, логические, гносеологические, психологические, лингвистические, семантические, коммуникативные аспекты понимания.

Значительная часть различных определений понимания имеет отношение к герменевтике – науке об интерпретации текстов, которая со временем стала рассматриваться как общая методология понимания. Так в «Словаре по логике» [Ивин, 1997б] понимание характеризуется как *универсальная операция мышления, связанная с усвоением нового содержания, включением его в систему устоявшихся идей и представлений.*

По сути, такая трактовка понимания как включения новых знаний в прошлый опыт восходит к одному из основателей герменевтики Ф. Шлейермахеру, который считал, что в основе любого понимания лежит соотнесение или сравнение с уже известным. Понимание есть подведение под известное понятие или отнесение понятия к определенной категории.

Это определение понимания близко к идеям операциональной теории интеллекта Ж.Пиаже [Пиаже, 1994], в частности, к понятию «ассимиляции», означающему усвоение нового материала на основе существующих когнитивных структур и схем поведения, а также действия с новыми предметами на основе уже сложившихся умений и навыков. Соответственно Г.Геффдинг считал, что характер понимания определяется качеством узнавания.

Еще одно классическое определение понимания, приводимое в Энциклопедическом словаре по философии [Философия, 2004], представляет собой его оценочную трактовку: понимание – это такая универсальная операция мышления, которая является оценкой объекта (текста, поведения, явления природы) с определенных позиций, на основе некоторого образца, стандарта, нормы, принципа и т.п. Таким образом, данная трактовка является аксиологической, т.е. опирается на теорию ценностей, поскольку любая оценка подразумевает подведение под ценность (или вывод из принятых ценностей и с использованием общих правил).

Аксиологическая концепция понимания опирается на принцип единства понимания и оценки, сформулированный одним из классиков герменевтики В.Дильтеем. В развитие этого принципа «отец-основатель» современной диалогии М.М.Бахтин писал, что безоценочное понимание невозможно [Бахтин,1979]. Нельзя разделить понимание и оценку: они одновременны и составляют единый целостный акт.

Таким образом, пониматься может все, для чего найден общий образец, т.е. не только тексты, но и действия, поступки, ситуации. Например, в последнем случае двумя базовыми операциями, делающими понимание возможным, являются поиск стандарта оценки (нормы) и обоснование его приложения к конкретной ситуации. Непонимание чаще всего обусловлено необоснованностью, неочевидностью или отсутствием такого стандарта.

Важнейшей познавательной процедурой, способствующей пониманию, является объяснение. Объяснение – это рассуждение, посылки которого содержат информацию, достаточную для выведения из нее описания объясняемого явления. Объяснение связывает объясняемое явление или событие с другими явлениями и событиями реального мира и указывает на закономерный характер этих связей. Объяснение и понимание часто трактуют как две универсальные операции мышления, взаимно дополняющие друг друга [Ивин,1997а]. Объяснение, рассматриваемое как подведение объясняемого явления под научный закон, пример или общую истину, опирается на описание и обеспечивает понимание, а понимание как подведение под общее правило или стандарт имеет нормативную базу (рисунок 1). При этом оценка уровня понимания вносит свои коррективы в характер объяснения.

Классическим вариантом объяснения является объяснение на *примерах*. Пример принципиально отличается от образца. *Пример* говорит о том, что имеет место в действительности, а *образец* – о том, что должно быть. Иными словами, примеры используются для поддержки описательных суждений, а ссылки на образцы служат как обоснование правил и норм.

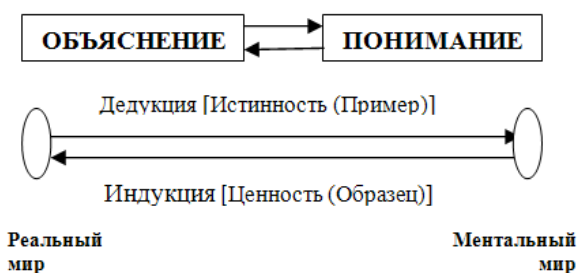


Рисунок 1 – Взаимосвязи и различия между объяснением и пониманием

В то же время, некоторые авторы, например, Г.Х. фон Вригт, указывают на факты нарушения комментарности этих процедур (например, есть сильное каузальное объяснение вида дедуктивного рассуждения, но нет дедуктивного каузального понимания). По-видимому, это обусловлено тем, что в отличие от объяснения понимание особым образом связано с интенциональностью и задачами интерпретатора [фон Вригт, 1986].

В психологическом словаре [Психология,1990] **понимание определяется как способность постичь смысл или значение чего-либо с целью достижения полезного результата.** Это выдержанное в традициях психосемиотики определение связывает понимание с процедурами раскрытия реального смысла, с соотношениями между значением, смыслом и ценностью.

Понимание нередко связывают с *рефлексией*. Еще Дж.Локк трактовал понимание как силу мышления, проявляющуюся в рефлексивных действиях ума, а Г.Лейбниц называл пониманием способность обладать отчетливыми идеями, рефлексировать о них, выводить из них разные следствия.

Понимание также рассматривают как процесс постановки и решения мыслительных или практических задач. Согласно Л.Витгенштейну [Витгенштейн,2009], можно показать свое понимание либо путем перевода из одной системы символов в другую, либо в действиях. Нельзя интерпретировать ни одного языкового явления без перевода его знаков в другие знаки той же системы или в знаки другой системы. Следовательно, понимание подразумевает перевод на другой язык или обращение к метаязыку, т.е. к знакам метауровня. Аналогично понимание некоторого сложного явления или ситуации, как правило, основано на их многомодельном представлении.

Однако помимо символического существует и чувственное понимание, например, *эмпатия*, связанная с вживанием человека в другое «я», или понимание как сопереживание, предполагающее идентификацию эмоций другого человека.

По Б.Спинозе, для развития и углубления понимания нужно располагать специальными инструментами; впоследствии Л.С.Выготский выберет в качестве этих инструментов знаки и назовет их «психологическими орудиями» (см.

[Выготский, 1982]). При этом функции знаков не ограничиваются поддержкой когнитивных и коммуникативных механизмов понимания; они также выступают как социальные регуляторы человеческого поведения.

2. Понимание как междисциплинарная проблема

2.1. Понимание и интерпретация в герменевтике

Наиболее древней наукой о понимании является *герменевтика* – учение об интерпретации текстов, восходящее к комментаторам библейских сказаний. Герменевтика в узком смысле означает толкование текста, разъяснение его содержания и установление его значения. При этом изучаются способы выявления скрытого содержания текста. Поиск такого неявно выраженного содержания происходит в результате обращения к знаниям, имеющим отношение к данному тексту. Цель классической герменевтики – понимание авторского текста во всем содержании, где двумя сторонами содержания являются значение и смысл.

В более широком плане, центральная проблема герменевтики связана с идеей языка и ориентированных на язык понятий, таких как: «значение», интенциональность», «интерпретация», «понимание» [фон Вригт, 1986].

В герменевтике имеются различные мнения по вопросу соотношения между пониманием и интерпретацией. Иногда понимание фактически отождествляется с интерпретацией, трактуемой как корректное истолкование текстового источника. Однако чаще эти категории разводят между собой, причем понимание берется в качестве базиса для всякой интерпретации, или, иначе, интерпретация рассматривается как особый аспект понимания, направленный на смысловое содержание текстов. Если понимание предполагает установление прямого, очевидного, конкретного смысла, то интерпретация охватывает такие операции как выявление скрытого смысла, придание нового смысла, порождение смысла в диалоге. Спутником интерпретации языковых выражений является многозначность (или даже неопределеннозначность по В.В.Мартынову [Мартынов, 2009]). Чем глубже уровень понимания текста, тем более интересные и нетривиальные интерпретации могут возникнуть.

Основной вопрос герменевтики с позиции современного ИИ касается смысла текста. Является ли смысл текста абсолютным и объективным, или его можно определить только в зависимости от исходного контекста и позиции интерпретатора? Иными словами, можно ли говорить о смылосодержании текста как замкнутой системы или смысл возникает только при взаимодействии «текст-интерпретатор-среда»?

Для рационалистической, объективистской школы герменевтики, опирающейся в первую

очередь на идеи Ф.Шлейермахера и В.Дильтея, текст имеет смысл независимо от интерпретации. Смысл един и объективен, тогда как его интерпретаторы многочисленны и субъективны. Соответственно, цель теорий интерпретации заключается в том, чтобы разработать методы, позволяющие провести объективный анализ того, что уже существует и заложено в тексте. В идеале речь идет о «полной деконтекстуализации» текста.

Напротив, М.Хайдеггер [Хайдеггер,1997] и Х.-Г. Гадамер [Гадамер,1988] ввели представление о «возникающем смысле». У них интерпретатор и интерпретируемое не существуют независимо друг от друга. Каждое прочтение текста или каждое прослушивание речи есть смыслообразование, возникающее в процессе интерпретации. Точнее, это смыслообразование определяется связями между интерпретацией и культурной традицией. В традиции (предании) содержится прошлое, которое постоянно актуализируется и обуславливает предвосхищение, формирование исходного мнения, проверяемого фактами. В этом русле Ф.Шеллинг указывал на бесконечное множество смыслов, заключенных в литературном произведении и заново формирующихся в сознании читателя.

Если у М.Хайдеггера понимание есть способ бытия человека, его отношения к миру, то подход Х.-Г.Гадамера опирается на представление о герменевтическом круге понимания. Отстаивается идея о циклическом нарастании понимания. Общий процесс понимания предполагает развертывание мерееологических отношений, т.е. переход от целого к части и обратно к целому. Задача состоит в том, чтобы концентрическими кругами расширять единство понятого смысла. Как уже отмечалось, понимание основано на уже имеющемся у нас знании, но то, что мы знаем, определяется нашими возможностями понимания. В целом, сама идея герменевтического круга связана с рассмотрением единства и взаимообусловленности языка, знания, понимания, объяснения и интерпретации.

Процесс интерпретации запускается конкретной герменевтической ситуацией, когда происходит взаимодействие «горизонтов» текста (в скрытом виде, его автора) и интерпретатора. Горизонты смещаются, ибо изменяются герменевтические ситуации. Интерпретатор расширяет свой горизонт, например, ставя вопросы, и создает условия для понимания. Понимание наступает тогда, когда происходит расширение и слияние горизонтов.

По Гадамеру, именно язык оказывается той универсальной средой, в которой осуществляется понимание. В то же время понимание есть плод действия, поскольку, находясь в герменевтической ситуации, интерпретатор тут же использует каждую порцию добытого им смысла в каких-либо целях.

В русле исследования языковой формы мышления Х.-Г. Гадамер проводит различие между герменевтикой и семантикой. Семантика описывает данную нам языковую действительность, как бы

наблюдая ее извне, благодаря чему стала возможной классификация используемых знаков. Заслуга семантического анализа состоит, прежде всего, в фиксации всеобщих структур языка. Герменевтика же сосредоточена на внутренней стороне этого обращения с миром знаков, в особенности, на таком глубоко внутреннем процессе как речь, которая извне предстает как освоение мира знаков.

Наконец, труды П.Рикера [Рикер,1995] знаменуют дальнейший поворот герменевтики к эпистемологической проблематике. В первую очередь, П. Рикер стремится установить связи между герменевтикой и семантикой. Согласно его эпистемологической программе, герменевтика характеризуется как «семантика многозначных выражений», а понимание всегда опосредуется знаками или символами.

Более того, рассуждения Рикера направлены на преодоление концептуального разрыва между герменевтикой и семиотикой. Всякий знак предполагает существование интерпретатора. Семиотическая коммуникация предполагает совместное участие двух интерпретаторов: как отправителя сообщения (адресанта), так и его получателя (адресата). Их различие заключается в том, что первый осуществляет операцию кодирования, а второй – декодирования. Производя декодирование, адресат должен, прежде всего, уловить целое. Так для адресанта не существует проблемы омонимии, поскольку он знает подразумеваемое под ним значение, тогда как адресат, пока у него нет опоры на контекст, вынужден использовать вероятностные (частотные) или нечеткие оценки.

Итак, в конце XX-го – начале XXI-го века герменевтика, трактуемая в широком смысле как общая теория понимания и интерпретации (текстов, действий, событий, ситуаций и пр.), поистине обрела второе рождение [Мартынов,2009], [Шульга,2002], [Winograd, 1986] и постепенно превращается в «науку-перекресток». Помимо идейного моста между герменевтикой и семиотикой появляются новые течения, такие как «когнитивная герменевтика», «компьютерная герменевтика» «логическая герменевтика» Б.Вольневича, и пр.

2.2. Проблема понимания в семиотике

Как уже отмечалось выше, в качестве исходных терминов для исследования проблемы понимания часто используются категории значения, смысла, ценности, истины... Так еще у П.Абеляра понимание раскрывалось через отношение значения к смыслу, а Э.Гуссерль характеризовал понимание как акт установления значения знака. Главный механизм понимания в процессах коммуникации – переход от знака и значения к смыслу на основе интенции. Налицо семиотические корни понимания: как правило, понимание предполагает опору на знак, а результатом понимания является смысл. Поэтому ниже мы изложим семиотические основы понимания и взаимопонимания.

Семиология (термин Ф. де Соссюра [Соссюр де, 2004]) или *семиотика* (по Дж. Локку и Ч. Пирсу [Пирс, 1999]) есть наука о знаковых системах. Она возникла как самостоятельная дисциплина в XX-м веке, хотя еще Дж. Локк ввел этот термин и писал, что важная задача учения о знаках (семиотики) состоит в рассмотрении природы знаков, которыми ум пользуется для понимания вещей или для передачи своего знания другим. Ранние семиотические идеи восходят к концепции «возможных миров» Г. Лейбница. Один из видных представителей русской школы семиотики Г. Г. Шпет понимал ее как «онтологическое учение о знаках вообще» (см. [Шпет, 1996], [Почепцов, 1998]).

Родоначальники теории знаковых систем исходили из разных предпосылок: так де Соссюр рассматривал ее как науку, изучающую знаковые системы в контексте социальной жизни [де Соссюр, 2004], а Пирс определял ее как логику в самом общем смысле, формальную доктрину знаковых систем [Пирс, 1999]. Однако оба были едины в том, что единицей любой семиотической системы является знак.

Знак – это объект, который представляет или замещает какой-либо другой объект или процесс. Для каждого знака обязательным является наличие двух сторон: означающей (материальной) стороны, воспринимаемой органами чувств, и означаемой стороны – значения, изучаемого в семантике. Так для слов естественного языка означающей стороной является написание слова, а означаемой стороной – само его значение. При этом знаки одной семиотической системы (к примеру, слова русского языка) могут быть означающей стороной для знаков другой семиотической системы (например, математического языка), надстроенной над ними. В целом, установление связи между означающим и означаемым является необходимой предпосылкой любого понимания.

Классификацию знаков в зависимости от вида отношений между означаемым и означающим построили Ч. С. Пирс [Пирс, 1999] и Ч. У. Моррис [Моррис, 1983]. Ими выделены различные типы знаков: изобразительные знаки, указатели и символы. Так знаки, основанные на реальном сходстве между означающим и означаемым, называются *изобразительными (иконическими)* знаками. Действие иконического знака основано на фактическом подобии означающего и означаемого, например, географической карты и местности, ярлыка с изображением принтера и принтера, и т.п. Внутри этой группы Ч. Пирсом различаются образы и диаграммы. У последних отношения между составными частями означающей стороны сходны с отношениями между составными частями означаемой стороны.

Знаки, в которых между означающей и означаемой сторонами устанавливаются отношения смежности, называют *указателями* или *индексами* (например, стрелка-указатель пути к определенному месту). Действие указателей опирается на

ассоциацию по смежности, например, дым есть указатель огня, замедление работы компьютера есть симптом вируса и т.п. Иными словами, в каждый момент времени указатель означает то, на что указывается.

В свою очередь, знаки, в которых отношение между означаемой и означающей сторонами условно, называются *символами*. Здесь связь между означаемым и означающим носит договорный характер: она определяется некоторым (часто неформальным) соглашением.

Превращение иконических знаков в символы соответствует переходу от поверхностного к глубокому пониманию.

Знаки также делятся на собственные и переносные. *Собственные* знаки используют для обозначения вещей, ради которых они изобретены, а *переносные* – тогда, когда сами вещи используют для обозначения чего-либо другого. В этом русле основатель прикладной семиотики Д. А. Поспелов ввел представление о механизме понимания в иерархии знаковых систем [Поспелов, 1970]. На нижнем уровне значениями знаков выступают объекты или предметы реального мира, а к верхнему уровню относятся такие знаки (метазнаки), значениями которых служат знаки семиотической системы нижнего уровня. Путем индукции нетрудно ввести системы знаков любого *k*-го уровня.

Согласно Ч. Пирсу, семиотика предполагает взаимодействие знака, объекта и интерпретатора. Таким образом, семиотический подход предполагает наличие, по крайней мере, трех сторон изучения знаковых систем: синтактики, семантики и прагматики или, иначе, плана выражения, плана содержания и плана значения [Моррис, 1983; Поспелов и др., 1999]. *Синтактика* есть раздел семиотики, который исследует отношения между знаками внутри данной знаковой системы. *Семантика* охватывает сферу отношений между знаками и тем, что они обозначают, а *прагматика* – сферу отношений между знаками и теми, кто ими пользуется. Таким образом, прагматика учитывает требования конкретного адресата сообщения, в то время как у семантики его нет.

Ниже остановимся подробнее на семантических и прагматических аспектах понимания, связанных с исследованием внешних и внутренних отношений в знаках. Семантика как общая теория значения подразделяется на *теорию референции* и *теорию смысла*. В теории референции рассматривают внешнее отношение знаковой формы к объектам реального мира (денотатам), а в основе теории смысла лежит внутренняя связь знаковой формы с соответствующими языковыми значениями и представлениями (смысл или интенционал знака). В первом случае семантическое отношение называют референцией (или десигнацией) знака, а во втором – означиванием (или сигнификацией).

Проблема понимания как соотношения значения и смысла имеет давнюю историю. Значение и смысл – это две ипостаси содержания. Так Г.Г.Шпет [Шпет, 1996] различает значение как указание на содержание выражения и смысл как обозначение предмета в его определенной квалификации.

В соответствии с концепцией значения Г.Фреге [Фреге, 2012], все языковые выражения суть *имена*, т.е. обозначения некоторых внеязыковых объектов (денотатов). Языковое выражение обозначает свой денотат и имеет свой *смысл* – содержание выражения, которое усваивается в процессе его понимания. Денотат и смысл определяют предметное и смысловое значения знака. Смысловая обработка знаков означает конструирование понимающим семантического представления объекта понимания.

Значения знаков можно определять посредством указания их *денотатов* – предметов или ситуаций, которые обозначаются знаками. В общем случае различают денотат и десигнат. Десигнат есть класс мыслимых объектов, к которым применим данный знак, т.е. класс объектов с определенными свойствами. Когда же объект, на который ссылаются, реально существует, он называется денотатом. Класс всех возможных денотатов образует *экстенционал* знака (объем понятия), а его *интенционал* выражает содержание, совокупность мыслимых признаков называемого предмета.

Разные выражения могут иметь один и тот же денотат, но различаться по смыслу. Также выражение может иметь смысл, но не иметь денотата.

Классическим инструментом семиотического анализа понимания является треугольник Фреге (рисунок 2). Здесь определение механизмов понимания сводится к определению связей между именем 1, понятием 2, десигнатом 3 и денотатом 3' (рисунок 2). Соответствующие направленные отношения определяют базовые семиотические операции: 1→2 – описание; 2→1 – обозначение; 1→3 – представление; 3→1 – именование; 2→3 – конкретизация; 3→2 – обобщение.

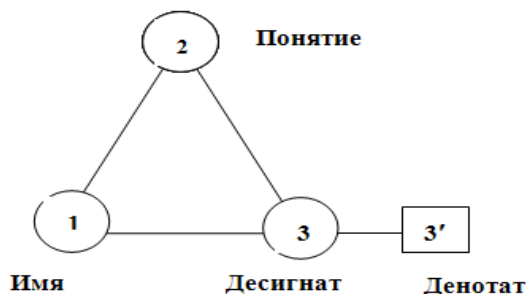


Рисунок 2 – Отношения в треугольнике Фреге

Прагматика – это раздел семиотики, изучающий отношение знаков к их интерпретаторам [Моррис, 1983]. Она охватывает совокупность условий, определяющих использование знака и учитывает влияние контекста на значение и смысл. Здесь слово

«контекст» трактуется в широком смысле как среда коммуникации, обеспечивающая взаимопонимание собеседников. Опора на контекст как фактор взаимопонимания означает пересечение у собеседников их семантических полей и установление определенного уровня абстракции. Потеря контекста приводит к недопониманию, интерпретированию мысли собеседника в ином смысле, нежели тот, который он подразумевает. В настоящее время семиотическая прагматика представляет собой весьма широкую область с нечеткими границами, куда входят такие подобласти как теория речевых актов Остина-Серла, максимы коммуникации Грайса, теория диалоговых сетей и пр. (см. [Тарасов, 2002]).

Согласно Ю.М.Лотману, основными вопросами для любой семиотической системы являются, во-первых, отношение к миру, лежащему за ее пределами, а, во-вторых, отношение статики к динамике [Лотман, 2010]. Динамика семиотической системы наиболее ярко выражена в процессе производства и функционирования знаков, который называется *семиозисом*. Здесь следует вспомнить введенное Ч.С.Пирсом определение семиозиса как постоянного движения знака. Оно лежит в основе общесемиотической концепции Ч.С.Пирса: все входящее в процесс семиозиса становится знаком [Пирс, 1999]).

На наш взгляд, хорошие перспективы в разработке формальных моделей понимания имеют подходы *алгебраической семиотики*, предложенные в работах Дж.Гогена [Goguen, 1999] и В.В. Мартынова [Мартынов, 2009].

Как известно, абстрактные алгебры занимаются изучением свойств алгебраических операций, заданных на множествах произвольной природы, а также алгебраическими структурами (множествами, вместе с алгебраическими операциями, которые удовлетворяют определенным свойствам). Большая часть исследований по абстрактной алгебре посвящены, так сказать, внутреннему устройству алгебраических структур. Однако сегодня активно распространяется и «внешнее» рассмотрение алгебраических структур, когда математики стремятся установить взаимосвязи между отдельными типами алгебраических структур, не определяя операции или элементы структур [Фрид, 1979]. Для этого используются элементы теории категорий, где определяют совокупности объектов и отображений (морфизмов) между ними.

В вышеупомянутых работах по алгебраической семиотике применяются эти взаимодополняющие подходы. Так В.В.Мартынов [Мартынов, 2009] дает алгебраическое представление универсального семантического кода УСК-6 на основе композиции элементарных импликаций и отрицаний соответственно, а также аксиом порождения и преобразования. В то же время Дж.Гоген [Goguen, 1999] рассматривает знаковые системы как алгебраические теории, наделенные некоторой дополнительной структурой, вместе с так

называемыми семиотическими морфизмами, т.е. отображениями алгебраических теорий, частично сохраняющими эту структуру.

2.3. Уровни понимания

В дидактике выявлен ряд основных параметров понимания, к которым относятся глубина, отчетливость, полнота, обоснованность [Коробов, 2005]. Следует отметить, что практически все эти параметры характеризуют элементарное понимание текстов и не затрагивают гештальт-механизмы понимания.

Полнота понимания означает максимальное выявление содержания в усваиваемом объеме информации. Ее определяют как отношение понятых человеком концептов и связей между ними

ко всем имеющимся в объекте понимания концептам и связям. Здесь обычно используют трехзначную прагматику понимания: {полное понимание, частичное понимание, полное непонимание}, хорошо известную педагогам и учащимся. С этой прагматикой связана следующая классификация уровней понимания: 1) возможность презентации (знать – не знаю, а рассказать – что-нибудь расскажу), 2) уровень частичного, рабочего понимания (понимаю все основное, но в общих чертах); 3) уровень полного понимания (понимаю все в мельчайших деталях).

Отчетливость понимания есть степень осмысления атрибутов, связей, ограничений в объекте понимания. Здесь можно взять за основу четырехзначную прагматику: {критическое понимание, интерпретирующее понимание, буквальное понимание, смутное понимание}. Как смутное, так и буквальное понимание текста не обеспечивают его полноценного перевода или адекватной интерпретации. На следующем уровне интерпретирующего понимания возможны: определение причинно-следственных связей, осуществление сравнений, выводов и обобщений, идентификация целей автора. Наконец, на уровне критического понимания производится оценка правдоподобия или качества информации, содержащейся в тексте. В случае перевода с одного языка на другой переводчик, способный работать на уровне критического понимания, может даже улучшить исходный текст.

В свое время, Л.Витгенштейн, чувствуя недостаточность понятия объяснения для развития понимания, ввел также понятия «обоснования» и «оправдания» [Витгенштейн, 2009]. Так утверждение считается оправданным, если оно не может быть отброшенным без того, чтобы не были затронуты другие важные утверждения. Оправдание связано с установлением ценностного отношения между утверждениями некоторой теории и возможно только внутри определенной «познавательной практики». Оправдание предполагает иерархию: в нем участвуют утверждения разных уровней,

причем оправдываемое утверждение относится к более низкому уровню.

В свою очередь, обоснование есть приведение комплекса аргументов или доводов, в силу которых следует принять некоторое утверждение или концепцию. Как и оправдание, оно предполагает проведение ряда мыслительных действий, касающихся не только рассматриваемой гипотезы, но и всей системы утверждений, элементом которой она является. Тогда *обоснованность понимания* – это наличие системы оснований, которые обуславливают уверенность в правильности понимания. Вариантом такого обоснования является непротиворечивость (паранепротиворечивость) системы утверждений, куда входит обосновываемое положение.

В книге [Знаков, 1999] выделены три уровня понимания по характеру мыслительных действий с содержанием понимаемого: 1) понимание-узнавание; 2) понимание-гипотеза, которую надо обосновать или оправдать; 3) понимание-соединение, когда необходимо собрать отдельные фрагменты понимаемого в единое целое.

Глубина понимания характеризуется степенью проникновения в сущность текста, явления или поведения. Глубину понимания обычно связывают с усвоением законов, принципов, правил, метафор, афоризмов, т.е. с тем, что может иметь глубокий смысл.

Несомненный интерес представляет анализ переходов в иерархической структуре DIKW «данные–информация–знания–метазнания» (Data–Information–Knowledge–Wisdom), представленной в координатах «понимание – деятельность» [Askoff, 1989]. Реализация этих переходов в когнитивных системах предполагает наличие многоуровневых механизмов понимания (рисунок 3): а) понимание как выявление связей между объектами или событиями, б) понимание как поиск и обоснование образцов для выработки нормативного или ситуативного поведения; в) понимание как охват и интерпретация ситуации в целом. При этом понимание как связей, так и правил и образцов обеспечивает формирование целенаправленных действий, а понимание ситуации в целом необходимо для организации деятельности.

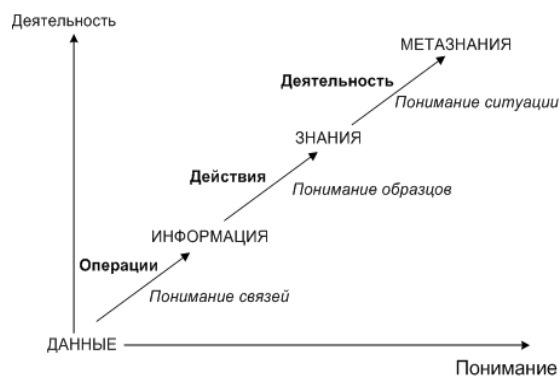


Рисунок 3 –Схема переходов «данные – информация – знания – метазнания» в координатах «понимание – деятельность»

В работе [Поспелов, 1989] выделены шесть уровней компьютерного понимания текстов. Следуя Д.А.Поспелову, мы (совместно с А.П.Калуцкой) построили близкую классификацию уровней понимания для когнитивных роботов.

Когнитивная робототехника есть важная прикладная область современного искусственного интеллекта. Ее задачи связаны с моделированием информационного взаимодействия автономных роботов с плохо определенной (и, чаще всего, экстремальной) внешней средой. Она изучает, как робот получает информацию о внешнем мире, в какой форме он ее представляет, как она хранится в памяти и преобразуется в мнения и знания, и как эти знания влияют на поведение робота. В центре внимания находятся задачи интеллектуального анализа данных и интеграции (комплексирования) информации, снимаемой с различных датчиков, построения системы целостного восприятия информации и формирования знаний об окружающей среде и самом роботе.

Когнитивные роботы обладают хорошо развитой и пополняемой символьной моделью внешнего мира, что достигается благодаря наличию у них, по крайней мере, трех различных источников информации: собственной базы знаний, системы датчиков, и интерфейса с человеком-пользователем на ограниченном естественном языке. При этом важнейшим требованием к такому роботу является обеспечение некоторого уровня понимания ситуации, в которой он находится, что требуется для успешного осуществления интерактивного диалогового управления роботом. Более подробно когнитивные науки будут представлены в разделе 3.

На первом уровне понимания робот, имеющий свою базу знаний, способен отвечать на все вопросы, содержащиеся в полученном им тексте (сообщении), и выполнять свою задачу.

Второй уровень понимания отличается от первого тем, что робот наряду с базой знаний использует также средства рассуждения и планирования действий, т.е., по сути, имеет встроенную экспертную систему.

На третьем уровне понимания сведения из базы знаний и предварительные планы действий дополняются обработкой сенсорной информации, позволяющей роботу соотнести их с реальной текущей ситуацией.

На четвертом уровне понимания в условиях интерактивного диалога с пользователем у робота формируются нормативные модели поведения, типовые сценарии процессов и программы действий, характерных для предметной области

Наконец, верхний пятый уровень понимания подразумевает наличие у робота знаний о человеке или роботе– партнере по взаимодействию, а также пополняемой информации о среде и процессах коммуникации (диалога).

2.4 Механизмы понимания: перцептивные и мыслительные операции

Для осуществления алгебраического подхода к моделированию механизмов понимания надо прежде всего определить состав перцептивных и мыслительных операций, связанных с пониманием. Здесь уместно обратиться к трудам отечественных и зарубежных классиков в области психологии восприятия (И.М.Сеченов, Ф.Кликс, Ж.Пиаже, Ю.М.Забродин, и др.) и психологии мышления (С.Л.Рубинштейн, А.В.Брушлинский, О.К.Тихомиров, и др.).

Предварительно заметим, что понятия операции в абстрактной алгебре значительно отличается от операции в психологии. Так алгебраическая операция на множестве X есть функция $\alpha: X^n \rightarrow X$, где n -арность операции. В приложениях алгебр, чаще всего, используют унарные ($n=1$) и бинарные ($n=2$) операции.

В абстрактной алгебре важную роль играют однотипные структуры. В однотипных алгебрах задана одна и та же система операций. Так, например, кольцо и решетку можно считать однотипными алгебраическими структурами, если в кольце рассматривать в качестве пары базовых операций сложение и умножения, а в решетке – объединение и пересечение. В самом деле, и кольцо, и решетку можно рассмотреть как базовое множество вместе с двумя бинарными операциями. Примером применения абстрактных алгебр в психологии служат группировки Ж.Пиаже [Пиаже, 1994].

В психологии же различают *действия* как единицы деятельности, направленные на достижение определенной цели, и *операции*, т.е. способы выполнения действий, определяемые ситуативно. При этом можно говорить об иерархии мыслительных и перцептивных операций.

В психологии восприятия выделяют четыре типа перцептивных операций или четыре уровня перцептивного действия: *обнаружение*, *различение*, *идентификация* и *опознание*. Первые два относятся к чисто перцептивным, а последние два – к опознавательным действиям.

Обнаружение – исходная фаза развития любого сенсорного процесса, когда субъект просто определяет, есть ли внешний стимул.

Базовая операция восприятия – это *различение*, или, собственно, восприятие. Ее конечный результат состоит в формировании перцептивного образа-эталона. При этом развитие перцептивного действия идет по линии выделения специфического сенсорного содержания в соответствии с особенностями предъявляемого материала и стоящей перед субъектом задачи.

Когда перцептивный образ сформирован, становится возможным опознавательное действие. Опознание включает сличение (сопоставление с образцом) и идентификацию. *Идентификация* есть

отождествление непосредственно воспринимаемого объекта с образом, хранящимся в памяти, или сопоставление одновременно воспринимаемых объектов.

Опознание включает также категоризацию (отнесение объекта к определённому классу объектов, воспринимавшихся ранее) и извлечение соответствующего эталона из памяти.

К числу основных *мыслительных операций*, связанных с пониманием, относятся: сравнение, анализ, синтез, абстрагирование, конкретизация, обобщение, ограничение [Рубинштейн, 1989], [Шадриков, 2006].

Мыслительная операция *сравнения* заключается в сопоставлении предметов и явлений, их свойств и отношений друг с другом с целью выявления сходства или различия между ними. На основе сравнения осуществляются *классификация* и *систематизация*.

По мнению К.Д.Ушинского сравнение есть основа всякого понимания и мышления. Для ясного понимания, какого-нибудь предмета следует найти его отличие от самых сходных с ним предметов и сходство с наиболее отдаленными от него объектами. Только тогда удастся выяснить существенные признаки предмета, т.е. понять его (см.[Рубинштейн, 1989]).

В качестве других первичных мыслительных операций, обеспечивающих исходное, частичное, поверхностное понимание предмета, выступают анализ и ограничение.

Анализ как расчленение целого на составные части позволяет разбить исходную задачу на подзадачи и получить достаточно ясное понимание выделенной подзадачи и частичное понимание общей задачи. В свою очередь, операция *ограничения*, связана с движением от родового к видовому понятию. Происходит переход от понятия с большим объемом к понятию с меньшим объемом, но большим содержанием (например, от понятия «понимание» к понятию «понимание поведения»). Здесь примерами формальных методов служат разбиения объектов на классы (классификации), индуцируемые отношениями эквивалентности, и варианты кластеризации на основе расстояний в пространствах признаков.

В то же время, такие операции как синтез, абстрагирование, обобщение способствуют более полному и глубокому пониманию.

Синтез как мыслительная операция соединения различных частей объекта опирается на отношение сходства и лежит в основе понимания-соединения.

Обобщение –это операция перехода от родового к родовому понятию, т.е. перехода от понятия с меньшим объемом к понятию с большим объемом за счет уменьшения содержания исходного понятия (например, переход от понятия «интерпретация текста» к понятию «интерпретация»). Речь идет о

мысленном объединении предметов или явлений по какому-либо признаку (системе признаков), например, по признаку «операционная структура»).

В частности, обобщение как распространение наблюдаемого факта, явления на другие явления, служит базой для интерпретации. Обобщение может осуществляться с помощью операций группирования.

Наконец, особо важную роль для развития понимания играет операция *абстрагирования*—одна из основных интеллектуальных операций, состоящая в мысленном выделении, вычлениении каких-либо важных свойств или признаков изучаемых объектов и отношений между ними и отвлечении от ряда других, второстепенных свойств и отношений. Происходит отсечение единичного, случайного или несущественного для дальнейших шагов работы. Термин *абстракция* означает результат абстрагирования.

Целями операции абстрагирования в плане обеспечения понимания, является поиск общих свойств, отождествление объектов, создание образцов и норм. Эта операция играет важную роль в процессе понимания условного смысла, метафор, пословиц и поговорок. Для того, чтобы понять этот смысл, необходимо отвлечься от той конкретной ситуации, которая в них описывается.

Операция *конкретизации* является обратной по отношению к абстрагированию. Конкретизация – это описание деталей, представление чего-то единичного, соответствующего данному понятию. Она всегда выступает как пример или как иллюстрация общего, связывая его с чувственным опытом и способствуя лучшему пониманию. Конкретизация предполагает сжатие информации об изучаемом объекте.

Эти две операции связаны с изменением требований к точности информации и знаний, обеспечивающих процесс понимания. Если абстрагирование предполагает в некотором смысле удаление от изучаемого объекта, рассмотрение его с определенного расстояния, укрупненно, то конкретизация, напротив, означает приближение к исследуемому объекту и выделение его мельчайших деталей.

Абстрагирование служит для построения самых общих и глубоких интерпретаций, тогда как конкретизация используется для понимания текущих ситуаций на базе ранее сформированных общих понятий.

На наш взгляд, решение проблемы понимания на различных уровнях обобщенности или абстрактности тесно связано с развитием теории грануляции информации и концепции гранулярных вычислений [Zadeh, 1997], [Bargiela et al., 2003], [Тарасов, 2014].].

Следует отметить, что в конце 1990-х годов Л.Заде в очередной основополагающей статье «На

пути к теории нечеткой грануляции информации...» определил понятие *гранулы* и заложил основы теории нечеткой информационной грануляции [Zadeh, 1997], а работавший тогда у него в Беркли Т.Лин ввел термин «гранулярные вычисления» и предложил оригинальный подход к построению гранулярных вычислений на основе систем окрестностей [Lin, 1997].

Термин «гранула» происходит от латинского слова *granum*, означающего «зерно». У Л.Заде гранула определяется формально как группа объектов, объединяемых отношениями сходства, близости, неразличимости или функциональности, т.е. отношениями, имеющими, по крайней мере, свойства рефлексивности и симметричности [Zadeh, 1997]. Смысл термина «гранулярный» (в частности, интервальный) легко пояснить путем его противопоставления слову «сингулярный» (т.е. одноэлементный, одноточечный). Грануляция как способность представления и оперирования информацией и знаниями на различных уровнях детализации представляет собой одно из ключевых свойств понимающих систем. Главными объектами теории грануляции выступают гранулы, уровни, иерархии, гранулярные структуры и соотношения между ними (эти соотношения и отображения между гранулярными структурами выражены стрелками на рисунке 4).

В настоящее время *гранулярные вычисления* позиционируются как новая концептуальная и вычислительная парадигма обработки и понимания информации, а также извлечения и обнаружения знаний. В широком смысле этот термин служит для обозначения различных методологий, теорий, методов, моделей и инструментальных средств, использующих гранулы при решении сложных задач. В них широко используются такие формальные объекты как подмножества, разбиения, покрытия, кластеры, интервалы. Речь идет о построении эффективных моделей вычислений для сложных прикладных областей, в которых имеются большие объемы данных, информации и знаний.

Согласно Ю.Яо [Yao, 2008], гранулярные вычисления можно рассматривать с трех сторон как: 1) философию человеческого мышления; 2) методологию понимания и решения задач; 3) набор антропоморфных методов компьютерного анализа информации.

Типичные содержательные интерпретации гранул – это часть целого, подзадача задачи, ограничение на значение переменной. Отсюда видно, что понятие «гранула» отражает уровень абстрагирования или уровень обобщения в процессе понимания. По сути, теория грануляции информации, направленная на построение иерархии концептуальных гранул и гранулярных структур, с ее операциями перехода от мелкозернистых гранул к крупнозернистым, и обратно (см. вертикальные линии со стрелками на рисунке 4), покрывает почти все ключевые перцептивные и мыслительные операции, рассмотренные выше. Здесь

крупнозернистые структуры соответствуют более высокому уровню абстрагирования и большей глубине понимания, а мелкозернистые структуры отвечают конкретизации и характеризуют более поверхностное, но и более точное понимание.

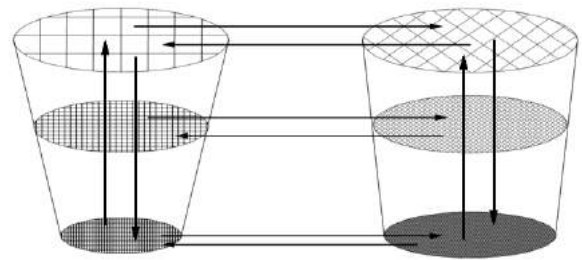


Рисунок 4 – Примеры трехуровневых гранулярных структур и соотношений между ними

3. Гибридные когнитивные микронауки и понимание

В разделах 1.1-1.4 был проведен системный, междисциплинарный анализ феномена понимания, его принципов и механизмов, который опирался на идеи и результаты герменевтики, семиотики, дидактики, психологии мышления. Были также очерчены наиболее абстрактные, алгебраические подходы к моделированию понимания. В то же время понимание есть одна из главных когнитивных способностей человека, и это требует привлечения целого ряда гибридных когнитивных микронаук.

Прежде, чем построить и представить такую когнитивистскую систему, в разделе 3.1. мы совершим краткий экскурс в историю ментализма и когнитивизма.

3.1. От ментализма к когнитивизму: основания когнитивных микронаук

Ментализм – это научная доктрина, согласно которой адекватная характеристика человеческого поведения и самой жизни невозможна без привлечения сознания или психики в целом как «средств объяснения». По сути, ментализм как философия сознания близок к *психологизму*, так как его адепты утверждают, что любые попытки объяснить психические свойства, состояния и процессы путем учета только физических и физиологических аспектов функционирования (как например, в *бихевиоризме*) являются признаком редукционизма, т.е. недопустимым упрощения существа дела. В соответствии с познавательными установками ментализма, для понимания человеческих действий и поступков требуется обращение к ментальным (т.е. описывающим ненаблюдаемый внутренний мир) категориям. Следует отметить, что в отличие от герменевтики, традиционная семиотика, появившаяся в эпоху логицизма и бихевиоризма, оказывается просто пронизанной антиментализмом [Daddesio, 1994].

Одним из современных вариантов ментализма является *когнитивизм*, развивающий трактовку

интеллекта как познания. Истоки когнитивных течений восходят к пониманию роли знаний как ведущего фактора, определяющего эффективность действий человека. Когнитивистский подход в любой области подразумевает исследование жизненного цикла знаний.

В то же время когнитивистский взгляд на психику предполагает выделение разнородных когнитивных единиц – когнитонов, характеризующих познание разных сторон психики – собственно ее когнитивных механизмов, а также отражение, компонентов организационной, коммуникативной, интенциональной подсистем. Когнитон – это общий термин для когнитивного описания поведения агента. Он был введен с целью единого представления семейств ментальных единиц, которые лежат в основе динамических структур человеческого интеллекта. Возникает проблема инженерии когнитонов как основы разработки когнитивных агентов, а также организации их взаимодействия и обеспечения взаимопонимания (подобно тому, как инженерия знаний лежит в основе разработки классических интеллектуальных систем). Инженерия когнитонов расширяет обычные подходы инженерии знаний, видоизменяя и адаптируя их к проблемам формализации интенциональных свойств агентов (желаний, намерений, обязательств и т.д.).

В книге [Тарасов,2002] описаны когнитоны представления, конативные когнитоны, когнитоны взаимодействия, организационные когнитоны. В число когнитонов представления входят знания, суждения, оценки (как логические, так и перцептивные), мнения. В состав конативных (интенциональных) когнитонов включены цели, предпочтения, стремления, требования, желания, намерения. К когнитонам взаимодействия относят запросы и ответы, сведения и решения, аргументы и контраргументы. Организационные когнитоны охватывают роли, задачи, нормы, соглашения, обязательства.

3.2. О системах когнитивных наук

В знаменитом отчете Converging Technologies for Improving Human Performance, подготовленном в 2002 г. М.Роко и У.Бейнбриджем во Всемирном центре оценки технологий (WTEC), был введен термин NBIC-конвергенция, означающий построение нового технологического уклада в результате схождения нано-, био-, инфо- и когно-технологий (NBIC по первым буквам областей: N -нано; B -био; I-инфо; C -когно). Это объясняет растущий интерес к современной когнитивной науке, обеспечивающей теоретическую базу для ускоренной реализации когнитивных технологий – пока еще наименее развитого компонента NBIC.

Сегодня когнитивная наука (когнитивистика) есть обширная междисциплинарная область, занимающаяся исследованием процессов познания и когнитивных систем. Она стремится ответить на вопросы, что такое познание и как оно работает,

каковы общие принципы организации и механизмы функционирования мозга. Когнитивистика как синергетическое научное направление объединяет ряд междисциплинарных и гибридных научных областей: обычно в ее состав включают (по Х. Гарднеру) следующие шесть компонентов: теорию познания (эпистемологию), нейрофизиологию (и, вообще, комплекс нейронаук), антропологию, когнитивную и экспериментальную психологию, структурную и когнитивную лингвистику, информатику и искусственный интеллект [Gardner, 1987], [Величковский, 2006] (рисунок 5). Позже в состав когнитивной науки стали также включать седьмой компонент – образование (дидактику).

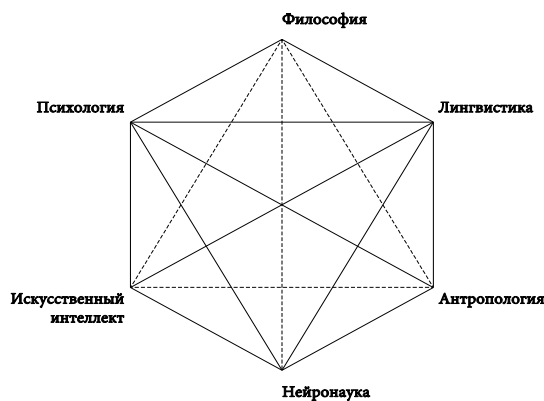


Рисунок 5 – Шестигранник Х.Гарднера: каноническая структура когнитивной науки

На наш взгляд, эта структура когнитивной науки отнюдь не универсальна. В общем случае, ее состав зависит от целей исследования и решаемых задач. В данной статье междисциплинарный подход к пониманию опирается на когнитивный базис, изображенный ниже на рисунке 6. Далее приведем краткое описание ряда гибридных когнитивных микронаук, включенных в состав этого базиса.



Рисунок 6 – Вариант системы когнитивных наук для междисциплинарного исследования понимания

3.3. Когнитивная психология

Когнитивная психология – одна из наиболее старых и развитых когнитивных микронаук, возникшая еще в середине XX-го века. Предмет когнитивной психологии составляют модели познавательных процессов, т.е. модели памяти,

внимания, восприятия, представления, мышления, понимания [Величковский, 2006], [Психология, 1990].

На начальном этапе своего развития когнитивная психология основывалась на «компьютерной метафоре» – аналогии между преобразованием информации в компьютере и познавательными процессами у человека. Возникли информационные теории мышления и интеллекта (см. [Тихомиров, 1984]). По У. Найссеру термин «познание» имеет отношение к любым процессам преобразования, хранения, обработки, сжатия, восстановления, применения информации, ранее поступившей на сенсорный вход.

Одна из важных задач когнитивной психологии заключается в демонстрации решающей роли знаний и механизмов познания в самоорганизации поведения человека. Здесь главным положением является тезис о неразрывной связи сознания и поведения: любая единица поведения, любой поступок регулируются сознанием.

Центральное место в когнитивной психологии занимает вопрос об организации знаний в памяти человека, в том числе, о соотношении вербальных и образных компонентов в процессах запоминания и мышления.

Понимание как ассимиляция рассматриваемого понятия связано с отношением между данным понятием и множеством объектов, атрибутов и значений в долговременной памяти.

В когнитивной психологии понимание включает построение внутреннего представления. Понимание природы агентом зависит от степени развитости у него внутренней модели внешней среды, а понимание поведения собеседника – от наличия и развитости внутренней модели этого собеседника.

3.4. Когнитивная лингвистика

Как это видно из самого названия «*когнитивная лингвистика*», данная научная область постулирует неразрывную связь между *знанием* и языком. При этом язык понимается и как единый механизм познания, и как проявление общей способности человека познавать мир, других людей и самого себя. В когнитивной лингвистике отбрасывается традиционное разделение лингвистики на фонологию, синтаксис, морфологию и пр. Здесь в центре внимания оказывается *когнитивная семантика* [Кузнецов, 2012], [Лакофф, 2004] [Talmy, 2000], связанная с конструированием смысла (концептуализацией) и представлением знаний как концептуальных структур. В ней исследуются соотношения между опытом, познанием и языком. В частности, значение изучается как когнитивный феномен. Ключевое положение теории значения Дж. Лакоффа состоит в том, что значение всегда включает интенциональность и человеческое понимание. Базовым для когнитивной лингвистики является интерпретирующий подход, что сближает ее с герменевтикой.

В результате когнитивной деятельности создается система смыслов, определяющая, что субъект знает и думает о мире («образ мира» у А.Н. Леонтьева). Весьма активно используемая лингвистами внеязыковая категория пресуппозиции оказывается напрямую связанной с моделью мира.

По сути, в когнитивной лингвистике язык рассматривается как средство быстрого доступа к познавательным процессам. При этом изучаются связи отдельных когнитивных процессов и способностей человека с языком и формы их взаимодействия.

Таким образом, одними из главных категорий когнитивной лингвистики являются категории «знания» и «понимание»; при этом исследуются как виды знаний, так и способы их языкового представления. Специалисты по когнитивной лингвистике стремятся понять, как протекают процессы восприятия, классификации, узнавания, осмысления, категоризации, абстрагирования, как происходит формирование и накопление знаний. Одним из главных инструментов познания является языковая коммуникация, связанная с обменом знаниями и обеспечением взаимопонимания.

В последние годы в русле когнитивной лингвистики также развивается «*когнитивная прагматика*» [Schmid, 2012], связанная с изучением роли и особенностей развертывания ментальных процессов в человеческой коммуникации. Если трактовать прагматику как исследование значения в некотором контексте, то когнитивная прагматика охватывает изучение когнитивных принципов и процессов, участвующих в определении контекстно-зависимого значения. Обратим особое внимание на то, что этапы становления и модели языка связаны с развитием его прагматики.

Следует отметить, что при описании процессов коммуникации в русле когнитивной прагматики центральное место занимают понятия «мнение», «обмен мнениями» и «разделяемое мнение», а также соответствующие формальные модели.

3.5. Когнитивная семиотика

Когнитивная семиотика представляет собой сравнительно научную парадигму, сочетающую знаковые и когнитивные структуры. Ее зарождение и развитие демонстрирует, что познание отнюдь не является антитезой теории знаков.

Некоторые авторы, например, К. Ойхлер, Ю.Р. Валькман, утверждают, что уже в трудах родоначальника семиотики Ч.С. Пирса (см. [Пирс, 1999]) содержатся зачатки когнитивной семиотики. В самом деле, Ч.С. Пирс допускал возможность взаимодействия между познанием и семиозисом; более того, сам семиозис включает у него две одинаково важные части: производство знаков и их интерпретацию.

В свою очередь, Ч.У. Моррис предложил свою классификацию знаков, непосредственно связанную

с типологией знаний (см. [Моррис, 1983]). Так он различал: а) знаки-идентификаторы (т.е. те знаки, которые выражают "где-знания?"); б) знаки-десигнаторы (знаки, отвечающие на вопрос "что такое?"); в) оценочные знаки, связанные с предпочтением, которые характеризуют "почему-знания?"; г) прескриптивные знаки, соответствующие "как-знаниям?"; д) формирующие знаки, или знаки систематизации (направляющие поведение интерпретатора в отношении других знаков).

Тем не менее, базовые принципы и подходы классической семиотики в явном виде не обращены к теории познания, и идеи опосредования знаков ментальными представлениями стали активно распространяться лишь в последней четверти XX-го века.

Когнитивная семиотика в нашей стране имеет глубокие корни; ее зарождение связано со школой семиотического моделирования и прикладной семиотики Д.А.Поспелова, Прикладная семиотика направлена на использование знаковых систем при разработке новых компьютерных, в том числе интеллектуальных, технологий. В русле интеграции семиотики и инженерии знаний было предложено опираться на знания как объекты знаковой природы. Стали появляться семиотические базы знаний, семиотические решатели, семиотические системы управления сложными объектами [Поспелов и др., 1999]. Все это можно рассматривать как интеграцию семиотики, когнитологии и формальных систем.

Еще в 1970г. Д.А.Поспелов ввел понятие «метазнака» [Поспелов, 1970]. Он писал, что кроме знаков, значениями которых выступают предметы или явления реального мира, можно рассматривать знаки знаков (метазнаки), значениями которых служат знаки семиотической системы первого уровня. Такую систему следует отнести к знаковой системе второго уровня. Путем индукции нетрудно ввести системы знаков любого k -го уровня. Введение метуровня в знаковые представления предлагает явным образом в них внутреннюю интерпретируемость действий.

Метазнак в отличие от обычного знака несет в себе «заряд активности» В треугольнике Фреге появляется дополнительная вершина, с которой связан фрагмент некоторой структуры на множестве знаков, которая играет роль денотата метазнака. Соответственно, возникает новая базовая семиотическая модель, которую Г.С.Осипов предложил называть «квадратом Поспелова» [Осипов, 2002]

Зарубежные работы по когнитивной семиотике появились существенно позже публикаций Д.А.Поспелова по семиотическому моделированию. Одним из основателей когнитивной семиотики считается Т.Даддезио, написавший книгу «О разуме и символах: релевантность когнитивных наук семиотике» [Daddesio, 1994]; среди других

часто цитируемых авторов выделим Й.Златева [Zlatev, 2012], Ж.П.Менье, П.Кейе.

В русскоязычной литературе проблемы когнитивной семиотики рассмотрены в монографии В.В.Мартынова «В центре сознания человека» [Мартынов, 2009] и статьях Ю.Р.Валькмана [Валькман, 2012], [Валькман, 2013]. В частности, Ю.Р.Валькман считает, что проблема «знак-значение-понимание-смысл» связана с одной стороны с семиотикой, а с другой стороны – с когнитологией, так как идет о понимании и познании.

Появление когнитивной семиотики инициирует движение когнитивных наук и семиотики встречным курсом. С одной стороны, когнитивная семиотика есть наука о получении знаний из знаковых структур, а с другой стороны, речь идет о синтезе знаковых конструкций, представляющих соответствующие структуры знаний [Валькман, 2012]. Типичным способом получения процедурных знаний является анализ связей в треугольнике Фреге или квадрате Поспелова.

В когнитивной семиотике центральное место занимает поиск и определение соответствия между знаковыми структурами и структурами знаний. Здесь показательным примером служит анализ соответствия знаков и фреймов и введенник в связи с этим структуры знака-фрейма в [Поспелов и др., 1999].

По нашему мнению, определение когнитивной семиотики следует расширить и не ограничиваться только уровнем знаний. С одной стороны, ее можно интерпретировать как науку о развитии моделей и методов познания и конструирования смысла на основе знаковых структур. С другой стороны, она занимается проблемами семиозиса, формирования семиотических отношений для когнитивных процессов и систем любого уровня (восприятие, представление, мышление).

Проиллюстрируем эту мысль на примере *когнитивного семиозиса* – процесса порождения и трансформации знаний, который определяет их синтаксис (представление в структурном мире), семантику (представление в ментальном мире) и прагматику (представление в реальном мире). Семиозис разворачивается внутри контура циркуляции знаний (рисунок 7), который типичен для целенаправленной системы где синтаксис характеризует структуру знаний, семантика – их смысл, содержание, а прагматика – их реальное воплощение в действиях или продуктах [Meystel, 1995].

Рецепторы обеспечивают входную информацию из внешнего мира, которая поступает в систему восприятия, где начинается процесс ее обработки и представления. Синтактика продуцирует начальные структуры знаний, которые затем сохраняются, и начинается процесс развертывания его смысла. Этот смысл вместе с целью агента определяют

стадию семантизации знаний. Смысл подключает процесс (программы) формирования поведения, в результате чего появляются программы действий. Действия приводят к изменениям внешней среды, которые становятся частью прагматики знаний. Эти изменения регистрируются рецепторами, и цикл повторяется.



Рисунок 7 – Общая схема когнитивного семиозиса

Каждый блок семиозиса на рисунке 7 можно трактовать как конечный автомат, который осуществляет переход от одной системы символов (знаков) к другой:

- из словаря Среда в словарь Восприятия;
- из сенсорного словаря Рецепторов в Словарь Базы Знаний;
- из словаря Восприятия в словарь Порождения Действий;
- из словаря Базы Знаний в словарь Эффекторов;
- из словаря Формирования Поведения в словарь Среда;
- из словаря Действий в словарь Рецепторов.

В каждом блоке должна обеспечиваться эффективность семиозиса с помощью специальных процедур обработки (сжатия) символьной информации. Сжатие информации обеспечивается тремя операциями: группирование, фокусировка и комбинаторный поиск. В результате такого сжатия группы символов, которые характеризуются свойством общности, заменяются единичными символами.

3.6. Когнитивная информатика

Когнитивная информатика – это открытый междисциплинарный научно-практический комплекс, развивающийся на стыке когнитивных наук и новых компьютерных технологий. Его главная цель состоит в исследовании и моделировании набора когнитивных процессов и способностей человека в интересах создания и практического внедрения *когнитивных компьютеров*, а на их основе, искусственных когнитивных систем, в которых будут аппаратно и

программно реализованы искусственные или имитированы естественные когнитивные процессы. Важная задача когнитивной информатики заключается в обеспечении программно-аппаратной поддержки процессов интеллектуального анализа текстовых и сенсорных данных, обнаружения знаний, машинного обучения, синтеза интеллектуальных сред и создания антропоцентрических интерфейсов.

Антропоморфное содержание когнитивной информатики связано с поддержкой и усилением человеческих способностей познания, обучения, понимания на базе когнитивно-информационных (СИ-) технологий нового поколения (с искусственными аппаратно-программно реализуемыми когнитивными структурами) обеспечивающих дружественный интерфейс для взаимопонимания и сотрудничества человека и искусственного агента.

3.7. Прагматические логики

На первый взгляд, включение многозначных логик в состав микронаук, обеспечивающих совместное междисциплинарное исследование понимания, выглядит странным. Однако анализ оснований прагматических логик позволяет сделать вывод об их полном соответствии трактовке понимания как оценки, основанной на некоторой норме, а также о тесных связях между с одной стороны прагматическими логиками и методами семиотики, герменевтики, когнитивной прагматики с другой.

В свое время, Л.Годдард и Р.Раутли [Goddard et al.,1973] высказали важную мысль о том, что в естественном языке разные семантические оценки высказываний зависят от контекста употребления, и поэтому многозначные логики могут служить в качестве полезной аппроксимации логической структуры языка. В этом плане ими была введена «логика значения и контекста» (Logic of Significance and Context), где промежуточные истинностные значения, отличные от истины и лжи, понимались либо как неполнота информации, либо как ее незначимость. В дальнейшем этот аппарат был развит в работах В.К.Финна и др. (см.[Финн, 2008]), где на базе алгебраического подхода были введены типы истинностных значений и предложена классификация логик значения.

Также в середине 1970-х годов в работах К.Айдукевича, Б.Н.Пятницына и др. появился термин «прагматические логики» (см.[Будбаева и др.,1974]). Если прагматическая логика у К. Айдукевича представляет собой скорее логическую методологию обучения, то А.А.Ивин [Ивин,1997а] Б.Н.Пятницын и другие ученые развили широкую трактовку прагматических логик как прикладных логических систем, учитывающих, в первую очередь, прагматический аспект исследований.

Классическими примерами таких логик служат индуктивная и вероятностная логики; к ним же

относятся логики оценок, логики норм, логики предпочтений, логики принятия решений, логики общения, иллюкативные логики и пр.

Далее будем использовать два близких термина: «прагматическая логика» и «логическая прагматика».

В основе прагматических логик находится *прагматическая концепция истинности* Ч.С.Пирса, согласно которой истинность знания определяется его полезностью при решении конкретных задач.

Прагматическая логика – это логическая система, опирающаяся на ценности (полезность), оценки, нормы. В свою очередь, различные логические прагматики связаны с нестандартными прагматическими оценками и интерпретациями истины, например, «измеренная истина» = норма в случае логического анализа результатов измерений, «согласованная истина» в логических моделях переговоров, и пр. При построении многозначных прагматических логик можно взять за основу подход Данна-Белнапа и его расширения.

Оценкой (оценочным суждением) называется высказывание, которое устанавливает абсолютную или относительную (сравнительную) ценность некоторых объектов. Оценки выражают с помощью аксиологических модальностей: относительных («лучше», «хуже», «равноценно») или абсолютных («отлично», «хорошо», «удовлетворительно», «плохо») [Ивин, 1997а]. Любая оценка может быть задана в виде фрейма-прототипа {Оценка, (Кто(агент оценки),), (Что (Кого) (объект оценки),...), (Какая (тип оценки),...), (Почему (основание оценки,...))}.

Уровень понимания агента стоящей перед ним задачи может определяться оценкой *результатов* его действий, причем эти действия должны не противоречить нормам поведения агентов.

Нормы – это различные социальные запреты и ограничения, накладываемые сообществом на отдельного агента. Нормативное высказывание служит для установления некоторой нормы поведения. С одной стороны, нормы есть частный случай оценок: их можно рассматривать как общественно апробированные и закрепленные оценки. Средством, превращающим оценку в норму, является угроза наказания; стандартизация норм осуществляется с помощью санкций. Так еще К.Менгер указывал на прямую связь между предписанием n и санкцией: «обязательно n , а если не n , то наказание или ухудшение». С другой стороны, формирование норм предполагает согласование разных мнений по этим нормам.

Тогда общая алгебраическая модель понимания (Comprehension) может выражаться четверкой

$$COM = \langle N, n, e, j \rangle$$

где N – множество норм, $n \in N$ – выбранная норма, e – стандарт (модальность) выбранной нормы n , j –

операция обоснования приложимости стандарта e к конкретной ситуации.

Теперь возьмем задачу понимания конкретного действия. Формально норму как предписание к действию можно выразить четверкой

$$N = \langle A, act, M, W \rangle$$

где A – множество агентов, которым адресована норма; $act \in ACT$ – действие, определенное на множестве действий ACT и являющееся объектом нормативной регуляции (содержание нормы), W – множество миров, в которых применима норма (условия приложения, обстоятельства, в которых должно или не должно выполняться действие); M – множество систем модальностей, связанных с действием, например, система норм, выраженных деонтическими модальностями: $M_N = \{O, P, B, Z\}$, O – «обязательно», P – «разрешено», B – «безразлично», Z – «запрещено».

В общем случае, осуществление логического подхода к пониманию соответствует выводу из принятых общих ценностей. Примером служит так называемый практический силлогизм:

- Агент a желает получить поощрение b .
- Для получения b надо совершить действие act .
- Следовательно a должен совершить действие act

Более интересный и показательный пример логической формализации понимания – это так называемый интенционально обусловленный вывод по Г.Х. фон Вригту [фон Вригт, 1986]. Здесь из оценки, устанавливающей намерение (цель) агента a , и описательного утверждения о причинно-следственной связи, выводится утверждение о том, что должен сделать агент, т.е. норма.

ЗАКЛЮЧЕНИЕ

В статье затронута фундаментальная проблема исследования и моделирования понимания в интересах разработки искусственных понимающих агентов и других классов интеллектуальных систем новых поколений. В ней продолжено обсуждение вопросов создания понимающих компьютеров и роботов, начатое почти 30 лет назад в монографии [Winograd et al., 1986]. Описаны классические подходы к пониманию, предложенные в семиотике, герменевтике, дидактике, психологии. Прослежены связи между базовыми категориями, относящимися к пониманию и образующими его онтологический каркас, такими как «ценность», «истина», «смысл», «значение», «интерпретация», «объяснение», «оправдание», «обоснование».

При этом мы сознательно опустили давно ведущиеся в искусственном интеллекте, начиная с Т.Винограда, работы по машинному пониманию естественного языка и сосредоточились на сравнительно новых объектах понимания –

событиях, действиях, ситуациях, измерениях и т.п., связанных с функционированием искусственных когнитивных агентов, интеллектуальных роботов, интеллектуальных сред и т.п. Также в статью не вошли онтологии и методы онтологического инжиниринга когнитивных систем: некоторые соображения по этому поводу были опубликованы ранее в нашей статье «Гранулярные нечеткие и лингвистические онтологии для обеспечения взаимопонимания между когнитивными агентами», опубликованной в материалах OSTIS-2012 [Тарасов и др., 2012].

Главное внимание в работе уделено построению системы гибридных когнитивных микронаук, непосредственно ориентированной на изучение комплексной проблемы понимания, и описанию ее компонентов в русле разработки инструментария когнитивной информатики. Перспективы наших дальнейших исследований связаны с развитием формальных подходов к моделированию понимания на базе абстрактных алгебр и прагматических логик.

Работа выполнена при финансовой поддержке РФФИ, проекты №14-07-00846 и №14-07-00653.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- [Бахтин, 1979] Бахтин, М.М. К методологии гуманитарных наук/ М.М.Бахтин// Эстетика словесного творчества. – М.: Искусство, 1979.
- [Будбаева и др., 1974] Будбаева, С.П. К исследованию и построению прагматических логик/ С.П.Будбаева, Б.Н.Пятницын// Философия и логика. – М.: Наука, 1974. – С.220-278.
- [Валькман,2012] Валькман, Ю.Р. О когнитивной семиотике/ Ю.Р.Валькман//Интеллектуальный анализ информации. Сборник трудов XII-й международной конференции им. Т.А.Таран (ИАИ-2012, Киев, 16-18 мая 2012 г.). – Киев: Просвіта, 2012. – С.19-30.
- [Валькман, 2013] Валькман, Ю.Р. Когнитивная семиотика: истоки и перспективы/ Ю.В.Валькман//Интегрированные модели и мягкие вычисления в искусственном интеллекте. Сборник научных трудов VII-й международной научно-практической конференции (ИММВ-2013, Коломна, 20-22 мая 2013 г.). – М.: Физматлит, 2013. – Т.1. – С.48-61.
- [Васильев, 1999] Васильев Л.Г. Три парадигмы понимания: Анализ литературы вопроса/ Л.Г.Васильев/ <http://www.konf-csu.narod.ru/ze/lib/vasilyev.html>.
- [Величковский, 2006] Величковский, Б.М. Когнитивная наука. Основы психологии познания. В 2-х томах/ Б.М.Величковский. – М.: Смысл/ Академия, 2006.
- [Витгенштейн, 2009] Витгенштейн, Л. Логико-философский трактат / Пер. с нем., 2-е изд./Л.Витгенштейн. – М.: Наука, 2009.
- [фон Вригт,1986] Вригт фон, Г.Х. Объяснение и понимание/ Г.Х. фон Вригт// Логико-философские исследования. Избранные труды: Пер. с англ. – М.: Прогресс, 1986.
- [Выготский, 1982] Выготский Л.С. Собрание сочинений: в 6 томах/ Л.С.Выготский. Т.1. – М.: Педагогика, 1982.
- [Гадамер, 1988] Гадамер, Х.-Г. Истина и метод: основы философской герменевтики: Пер. с нем./ Х.-Г.Гадамер. – М.: Прогресс, 1988.
- [Голенков и др., 2012] Голенков В.В. Графодинамические модели параллельной обработки знаний: принципы построения, реализации и проектирования/ В.В.Голенков, Н.А. Гулякина Графодинамические модели параллельной обработки знаний: принципы построения, реализации и проектирования// Открытые семантические технологии проектирования интеллектуальных систем. Материалы II-й международной научно-технической конференции (Минск, БГУИР, 16-18 февраля 2012 г.). – Минск: Изд-во БГУИР, 2012. – С.23-52.
- [Голенков и др., 2013] Голенков В.В. Открытый проект, направленный на создание технологии компонентного проектирования интеллектуальных систем/ В.В.Голенков, Н.А.

Гулякина//Открытые семантические технологии проектирования интеллектуальных систем. Материалы III-й международной научно-технической конференции (Минск,БГУИР,21-23 февраля 2013 г.). – Минск: Изд-во БГУИР, 2013. – С.55-77.

[Голенков и др., 2014] Голенков В.В. Структуризация смыслового пространства/ В.В.Голенков, Н.А.Гулякина // Открытые семантические технологии проектирования интеллектуальных систем. Материалы IV-й международной научно-технической конференции (Минск, БГУИР, 20-22 февраля 2014 г.). – Минск: Изд-во БГУИР, 2014. – С.65-78.

[Ефименко и др., 2014] Ефименко И.В. УСК Мартынова – тридцать лет спустя/ И.В.Ефименко, В.Ф.Хорошевский // Открытые семантические технологии проектирования интеллектуальных систем. Материалы IV-й Международной научно-технической конференции (Минск,БГУИР,20-22 февраля 2014 г.). – Минск: БГУИР, 2014. – С.29-38.

[Знаков,1999] Знаков В.В. Понимание в познании и общении/ В.В.Знаков. – М.: Изд-во «Институт психологии РАН», 1999.

[Ивин, 1997а] Ивин, А.А. Основы теории аргументации/ А.А.Ивин. – М.: ГИЦ ВЛАДОС,1997.

[Ивин, 1997б] Ивин, А.А. Словарь по логике/ А.А.Ивин, А.Л.Никифоров. – М.: ГИЦ ВЛАДОС,1997.

[ИОС и ВУО, 2001] Интеллектуальные обучающие системы и виртуальные учебные организации/ Под ред. В.В.Голенкова и В.Б.Тарасова. – Мн.: Изд-во БГУИР, 2001.

[Коробов,2005] Коробов, Е.Т. Понимание как дидактическая проблема/ Е.Т.Коробов// Московский психологический журнал, 2005, №11-12.

[Кузнецов, 2012] Кузнецов О.П. Когнитивная семантика и искусственный интеллект/ О.П.Кузнецов// Искусственный интеллект и принятие решений, 2012, №4, с.32-42.

[Лакофф, 2004] Лакофф Дж. Женщины, огонь и опасные вещи: Что категории языка говорят нам о мышлении: Пер. с англ./ Дж. Лакофф// – М.: Языки славянской культуры, 2004

[Лотман, 2010] Лотман, Ю.М. Семиосфера/ Ю.М.Лотман. – СПб.: Искусство-СПБ, 2010.

[Мартынов,1974] Мартынов, В.В. Семиологические основы информатики/ В.В.Мартынов. – Мн.: Наука и техника, 1974.

[Мартынов, 1978] Мартынов, В.В. Семиологические проблемы искусственного интеллекта/ В.В.Мартынов// Известия АН СССР: Серия литературы и языка, 1978, том 37, №1, с.3-9.

[Мартынов, 2009] Мартынов, В.В. В центре сознания человека/ В.В.Мартынов. – Мн.: Изд-во БГУ, 2009.

[Моррис, 1983] Моррис, Ч.У. Основания теории знаков/ Ч.У.Моррис//Семиотика:Антология. –М.:Академический проект, 2001. – С.45-97.

[НИИ,1998] Новости искусственного интеллекта, 1998, №2, с.164-174.

[Осипов, 2002] Осипов, Г.С. От ситуационного управления к прикладной семиотике/ Г.С.Осипов// Новости искусственного интеллекта, 2002, №6, с.3-7.

[Пиаже, 1994] Пиаже, Ж. Избранные психологические труды: Пер. с франц. и англ./ Ж.Пиаже. – М.: МПА, 1994.

[Пирс,1999] Пирс,Ч.С. Логика как семиотика: теория знаков /Ч.С.Пирс//Метафизические исследования. Вып.11. Язык. – СПб: Алетейя, 1999. – С.199-217.

[Поспелов, 1970] Поспелов, Д.А. Системный подход к моделированию мыслительной деятельности/ Д.А.Поспелов// Проблемы методологии системного исследования. – М.: Мысль, 1970. – С.333-358.

[Поспелов, 1976] Поспелов, Д.А. Семиотические модели: успехи и перспективы/ Д.А.Поспелов// Кибернетика, 1976, №6, с.114-123.

[Поспелов,1989] Поспелов,Д.А.Интеллектуальные интерфейсы для ЭВМ новых поколений/ Д.А.Поспелов// Электронная вычислительная техника. Сборник статей. Вып.3. – М.: Радио и связь, 1989. – С.4-20.

[Поспелов и др., 1999] Поспелов, Д.А. Прикладная семиотика/ Д.А.Поспелов, Г.С.Осипов//Новости искусственного интеллекта, 1999, №1, с.9-35.

[Почепцов,1998] Почепцов Г.Г. История русской семиотики / Г.Г.Почепцов. – М.: Лабиринт, 1998.

[Психология, 1990] Психология. Словарь/ Под общ. ред. А.В.Петровского, М.Г.Ярошевского. – 2-е изд., испр и доп. – М.: Политиздат, 1990.

[Рикер, 1995] Рикер, П. Конфликт интерпретаций. Очерки герменевтики: Пер. с франц./ П.Рикер. –М.: КАНОН-пресс, 2002.

[Рубинштейн, 1989] Рубинштейн? С.Л. Основы общей психологии: В 2 т. Т. I/ С.Л.Рубинштейн. – М.: Педагогика, 1989.

[СФС, 2004] Современный философский словарь.–3-е изд., испр и доп. –М.: Академический проект, 2004.

[де Соссюр, 2004] Соссюр де, Ф. Курс общей лингвистики/ Ф. де Соссюр. – М.: Едиториал УРСС, 2004.

[Тарасов, 2002] Тарасов, В.Б. От многоагентных систем к интеллектуальным организациям: философия, психология, информатика/ В.Б.Тарасов. – М.: Едиториал УРСС, 2002.

[Тарасов, 2014а] Тарасов, В.Б. От семантического кода к когнитивной лингвистике, семиотике и информатике: наследие В.В.Мартынова/В.Б.Тарасов//Открытые семантические технологии проектирования интеллектуальных систем. Материалы IV-й Международной научно-технической конференции (Минск, БГУИР, 20-22 февраля 2014 г.). – Минск: БГУИР, 2014. – С.39-48.

[Тарасов, 2014б] Тарасов, В.Б. Грануляция информации: основа когнитивных процессов и предпосылка создания интеллектуальных систем новых поколений/ В.Б.Тарасов, В.Г.Редько и др.// Подходы к моделированию мышления. – М.: Изд-во УРСС, 2014. – С.219-261.

[Тарасов и др., 2012] Тарасов, В.Б. Гранулярные, нечеткие и лингвистические онтологии для обеспечения взаимопонимания между когнитивными агентами/ В.Б.Тарасов, А.П.Калуцкая, М.Н.Святкина// Открытые семантические технологии проектирования интеллектуальных систем. Материалы II-й Международной научно-технической конференции (Минск, БГУИР, 16-18 февраля 2012 г.). – Минск: БГУИР, 2012. – С.267-278.

[Философия, 2004] Философия: Энциклопедический словарь. – М.: Гардарики, 2004.

[Финн, 2008] Финн, В.К. Многозначные логики и их применения. Том 1. Логические исчисления, алгебры и функциональные свойства/ В.К.Финн (ред.-сост.). – М.: Изд-во ЛКИ, 2008.

[Фреге, 2012] Фреге, Г. Логика и логическая семантика: Пер. с нем. 2-е изд./ Г.Фреге. – М.: Книжный дом «ЛИБРОКОМ», 2012.

[Фрид, 1979] Фрид, Э. Элементарное введение в абстрактную алгебру: Пер. с венгер./Э.Фрид. – М.: Мир, 1979.

[Хайдеггер, 2013] Хайдеггер, М. Бытие и время: Пер. с нем./М.Хайдеггер. – М.: Академический проект, 2013.

[Чудова, 2012] Чудова, Н.В. Понимание: предмет исследования и объект моделирования/ Н.В.Чудова// Искусственный интеллект и принятие решений, 2012, №4, с.3-31.

[Шадриков, 2006] Шадриков, В.Д. Интеллектуальные операции/ В.Д.Шадриков. – М.: Логос, 2006.

[Шпет, 1996] Шпет, Г.Г. Явление и смысл/Г.Г.Шпет.– Омск: Водолей, 1996.

[Шульга, 2002] Шульга, Е.Н. Когнитивная герменевтика/ Е.Н.Шульга. – М.: Изд-во ИФ РАН, 2002.

[Ackoff, 1989] Ackoff, R. From Data to Wisdom/ R.Ackoff// Journal of Applied Systems Analysis, 1989, vol.16, p.3-9.

[Ajdukiewicz, 1974] Ajdukiewicz, K. Pragmatic Logic/ K. Ajdukiewicz. – Warsaw: D. Reidel Publishing Company, 1974.

[Daddesio, 1994] Daddesio, Th.C. On Minds and Symbols: the Relevance of Cognitive Science to Semiotics/ Th.C. Daddesio. – Berlin: Mouton de Gruyter, 1994.

[Gardner, 1987] Gardner, H.E. The Mind's New Science: a History of Cognitive Revolution/ H.E.Gardner. – New York: Basic Books, 1987.

[Goddard et al., 1973] Goddard, L. The Logic of Significance and Context/ L.Goddard, R.Routley. – Edinburgh: Scottish Academic Press, 1973.

[Goguen. 1999] Goguen, J. An Introduction to Algebraic Semiotics with Applications to User Interface Design/ J.Goguen// Computation for Metaphors, Analogy and Agents. Lecture Notes in Artificial Intelligence. Vol.1562/ Ed.by C.Nehaniv. – Berlin: Springer, 1999, p.242-291.

[Lin, 1998] Lin, T.Y. Granular Computing on Binary Relations I: Data Mining and Neighborhood System/ T.Y.Lin// Rough Sets in Knowledge Discovery/ Ed. by A. Skowron and L. Polkowski. Heidelberg: Physica-Verlag, 1998. – P.107-140.

[Meystel, 1995] Meystel, A. Semiotic Modeling and Situation Analysis: an Introduction/ A.Meystel. – Bala Cynwyd, PA: AdRem Inc., 1995.

[Schmid, 2012] Schmid H.-J, Cognitive Pragmatics/ H.-J. Schmid. – Berlin: Mouton de Gruyter, 1994.

[Talmy, 2000] Talmy, L. Toward a Cognitive Semantics/ L.Talmy. – Cambridge MA, MIT Press, 2000.

[Winograd et al., 1986] Winograd, T. Understanding Computers and Cognition: a New Foundation for Design/ T.Winograd, F.Flores. – Norwood: Ablex, 1986.

[Yao, 2008] Yao, Y.Y., Granular Computing: Past, Present, and Future/Y.Y.Yao/ Proceedings of the 2008 IEEE International Conference on Granular Computing (Hangzhou, China, August 26-28, 2008). – P.80-85.

[Zadeh, 1997] Zadeh, L.A. Toward a Theory of Fuzzy Information Granulation and its Centrality in Human Reasoning and Fuzzy Logic/ L.A.Zadeh// Fuzzy Sets and Systems, 1997, vol.90, p.111-127.

[Zlatev, 2012] Zlatev, J. Cognitive Semiotics: an Emerging Field for Transdisciplinary Study of Meaning/ J.Zlatev// The Public Journal of Semiotics/ 2012, vol IV, №1.

PROBLEM OF UNDERSTANDING: PRESENT AND FUTURE OF ARTIFICIAL INTELLIGENCE

Tarassov V.B.

*Bauman Moscow State Technical University,
Moscow, Russia*

Vbulbov@yahoo.com

Some results of preliminary investigation in the field of transdisciplinary problem of understanding (comprehension) are presented. This problem is faced in the context of OSTIS project with the purpose of developing artificial comprehensive agents and other classes of next generation cognitive intelligent systems. Here a special interest is taken in new objects of understanding such as behavior, actions, situations, measurements, etc. To answer the question «What is comprehension?» we collected and analyzed basic definitions of understanding. Some classical understanding concepts, which have been suggested in hermeneutics, semiotics, didactics, psychology, are considered. Also key categories related to understanding such as meaning, sense, value, truth, interpretation, explanation, justification are discussed. A few classifications of understanding levels are made; an appropriate hierarchy for understanding robot is introduced. A short review of perceptual and thinking operations is presented and their links with information granulation concepts are revealed. The main attention in this paper is paid to constructing a new system of hybrid cognitive micro-sciences different from canonical Gardner's system. This new system is intended to thoroughly study the problem of understanding. Some formal approaches to modeling comprehension on the basis of abstract algebras and pragmatic logics are proposed.

Keywords: Artificial Intelligence, Knowledge, Understanding, Comprehension, Hermeneutics, Meaning, Sense, Pragmatics, Interpretation, Cognitive Semiotics, Cognitive Semantics, Cognitive Informatics.



OSTIS-2015

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822:514

СЕМАНТИЧЕСКАЯ ТЕХНОЛОГИЯ КАРТИРОВАНИЯ СЕМАНТИЧЕСКИХ ТЕХНОЛОГИЙ (Наукометрический анализ конференций OSTIS)

Хорошевский В.Ф.* , Ефименко И.В.**

* *Вычислительный центр им. А.А. Дородницына РАН, г. Москва, Россия*
khor@ccas.ru

** *Факультет гуманитарных наук НИУ ВШЭ, г. Москва, Россия*
iefimenko@hse.ru

Обсуждаются вопросы картирования научных направлений на основе методов наукометрии с использованием семантических технологий. Исходными данными для проведения исследования является корпус статей, опубликованных в трудах конференций серии OSTIS «Open Semantic Technologies for Intelligent Systems». Дается краткое описание основных понятий наукометрии и обосновывается важность применения семантических технологий в решении возникающих здесь задач. Предлагаются модели, методы и средства анализа научных направлений и выявления центров компетенций и центров превосходства на базе семантизации методов наукометрии. Приводятся результаты семантического картирования предметной области семантических технологий, специфицированной в трудах конференций серии OSTIS.

Ключевые слова: наукометрия; онтологическое моделирование; карта науки; скрытые коллективы; центры превосходства и компетенций; семантические технологии; извлечение информации из текстов

Введение

Исследования и разработки по картированию научных направлений на основе методов и средств наукометрии активно развиваются во всем мире [Small, 2010; Borner et al., 2012; Boyack et al., 2014]. Для оценки ситуации в той или иной области, как правило, используются статистические методы библиометрического анализа публикаций [Boyack et al., 2005; Klavans et al., 2006; Shibata, et al., 2008], которые далеко не всегда адекватны задачам построения действительно информативных карт науки. В связи с этим в последнее время в проектах по наукометрии все большее внимание уделяется поиску новых методов картирования научных направлений и способов выявления центров компетенции и превосходства в различных областях науки и техники [Boyack, 2009; Upham et al., 2010; Klavans et al., 2010; Erdi et al., 2013].

При этом «горячими точками» исследований и разработок является применение методов интеллектуального анализа данных [Паклин и др., 2009; Witten et al., 2011], компьютерной лингвистики и, в частности, извлечения информации из текстов [Li, et al., 2011; Boyack et al., 2013; Efimenko et al., 2014], а также специальных

средств визуализации результатов библиометрии [Borner et al., 2003; Klavans et al., 2011; Van Eck et al., 2014].

С учетом вышесказанного, целью настоящей работы является обсуждение вопросов построения семантических карт научных направлений на базе совместного использования методов наукометрии и семантических технологий, а в качестве исходных данных для проведения исследования выступает корпус статей, опубликованных в трудах конференций серии OSTIS.

1. Основные понятия наукометрии

1.1. Предварительные замечания

В рамках общего направления «Информатика» одной из востребованных в настоящее время научных дисциплин, связанной с количественными измерениями хранимой и используемой информации является **инфометрия**¹, как правило, включающая:

- **библиометрию (Bibliometrics)**, которая занимается изучением документов на основе

¹ <http://en.m.wikipedia.org/wiki/Informetrics>

количественного анализа первичных и вторичных источников информации с помощью формализованных методов с целью получения данных об эффективности, динамике, структуре и закономерностях развития исследуемых областей;

- **наукометрию (Scientometrics)** – научную дисциплину, связанную с изучением количественных методов развития науки как информационного процесса;

- **вебометрику (Webometrics)** – раздел информатики, где исследуются количественные аспекты конструирования и использования информационных ресурсов, структур и технологий применительно к Интернет;

- **альтметрику (Altmetrics)** – раздел информетрии, где исследуются различные аспекты использования открытых информационных ресурсов, включая социальные сети и другие источники информации.

В настоящей работе авторы, в основном, концентрируются на вопросах библиометрии и наукометрии. Поэтому в оставшейся части данного раздела обсуждаются понятия и проблемы именно этих направлений информетрии.

1.2. Основные понятия и индикаторы

Как известно, сам термин «наукометрия» был впервые введен в монографии [Налимов и др., 1969], где данное понятие связывалось с изучением эволюции науки через многочисленные измерения и статистическую обработку информации. Наукометрию часто применяют как абсолютную основу оценки выполнения и финансирования различных организационных единиц (институтов, команд, индивидуумов, а также проектов и разработок).

В западной науке данное направление (Scientometrics) трактуют шире, включая в него изучение, измерения и аналитику в области науки, технологий и инноваций для использования в контексте научной политики и управления.

Основным понятием библиометрии и наукометрии является **индекс цитирования** научных статей (**ИЦ**) — реферативная база данных научных публикаций, индексирующая ссылки, указанные в пристатейных списках этих публикаций и предоставляющая количественные показатели этих ссылок. Заметим, что в России распространена особая интерпретация данного понятия, как показателя, указывающего на значимость статьи и вычисляющегося на основе последующих публикаций, ссылающихся на данную работу.

История создания индексов (или указателей) научного цитирования начинается с 70-х годов XIX века, когда практически одновременно появляются индекс юридических документов Shepard's Citations (1873 г.) и индекс научных публикаций по медицине Index Medicus (1879 г.). Последний просуществовал вплоть до 2004 г.

В 1960 году Институт научной информации (ISI),

основанный Юджином Гарфилдом, ввёл индекс цитирования для статей, опубликованных в научных журналах, положив начало «Science Citation Index (SCI)», а затем включив в него индексы цитирования по общественным наукам («Social Sciences Citation Index», SSCI) и искусствам («Arts and Humanities Citation Index», АНЦИ). Начиная с 2006 г. появились другие источники подобных данных, например, Google Scholar. Наиболее известными ИЦ в настоящее время являются библиометрические базы Web of Science и SCOPUS.

В 1974 году в ВИНТИ были предприняты попытки создания отечественного указателя научного цитирования (УНЦ), который в технологическом плане должен был стать «аналогом» SCI [Егоров и др., 2006]. Работы по библиометрии и наукометрии активно велись и в других организациях России [Налимов и др., 1971; Евстигнеев, 1987; Маршакова, 1988; Хайтун, 1989], но с приходом на отечественный рынок БД и сервисов Web of Science, SCOPUS, PubMed и некоторых других отечественные исследования и разработки в данной области «сходят на нет». И только в последнее время здесь намечается некоторое «оживление» ситуации [Кулинич, 2011; Крюков и др., 2013; Хорошевский, 2012а; Хорошевский и др., 2014; Efimenko et al., 2014].

В 1987 Китай запустил проект по созданию Китайского индекса научного цитирования Chinese Science Citation Index, а в следующем, 1988 появился его конкурент — China Scientific and Technical Papers and Citations. В 1997 начата разработка китайского индекса цитирования по общественным наукам Chinese Social Sciences Citation Index.

В 1995 году Япония приступила к созданию национального индекса цитирования Citation Database for Japanese Papers, разработчиком которого является Национальный институт информатики Японии.

Разработки национальных индексов ведутся на Тайване (Taiwan Humanities Citation Index), а также в ряде европейских стран (Польша, Испания).

Базовые индикаторы и показатели библиометрии и наукометрии², как правило, включают:

- **g-индекс** (2006 г., Leo Eggh) – индекс для измерения научной продуктивности, который рассчитывается на основе распределения цитирований, полученных публикациями ученого следующим образом: для данного множества статей, отсортированного в порядке убывания количества цитирований, которые получили эти статьи, g-индекс это наибольшее число, такое что g самых цитируемых статей получили (суммарно) не менее g² цитирований.

- **h-индекс** (индекс Хирша) – показатель, предложенный в 2005 г. аргентино-американским физиком Хорхе Хиршем из Калифорнийского

² <https://ru.wikipedia.org/wiki/Категория:Наукометрия>

университета в Сан-Диего, который является количественной характеристикой продуктивности ученого, группы ученых, научной организации или страны в целом, основанной на количестве публикаций и количестве цитирований этих публикаций. Согласно Хиршу, ученый имеет индекс h , если h из его N_p статей цитируются как минимум h раз каждая, в то время как оставшиеся ($N_p - h$) статей цитируются не более, чем h раз каждая.

- **i -индекс** (Предложен в 2006 г. независимо М. Космульским и Г. Пратхапом) – индекс публикационной активности научной организации, рассчитываемый на основе библиометрических показателей с использованием распределения индекса Хирша ученых из данной научной организации. Научная организация имеет индекс i , если не менее i ученых из этой организации имеют h -индекс не менее i .

- **Исследовательские фронты**, основанные на использовании методов библиографических связей (ретроспективный метод) и/или коцитирования (проспективный метод).

- **Импакт-фактор (ИФ или IF)** - численный показатель важности научного журнала. С 60-х годов прошлого века он ежегодно рассчитывается Институтом научной информации (ISI), который в 1992 г. был приобретен корпорацией Thomson, ныне называется Thomson Scientific и публикуется в журнале «Journal Citation Report». Расчет импакт-фактора основан на 3-х летнем периоде: так, например, импакт-фактор журнала в 2011 году – $IF_{2011} = A/B$, где: A - число цитирований в течение 2011 г. в журналах, отслеживаемых ISI, статей, опубликованных в данном журнале в 2009–2010 гг.; B — число статей, опубликованных в данном журнале в 2009-2010 гг.

В соответствии с ИФ (в основном в других странах, но в последнее время и в России) оценивают уровень журналов, качество статей, опубликованных в них, дают финансовую поддержку исследователям и даже принимают сотрудников на работу.

К основным библиометрическим базам и сервисам относятся:

- **Web of Knowledge (Web of Science)**³ – поисковая платформа, объединяющая реферативные базы данных публикаций в научных журналах и патентов, в том числе базы, учитывающие взаимное цитирование публикаций, разрабатываемая и предоставляемая компанией Thomson Reuters. Web of Knowledge охватывает материалы по естественным, техническим, биологическим, общественным, гуманитарным наукам и искусству. Платформа обладает встроенными возможностями поиска, анализа и управления библиографической информацией.

- **SCOPUS**⁴ – библиографическая и реферативная база данных и инструмент для отслеживания цитируемости статей,

опубликованных в научных изданиях. Индексирует 18 тыс. названий научных изданий по техническим, медицинским и гуманитарным наукам 5 тыс. издателей, включая научные журналы, материалы конференций и серийные книжные издания. Разработчиком и владельцем Scopus является издательская корпорация Elsevier. В отличие от Web of Knowledge, в Scopus не используется понятие импакт-факторов, но очень широко применяется индекс Хирша.

- **Академия Google (Google Scholar)**⁵ – свободно доступная поисковая система, обеспечивающая полнотекстовый поиск научных публикаций всех форматов и дисциплин. Система работает с ноября 2004 года, первоначально в статусе бета-версии. Индекс Google Scholar включает в себя большинство рецензируемых онлайн журналов Европы и Америки крупнейших научных издательств. По функциям он похож на свободно доступные системы Scirus от Elsevier, CiteSeerX и getCITED, а также на платные сервисы Scopus и Web of Science.

- **РИНЦ**⁶ – библиографическая база данных научных публикаций российских ученых. Для получения необходимых пользователю данных о публикациях и цитируемости статей на основе базы данных РИНЦ разработан аналитический инструмент ScienceIndex. Проект РИНЦ разрабатывается с 2005 года компанией «Научная электронная библиотека» (ELIBRARY.ru)

Базис библиометрических баз и сервисов составляет инструментарий наукометрии. При этом типология таких инструментов включает

- средства загрузки данных;
- средства аналитики на данных и
- средства визуализации результатов.

Известными примерами инструментов наукометрии являются, в частности:

- Аналитические сервисы Web of Science и SCOPUS.
- Системы CitNet Explorer и VOS Viewer из Лейденского университета.
- Коммерческая платформа VantagePoint.
- Аналитические инструменты компании SciTech Strategies и др.

Мощными инструментами визуализации результатов наукометрии являются сети цитирования и коцитирования, диаграммы взаимовлияния научных направлений (Win-Win партнерства), сети динамики развития научных направлений, а также геоландшафты, обеспечивающие поиск партнеров, выявление лидерства стран, распространения научных идей, научных школ и виртуальных международных коллективов.

В качестве примеров, на Рис. 1-4 приводятся скриншоты, иллюстрирующие перечисленные выше

³ <http://thomsonreuters.com/about-us/>

⁴ <http://www.scopus.com/>

⁵ <http://scholar.google.ru/>

⁶ <http://elibrary.ru/>

сервисы наукометрии.

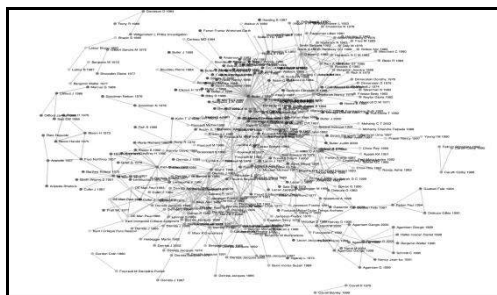


Рисунок 1 – Фрагмент сети цитирования и коцитирования

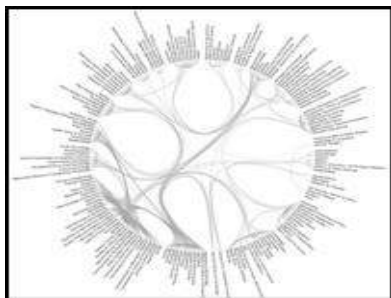


Рисунок 2 – Пример Win-Win партнерства

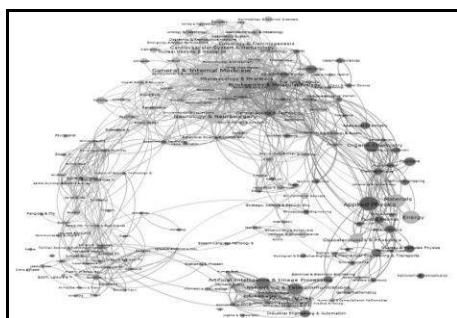


Рисунок 3 – Пример динамики развития научных направлений



Рисунок 4 – Пример геоландшафта научных направлений

1.3. Наукометрические БД и сервисы

В настоящее время работы по теории и практике наукометрии активно развиваются во всем мире, а особое внимание уделяется инструментальным средствам. Не имея возможности в настоящей работе дать даже краткий обзор исследовательской активности в данной области, мы представим ниже лишь несколько интересных систем и сервисов наукометрии.

1.3.1. Модели и инструменты SciTech Strategies

Модели, методы и инструменты картирования науки и технологий от компании SciTech Strategies⁷

⁷ <http://www.mapofscience.com/>

из США известны во всем мире. И даже само понятие карт науки было введено основателями этой компании К. Бояком (K. Boyack) и Р. Клэвансом (R. Klavans).

Для примера, на Рис. 5 приведена экранная форма библиотеки карт науки от компании SciTech Strategies.

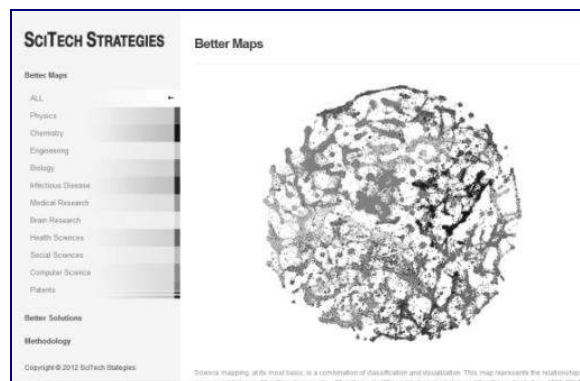


Рисунок 5 – Библиотека карт науки SciTech Strategies

В настоящее время К. Бояк, Р. Клэванс и их сотрудники активно работают над моделями наукометрии нового поколения [Boyack et al., 2014; Klavans et al., 2014a; Klavans et al., 2014b].

1.3.2. Наукометрические инструменты Лейденского университета

Центр исследований науки и технологий Лейденского университета в Нидерландах является одной из известных европейских точек превосходства и компетенций в данной области, а наукометрические инструменты CitNet Explorer и VOS Viewer [Van Eck et al., 2014], разработанные здесь, активно используются в разных странах.

Основные функционалы базового инструмента CitNet Explorer – следующие:

- В части данных поддерживаются
 - импорт из БД Web of Science,
 - экспорт в файлы Rajek-формата,
 - работа с «большими данными» (сети 10¹⁶ публикаций и 10¹⁷ ссылок).
- При визуализации поддерживается
 - Масштабирование и скроллинг сетей в стиле Google Maps,
 - отображение сетей косвенного цитирования,
 - экспорт скриншотов с результатами в Word или PowerPoint.
- Аналитика с помощью библиотеки алгоритмов для идентификации связанных компонент, кластеров, ядерных публикаций, кратчайших (максимальных) путей цитирования поддерживает:
 - фильтрацию публикаций по времени,
 - функции drill down и expand для навигации по сетям цитирования.

Примеры экранных форм CitNet Explorer и VOS Viewer представлены на Рис. 6-7.

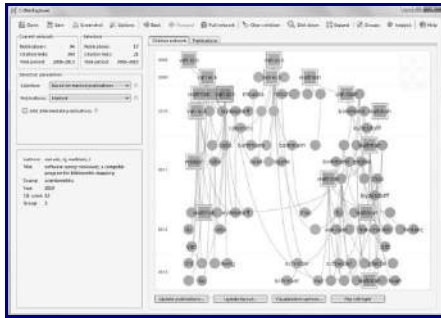


Рисунок 6 –Пример экранной формы CitNet Explorer

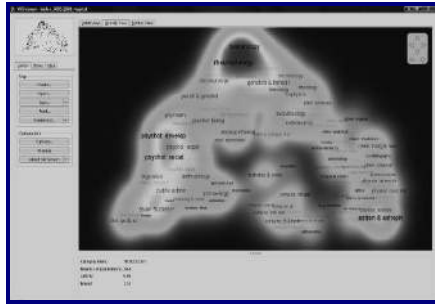


Рисунок 7 – Пример экранной формы VOS Viewer

Достоинством инструментов CitNet Explorer и VOS Viewer является и то, что, в отличие от многих других, они поставляются как программное обеспечение с открытым кодом, а недостатком – невозможность работы с русскоязычными текстами.

1.3.3. Коммерческая система VantagePoint

Флагманским продуктом компании VantagePoint⁸ является одноименная библиометрическая платформа, которая (по утверждению разработчиков) является мощным инструментом извлечения знаний из результатов поиска патентов и БД литературы, а визуальные VantagePoint-перспективы, скриншоты которых представлены на Рис. 8, позволяют быстро найти ответы на вопросы WHO, WHAT, WHEN и WHERE, обеспечивая поиск критических паттернов.

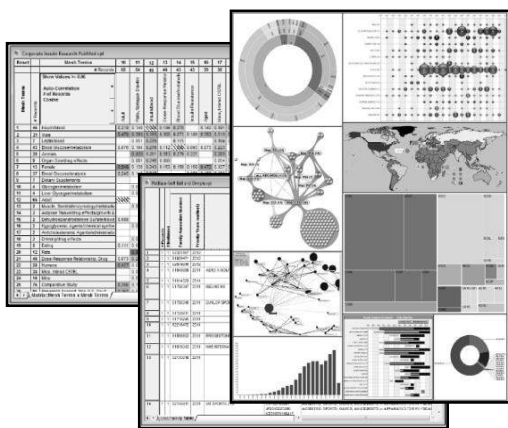


Рисунок 8 – Примеры VantagePoint-перспектив

1.3.4. Аналитический инструмент ScienceIndex РИНЦ

Как отмечалось выше, российский индекс научного цитирования (РИНЦ) в настоящее время

формируется на платформе e-Library, где основным аналитическим инструментом является ScienceIndex [Юрков, 2015], основные сервисы которого представлены на Рис. 9.

СТАТИСТИЧЕСКИЕ ОТЧЕТЫ	
■ Распределение публикаций по тематике	■ Распределение цитирующих публикаций по тематике
■ Распределение публикаций по ключевым словам	■ Распределение цитирующих публикаций по ключевым словам
■ Распределение публикаций по журналам	■ Распределение цитирующих публикаций по журналам
■ Распределение публикаций по организациям	■ Распределение цитирующих публикаций по организациям
■ Распределение публикаций по соавторам	■ Распределение цитирующих публикаций по соавторам
■ Распределение публикаций по годам	■ Распределение цитирующих публикаций по годам
■ Распределение публикаций по типу цитирований	■ Распределение цитирующих публикаций по типу цитирований
■ Распределение публикаций по числу соавторов	■ Распределение цитирований по годам цитируемых публикаций
■ Распределение цитирований по годам цитирующих публикаций	■ Распределение цитирований по годам цитируемых публикаций
■ Распределение цитирований по тематике цитирующих публикаций	■ Распределение цитирований по соавторам цитируемых публикаций
■ Распределение цитирований по цитирующим журналам	■ Распределение цитирований по типу цитирующих публикаций

Рисунок 9 – Основные сервисы ScienceIndex РИНЦ

К сожалению, в настоящее время в БД РИНЦ проиндексировано мало (по сравнению с мировыми библиометрическими БД) документов, что существенно затрудняет проведение аналитических исследований.

1.3.5. Заключительные замечания

Представленный выше краткий обзор основных понятий, методов и инструментов наукометрии позволяет сформулировать несколько положений.

Во-первых, можно констатировать, что важность наукометрии как одного из инструментов для оценки состояния исследований и разработок в научно-технической сфере и поддержки принятия решений в данной области уже практически осознана во всем мире. При этом для России основными задачами наукометрии являются:

- Выявление новых направлений научно-технического прогресса.
- Выявление центров компетенции и превосходства в прорывных направлениях научно-технического прогресса.
- Отображение научно-технологических ресурсов страны на мировые тренды научно-технического прогресса.
- Отображение научно-технологических ресурсов страны и мировых трендов на цели страны и доступные средства.
- Формирование целевых программ научно-технического прогресса.
- Планирование процессов поддержки целевых программ научно-технического прогресса.
- Оценка результатов выполнения целевых программ и формирование новых целей.

Во-вторых, можно отметить, что в рамках наукометрии активно развиваются следующие научно-технологические направления:

- Библиометрия и наукометрия научно-технической сферы, включая
 - индикативные методы библиометрии;
 - анализ исследовательских фронтов;

⁸ <https://www.thevantagepoint.com/>

- выявление новых научно-технологических трендов.
- Форсайт в научно-технической сфере, в том числе
 - экспертные панели;
 - дорожное картирование.
- Системный анализ и исследование операций, включая
 - модели и методы прогнозирования;
 - методы экспертных оценок и согласования мнений;
 - методы планирования и оптимизации.

Основными проблемами в данной области, по нашему мнению, являются:

- Неполнота и «зашумленность» исходных данных.
- Преимущественная ориентация на статистические методы анализа процессов наукометрии.
- Отсутствие новых моделей и методов анализа процессов наукометрии.
- Недостаточно активное использование в научно-технической сфере уже существующих математических моделей и методов поддержки принятия решений.
- Слабое использование математических методов управления научно-технической сферой.

Представляется, что в свете указанных целей, задач и проблем семантические технологии могут стать драйвером создания новых методов мониторинга исследований и разработок в научно-технической сфере и драйвером создания новых методов поддержки принятия решений.

В заключение хотелось бы отметить, что научно-технологическое направление наукометрии нового поколения находится в настоящее время лишь в начале пути и требует серьезных междисциплинарных исследований и разработок.

2. Модели, методы и средства анализа научных направлений и выявления центров компетенций и превосходства

2.1. Предварительные замечания

Как отмечалось в работах [Wang, et al., 2010; Li, et al., 2011; Erdi, et al., 2013; Хорошевский и др., 2014], для построения карт науки целесообразно использовать гибридный подход, где методы кластеризации и классификации работают на данных, сформированных с помощью методов извлечения информации из текстов под управлением онтологий.

В основе предлагаемого в настоящей работе подхода к выделению из публикаций семантически значимой системы терминов предметной области, которые специфицируют характеристические вектора публикаций, лежит принцип «черного ящика» [Efimenko, et al., 2014], а интеграция результатов обработки коллекций отдельных жанров

осуществляется на основе результатов статистической обработки коллекций публикаций. Спецификой предлагаемого подхода является и то, что после лингвистической и статистической обработки текстов происходит автоматическая генерация семантических представлений результатов, которые затем отражаются в экземплярную часть OWL-онтологии предметной области, в XML-файлы для системы кластеризации Caquot, а также в спецификации графического представления карты научного направления и выявления временных рядов терминов.

2.2. Онтологическая модель направления

Для автоматизации процессов построения карты научного направления OSTIS была использована система онтологических моделей, описанных в работе [Хорошевский и др., 2014]. Для примера, на Рис. 10 представлен фрагмент используемой в настоящем исследовании онтологической модели. Заметим, что эта система моделей используется как на этапе извлечения информации из текстов, так и в процессе генерации результатов обработки.

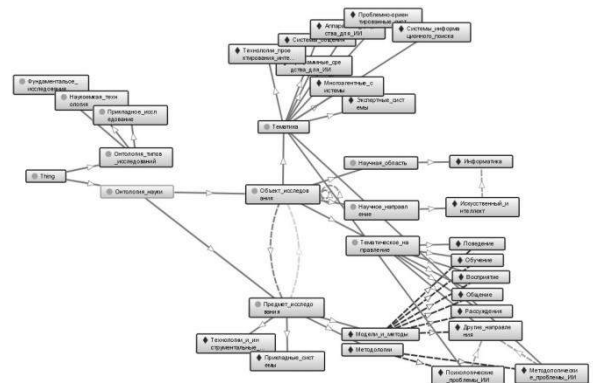


Рисунок 10 – Фрагмент онтологической модели

2.3. Технология построения карт научных направлений

Как показывает анализ литературы по средствам автоматизированного построения карт науки, в общей схеме обработки информации целесообразно выделять этапы

- выявления центров компетенции (включая организации и авторские коллективы) в предметной области,
- формирования репрезентативной коллекции документов с учетом выявленных на предыдущем этапе центров превосходства и компетенции и
- собственно обработки сформированных коллекций документов.

Технология построения семантических карт научного направления представлена на Рис. 11.



Рисунок 11 – Технологии построения семантических карт научного направления

В качестве инструментария для извлечения информации из текстов использована платформа GATE, расширенная плагином русской морфологии, а также специально разработанными модулями обработки именных групп и генерации статистических портретов документов.

Для удобства анализа результатов обработки коллекций документов был разработан специальный модуль генерации OWL-представлений, которые загружались в систему Protégé⁹ и использовались для дальнейшего анализа. При построении карты научного направления OSTIS в технологической цепочке использовался кластеризатор Carrot¹⁰.

2.4. Методика обработки информации и анализа результатов

Методика обработки коллекции публикаций OSTIS состояла в следующем:

- Труды конференций за каждый год конвертировались в XML-представление, принятое в качестве стандарта для публикаций Web of Science.
- Сконвертированные труды обрабатывались гибридным модулем извлечения информации из текстов, на выходе которого формировались статистические и семантические портреты публикаций данного тома.
 - Результаты загружались в систему Carrot для обработки с помощью разных методов кластеризации и визуализации состава отдельных кластеров и взаимосвязей между ними;
 - Protégé для построения карт геосемантики направлений и выявления в нем «скрытых коллективов».
- На статистических портретах отдельных конференций строились временные ряды терминов направления OSTIS.

Анализ полученных результатов выполнялся экспертами, которые формировали заключение по картированию научного направления и центрах компетенций и превосходства в данной области.

⁹ <http://www.protege.com>

¹⁰ <http://www.Carrot.com>

3. Наукометрический анализ конференций OSTIS

3.1. Корпус публикаций OSTIS

Как отмечалось выше, для наукометрического анализа направления «Открытые семантические технологии проектирования интеллектуальных систем» (OSTIS) были выбраны труды профильных конференций этой серии за 2012-2014 гг. Общий объем OSTIS-корпуса составил 273 статьи.

Все опубликованные статьи были обработаны с помощью специально разработанного конвертора, на вход которого подавались исходные тексты, а на выходе формировались их XML-представления. Для примера, на Рис. 12 приведен фрагмент одного из результатов таких XML-представлений.

```
<?xml version="1.0" encoding="UTF-8"?>
<Document>
  <meta>
    <prop name="uid">OSTIS_2013.txt</prop> ...
  </meta>
  <record>
    <PT>J</PT>
    .....
    <AF>Грибова В.В., Клешев А.С.</AF>
    <TI>ОБЛАЧНЫЕ ...</TI>
    <LA>Russian</LA>
    <AB>В работе описана ...</AB>
    <C1>Грибова В.В., Клешев А.С.] Федеральное
      государственное ..., г. Владивосток, Россия.</C1>
    <CR>
      [Bachant et al., 1984] Bachant, J. McDermott, J.
      R1 revisited: four years ...
      .....
      [Грибова и др., 2011] Грибова В.В., Клешев А.С.,
      Крылов и др. Проект IACPaas – развиваемый ...
    </CR>
    <PY>2013</PY>
    <WC>интеллектуальные системы; облачные технологии;
      семантические технологии; онтологии; базы знаний</WC>
  </record>
  .....
</Document>
```

Рисунок 12 – Фрагмент XML-представления статьи

В соответствии с представленной выше технологией полученные XML-представления обрабатывались гибридными IE-системами с целью выявления семантически значимых объектов и связей между ними, что позволило построить статистические и семантические портреты OSTIS, а также провести его наукометрический анализ.

3.2. Статистические портреты OSTIS

3.2.1. Общий портрет направления

Структура публикаций по исследованиям и разработкам в области открытых семантических технологий для проектирования интеллектуальных систем, как она представлена в трудах конференций серии OSTIS (после авторского обобщения тематики и направлений секций), зафиксирована в Табл. 1.

Понятно, что использованные для заполнения таблицы исходные рубрикаторы секций, не претендует на полноту охвата данной предметной области, но дают основу для оценки ситуации. Заметим также, что таблица формировалась на основе оглавлений трудов конференций и не всегда

соответствует общему числу действительно обработанных статей.

Таблица 1 – Структура публикаций конференций OSTIS

	2012	2013	2014
Семантические модели представления и обработки знаний и их программная и аппаратная реализация	13	16	10
Семантические технологии проектирования баз знаний, программ и пакетов программ.	13	16	
Семантические модели поиска, классификации/кластеризации и решения задач.	9		9
Компьютерная лингвистика и семантические технологии проектирования мультимодальных и ЕЯ-интерфесов.	16	20	16
Онтологическое моделирование и онтологический инжиниринг.	6	10	15
Прикладные интеллектуальные системы.	12		31
Логико-семантические модели.		20	10
Научное наследие Мартынова.			4
Когнитивное моделирование.			6

3.2.2. Статистика авторов и геостатистика

Статистические портреты, представленные в Табл. 2, сформированы на основании авторских указателей в трудах конференций OSTIS, а остальные параметры получены в результате автоматической обработки OSTIS-корпуса.

Таблица 2 – Статистика авторов и геостатистика

	2012	2013	2014
Авторы публикаций	130	169	171
Страны	6	6	6
Города	20	27	30
Организации ¹¹	53	57	77

Заметим, что в каждой из конференций серии OSTIS участвовало 6 стран, но лишь 4 из них (Россия, Беларусь, Украина и Казахстан) были участниками всех конференций. Остальные две страны – следующие: Болгария, Испания (2012); Китай, Таджикистан (2013); США, Латвия (2014).

3.3. Семантические портреты OSTIS

В соответствии с методикой картирования научных направлений, представленной выше, для построения семантических портретов OSTIS все публикации одного года обрабатывались совместно под управлением онтологии предметной области (Рис. 10). Полученные результаты обсуждаются в следующих подразделах.

3.3.1. Геосемантика

Для выявления геосемантики OSTIS из общей онтологии направления с помощью специальных запросов были выделены все объекты типа Location, связанные отношениями locatedIn, в результате чего были сформированы общие карты, представленные на Рис. 13.

Для более детального обсуждения полученных результатов из общих карт были сформированы карты странового участия в конференциях серии OSTIS. Для примера на Рис. 14 представлена карта геосемантики Беларуси, которая демонстрирует явно выраженное увеличение представительства организаций из этой страны, что, в свою очередь, демонстрирует развитие белорусского кластера исследований и разработок в данной области

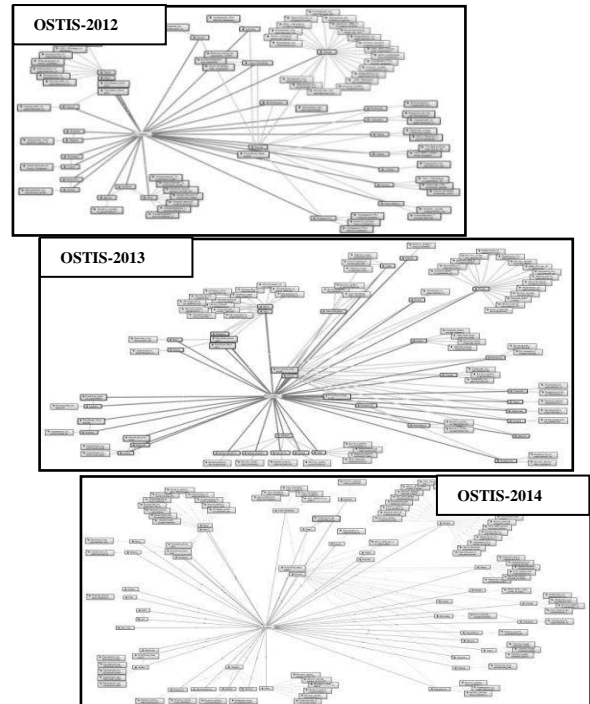


Рисунок 13 – Семантическая карта OSTIS: «Страны-Города-Организации»

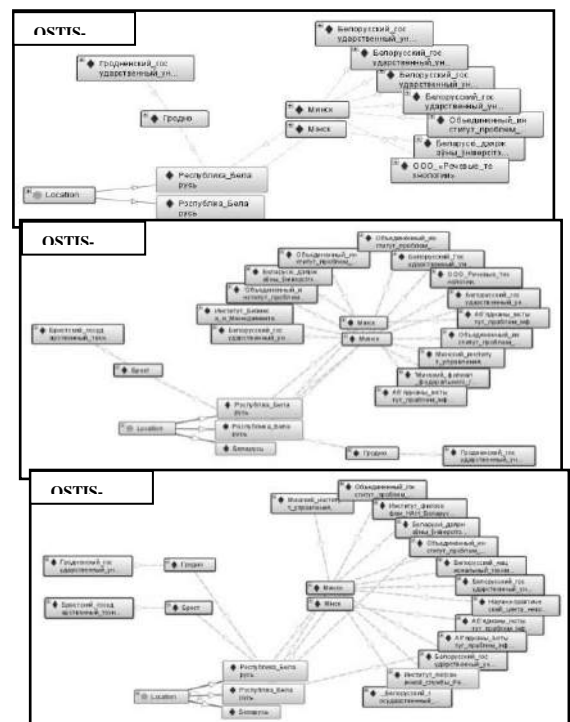


Рисунок 14 – Семантическая карта OSTIS: «Беларусь-Города-Организации»

¹¹ При подсчете числа синонимы названий организаций не учитывались.

Аналогичные тренды наблюдаются и для представительства организаций, участвовавших в конференциях серии OSTIS из других стран.

3.3.2. «Скрытые коллективы»

Для поиска в корпусе авторов, представленных в трудах конференций серии OSTIS, из общей онтологии направления с помощью специальных запросов были выделены все объекты типа Author и Reference, связанные отношениями beCoauthor и referencedBy. locatedIn, в результате чего были сформированы общие карты цитирования. Для примера на Рис. 15 показана такая карта для конференции OSTIS-2012.

Как показывает анализ данной карты цитирования, «хорошей» визуализации результатов

выявления «скрытых коллективов» с помощью стандартных средств системы онтологического инжиниринга Protégé (в силу значительного числа присутствующих в онтологии объектов и отношений) достичь практически невозможно.

Поэтому в данном случае авторы пошли по пути уменьшения структурной сложности визуализации за счет выделения с помощью специальных запросов фрагментов сетей цитирования для отдельных авторов, представленных в трудах конференций серии OSTIS.

Для примера, на Рис. 16 показана сети цитирования для автора Голенков, а на Рис. 17 и Рис. 18 – для авторов Гаврилова и Хорошевский.

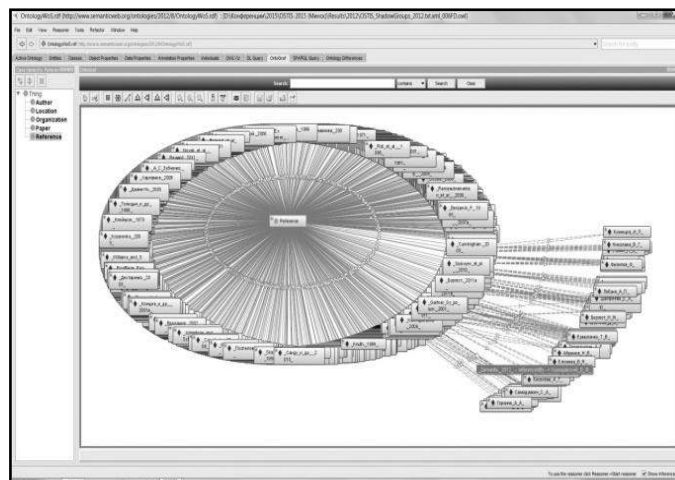


Рисунок 15 – Семантическая сеть цитирования (OSTIS-2012)

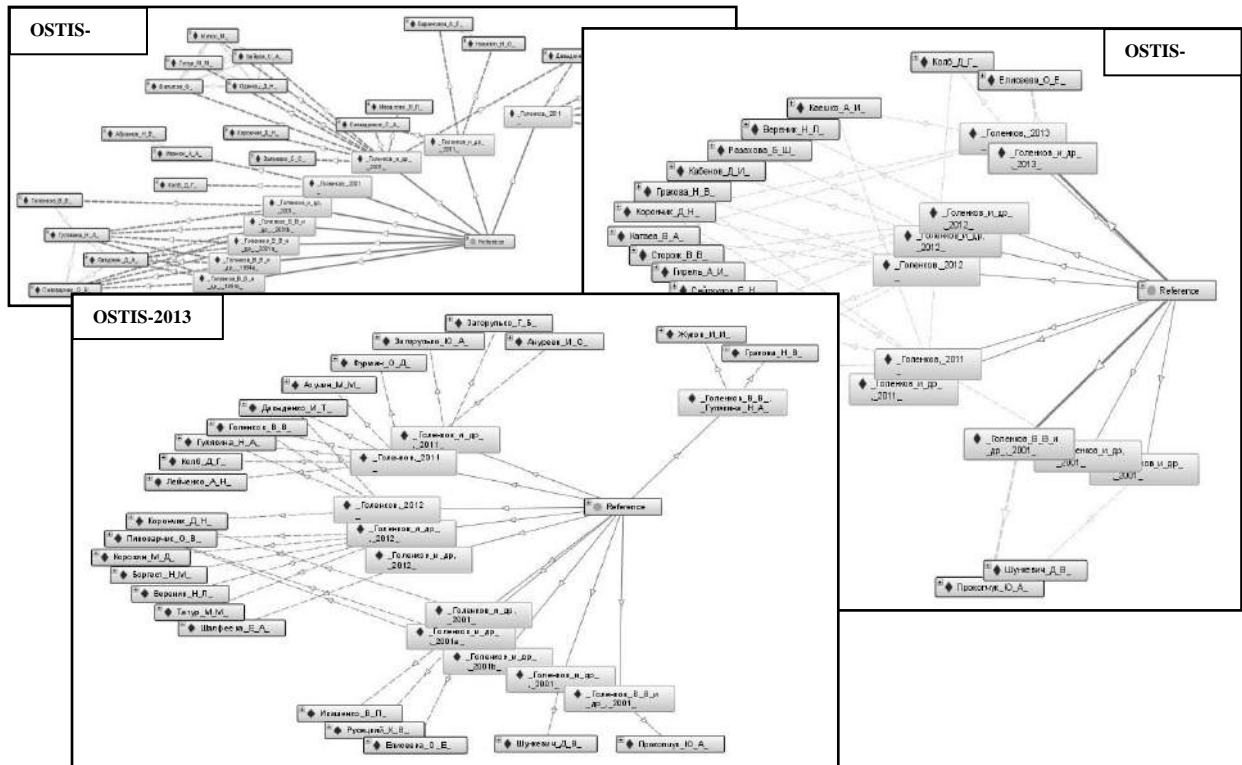


Рисунок 16 – Динамика развития сетей цитирования для автора Голенков

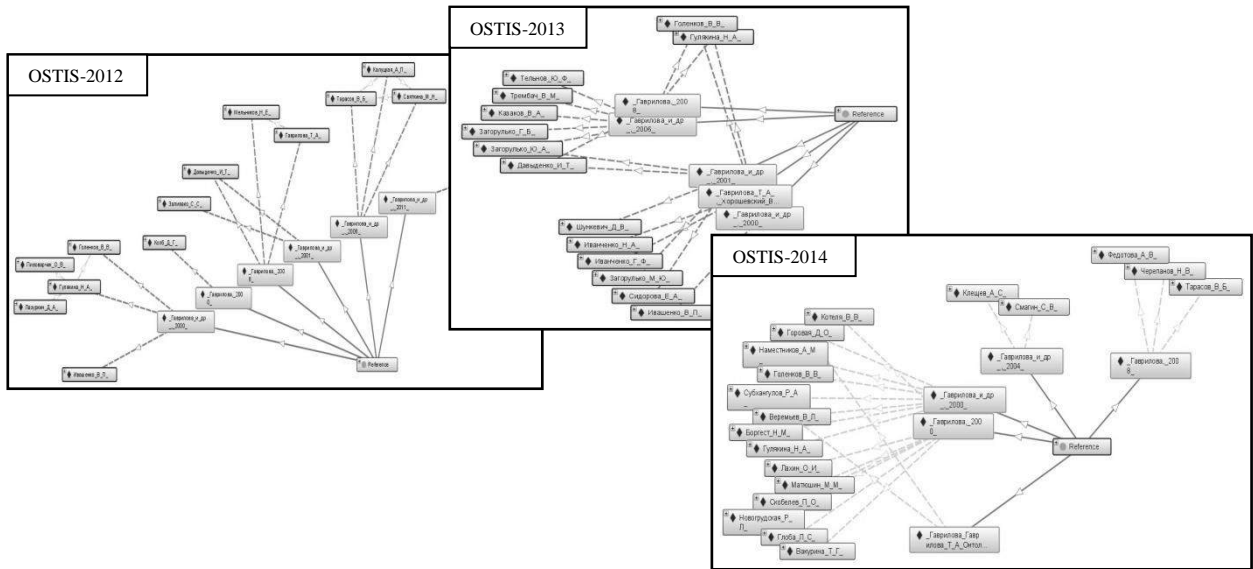


Рисунок 17 – Динамика развития сетей цитирования для автора Гаврилова

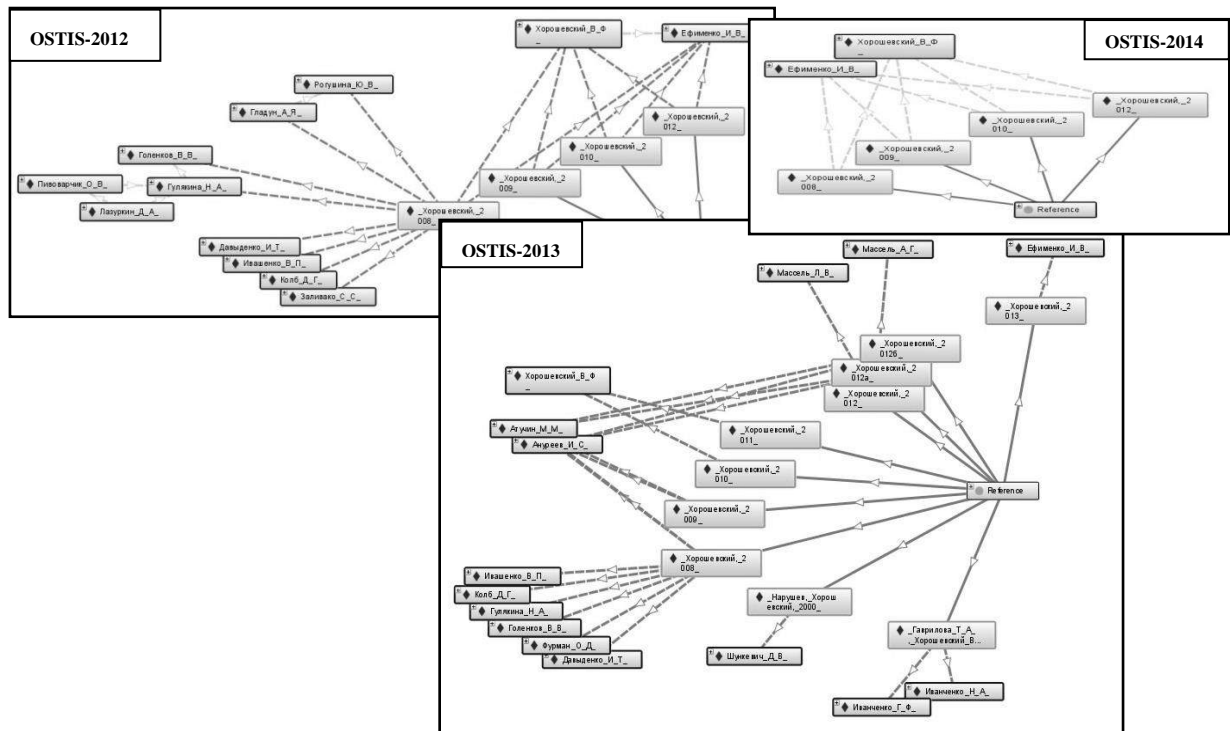


Рисунок 18 – Динамика развития сетей цитирования для автора Хорошевский

Как показывает анализ сетей цитирования, представленных на Рис. 16, для данного автора наблюдается хорошая динамика изменения состава цитируемых публикаций. Так, например, в 2012 г. основные цитирования датированы 1998г. и 2001г., в 2013 и 2014 гг. наблюдается сдвиг датирования цитируемых работ на 2011г. и 2012г. В 2014г. появляются цитирования работ данного автора (и с соавторами), опубликованных в 2013г. Таким образом, можно констатировать, что авторы публикаций в трудах конференций серии OSTIS достаточно активно используют результаты работ В.В. Голенкова (и с соавторами) и отслеживают новые работы данного автора.

Как показывает анализ сетей цитирования, представленных на Рис. 17-18, для данных авторов

наблюдается иная динамика изменения состава цитируемых публикаций.

Так, для Т.А. Гавриловой (одного из самых известных в России и за ее пределами специалиста в области онтологического инжиниринга) наиболее цитируемой публикацией в трудах всех конференций серии OSTIS является совместная с В.Ф. Хорошевским монография [Гаврилова и др., 2000] и переиздание этой монографии в 2001г., а из других публикаций цитируются только публикация по онтологическому инжинирингу на сайте¹² и работы [Гаврилова и др., 2006; Гаврилова и др., 2008]. Такая ситуация может объясняться, с одной стороны, важностью выше упомянутой монографии,

¹² <http://www.big.spb.ru/publications/bigspb/km/>

которая уже стала классической, а с другой – недостаточным доступом к другим публикациям данного автора.

Несколько другая, но, вместе с тем, тоже не лучшая ситуация наблюдается и для динамики цитирования в трудах конференций серии OSTIS публикаций В.Ф. Хорошевского. И здесь, в основном, цитируется уже упоминавшаяся монография по базам знаний интеллектуальных систем, а также обзорные работы [Хорошевский, 2008; Хорошевский, 2009; Хорошевский, 2012a]. При этом достаточно свежие работы данного автора в области семантических технологий [Хорошевский, 2012b; Хорошевский, 2013] имеют либо автоссылки, либо ссылки его соавторов по другим работам.

Вместе с тем, анализ сетей цитирования, полученных в результате обработки трудов конференций серии OSTIS, показывает, что таких специалистов, как В.В. Голенков, Т.А. Гаврилова и В.Ф. Хорошевский (с соавторами) можно идентифицировать в качестве центров «скрытых коллективов» в области семантических технологий.

Кластеры направлений

Кластеризация направлений, представленных в трудах конференций серии OSTIS, осуществлялась с использованием гибридного подхода. При этом целью обработки каждой статьи является

формирование ее семантического портрета в виде «мешка слов» (Bag of Words). В нашем случае такие портреты формируются из авторских ключевых слов, а также из статистически значимых терминов, выделенных из названий статей, аннотаций и названий работ в ссылках. Для каждого года OSTIS семантические портреты отдельных статей объединяются в общий портрет, фрагмент которого приведен на Рис. 19.

```
<?xml version="1.0" encoding="UTF-8"?>
<searchresult>
  <query>OSTIS_2013.txt.xml_000E3</query>
  <document id="СИСТЕМА-УПРАВЛЕНИЯ...">
    <title>СИСТЕМА-УПРАВЛЕНИЯ...</title>
    <url>D:/ScienceMap/СИСТЕМА-УПРАВЛЕНИЯ...СИСТ
    <snippet>система управление проектирование интеллект
  </document>
  .....
  <document id="ПОСТРОЕНИЕ...">
    <title>ПОСТРОЕНИЕ...</title>
    <url>D:/ScienceMap/ПОСТРОЕНИЕ...ПРОЦЕССОРА.txt
    <snippet>аппаратный реализации; интеллектуальный сис
  </document>
</searchresult>
```

Рисунок 19 – Фрагмент семантического портрета OSTIS-2013

Полученные портреты обрабатывались по годам с помощью кластеризатора Carrot. Ниже обсуждаются результаты кластеризации портретов OSTIS, представленные на Рис. 20-21.



Рисунок 20 – FoamTree представление результатов кластеризации направлений OSTIS (метод STC)

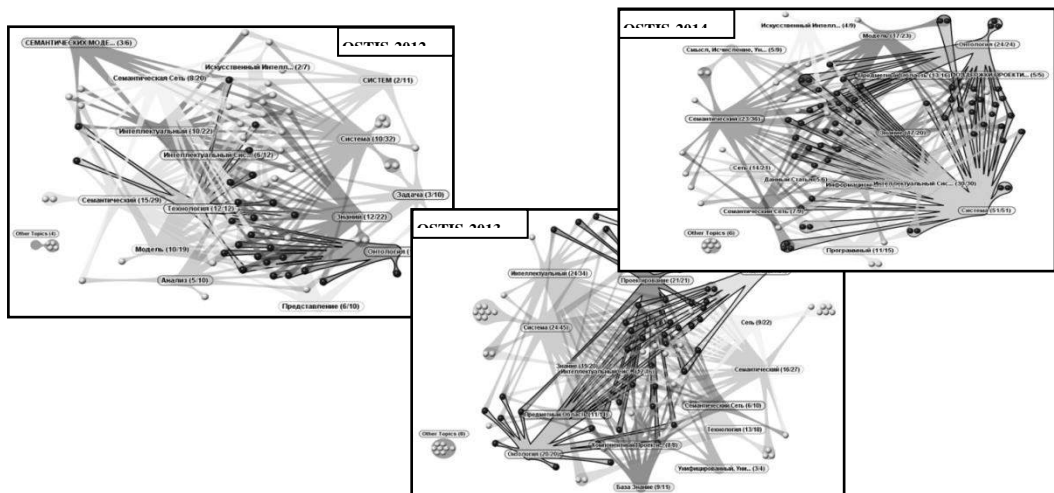


Рисунок 21 – AdunaMap представление результатов кластеризации направлений OSTIS (метод STC)

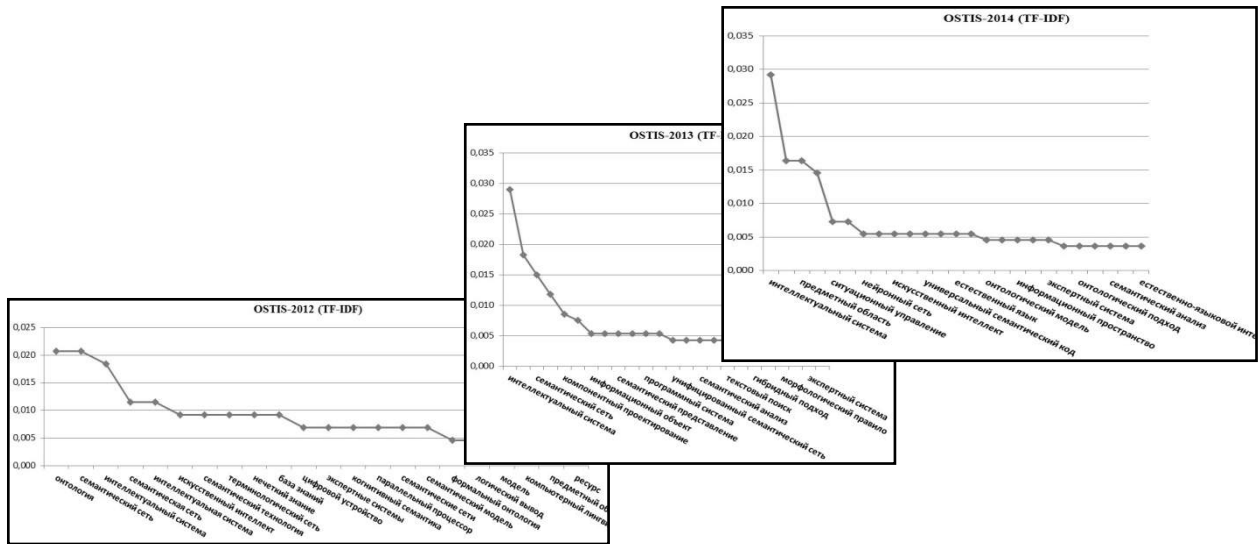


Рисунок 22 – Представление 25 наиболее частотных терминов OSTIS по параметру TF-IDF

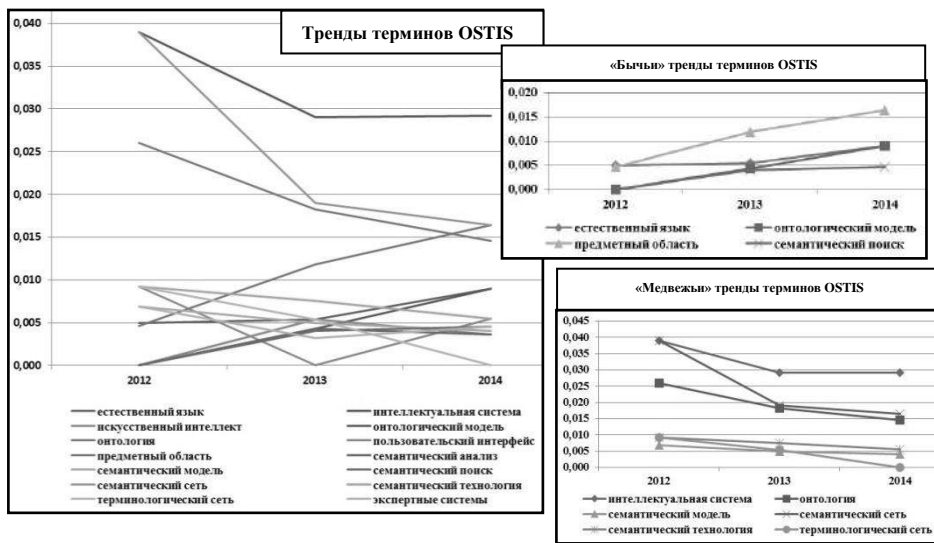


Рисунок 23 – Паттерны данных терминов OSTIS

Результаты кластеризации, представленные в виде, аналогичном картам Кохонена (Рис. 20), показывают, что наиболее значимыми для данного направления являются топики «Система», «Модель» и «Онтология», что, впрочем, является ожидаемым.

Гораздо более интересным является представление тех же результатов в виде карт взаимосвязи отдельных публикаций портретов OSTIS с различными кластерами (Рис. 21). Во всех случаях здесь наблюдается значительный «вклад» портретов отдельных публикаций в кластеры «Онтология», «Система» и «Модель». Отсюда следует, что в конференциях серии OSTIS значительное число работ представляет результаты на стыке направлений, что подтверждается экспертами в области всего направления OSTIS.

Паттерны данных терминов

Использование гибридных методов наукометрии позволило авторам провести еще одно исследование полученных результатов – анализ паттернов данных, построенных на статистически значимых множествах терминов, выявленных в портретах по

отдельным годам OSTIS-конференций. Результаты этого исследования обсуждаются ниже и представлены на Рис. 22-23.

Представленные на Рис. 22 кривые показывают, что по параметру TF-IDF наиболее значимыми терминами в 2012г. были «Онтология», «Семантические сети» и «Интеллектуальные системы», в 2013г. – «Интеллектуальные системы», «Семантические сети» и «Компонентное проектирование», а в 2014г. – «Интеллектуальные системы», «Предметная область» и «Ситуационное управление».

Паттерны данных терминов OSTIS, представленные на Рис. 23, демонстрируют разное «поведение». Так, термины «Естественный язык», «Предметная область», «Онтологическая модель» и, частично, «Семантический поиск» относятся к «бычьим» трендам, а термины «Интеллектуальные системы», «Онтология», «Семантическая модель», «Семантическая сеть», «Семантические технологии» и «Терминологические сети» – к «медвежьим» трендам. Следует сразу отметить, что

для серьезных выводов и обобщений данных мало, но можно предположить, что отмеченные тенденции связаны с естественной диверсификацией терминов.

Проблемы и направления дальнейших исследований

В современной наукометрии существует много проблем, среди которых можно выделить:

- организационно-технические проблемы и
- проблемы чисто научного плана.

В первом классе проблем явно доминируют сложности получения представительных и качественных электронных корпусов исходных данных, поскольку публикации существенно отличаются и по форматам, и по качеству их оформления. Для примера можно отметить, что авторы статей, опубликованных в трудах конференций серии OSTIS, далеко не всегда следуют заданным организаторами стандартам. Особенно часто наблюдаются несоответствие и ошибки в оформлении списков литературы к статьям, а также «вольная» трактовка авторами названий организаций, в которых они работают, и геоимен. Следует также отметить недооценку авторами таких разделов статьи, как аннотация и списки ключевых терминов. К сожалению, эти проблемы в большей степени характерны для нашей действительности. В странах с устоявшимися научными традициями большинство из указанных проблем уже решены – как за счет автоматизации процессов приема авторских материалов для публикаций, так и более ответственным отношением самих авторов к процессам подготовки и оформления статей.

Вместе с тем, в современной наукометрии существуют и чисто научные проблемы, к числу которых, в первую очередь, относятся:

- разработка эффективных и надежных методов идентификации авторов и их аффилиций,
- разработка новых моделей семантизации публикаций,
- разработка новых моделей и методов оценки публикационной активности,
- разработка новых моделей и методов выявления и оценки перспективных направлений исследований и разработок, а также
- создание инструментов наукометрии нового поколения, где классические методы были бы интегрированы с методами искусственного интеллекта и компьютерной лингвистики.

С учетом вышесказанного и анализа полученных в рамках настоящего исследования результатов, авторы планируют в дальнейшем сосредоточиться на разработке и реализации более мощных и адекватных методов и средств использования семантических технологий в наукометрии.

Заключение

В работе представлено обсуждение вопросов

картирования научных направлений с использованием семантических технологий на примере анализа корпуса статей, опубликованных в трудах конференций серии OSTIS «Open Semantic Technologies for Intelligent Systems». Для понимания изложения дано краткое описание основных понятий наукометрии и обоснована важность применения семантических технологий в решении возникающих здесь задач. Полученные результаты показывают, что предложенные модели, методы и средства анализа научных направлений и выявления центров компетенций и центров превосходства на базе семантизации методов наукометрии позволяют оценить и статику, и динамику развития исследований и разработок в определенных предметных областях, что должно обеспечить автоматизацию процессов поддержки принятия решений.

Авторы считают своим приятным долгом поблагодарить административную группу сайта OSTIS за оперативное предоставление электронных версий трудов всех конференций OSTIS без чего настоящее исследование было бы невозможным.

Работа выполнена при частичной поддержке гранта РФФИ № 15-01-06819 «Исследование и разработка онтологических моделей центров компетенции/превосходства в прорывных научно-технологических направлениях на основе мониторинга разнородных информационных ресурсов».

Библиографический список

- [Borner et al., 2003] Borner, K., Chen, C., & Boyack, K. W. (2003). Visualizing knowledge domains. *Annual Review of Information Science and Technology* 37, 179-255.
- [Borner et al., 2012] Borner, K., Boyack, K. W., Milojevic, S., & Morris, S. (2012). An introduction to modeling science: Basic model types, key definitions, and a general framework for the comparison of process models. / *Modeling Science Dynamics (Understanding Complex Systems)*, Springer-Verlag, p 3-22.
- [Boyack et al., 2005] Boyack, K. W., Klavans, R., & Borner, K. (2005). Mapping the backbone of science. *Scientometrics*, 64(3), 351-374.
- [Boyack et al., 2013] Boyack, K. W., Small, H., & Klavans, R. (2013). Improving the accuracy of co-citation clustering using full text. *Journal of the American Society for Information Science and Technology*, 64(9), 1759-1767.
- [Boyack et al., 2014] Boyack, K. W., & Klavans, R. (2014). Creation of a highly detailed, dynamic, global model and map of science. *Journal of the Association for Information Science and Technology*, 65(4), 670-685.
- [Boyack, 2009] Boyack, K. W. (2009). Using detailed maps of science to identify potential collaborations. *Scientometrics*, 79(1), 27-44.
- [Efimenko et al., 2014] Efimenko I., Khoroshevsky V. New Technology Trends Watch: an Approach and Case Study. In: *Proc. of AIMS-2014*.
- [Erdi et al., 2013] Erdi P., Makovi K., and et al. Prediction of emerging technologies based on analysis of the US patent citation network. *Scientometrics*, 2013, 95 (1), pp. 225-242 (2013).
- [Klavans et al., 2006] Klavans, R., & Boyack, K. W. (2006). Quantitative evaluation of large maps of science. *Scientometrics*, 68(3), 475-499.
- [Klavans et al., 2010] Klavans, R., & Boyack, K. W. (2010). Toward an objective, reliable, and accurate method for measuring research leadership. *Scientometrics*, 82(3), 539-553.
- [Klavans et al., 2011] Klavans, R., & Boyack, K. W. (2011). Using global mapping to create more accurate document-level maps

of research fields. Journal of the American Society for Information Science and Technology, 62(1), 1-18.

[Klavans et al., 2014a] Klavans R., Boyack K. W. Scientific Superstars and their Effect on the Evolution of Science. // Proc. Of ENID STI Conference, Rome, Italy, 2011, p. 7-9.

[Klavans et al., 2014b] Klavans R., Boyack K.W., Small H. Indicators and precursors of hot science. // Proc. Of 17th International Conference on Science and Technology Indicators, Leiden, Netherlands, 2014, p.475-487.

[Li, et al., 2011] Li H., Xu F., Uszkoreit H.: TechWatchTool: Innovation and Trend Monitoring. In: Proc. of the International Conference on Recent Advances in Natural Language Processing 2011 RANLP 2011, Tislar, Bulgaria, pp. 660-665 (2011).

[Shibata, et al., 2008] Shibata N., Kajikawa Yu., Takeda Y., Matsushima K.: Detecting emerging research fronts based on topological measures in citation networks of scientific publications, Technovation, Vol. 28, Issue 11, November 2008, pp. 758–775 (2008).

[Small, 2010] Small, H. (2010). Maps of science as interdisciplinary discourse: Co-citation contexts and the role of analogy. Scientometrics, 83(3), 835-849.

[Upham et al., 2010] Upham, S. P., & Small, H. (2010). Emerging research fronts in science and technology: Patterns of new knowledge development. Scientometrics, 83(1), 15-38.

[Van Eck et al., 2014] Van Eck, N.J., & Waltman, L. (2014). CitNetExplorer: A new software tool for analyzing and visualizing citation networks. Journal of Informetrics, 8(4), 802-823.

[Wang et al., 2010] Wang et al. Identifying technology trends for RD planning using TRIZ and text mining, RD Management, vol. 40, N 5, 2010.

[Witten et al., 2011] Witten Ian H., Frank Eibe and Hall Mark A. Data Mining: Practical Machine Learning Tools and Techniques. — 3rd Edition. — Morgan Kaufmann, 2011. — P. 664.

[Гаврилова и др., 2000] Гаврилова, Т.А. Базы знаний интеллектуальных систем/ Т.А. Гаврилова, В.Ф. Хорошевский //СПб – Питер, 2000 г., 384 с.

[Гаврилова и др., 2006] Гаврилова, Т.А. Модели и методы формирования онтологий / Т.А. Гаврилова, Д.В. Кудрявцев, В.А. Горовой // Научно-технические ведомости СПбГПУ, № 4, 2006. – С.21-28.

[Гаврилова и др., 2008] Гаврилова, Т. А. Визуальные методы работы со знаниями: попытка обзора / Т. А. Гаврилова, Н. А. Гулякина // Искусственный интеллект и принятие решений, 2008, № 1, С. 15-21.

[Евстигнеев, 1987] Евстигнеев В.А. Методы теории графов в наукометрии: исследование структуры пространства журналов и незримых коллективов в программировании. // Новосибирск, 1987. (Препр. АН СССР. Новосибир. филиал. ИТМ и ВТ им. С.А.Лебедева; № 4).

[Егоров и др., 2006] Егоров В. С., Пожидаев А. В., Чернобровская Т. Н. Систематизация и использование сведений о научных мероприятиях в автоматизированной технологии ВИНТИ. // НТИ. Сер. 1. – 2006. – №4.– С.17-23.

[Крюков и др., 2013] Крюков К.В., Кузнецов О.П., Суховеров В.С. О понятии формальной компетентности научных сотрудников. // Труды международной конференции OSTIS-2013, Минск, Беларусь, 2013.

[Кулинич, 2011] Кулинич А.А. Компьютерные системы анализа ситуаций и поддержки принятия решений на основе когнитивных карт: подходы и методы./ Проблемы управления, 2011, № 4 С.31-45.

[Маршакова, 1988] Маршакова И.В. Система цитирования научной литературы как средство слежения за развитием науки. — М.: Наука, 1988. 288 с.

[Налимов и др., 1969] Налимов В.В. Мульченко З. М. Наукометрия. Изучение науки как информационного процесса / В. В. Налимов, З. М. Мульченко. — М.: Наука, 1969. — 192 с.

[Налимов и др., 1971] Налимов В.В., Кордон И.В., Корнеева А.Я. Географическое распределение научной информации // Информационные материалы Научного совета по комплексной проблеме "Кибернетика" АН СССР. 1971. № 2 (49). С. 3-37.

[Паклин и др., 2009] Паклин Н. Б., Орешков В. И. Бизнес-аналитика: от данных к знаниям (+ CD). — СПб.: Изд. Питер, 2009. — 624 с.

[Хайтун, 1989] Хайтун С.Д. Проблемы количественного анализа науки. - М.: Наука, 1989. 280 с.

[Хорошевский и др., 2014] Хорошевский В.Ф., Ефименко И.В. Искусственный интеллект: карта научного направления в трудах конференций РАИИ. // Труды 14-й национальной конференции по искусственному интеллекту с международным участием, КИИ-2014, Том 1. С. 160-168. Казань, Россия

[Хорошевский, 2008] Хорошевский, В.Ф. Пространства знаний в сети Интернет и Semantic Web (Часть 1) // Искусственный интеллект и принятие решений. - 2008. - № 1. - С.80-97.

[Хорошевский, 2009] Хорошевский В.Ф. Пространства знаний в сети Интернет и Semantic Web (Часть 2) // Искусственный Интеллект и Принятие решений, № 4, 2009, с.15-36.

[Хорошевский, 2012b] Хорошевский В.Ф. Выявление новых технологических трендов: проблемы и перспективы. // Труды 13-й Конференции по Искусственному Интеллекту с международным участием, КИИ-2012, Том 1, с. 252-259. Белгород, Россия, 2012. Изд-во БГТУ, 2012.

[Хорошевский, 2012a] Хорошевский В.Ф. Пространства знаний в сети Интернет и Semantic Web (Часть 3) // Искусственный Интеллект и Принятие решений, № 1, 2012, с.3-38.

[Хорошевский, 2013] Хорошевский В. Ф. Автоматизация процессов выявления технологических трендов в системе АРМ Тренд. // Материалы III международной научно- технической конференции «Открытые семантические технологии проектирования интеллектуальных систем (OSTIS-2012)». Минск, 2013. С. 233–241.

[Юрков, 2015] Юрков А. В. Интернет-аналитика для прикладной наукометрии. // Труды международной конференции SCIENCE INDEX 2015: аналитические инструменты и сервисы для оценки научной деятельности, 17-24 января 2015 г., Андорра. http://elibrary.ru/projects/conference/andorra2015/conf_2015_1_presentations.asp

SCIENTOMETRICS OF THE DOMAIN: OSTIS-CONFERENCES CASE STUDIES

Khoroshevsky V.F. *, Efimenko I.V. **

**Dorodnicyn Computing Centre RAS, Moscow,
Russia*

khor@ccas.ru

***Faculties of Humanities, NRU HSE, Moscow,
Russia*

iefimenko@hse.ru

The paper discusses “maps of science” based on advanced scientometrics methods using semantic technologies. Text collections of articles published in the proceedings of OSTIS Conference («Open Semantic Technologies for Intelligent Systems», 2012-2014) are used as input data. The proposed models, methods and tools are oriented towards identification of centers of excellence based on “scientometrics semantization” techniques. The results obtained allow users to analyze static and dynamic aspects of research and development in specific S&T fields, which is a basis for the decision support.



OSTIS-2015

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822:514

СЕМАНТИЧЕСКАЯ ТЕХНОЛОГИЯ КОМПОНЕНТНОГО ПРОЕКТИРОВАНИЯ СИСТЕМ, УПРАВЛЯЕМЫХ ЗНАНИЯМИ

Голенков В.В., Гулякина Н.А.

*Белорусский государственный университет информатики и радиоэлектроники,
г. Минск, Республика Беларусь*

golen@bsuir.by
guliakina@bsuir.by

В работе рассматривается итог пятилетнего развития Проекта OSTIS, направленного на создание Открытой семантической технологии проектирования интеллектуальных систем. В основе указанной технологии лежит представление знаний в виде унифицированных семантических сетей с теоретико-множественной интерпретацией. В работе рассматриваются классы систем, основанных на знаниях, и систем, управляемых знаниями.

Ключевые слова: системы, управляемые знаниями, компонентное проектирование, технология проектирования, представление знаний.

Введение

Данная работа подводит итог пятилетнего развития Проекта OSTIS (Open Semantic Technology for Intelligent Systems), целями которого являются:

- Разработка Технологии OSTIS, позволяющей быстро и качественно разрабатывать семантически совместимые компьютерные системы, управляемые знаниями, и способные создавать временные коллективы компьютерных систем для распределенного решения сложных задач.
- Определение круга актуальных приложений таких систем.
- Использование Технологии OSTIS как основы практико-ориентированной подготовки студентов и магистрантов.
- Уточнение принципов взаимодействия науки, образования, инженерии и бизнеса в развитии рынка компьютерных систем, управляемых знаниями.

Под семантической технологией компонентного проектирования компьютерных систем мы понимаем комплекс согласованных частных технологий, обеспечивающий целостное проектирование компьютерных систем, управляемых знаниями, и включающий:

- семантическую технологию компонентного проектирования баз знаний;

- семантическую технологию компонентного проектирования программ, ориентированных на обработку баз знаний;
- семантическую технологию компонентного проектирования решателей задач в компьютерных системах, управляемых знаниями;
- семантическую технологию компонентного проектирования пользовательских интерфейсов, обеспечивающих общение компьютерных систем, управляемых знаниями, с пользователями (в т.ч. и на естественных языках);
- семантическую технологию компонентного проектирования подсистем мультисенсорного восприятия и анализа информации о внешней среде компьютерных систем, управляемых знаниями;
- семантическую технологию компонентного проектирования подсистем координируемого воздействия на внешнюю среду компьютерных систем, управляемых знаниями.

1. Компьютерные системы, управляемые знаниями

Компьютерная система, управляемая знаниями, – система, в основе которой лежит представленная унифицированным образом база знаний, содержащая в систематизированном виде всю информацию, используемую этой системой. Таким образом, в компьютерных системах,

управляемых знаниями, вся используемая и обрабатываемая информация представляется в виде семантически структурированной целостной базы знаний, отражающей на смысловом уровне полную картину Мира, в котором "живет" эта компьютерная система. Решатель задач компьютерной системы, управляемой знаниями, (машина обработки знаний) представляет собой коллектив самостоятельных агентов, взаимодействующих между собой только через базу знаний. Некоторые агенты могут взаимодействовать и с внешней средой с помощью рецепторных и эффекторных средств. Указанный характер взаимодействия рассматриваемых агентов позволяет трактовать систему, управляемую знаниями, как систему, управляемую своей базой знаний. Память компьютерных систем, управляемых знаниями, носит структурно перестраиваемый ассоциативный характер и в перспективе может быть реализована не только программно (как в настоящий момент), но и аппаратно. Для реализации обработки знаний в такой памяти (в частности, для описания поведения агентов) необходимы языки программирования, ориентированные на обработку знаний в структурно перестраиваемой ассоциативной памяти, программы которых должны рассматриваться как частный вид знаний, хранимых в составе единой базы знаний. Интерпретатор одного из таких языков программирования (базового языка программирования) при аппаратной реализации структурно перестраиваемой ассоциативной памяти должен быть реализован тоже аппаратно. Существенно то, что при переходе на любую новую платформу реализации структурно перестраиваемой памяти и интерпретатора соответствующего базового языка программирования (в том числе, и при переходе на аппаратную их реализацию) никаких изменений в базу знаний и в решатель задач конкретной системы, управляемой знаниями, вносить не потребуется.

система, управляемая знаниями

- = система, управляемая своей базой знаний
- = система с активной базой знаний, ситуации и события в которой инициируют процессы ее обработки, выполняемые коллективом агентов
- = система, в основе которой лежит база знаний, управляющая деятельностью работающих над ней агентов
- = многоагентная компьютерная система, агенты которой управляются ее базой знаний
- = многоагентная компьютерная система, агенты которой взаимодействуют только через ее базу знаний
- = многоагентная компьютерная система, агенты которой обмениваются информацией только через ее базу знаний
- = многоагентная компьютерная система, агенты которой инициируются событиями или

ситуациями либо в базе знаний, либо во внешней среде

- = система, в основе которой лежит база знаний и коллектив работающих над ней и взаимодействующих через нее агентов
- = система, основанная на семантически структурированных данных и метаданных, управляющих процессом их обработки
- = система, основанная на связанных, формально представленных, хорошо структурированных данных и метаданных, управляющих процессом их обработки
- = система, в основе которой лежит база знаний, содержащая всю используемую ею информацию и организующая взаимодействие работающих над ней агентов
- = система, архитектура которой состоит из базы знаний и коллектива агентов, работающих над этой базой знаний и взаимодействующих только через нее
- = система, в которой хранящиеся в ней знания полностью управляют ее деятельностью, инициируя активность различных агентов, осуществляющих обработку этих знаний
- = система, в которой управляющими воздействиями являются события или ситуации, которые возникают в обрабатываемой базе знаний и которые инициируют деятельность соответствующих агентов, работающих над этой базой знаний (каждый агент реагирует на свой класс ситуаций или событий)
- = система, обрабатываемая база знаний которой выполняет роль среды, управляющей деятельностью и взаимодействием агентов, работающих над этой базой знаний
- = система, обрабатываемая база знаний которой выполняет роль реконфигурируемого (структурно перестраиваемого) коммутатора, обеспечивающего взаимодействие агентов, осуществляющих реконфигурацию этого коммутатора
- = система, состоящая из изменяемой (обрабатываемой) базы знаний и коллектива самостоятельных агентов, осуществляющих обработку этой базы знаний и взаимодействующих между собой только через указанную базу знаний

Не следует отождествлять понятие системы, управляемой знаниями, и понятие системы, основанной на знаниях. Понятие системы, основанной на знаниях, является более общим, т.к. база знаний здесь не обязательно должна быть активной и семантически структурированной. Это означает, что управление в системе, основанной на знаниях, может осуществляться программой, а не ситуациями и событиями в обрабатываемой базе знаний. А это, в свою очередь, означает, что обработка информации в системе, основанной на знаниях, не обязательно осуществляется коллективом агентов, работающих над базой знаний и взаимодействующих через нее. Таким

образом, системы, управляемые знаниями, следует считать важным этапом эволюции систем, основанных на знаниях, который обеспечивает существенно более высокий уровень их гибкости.

Не следует также путать понятие системы, управляемой знаниями, и понятие системы управления знаниями [Гаврилова и др., 2007], [Тузовский и др., 2005]. Система управления знаниями является частного вида системой, основанной на знаниях, которая представляет собой корпоративную систему, автоматизирующую процессы создания, распространения и использования информации внутри некоторого предприятия [Джанетто и др., 2005].

Понятие системы, управляемой знаниями, не следует путать с понятием интеллектуальной системы, т.е. системы, способной решать интеллектуальные задачи. Хотя очевидно, что, если интеллектуальную систему построить как систему, управляемую знаниями, то она приобретет способность существенно быстрее развиваться.

Переход от традиционного построения компьютерных систем к системам, построенным по архитектуре систем, управляемых знаниями, представляет собой переход к принципиально иным методологическим принципам построения компьютерных систем. В основе традиционного подхода к построению компьютерных систем лежит доминирование программ, в результате чего структуризация обрабатываемых данных "подстраивается" под программы. Поэтому при заранее не предусмотренном расширении функциональных возможностей системы нередко возникает необходимость либо в корректировке структур данных и, следовательно, корректировке уже написанных программ, либо в эклектичных, специализированных надстройках над изначально продуманными структурами данных, что приводит к усложнению понимания (разработчиками) смысла такой структуризации обрабатываемых данных. Основой построения компьютерных систем, управляемых знаниями, лежит доминирование структуризации обрабатываемой информации над программами. При этом структуризация обрабатываемой информации осуществляется независимо от программ и определяется исключительно смыслом этой информации.

Для того, чтобы в компьютерной системе в полной мере обеспечить управление знаниями, в состав базы знаний системы, в частности, должны входить:

- формулировки задач, решаемых компьютерной системой;
- планы решения этих задач, представляющие собой описание различных вариантов сведения решаемых задач к иерархической системе подзадач;

- указание активных задач, решаемых в текущий момент (как атомарных задач, так и задач, сводимых к подзадачам).

Переход современных компьютерных систем к системам, основанным на знаниях, является важнейшей тенденцией перехода к следующему поколению компьютерных систем. При этом существенно подчеркнуть то, что любую современную компьютерную систему можно реализовать по архитектуре систем, управляемых знаниями. Такая трансформация современных компьютерных систем существенно повысит их качество и конкурентоспособность. В результате трансформации современных компьютерных систем в функционально эквивалентные системы, управляемые знаниями, мы получим различные, но семантически совместимые системы, отличающиеся структурной сложностью баз знаний, функциональной сложностью агентов обработки знаний, и способные к формированию временных коллективов компьютерных систем при необходимости коллективного решения сложных задач.

Таким образом, технология проектирования компьютерных систем, управляемых знаниями – это технология, ориентированная на разработку систем, которые не обязательно должны уметь решать интеллектуальные задачи. Использование такой технологии для проектирования любых компьютерных систем на первый взгляд кажется, по сравнению с традиционными технологиями, более сложным вариантом их разработки, требующим более глубокого проникновения в семантику обрабатываемой информации и в суть реализуемых информационных процессов. Но логико-семантическая унификация архитектур компьютерных систем на основе унифицированного семантического представления обрабатываемой информации переводит эти системы на принципиально новый эволюционный уровень и создает условия для разработки более эффективных технологий их проектирования.

Актуальность систем, управляемых знаниями, обусловлена тем, что в настоящее время все большую важность приобретают задачи, для решения которых априори трудно локализовать информацию, достаточную для их решения. В ходе решения таких задач может потребоваться заранее непредсказуемая информация, хранимая в памяти системы, удовлетворяющая тем или иным требованиям. Образно говоря, исходными данными таких задач является вся информация, хранимая в памяти системы. Примерами указанных задач являются:

- информационно-поисковые задачи;
- задачи интеграции информации;
- задачи, для решения которых априори не известны способы (программы) их решения и эти способы надо найти, скомбинировать, скомбинировать из множества хранимых в памяти программ;

- задачи анализа качества всей совокупности хранимой в памяти информации;
- задачи логического вывода.

Традиционные компьютерные системы в основном ориентированы на решение задач, исходные данные которых локализованы (полностью заданы). В ходе решения таких задач потребность в дополнительной информации отсутствует. Для задач такого вида могут существовать специализированные решатели, которые у себя могут использовать удобные для них специализированные способы представления и хранения обрабатываемых ими данных. Для того, чтобы такие специализированные решатели интегрировать в состав компьютерной системы, достаточно организовать обмен конечного количества исходных (передаваемых) и возвращаемых параметров с априори известной их семантической интерпретацией или обмен конечных информационных конструкций, представленных в оговоренном формате.

К сожалению, для задач первого вида такой принцип взаимодействий компьютерной системы с решателями задач невозможен. Обмен информацией между решателями задач первого вида возможен только через общую для них память на основе унифицированного представления обрабатываемой ими информации. Более того, чтобы достаточно быстро можно было найти информацию, востребованную в ходе решения задачи, вся хранимая в системе информация должна быть структурирована, систематизирована, т.е. преобразована в целостный информационный контент.

Актуальность трансформации традиционных компьютерных систем в функционально эквивалентные системы, управляемые знаниями, обусловлена также тем, что переход на унифицированное представление различного вида знаний создает реальные условия для устранения дублирования информации в различных системах, т.е. для ликвидации "вавилонского столпотворения", обусловленного многообразием форм представления одной и той же информации в разных системах. Это, в свою очередь, дает возможность унифицировать и средства обработки знаний.

Многообразие форм решения одних и тех же задач в разных системах, обусловленное многообразием форм представления обрабатываемой информации, является важнейшим препятствием совершенствования технологий проектирования компьютерных систем.

Методика компонентного проектирования, основанная на постоянно расширяемых библиотеках многократно используемых компонентов, является для компьютерных систем, управляемых знаниями, значительно более широкого применима, чем для традиционных

компьютерных систем. Это обусловлено следующими причинами:

- Унификация логико-семантической архитектуры компьютерных систем, управляемых знаниями, и унификация представления знаний дает возможность существенно расширить библиотеки многократно используемых (reusable) компонентов. Проектирование компьютерных систем становится более "крупноблочным" и следовательно, значительно более быстрым. В частности, компьютерные системы, управляемые знаниями, дают возможность создавать библиотеки семантически целостных и достаточно крупных многократно используемых компонентов баз знаний. Отсутствие семантически унифицированного представления обрабатываемой информации в традиционных компьютерных системах такой возможности не предоставляет.
- Агенты, работающие над базой знаний компьютерных систем, управляемых знаниями, часто имеют предметно независимый характер, что делает их многократно используемыми компонентами в самых различных компьютерных системах, управляемых знаниями.
- Компьютерные системы, управляемые знаниями, могут легко интегрировать новые полезные для них компоненты, вводимые в состав постоянно расширяемых библиотек многократно используемых компонентов. Таким образом, каждая компьютерная система, управляемая знаниями, может эволюционировать в ходе своей эксплуатации как в результате деятельности своих разработчиков, которые должны постоянно ее совершенствовать, так и на основе постоянно расширяемых библиотек многократно используемых компонентов.

Актуальность трансформации традиционных компьютерных систем в системы, управляемые знаниями, обусловлена также тем, что для обеспечения их эффективной эксплуатации и совершенствования они должны в своей памяти хранить хорошо систематизированную и формализованную документацию, которая должна стать важной частью информационного контента компьютерной системы. Компьютерные системы должны становиться более понятными, прозрачными, комфортными для человека – как для конечного (в том числе, начинающего) пользователя, так и для разработчика (в том числе, начинающего). Каждый пользователь должен иметь возможность получить полную информацию:

- о структуре и возможностях текущей версии системы (в т.ч. о ее пользовательском интерфейсе);
- о методике эффективной эксплуатации системы;

- об истории совершенствования системы (о прошлых ее версиях);
- о планах совершенствования системы;
- о команде разработчиков и распределении проектных задач и обязанностей;
- о методике и организации коллективной разработки (совершенствовании) системы.

Качество каждой компьютерной системы можно оценивать в трех аспектах:

- как уровень приобретенных ею знаний и навыков;
- как скорость повышения уровня приобретаемых знаний и навыков (имеется ввиду максимально возможная скорость при благоприятных обстоятельствах);
- как набор и степень ограничений, накладываемых на уровень знаний и навыков, которые может приобрести компьютерная система в процессе своего совершенствования (в том числе, и руками ее разработчиков).

Для того, чтобы компьютерная система могла обладать высокой скоростью повышения уровня приобретаемых ею знаний и навыков (т.е. высоким уровнем обучаемости), она должна быть гибкой (легко наращиваемой, легко модифицируемой). Гибкость системы обеспечивается такой ее архитектурой, компоненты которой могут модифицироваться в достаточной степени независимо друг от друга. При этом те факторы, которые обеспечивают связь (взаимодействие) компонентов и требуют четкого согласования для обеспечения целостности системы, четко регламентируются.

Уровень приобретенных системой знаний и навыков может достигнуть способности решать интеллектуальные задачи. Такие системы обычно называют интеллектуальными.

Таким образом, следует отличать

- Общий объем и тематику приобретенных системой знаний и навыков.
- Способность системы быстро приобретать (усваивать, схватывать) новые знания и навыки.

Уровень интеллекта системы определяется скоростью ее обучения (обучаемостью), а качество (уровень) приобретенных знаний и навыков определяется уровнем сложности задач, которые способна решать система. К задачам высшего уровня сложности относятся интеллектуальные задачи.

Можно говорить о классификации компьютерных систем, управляемых знаниями, по следующим признакам:

- уровень интеллектуальности средств обработки знаний;
- уровень развития рецепторно-эффекторных средств взаимодействия с внешней средой.

По первому признаку классификации можно выделить:

- компьютерные системы, управляемые знаниями, не имеющие интеллектуальных решателей задач;
- интеллектуальные системы, управляемые знаниями.

По второму признаку классификации выделим:

- компьютерные системы, управляемые знаниями, не имеющие развитых рецепторно-эффекторных средств взаимодействия с внешней средой;
- робототехнические компьютерные системы, управляемые знаниями.

Особо подчеркнем то, что массовый перевод современных компьютерных систем на уровень систем, управляемых знаниями, необходимо начинать с систем, управляемых знаниями, не имеющих интеллектуальных решателей задач и развитых рецепторно-эффекторных средств. Система, управляемая знаниями, не обязана на каждом этапе своей эволюции уметь решать интеллектуальные задачи, но обязана быть гибкой (легко модифицируемой, легко совершенствуемой) и обязана при необходимости легко приобрести умение решать интеллектуальные задачи.

2. Принципы внутреннего представления знаний в системах, управляемых знаниями

В разработке компьютерных систем, управляемых знаниями, ключевую роль играет способ внутреннего представления знаний. Качество представления знаний компьютерной системы определяется тем, насколько приблизилось это представление к смысловому (семантическому) представлению. Наш подход к формализации смысла представляемых знаний основан на следующих положениях:

- Все известные языки выполняют две функции – коммуникативную (как средство обмена сообщениями между субъектами) и когнитивную (как средство представления информационной модели описываемого Мира).
- Язык внутреннего представления знаний в памяти компьютерной системы не обязан выполнять коммуникативную функцию. От языка внутреннего представления знаний требуется только то, чтобы он обеспечил хранение знаний в удобном для их обработки виде. Удобство обработки хранимых в памяти знаний определяется:
 - (1) простотой процедур информационного поиска фрагментов хранимой базы знаний, удовлетворяющих заданным требованиям;
 - (2) простотой процедур интеграции новых знаний, добавляемых в базу знаний;
 - (3) простотой реализации процедур логического вывода.

Таким образом, все, что обеспечивает только коммуникативные функции языка, из языка внутреннего представления знаний может быть исключено. Язык внутреннего представления знаний в памяти компьютерной системы, основанный на формализации смысла этих знаний, должен выполнять только когнитивную функцию – быть средством представления внутренней информационной модели некоторого описываемого Мира (в том числе и внешней среды соответствующей компьютерной системы).

- Знаки, входящие в состав внутреннего представления знаний, не должны иметь внутренней структуры, т.е. не должны быть представлены в виде некоторого имени соответствующей (обозначаемой) сущности. Смысл каждого знака определяется исключительно его связями с другими знаками, входящими в состав внутреннего представления базы знаний компьютерной системы. В отличие от этого для семантического анализа и понимания сообщений (внешних текстов) нужна структурированность и легкая распознаваемость знаков. По аналогичности структур, изображающих знаки (например, строк символов), определяется синтаксическая эквивалентность знаков, хотя в неформальных языках она не всегда совпадает с их семантической эквивалентностью (т.е. с их синонимией).
- В рамках внутреннего смыслового представления базы знаний компьютерной системы исключается синонимия (дублирование) знаков. Внутренние знаки, обозначающие одну и ту же сущность, должны быть склеены. Как следствие этого в рамках каждой базы знаний исключается семантическая эквивалентность (дублирование) входящих в ее состав фрагментов, т.е. фрагментов, несущих одну и ту же информацию. При этом сохраняется возможность существования логически эквивалентных фрагментов баз знаний, когда один фрагмент является логическим следствием второго и наоборот.
- Тексты языка внутреннего смыслового представления знаний должны быть нелинейными в отличие от привычных текстов, т.к. нелинейные тексты способны более адекватно отразить описываемый ими Мир, который по своей природе нелинеен. Каждая описываемая сущность может быть связана неограниченным числом связей с другими сущностями. Более того, для любой группы сущностей всегда существует связывающая их связь – все в Мире взаимосвязано. Вопрос в том, какие связи целесообразно, а какие нецелесообразно явно представлять в базе знаний. Линейность

привычных текстов – это результат проецирования нелинейного Мира на линейное (одномерное) пространство, что требует дополнительных специальных языковых средств, обеспечивающих не само описание этого Мира, а его проецирование на линейное пространство.

- В рамках языка внутреннего смыслового представления знаний вводятся мощные и простые средства перехода от информации к метаинформации (в частности, от слабо структурированных данных к связанным данным). Для этого тексты, входящие в состав базы знаний, также рассматриваются как описываемые сущности, для обозначения которых вводятся соответствующие знаки, каждый из которых трактуется как знак множества всех знаков, входящих в состав обозначаемого текста, включая и входящие в него знаки всевозможных связей.
 - Язык внутреннего смыслового представления знаний должен быть универсальным, т.е. должен обеспечить представление всевозможного вида знаний:
 - спецификаций самых различных сущностей;
 - документаций различных технических систем (в т.ч. систем, построенных по технологии OSTIS);
 - различных предметных областей (как статических, так и динамических);
 - различного вида онтологий предметных областей;
 - текстов высказываний;
 - текстов доказательства теорем;
 - формулировок задач;
 - формулировок классов задач;
 - текстов решений конкретных задач;
 - способов решения различных классов задач;
 - описаний историй эволюции различных систем (в т.ч. систем, разрабатываемых по технологии OSTIS);
 - описаний проектов, направленных на создание и совершенствование различных технических систем (в т.ч. систем, разрабатываемых по технологии OSTIS).
- Использование универсального языка внутреннего представления знаний с возможностью его неограниченного расширения, если возникает необходимость для представления новых видов знаний, создает условия для неограниченного расширения областей применения компьютерных систем, построенных на основе такого внутреннего языка представления знаний.
- В рамках языка внутреннего смыслового представления знаний имена, термины, обозначения, используемые при передаче и

приеме внешних сообщений, также являются самостоятельными описываемыми сущностями, имеющими свои внутренние знаки, которые соответствующими отношениями связываются с внутренними знаками тех сущностей, которые именуется этими внешними обозначениями. Все внешние языки являются для базы знаний частью описываемого ею внешнего Мира.

- Атомарными фрагментами внутреннего смыслового представления базы знаний являются только знаки. При этом каждый внутренний знак сам может быть описываемой сущностью (имеется в виду сам знак, а не сущность, обозначаемая этим знаком). Кроме того, каждая связь между описываемыми сущностями сама также является описываемой сущностью, которая во внутреннем представлении имеет свой внутренний знак и которая трактуется как множество, элементами которого являются знаки сущностей, связываемых описываемой связью. Таким образом, из внутреннего представления знаний исключается все, что имеет отношение не к представлению смысла, а к используемой форме представления. Так, например, во внутреннем представлении знаний нет не только букв, слов, словосочетаний, но и разделителей, ограничителей, предлогов, союзов, местоимений, склонений, спряжений и т.п.
- Язык внутреннего смыслового представления знаний должен быть не только удобным для обработки знаний в компьютерной системе, но должен быть понятным и "прозрачным" как для разработчика компьютерной системы, так и для ее конечного пользователя. Для этого, наряду с разработкой языка внутреннего смыслового представления знаний, должны быть разработаны близкие ему внешние языки, в основе которых должны лежать простые и быстро усваиваемые процедуры трансляции текстов с внутреннего во внешние формы их представления.
- Типология знаков, входящих в состав внутреннего смыслового представления знаний, полностью определяется типологией сущностей, обозначаемых этими знаками. При этом особо выделяется базовая типология описываемых сущностей, которая задает синтаксическую типологию (алфавит) внутренних знаков.

Способ внутреннего представления знаний, используемый в технологии OSTIS, назван SC-кодом (Semantic Code). Знаки, входящие в состав текстов SC-кода, называются sc-элементами. Каждый sc-элемент можно считать инвариантом всего многообразия форм представления (во всевозможных языках и знаковых конструкциях) той сущности, которая

обозначается этим sc-элементом. Таким инвариантом является только то, что указанный sc-элемент обозначает соответствующую ему сущность. Поэтому sc-элемент не имеет формы. В этом смысле он абстрагируется от формы своего представления в рамках той или иной знаковой конструкции.

Каждому sc-элементу взаимно однозначно соответствует некоторый класс синонимичных знаков – множество всевозможных вхождений знаков обозначаемой сущности в самые различные внешние знаковые конструкции (тексты).

Текст SC-кода (sc-текст) с синтаксической точки зрения – это абстрактная знаковая конструкция, представляющая собой множество sc-элементов, связанных между собой отношениями инцидентности. С семантической точки зрения sc-текст – это абстрактная знаковая конструкция, являющаяся инвариантом всего многообразия форм представления соответствующей информации, т.е. инвариантом всего соответствующего класса семантически эквивалентных знаковых конструкций. Следовательно, каждому sc-тексту взаимно однозначно соответствует некоторый класс всевозможных семантически эквивалентных внешних текстов, представляющих заданную информацию в самых различных языках.

Рассмотрим типологию *sc-элементов* как типологию сущностей, обозначаемых *sc-элементами*.

Подчеркнем, что универсальный характер *SC-кода* означает то, что с помощью *sc-элемента* можно обозначить абсолютно любую сущность – как реальную, так и абстрактную.

Приведем описание основных классов *sc-элементов* средствами *SC-кода*, а точнее, средствами *SCg-кода* и *SCn-кода*, которые являются языками эквивалентными и близкими *SC-коду*.

Алфавит SC-кода

= Семейство классов синтаксически выделенных *sc-элементов*

= Семейство классов *sc-элементов*, элементы каждого из которых имеют одинаковые синтаксические метки

= Семейство классов эквивалентности для неявно заданного отношения эквивалентности "быть *sc-элементами*, имеющими одинаковую синтаксическую метку"

На рис. 1 представлен sc-текст, описывающий связь между *SC-кодом* и *Алфавитом SC-кода* с помощью отношения *алфавит** и отношения *множество атомарных фрагментов текстов**.

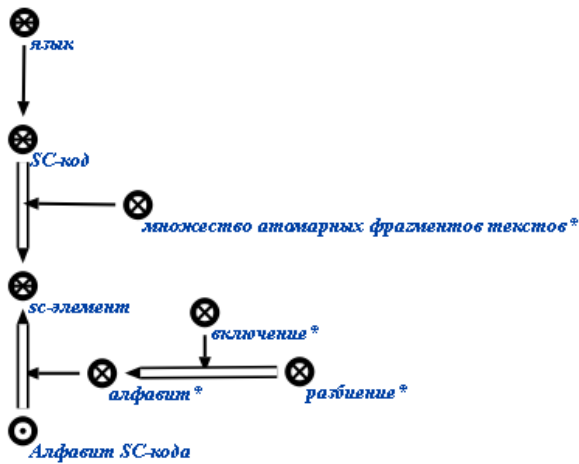


Рисунок 1 – Связь между SC-кодом и Алфавитом SC-кода

Поясним некоторые используемые в рис. 1 понятия.

язык

= множество знаковых конструкций, удовлетворяющих общим для этого множества синтаксическим и семантическим правилам

SC-код

= Множество текстов SC-кода, являющихся синтаксически и семантически унифицированными семантическими сетями, имеющих базовую теоретико-множественную интерпретацию

= Множество sc-текстов

= sc-текст

В SC-коде синонимия* между sc-элементами отсутствует, а, точнее, должна выявляться и устраняться путем склеивания синонимичных sc-элементов, либо путем явного указания факта их синонимии.

синонимия*

= быть синонимичными sc-элементами*

= синонимия sc-элементов*

= отношение, с помощью которого явно указывается факт синонимии некоторых sc-элементов*

∈ ключевой sc-элемент*:

...

∈ sc-примечание

<= трансляция sc-текста*:

...

С Русский язык

∃ хранимый файл*:

[Явно указываемая синонимия sc-элементов может использоваться для повышения живучести системы, построенной на основе SC-кода.

Это, своего рода, резервное копирование (дублирование) sc-элементов, которое должно учитываться при информационном поиске.

Заметим также, что синонимичными друг другу могут быть не только константные, но и переменные sc-элементы.]

Кроме того существует синонимия между синтаксически выделенными классами sc-элементов и соответствующими им семантически выделенными классами sc-элементов. Дело в том, что sc-элементы, принадлежащие разным синтаксически выделенным классам, не могут пересекаться, то есть не могут иметь сразу несколько синтаксических меток. Тогда, как соответствующие им семантически выделенные классы sc-элементов могут не просто пересекаться, а могут входить один в другой.

На рис. 2 представлены элементы Алфавита SC-кода, т.е. синтаксически выделяемые классы sc-элементов и синонимичные им семантически выделяемые классы sc-элементов.

Здесь, например, синонимия синтаксически задаваемого класса sc-узлов и класса всех sc-элементов означает то, что любой sc-элемент может быть представлен в форме sc-узла.

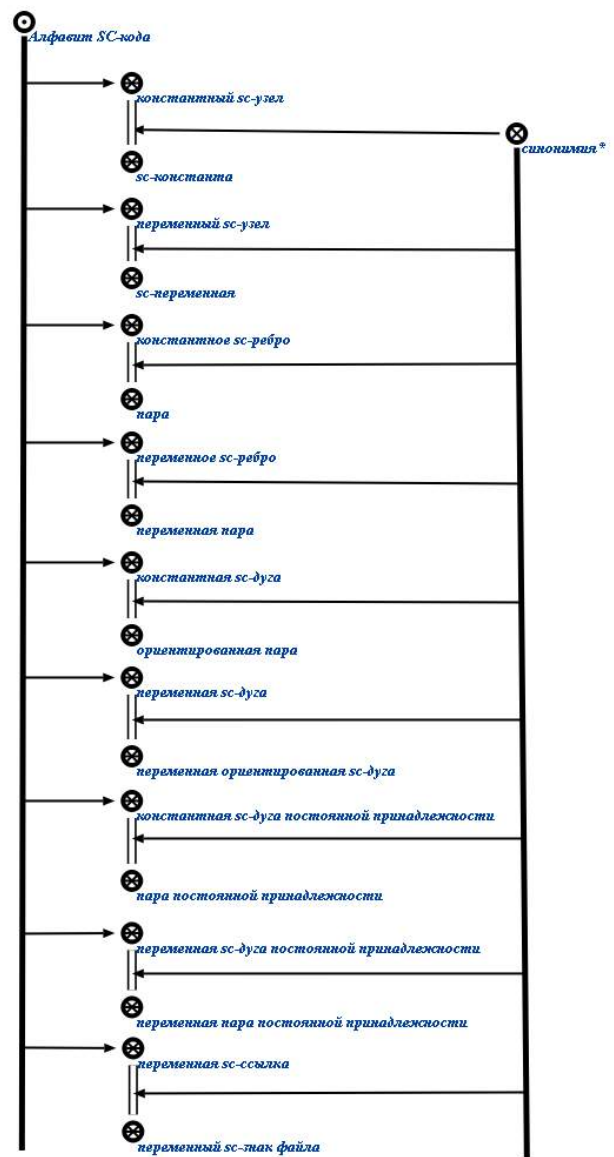


Рисунок 2 – Алфавит SC-кода

sc-коннектор

= синтаксически атомарная связка sc-элементов

<= разбиение*:

- {
 - sc-ребро
 - <= разбиение*:
 - {
 - константное sc-ребро
 - переменное sc-ребро
 - sc-дуга
 - <= разбиение*:
 - {
 - константная sc-дуга общего вида
 - переменная sc-дуга общего вида
 - константная sc-дуга постоянной принадлежности
 - переменная sc-дуга постоянной принадлежности

sc-элемент

= sc-знак

= знак сущности

= знак, входящий в текст SC-кода

= сущность

= sc-знак сущности

= знак, входящий в состав текста SC-кода

= Множество всевозможных атомарных фрагментов текста

<= разбиение*:

- {
 - sc-константа
 - sc-переменная

<= разбиение*:

- {
 - материальная или квазиматериальная сущность
 - абстрактная сущность

sc-константа

= константный sc-элемент

<= разбиение*:

- {
 - терминальная sc-константа
 - множество

<= разбиение*:

- {
 - вымышленная сущность
 - = придуманная сущность
 - невымышленная сущность
 - = действительно существующая сущность

<= разбиение*:

- {
 - постоянная сущность
 - = постоянно существующая сущность
 - = стационарная сущность

- временная сущность
 - = временно существующая сущность
 - = нестационарная сущность

терминальная sc-константа

= терминальная сущность

= первичная sc-константа

= первичная сущность

= sc-константа, не являющаяся знаком множества

= терминальная константная сущность

<= разбиение*:

- {
 - материальная или квазиматериальная сущность
 - sc-знак файла
 - абстрактная терминальная сущность
 - = абстрактная константная (конкретная, фиксированная) сущность, не являющаяся множеством
 - ⊃ действительное число
 - ⊃ геометрическая точка
 - ⊃ точка маршрута в графовой структуре

материальная или квазиматериальная сущность

<= разбиение*:

- {
 - материальная сущность
 - квазиматериальная сущность
 - = вымышленная сущность, подобная материальной
 - = псевдоматериальная сущность
 - = sc-знак квазиматериальной сущности
 - = придуманная сущность, подобная материальной

материальная сущность

= естественная сущность

= объект исследования естественной науки

= физическая сущность

= материальный объект

= реальная сущность

= знак материальной сущности

= sc-знак материальной сущности

= sc-элемент, обозначающий конкретную материальную сущность

sc-знак файла

= константный sc-знак файла

= sc-константа, обозначающая файл, хранимый либо во внешней файловой памяти sc-системы, либо в другом месте

= sc-знак конкретного (фиксированного) файла

= конкретный файл

= файл

= электронный образ внешней информационной конструкции

= внешняя информационная конструкция

⊃ sc-идентификатор

= *sc*-знак файла, в котором хранится в определенном формате электронный образ одного из экземпляров класса синтаксически эквивалентных конструкций, все или многие из которых, входя во внешние тексты, обозначают ту же сущность, что и соответствующий им *sc*-элемент

= *sc*-знак внешнего идентификатора некоторого *sc*-элемента

= внешний идентификатор *sc*-элемента

= *sc*-знак *sc*-идентификатора

множество

= *sc*-множество

= множество *sc*-элементов

= *sc*-знак множества *sc*-элементов

= множество знаков описываемых сущностей

= текст SC-кода

= *sc*-текст

= SC-код

<= разбиение*:

- {
- *связка*
- *класс*
- *структура*
- }

∈ ключевой *sc*-элемент':

...

∈ *sc*-примечание

<= трансляция *sc*-текста*:

...

⊂ *Русский язык*

∃ хранимый файл':

[Существенно подчеркнуть то, что любой текст SC-кода представляет собой множество *sc*-элементов, связанных между собой отношениями инцидентности *sc*-коннекторов с их компонентами. Таким образом, понятия текста SC-кода и множества *sc*-элементов являются синонимами.]

∈ ключевой *sc*-элемент':

...

∈ *sc*-примечание

<= трансляция *sc*-текста*:

...

⊂ *Русский язык*

∃ хранимый файл':

[Заметим, что все семантические виды множеств (и связки, и классы, и структуры) могут быть компонентами связок любых теоретико-множественных отношений (включение*, объединение*, разбиение*, пересечение* и т.д.).]

связка

= связь между сущностями

= *sc*-связка

= связка *sc*-элементов

= *sc*-знак связи

<= разбиение*:

- {
- *небинарная связка*

- *пара*

}

<= разбиение*:

{

- *постоянная связка*
= *стационарная связка*
- *временная связка*
= *нестационарная связка*

}

небинарная связка

= *небинарная sc*-связка

= *небинарная связка sc*-элементов

= *sc*-связка, имеющая мощность либо равную единице, либо большую, чем два

⊃ *синглетон*

= *унарная связка*

= *одномощное множество sc*-элементов

⊃ *тройка*

= *тернарная связка*

пара

= *бинарная связка*

= *пара sc*-элементов

= *sc*-пара

= *константная пара sc*-элементов

= *двухмощное множество*

<= разбиение*:

{

- *неориентированная пара*
- *ориентированная пара*

}

ориентированная пара

= *ориентированная бинарная связка*

= *ориентированная пара sc*-элементов

= *константная ориентированная пара sc*-элементов

⊃ *пара постоянной принадлежности*

= *пара стационарной принадлежности*

= *sc*-пара стационарной принадлежности

= *Отношение постоянной принадлежности*

= *Отношение стационарной принадлежности*

⊃ *пара временной принадлежности*

= *пара нестационарной принадлежности*

= *Отношение временной принадлежности*

⊃ *пара постоянной непринадлежности*

⊃ *пара временной непринадлежности*

⊃ *пара постоянной нечеткой принадлежности*

⊃ *пара временной нечеткой принадлежности*

⊃ *пара принадлежности или непринадлежности*

<= разбиение*:

{

- *пара постоянной принадлежности или непринадлежности*
- *пара временной принадлежности или непринадлежности*

}

<= разбиение*:

{

- *пара принадлежности*
- *пара непринадлежности*

}

- пара нечеткой принадлежности

}

класс

- = *sc*-класс
- = класс *sc*-элементов
- = *sc*-понятие
- = *sc*-концепт
- = множество всех тех и только тех *sc*-элементов, которые обозначают сущности, эквивалентные в заданном смысле, т.е. удовлетворяющие соответствующим требованиям
- ⊃ класс терминальных сущностей
- ⊃ класс материальных или квазиматериальных сущностей
- ⊃ класс абстрактных терминальных сущностей
- ⊃ отношение
- ⊃ класс структур
- ⊃ класс классов
- ⊃ параметр
- ⊃ величина

отношение

- = *sc*-отношение
- = класс связей
- = класс связей, эквивалентных в заданном смысле
- = отношение, заданное на множестве *sc*-элементов
- = класс связей, связывающих *sc*-элементы

<= разбиение*:

{

- бинарное отношение
 - = класс пар *sc*-элементов
 - ⊃ квазибинарное отношение
 - ⊃ сложение*
 - ⊃ умножение*
 - ⊃ объединение*
 - ⊃ пересечение*
 - ⊃ соответствие*
 - ⊃ изоморфизм*
 - ⊃ гомоморфизм*
 - ⊃ биекция*
 - ⊃ ролевое отношение
- класс небинарных связей одинаковой мощности
- класс связей разной мощности

}

ролевое отношение

- = отношение, являющееся подмножеством множества пар постоянной и-или временной принадлежности
- ⊃ пара постоянной принадлежности
- ⊃ ключевой *sc*-элемент'
- = ключевой элемент *sc*-структуры'
- ⊃ образец'
- = контрольный экземпляр'
- ⊃ пример'
- ⊃ хранимый файл'
- = экземпляр класса синтаксически эквивалентных информационных конструкций, хранимый в виде доступного файла'

параметр

- = характеристика
 - = свойство
 - = семейство классов сущностей, имеющих одинаковое или почти одинаковое значение заданного параметра
 - = признак классификации сущности некоторого класса
- <= разбиение*:
- {
- измеряемый параметр
 - = количественный параметр
 - неизмеряемый параметр
 - = качественный параметр
- }

величина

- = значение какого-либо параметра
- = класс сущностей, имеющих одинаковое или почти одинаковое значение какого-либо параметра
- ∈ ключевой *sc*-элемент':
- ...
- ∈ *sc*-примечание
- <= трансляция *sc*-текста*:
- ...
- ⊃ Русский язык
- ⊃ хранимый файл':
- [Каждая величина связывается с соответствующим ей параметром с помощью следующих ролевых отношений:
 - точная величина'
 - неточная величина'
 - величина в фиксированном интервале']

структура

- = *sc*-структура
 - = структура *sc*-элементов
- <= разбиение*:
- {
- тривиальная структура
 - = дискретная структура
 - = структура, не содержащая связей, связывающих элементы этой структуры
 - ⊃ геометрическая фигура
 - нетривиальная структура
 - = структура, содержащая связи, связывающие элементы этой структуры
- }
- <= разбиение*:
- {
- связная структура
 - несвязная структура
 - ⊃ тривиальная структура
- }

sc-переменная

- = *sc*-знак переменной сущности
- = переменная сущность
- = обозначение произвольной сущности, которое пробегает некоторое дополнительно уточняемое множество знаков
- = *sc*-обозначение произвольной сущности,

которому соответствует некоторое дополнительно уточняемое множество *sc*-элементов, являющееся областью возможных значений этого *sc*-обозначения
∈ ключевой *sc*-элемент':

...
∈ *sc*-сравнение
<= трансляция *sc*-текста*:
...
⊂ Русский язык
∃ хранимый файл':
[В отличие от переменной, область возможных значений каждой константы состоит только из самой этой константы]
⊃ *sc*-переменная первого порядка
= *sc*-переменная, значениями которой являются только *sc*-константы
⊃ *sc*-переменная второго порядка
= *sc*-метапеременная
= *sc*-переменная, значениями которой являются только *sc*-переменные первого порядка
⊃ *sc*-переменная смешанного порядка
= *sc*-переменная, значениями которой являются как *sc*-константы, так и *sc*-переменные первого порядка

абстрактная сущность

= *sc*-знак абстрактной сущности

<= разбиение*:

- {
 - *sc*-переменная
= *sc*-знак произвольной сущности из некоторого дополнительно уточняемого множества сущностей
 - множество
= *sc*-множество
= множество *sc*-элементов
= *sc*-знак множества *sc*-знаков
 - абстрактная терминальная сущность
= первичный элемент абстрактной структуры}

временная сущность

= временно существующая сущность

= нестационарная сущность

= процесс

<= разбиение*:

- {
 - материальная сущность
 - квазиматериальная сущность
 - временная связка
= временно существующая связь
= нестационарная связка
 - ситуация
= временно существующая структура
= нестационарная структура
= состояние динамической системы
 - событие
= переходный процесс
= процесс перехода от одного состояния к другому
⊃ действие}

= воздействие

= событие, осуществляемое неким целенаправленным активным субъектом

<= разбиение*:

- {
 - сущность прошлого времени
= прошлая сущность
= сущность, прекратившая свое существование
= сущность, существовавшая в прошлом времени
= сущность "жившая" в прошлом времени
 - сущность настоящего времени
= сущность, существующая в текущий момент времени
= Множество всех сущностей, существующих в текущий момент
= Текущее состояние окружающего Мира
 - сущность будущего времени
= будущая сущность}

При классификации знаков по признаку постоянства обозначаемых ими сущностей следует отличать постоянство (вечность и временность) обозначаемых сущностей и постоянство самих знаков, хранимых в памяти компьютерной системы.

В общем случае можно утверждать, что ничего вечного в базе знаний нет. Просто одни знаки "живут" (присутствуют, хранятся) в базе знаний дольше, другие имеют короткую "жизнь" (входя, например, в состав временной, вспомогательной информации, используемой для решения конкретной задачи и теряющей ценность после решения этой задачи).

Таким образом, не следует путать временный характер существования некоторой описываемой сущности и временный характер присутствия знака этой сущности в рамках текущего состояния базы знаний. Из состава текущего состояния базы знаний может быть удален любой знак, обозначающий как временно существующую, так и постоянно существующую сущность.

Для описания бывшей, настоящей и будущей принадлежности того или иного *sc*-элемента текущему состоянию базы знаний той или иной компьютерной системы вводятся следующие понятия.

текущее состояние базы знаний

= множество *sc*-элементов, входящих в состав текущего состояния базы знаний
= Семейство текущих состояний баз знаний различных компьютерных систем
= множество настоящих знаков базы знаний соответствующей системы
∃ Текущее состояние базы знаний IMS
∃ Текущее состояние базы знаний ИСС по геометрии

множество sc-элементов, удаленных из текущего состояния базы знаний

- = множество знаков, логически удаленных из базы знаний соответствующей системы
- = множество бывших знаков базы знаний
- = множество прошлых знаков базы знаний
- ⊃ Множество sc-элементов, удаленных из текущего состояния базы знаний IMS
- ⊃ Множество sc-элементов, удаленных из текущего состояния базы знаний ИСС по геометрии

множество sc-элементов, которые планируется ввести в текущее состояние базы знаний

- = множество будущих знаков базы знаний
- ⊃ Множество sc-элементов, которые планируется ввести в текущее состояние базы знаний IMS
- ⊃ Множество sc-элементов, которые планируется ввести в текущее состояние базы знаний ИСС по геометрии

На рис. 3 в качестве примера приведен фрагмент спецификации Метасистемы IMS, в котором используются введенные выше понятия.

В этом рисунке особое место занимает sc-узел с идентификатором "Я". Этот узел является sc-знаком синглтона, единственным элементом которого является sc-знак той компьютерной системы, в базу знаний которой указанный sc-узел входит. Таким образом, рассматриваемый синглетон есть не что иное, как указатель на самого себя, на собственное Я.

Рассмотрим кратко типологию знаний, входящих в состав внутреннего представления базы знаний, которая полностью определяется (задается) типологией знаков, обозначающих эти знания. К числу основных видов знаний, представленных в SC-коде, относятся предметные области, онтологии, семантические окрестности, sc-тексты высказываний, sc-тексты доказательств, задачи, планы решения конкретных задач, sc-тексты обоснования решений задач, обобщенные формулировки задач соответствующих классов, способы решения задач соответствующих классов, разделы баз знаний.

знание

- = семантически осмысленная и целостная sc-структура
- = sc-знание
- = знание, представленное в SC-коде
- ⊂ структура
- ⊃ предметная область
- ⊃ онтология
- ⊃ логическая система понятий и их определений
- ⊃ логическая система утверждений и их доказательств
- ⊃ семантическая окрестность
- ⊃ sc-сравнение
- ⊃ sc-текст высказывания
- ⊃ sc-текст доказательства

⊃ задача

- = sc-текст формулировки задачи
- = sc-формулировка задачи
- ⊃ план решения задачи
- = sc-текст плана решения задачи
- ⊃ sc-текст обоснования решения задачи
- ⊃ обобщенная формулировка задач соответствующего класса
- = sc-текст обобщенной формулировки задач некоторого класса
- ⊃ способ решения задач соответствующего класса
- = sc-текст способа решения задач
- = sc-текст обобщенного плана решения задач некоторого класса
- ⊃ раздел базы знаний

Важнейшим отношением, заданным на множестве знаний является отношение *быть метазнанием**, описывающее переход от знаний к описывающим их метазнаниям (в частности, от данных – к связанным данным, от неструктурированных данных – к структурированным).

метазнание*

- = *быть метазнанием**
- = Бинарное ориентированное отношение, связки которого связывают некоторое исходное знание со знанием, которое является метаотношением исходного знания, его спецификацией, описанием его структуры*
- ⊃ метазнание, основанное на выделении тривиальных структур из терминальных элементов исходного знания*
- ⊃ метазнание, основанное на выделении нетривиальных структур, являющихся фрагментами исходного знания*

Примером связи между знанием и соответствующим ему метазнанием является переход от некоторого исходного знания к описанию его декомпозиции (сегментации) на некоторые части с указанием связей между этими частями.

На рис. 4 приведены примеры использования отношения *метазнание** (*быть метазнанием**), а также частных видов этого отношения.

Важнейшим ролевым отношением, связывающим sc-знаки знаний с компонентами этих знаний, является отношение *ключевой sc-элемент'* (*быть ключевым sc-элементом'*).

предметная область

- = sc-модель предметной области
- ⊂ знание
- ⊂= разбиение*:

- {
- статическая предметная область
- = предметная область, в состав которой входят постоянно существующие объекты исследования, постоянно существующие связи и структуры
- = стационарная предметная область

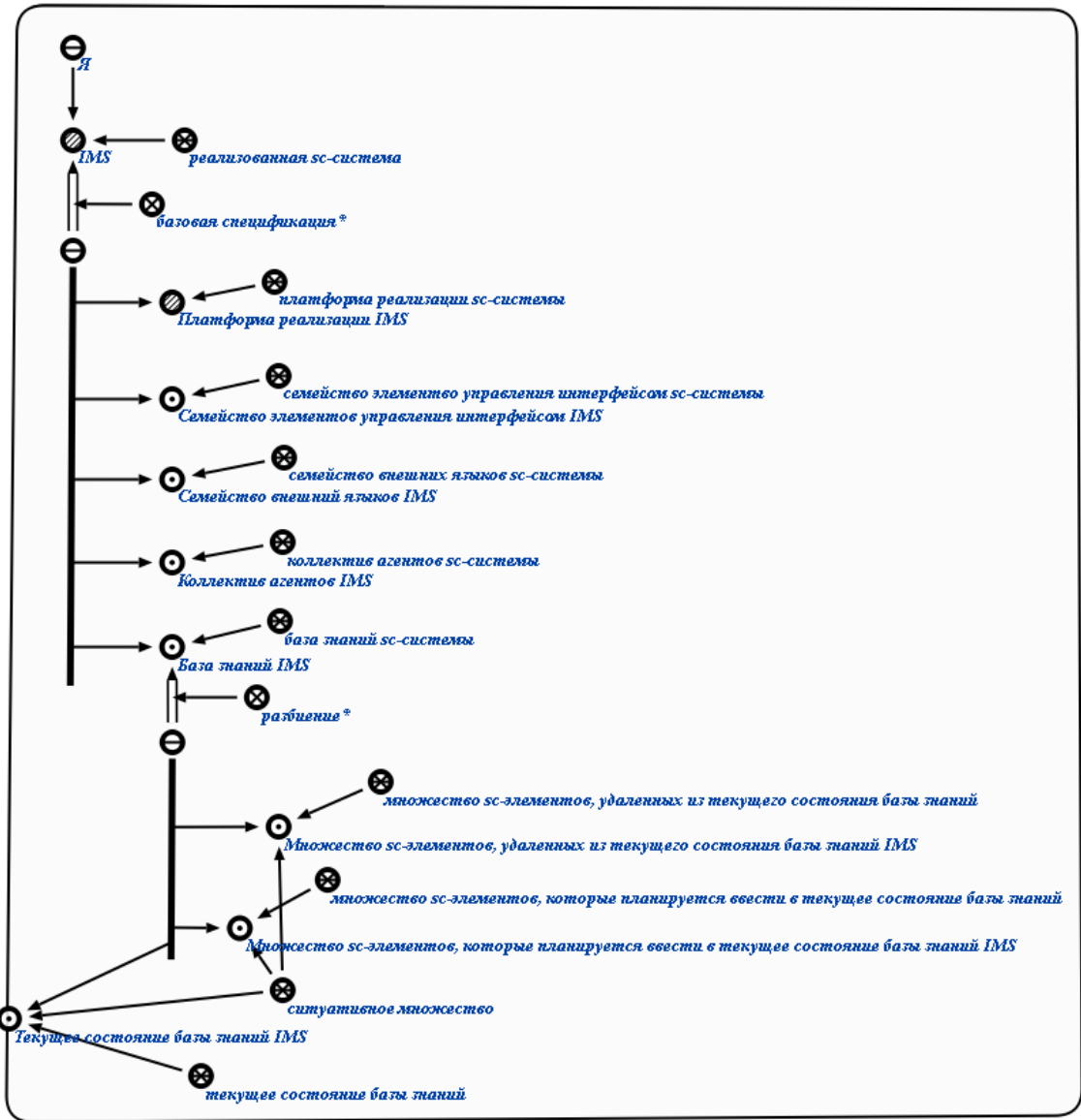


Рисунок 3 – Текст SC-кода, описывающего базовую спецификацию IMS

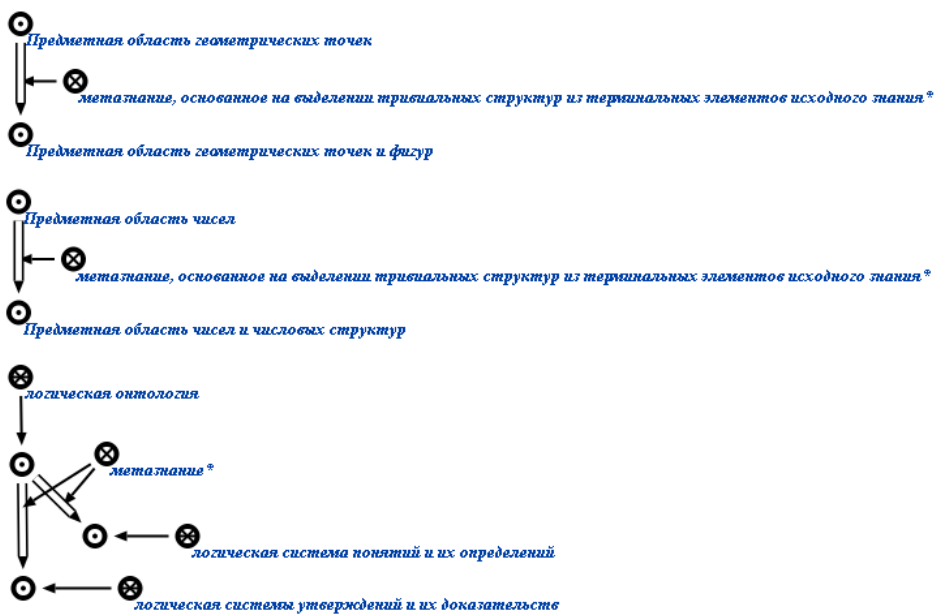


Рисунок 4 – Примеры связей, связывающих знания и соответствующие им метазнания

- динамическая предметная область
= нестационарная предметная область
- }
∈ ключевой sc-элемент':
- ...
- ∈ sc-примечание
- <= трансляция sc-текста*:
- ...
- С Русский язык
- ∃ хранимый файл':
- [Каждая предметная область задает соответствующий специализированный язык (sc-язык), представляющий собой множество всевозможных фрагментов (подструктур), включаемых в состав этой предметной области. Так, например, каждая теоретико-множественная онтология является текстом языка, задаваемого Предметной областью множеств.]
- ∃ Предметная область предметных областей
- = Предметная область, задающая sc-язык структурных спецификаций предметных областей
- ∃ Предметная область множеств
- = Предметная область, которая задает теоретико-множественный sc-язык, используемый, в частности, для записи теоретико-множественных онтологий
- ∃ Предметная область sc-связок и отношений
- ∃ Предметная область логических формул
- ∃ Предметная область sc-идентификаторов
- ∃ Предметная область параметров, величин и шкал
- ∃ Предметная область scp-программ

Предметная область логических формул
= Предметная область, определяющая специализированный sc-язык логических формул, высказываний и формальных теорий (логических онтологий)
= Предметная область, являющаяся объединением всевозможных текстов логического sc-языка (Языка SCL)

Предметная область scp-программ
= Предметная область, определяющая специализированный sc-язык SCP

SCP
= Semantic Code Programming
= Специализированный sc-язык, являющийся базовым языком программирования, ориентированным на обработку sc-текстов

Предметная область параметров, величин и шкал
= Предметная область, задающая sc-язык параметрических моделей всевозможных сущностей
= Предметная область параметров и их значений
= Предметная область, представляющая собой результат объединения всевозможных параметрических моделей
= Предметная область параметрического

- описания всевозможных сущностей
- ∃ максимальный класс объектов исследования':
- параметр
= признак классификации
= характеристика
- величина
= значение параметра
- шкала
= множество результатов измерения величины
- ∃ исследуемое отношение':
- точность*
= Отношение, связывающее каждую неточную величину с той точной величиной, которая задает половину интервала, в рамках которого указанная неточная величина находится
- измерение*
- единица измерения*
- шкала измерения*
- ∃ немаксимальный класс объектов исследования':
- количественный параметр
- качественный параметр
- точная величина
- неточная величина
- величина в фиксированном интервале
- дискретная шкала
- непрерывная шкала

На множестве предметных областей заданы следующие нерелевые отношения:

- частная предметная область*
∈ антитранзитивное отношение
- онтология*
= быть онтологией заданной предметной области
- включение*

На рис. 5 приведен пример использования указанных отношений.

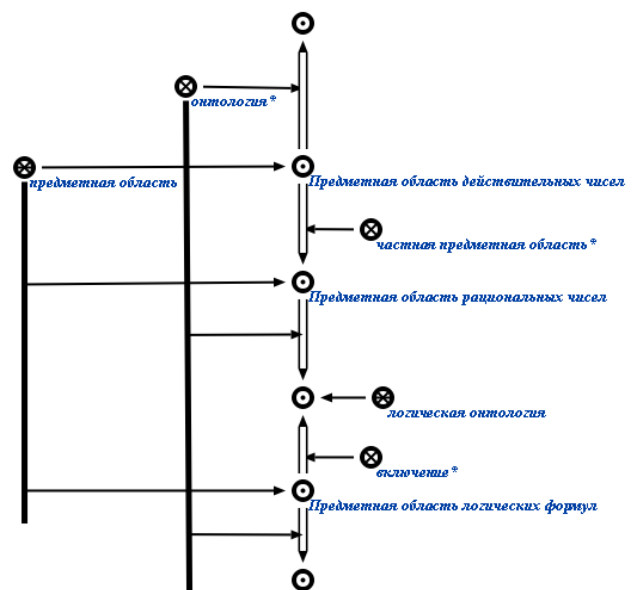


Рисунок 5 – Пример использования отношений, заданных на множестве предметных областей

частная предметная область*

= быть частной предметной областью*

<= разбиение*:

- {
- частная предметная область, являющаяся результатом сужения класса исследуемых объектов*
- частная предметная область, являющаяся результатом сужения набора отношений, заданных на объектах исследования или вторичных им объектах*
- }

К числу ролевых отношений, указывающих роли компонентов предметных областей, относятся:

- *максимальный класс объектов исследования'*
- *немаксимальный класс объектов исследования'*
- *исследуемое понятие'*
= понятие, исследуемое в данной предметной области'
- *понятие, исследуемое в другой предметной области'*
⊃ *понятие, исследуемое в частной предметной области'*
= *понятие, определяемое и исследуемое в частной предметной области'*
= *понятие, исследуемое в предметной области, которая связана с заданной связкой отношения быть частной предметной областью'*
- *ключевой объект исследования'*
= *объект исследования, обладающий особыми свойствами'*

К числу ключевых объектов исследования', например, в рамках Предметной области комплексных чисел относятся числа ноль, единица, мнимая единица.

максимальный класс объектов исследования'

⊂ *ключевой sc-элемент предметной области'*

∈ *ключевой sc-элемент'*:

...

∈ *sc-примечание*

<= *трансляция sc-текста**:

...

⊂ *Русский язык*

⊃ *хранимый файл'*:

[Предметная область может иметь несколько максимальных классов объектов исследования, если нет смысла объединять эти классы в одно общее понятие.]

онтология

= *sc-онтология*

= *sc-модель онтологии предметной области*

= *онтология предметной области*

= *онтология предметной области, представленная в SC-коде*

= *sc-текст онтологии*

= *спецификация предметной области*

⊂ *знание*

<= *разбиение**:

{

- *интегрированная онтология*
= *объединение онтологий различного вида, специфицирующих одну и ту же предметную область*
- *структурная спецификация предметной области*
- *теоретико-множественная онтология*
- *терминологическая онтология*
- *логическая онтология*
- *онтология задач и решений задач*
- *онтология классов задач и способов решения задач*

}

Связь между предметной областью и ее онтологией задается отношением *онтология** (*быть онтологией**), которое является частным видом отношения *метазнание** (*быть метазнанием**).

семантическая окрестность

= *sc-текст, посвященный описанию некоторой сущности, знак которой является ключевым sc-элементом указанного sc-текста*

= *sc-окрестность*

= *описание (спецификация) некоторой сущности, выполненное с той или иной степенью детализации и в том или ином ракурсе*

⊃ *базовая семантическая окрестность*

= *базовая семантическая окрестность в рамках текущего состояния базы знаний*

⊃ *общая семантическая окрестность*

= *общая семантическая окрестность в рамках текущего состояния базы знаний*

⊃ *sc-текст определения*

⊃ *sc-пояснение*

⊃ *sc-примечание*

⊃ *параметрическая модель*

⊃ *документация*

⊃ *документация технической системы*

⊃ *документация проекта*

⊃ *документация технологии*

⊃ *биография*

⊃ *медицинская карта пациента*

Каждая базовая семантическая окрестность включает в себя:

- все классы, которым принадлежит знак описываемой сущности;
- все связки, компонентами которых является знак описываемой сущности с указанием (1) ролей этого знака, (2) классов, которым принадлежат указанные связки и (3) остальных компонентов указанных связок;
- все структуры, элементами которых является знак описываемой сущности с указанием (1) ролей этого знака в рамках указанных структур, (2) классам, которым принадлежат эти структуры.

Кроме того:

- если описываемая сущность является связкой (связкой), то в ее базовую семантическую

- окрестность включаются все компоненты этой связи с указанием их ролей;
- если описываемая сущность является конечной структурой, то в ее базовую семантическую окрестность включаются все компоненты этой структуры с указанием их ролей, если таковые имеются;
 - если описываемая сущность является бесконечной структурой (например, бесконечной предметной областью), то в ее базовую семантическую окрестность включаются все ключевые sc-элементы этой структуры с уточнением их ролей в рамках этой структуры;
 - если описываемая сущность является классом, то в ее базовую семантическую окрестность включаются все ключевые элементы этого класса, отмеченные такими ролевыми отношениями, как *пример*, *образец*, *контрольный экземпляр*.

Каждая *общая семантическая окрестность* заданной описываемой сущности полностью включает в себя *базовую семантическую окрестность* этой сущности, но, кроме этого, в *общую семантическую окрестность* включаются:

- *параметрическая модель* описываемой сущности;
- все *sc-тексты высказываний*, описывающих свойства данной сущности;
- идентификация описываемой сущности в различных внешних информационных конструкциях и, если описываемая сущность является классом, то и правила построения таких идентификаторов для элементов указанного класса;
- история исследования описываемой сущности или история ее создания, если она является артефактом, с указанием конкретных персон, принимавших участие в этих процессах, и, в первую очередь, авторов соответствующих исследований или разработок.

параметрическая модель

= *параметрическая модель некоторой сущности*

= *спецификация заданной сущности через указание ее "координат" в абстрактном параметрическом пространстве (в пространстве признаков)*

= *параметрическая спецификация*

⊂ *семантическая окрестность*

Множество всевозможных параметрических моделей можно считать специализированным языком параметрических моделей. Объединением всех sc-текстов этого языка является *Предметная область параметров, величин и шкал*.

sc-сравнение

= *sc-текст, посвященный сравнительному анализу конечного числа (чаще всего, двух) сущностей, знаки которых являются ключевыми элементами указанного sc-текста*

раздел базы знаний

= *sc-текст раздела базы знаний*

= *фрагмент базы знаний, совершенствование которого может осуществляться в достаточной степени независимо от других специально выделенных фрагментов базы знаний*

= *специально выделенный фрагмент базы знаний, которому однозначно ставится в соответствие персона, администрирующая этот фрагмент и несущая полную персональную ответственность за его актуализацию, обновление и совершенствование*

⊃ *база знаний*

= *целостная база знаний компьютерной системы, управляемой знаниями*

= *sc-модель базы знаний*

= *база знаний компьютерной системы, представленная в SC-коде*

∈ *ключевой sc-элемент*':

...

∈ *sc-примечание*

<= *трансляция sc-текста* *:

...

⊂ *Русский язык*

⊃ *хранимый файл*':

[Разделом базы знаний может быть предметная область, онтология, документация, обоснование разработки, документация. Частным видом разделов баз знаний являются сами базы знаний, которые должны удовлетворять целому набору требований, обеспечивающих их целостность.]

3. Основные положения предлагаемой технологии, ориентированной на проектирование систем, управляемых знаниями, и принципы организации проекта, направленного на создание и использование этой технологии

В настоящее время одним из главных направлений развития искусственного интеллекта является создание общей, доступной и эффективной технологии комплексного проектирования систем, управляемых знаниями. К основным принципам, лежащим в основе любой эффективной технологии (не только проектирования компьютерных систем) можно отнести:

- Четкое отделение этапа проектирования от этапа производства и соответствующее этому четкое отделение технологии проектирования от технологии производства
- Унификация результатов проектирования и, соответственно этому, независимость проектирования от производства (для компьютерных систем – платформенная независимость логического проектирования)

- Компонентное (модульное) проектирование, на основе постоянно пополняемых библиотек многократно используемых компонентов
- Постоянное совершенствование технологии как со стороны науки, так и со стороны инженерии (должен накапливаться и систематизироваться проектный опыт), поэтому технология должна быть доступной, открытой и должна быть результатом широкого сотрудничества всех участников.

В соответствии с этим, к основным положениям технологии OSTIS, ориентированной на проектирование систем, управляемых знаниями, можно отнести следующее.

3.1. Базовая графодинамическая модель обработки информации

В качестве формальной основы проектируемых интеллектуальных систем предлагается использовать графодинамические модели обработки информации специального вида, ориентированные на параллельную асинхронную обработку информации.

Графодинамическая модель обработки информации трактует процесс обработки информации как процесс преобразования графовой структуры, в ходе которого меняется не только состояние элементов этой графовой структуры, но и конфигурация этой структуры (появляются или удаляются её вершины, а также связи между ними). Заметим, что для создания графодинамических моделей обработки информации недостаточно тех видов графовых структур, которые в настоящее время исследуются в теории графов [Касьянов, 2003]. Общее определение графовой структуры, на основе которого можно строить практически полезные графодинамические модели обработки информации приведено в работе [Голенков и др, 2001].

Без организации параллельной обработки информации невозможно рассчитывать на необходимую производительность подавляющего числа практически полезных интеллектуальных систем. Асинхронные модели обработки информации являются более гибкими, их легче интегрировать и наращивать новыми функциональными возможностями. Графодинамическая модель параллельной асинхронной обработки информации, которую будем также называть графодинамической параллельной асинхронной машиной, трактуется нами как абстрактная многоагентная система, состоящая из:

- абстрактной графодинамической памяти, в которой хранятся обрабатываемые графовые структуры;
- коллектива агентов, работающих над общей для них графодинамической памятью и обменивающихся информацией только через

эту память (в т.ч. и для координации своих действий).

Графодинамическая память носит реконфигурируемый, структурно перестраиваемый характер, поскольку процесс обработки графовых структур в конечном счете сводится к генерации и удалению различных элементов графовых структур, а также к генерации и удалению пар инцидентности между этими элементами. Другими словами, процесс обработки информации в графодинамической памяти сводится не только к изменению состояния элементов памяти, но и к изменению конфигурации связей между ними.

Агенты, работающие над общей графодинамической памятью, делятся на три вида:

- внутренние агенты, каждый из которых реагирует на определенного вида ситуации или события в графодинамической памяти и осуществляет изменение состояния графодинамической памяти, соответствующее своему функциональному назначению;
- рецепторные агенты, каждый из которых реагирует на определенные события во внешней среде и осуществляет первичное отражение этих событий в графодинамической памяти;
- эффекторные агенты, каждый из которых реагирует на определенного вида команды, формируемые внутренними агентами в графодинамической памяти, и осуществляет соответствующее изменение материального (физического) состояния интеллектуальной системы, которое определенным образом влияет на изменение её внешней среды.

Агенты могут работать параллельно, если одновременно возникают условия инициирования агентов. Асинхронность деятельности внутренних агентов заключается в том, что наличие условия инициирования агента ещё не означает начала его работы.

3.2. Унификация семантических моделей обработки знаний

В основе предлагаемого подхода к созданию технологии проектирования систем, управляемых знаниями, лежит принцип унификации семантических моделей интеграции знаний и семантических моделей интеграции целых систем, управляемых знаниями.

На основе унифицированных семантических сетей необходимо обеспечить построение унифицированных семантических моделей интеграции знаний (понимания знаний) и использовать эти модели, во-первых, как основу процесса приобретения интеллектуальной системой новых знаний как со стороны конечных пользователей, так и со стороны разработчиков; во-вторых, как основу интеграции программ и

различных семантических моделей расширения задач; в-третьих, как основу интеграции абстрактных логико-семантических моделей интеллектуальных систем.

Унифицированный способ кодирования различных видов знаний необходим для обеспечения совместимости различных видов знаний и различных языков представления знаний, что, в свою очередь необходимо для интеграции знаний [Ивашенко, 2012], а также для интеграции различных языков представления знаний. Без приведения интегрируемых знаний к одинаковой (общей, унифицированной) форме интеграция невозможна.

Унификация семантических моделей обработки знаний предполагает:

- унификацию структуризации баз знаний и типологии знаний;
- унификацию моделей информационного поиска;
- унификацию моделей интеграции знаний;
- унификацию моделей решения задач;
- унификацию моделей пользовательских интерфейсов;
- унификацию визуализации баз знаний.

3.3. Компонентное проектирование интеллектуальных систем

Проектируемые системы, управляемые знаниями, обладают высоким уровнем гибкости, их разработка в предлагаемой технологии осуществляется поэтапно, переходя от одной целостной версии системы к другой. При этом стартовая версия системы может быть ядром соответствующего класса систем, входящим в библиотеку многократно используемых компонентов. Предлагаемая технология концентрирует внимание не на создании стартовых версий разрабатываемых систем (это превращается в достаточно простую их сборку из имеющихся многократно используемых компонентов), а на расширении и совершенствовании текущих версий разрабатываемых систем в ходе их эксплуатации.

Проектирование по предлагаемой технологии имеет компонентный характер (модульный), основанный на библиотеке многократно используемых компонентов. Это многократное использование типовых компонентов уже разработанных компьютерных систем и модульное проектирование, заключающееся в сборке (синтезе) проектируемых систем из крупных ранее разработанных компонентов. Организация такого сборочного модульного проектирования требует создания, постоянного пополнения и сопровождения специальной библиотеки многократно используемых типовых ip-компонентов (компонентов интеллектуальной собственности – Intellectual Property Core).

Библиотека ip-компонентов – это, своего рода, систематизированный каталог комплектации для сборочного проектирования компьютерных систем, формальное описание коллективного опыта всех разработчиков таких систем.

3.4. Доступность и открытость технологии

В предлагаемой технологии осуществляется постоянное накопление и систематизация накапливаемого проектного опыта по разработке различных классов и различных видов компонентов систем, управляемых знаниями. Такое накопление проектного опыта осуществляется самими разработчиками прикладных систем. Конструктивным результатом этого процесса является расширение соответствующих разделов библиотеки многократно используемых компонентов.

Предлагаемая технология носит открытый и доступный характер как для ее пользователей (разработчиков прикладных систем, управляемых знаниями), так и для тех, кто желает участвовать в ее развитии.

Доступность предлагаемой технологии для ее понимания и усвоения обеспечивается тем, что документация по технологии оформляется как раздел базы знаний метасистемы, благодаря чему метасистема может выполнять функцию интеллектуальной справочной системы по предлагаемой технологии.

Реализация предлагаемой технологии осуществляется в виде метасистемы, управляемой знаниями, которая осуществляет поддержку проектирования систем по предлагаемой технологии, которая сама реализуется по этой же технологии и которая, как и другие системы, разрабатываемые по предлагаемой технологии, постоянно расширяется и совершенствуется. Таким образом, от гибкости (модифицируемости) проектируемых систем, управляемых знаниями, мы переходим к гибкости самой технологии их проектирования. Этой является важнейшим достоинством предлагаемой технологии, поскольку для любой технологии важен не столько текущий уровень ее развития, сколько трудоемкость и скорость ее перехода на новый уровень.

3.5. Принципы организации проекта OSTIS

Проект OSTIS, направленный на создание семантической технологии компонентного проектирования компьютерных систем, управляемых знаниями, и на применение этой технологии имеет следующие особенности:

- Предлагаемая технология является универсальной, т.к. любую компьютерную систему можно построить на ее основе, сутью предлагаемой технологии является новая парадигма построения компьютерных систем,

в которой доминирующую роль играют не программы, а обрабатываемая информация, имеющая в памяти системы онтологически унифицированное смысловое представление, основанное на согласованной системе понятий. При этом хранимая в памяти информация включает в себя как непосредственно обрабатываемую часть, так и часть, обеспечивающую управление процессом обработки информации, т.е. управление агентами, работающими над памятью (их инициирование и организацию взаимодействия).

- Развитие и расширение сфер применения предлагаемой технологии проектирования систем, управляемых знаниями, есть не что иное, как переход к компьютерным системам принципиально нового поколения, обладающих существенными преимуществами по сравнению с системами, построенными по традиционным технологиям.
- Развитие и расширение сфер применения предлагаемой технологии проектирования систем, управляемых знаниями, и, в частности, интеллектуальных систем (ИС), требует комплексного подхода и согласованного взаимодействия науки, образования, инженерии и бизнеса, см. рис.6.

Подчеркнем, что технология проектирования технических систем любого класса является ключевым фактором взаимодействия и продуктом сотрудничества соответствующей области науки, образования, инженерии и бизнеса.

Разработка большинства компьютерных систем сталкивается с недостаточностью развития используемых технологий. Поэтому разработка каждой такой системы должна включать в себя совершенствование используемой технологии – это необходимое условие создания конкурентоспособной системы. Из этого следует, что технологии проектирования сложных компьютерных систем должны быть гибкими (легко расширяемыми и модифицируемыми) и должен быть организован открытый массовый процесс развития каждой такой технологии, охватывающий всех ее пользователей. Если разработчики прикладных компьютерных систем сами не участвуют в развитии соответствующей технологии, то их квалификация и, следовательно, конкурентоспособность разрабатываемых ими систем будет падать.

Если обучение студентов и магистрантов разработке прикладных компьютерных систем на основе заданных технологий не сочетается с формированием у них навыков совершенствования этих технологий, то у молодых специалистов не появится способность быстро адаптироваться к новым тенденциям и новым уровням развития информационных технологий. Для того, чтобы студентов и магистрантов можно было учить не только использовать, но и совершенствовать различные технологии, эти технологии должны быть открытыми. Можно выделить три уровня подготовки молодых специалистов по разработке сложных компьютерных систем:

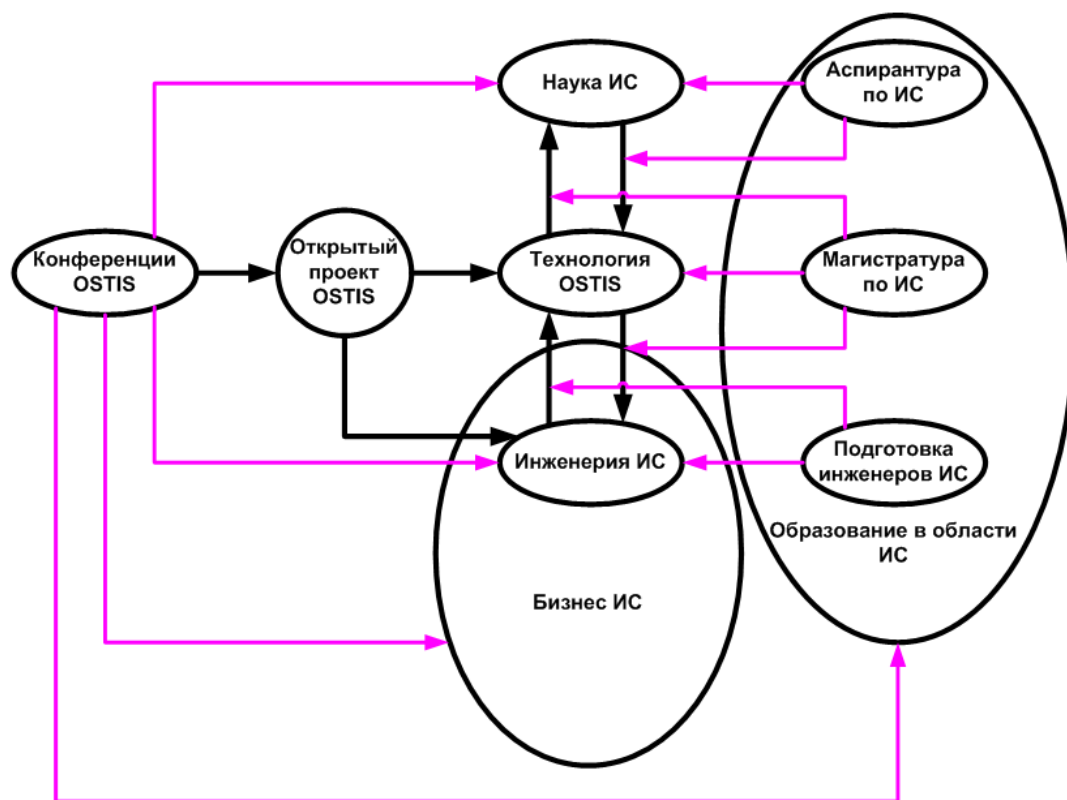


Рисунок 6 – Проект OSTIS как модель взаимодействия науки, образования, инженерии и бизнеса, направленного на развитие рынка совместимых компьютерных систем, управляемых знаниями

- (1) способность выполнять четко поставленные и не всегда простые индивидуальные проектные задания;
- (2) способность эффективно участвовать в коллективных решениях нечетко поставленных проектных задач;
- (3) способность эффективно участвовать в совершенствовании используемой технологии, как минимум, путем передачи в нее своего приобретенного проектного опыта в виде различных типовых решений.

4. Актуальные направления применения компьютерных систем, управляемых знаниями

Как уже было отмечено, любую компьютерную систему можно построить по архитектуре систем, управляемых знаниями, что даже без всякой интеллектуализации решателей задач переведет эти системы на качественно новый уровень, позволяющий существенно увеличить темпы их совершенствования в ходе эксплуатации и обеспечить их полную семантическую совместимость. Есть целый ряд областей применения компьютерных систем, настоятельно требующих того, чтобы эти системы были построены на основе знаний. Прежде всего, это те области применения компьютерных систем, в которых качество, структуризация и легкая модифицируемость контента играет решающую роль. Перечислим некоторые из таких областей.

- Мир вещей
- Умный дом
- Интеллектуальные робототехнические системы различного назначения
- Системы ситуационного управления, интеллектуального управления
- Мир Интернета
- Персональные агенты (ассистенты), живущие в интернете и способные вступать во взаимодействие с другими интернет-ориентированными агентами
- Интеллектуальные справочные системы различного назначения, способные отвечать на любые вопросы и решать любые задачи в заданной предметной области:
 - web-сайты с семантически структурированным контентом и с полным решением проблемы поиска информации, содержащейся в этом сайте;
 - интеллектуальные справочные системы учебного назначения, способные выполнить роль консультанта по заданной учебной дисциплине. В частности, актуальным является создание комплекса семантически совместимых интеллектуальных справочных систем по всем школьным дисциплинам и по общеобразовательным дисциплинам технических вузов (теория множеств, теория графов, теория отношений,

- абстрактная алгебра, математическая логика, теория алгоритмов и программ и др.);
- интеллектуальные help-системы (семантически структурированные руководства пользователей различных технических систем);
- порталы научных знаний, обеспечивающие постоянно совершенствуемую систематизацию и структуризацию научных знаний в соответствующих областях;
- семантически структурированный аналог Википедии, которая может полностью сохранить все существующие естественно-языковые статьи, сделав над ними семантически структурированную надстройку путем частичного или полного перевода этих статей на внутренний язык;
- семантически структурированные электронные научные журналы, актуализируемые соответствующими редакционными коллегиями, и автоматизирующие решение таких трудоемких задач, как рецензирование (в т.ч. и верификация доказательств) различных научных статей, которые должны быть представлены в формализованном виде (близком к внутреннему смысловому представлению знаний)
- Интеллектуальные корпоративные системы, управляемые знаниями
- Интеллектуальные системы автоматизации проектирования различных классов технических систем, управляемые знаниями, и способные к систематизации постоянно накапливаемого проектного опыта
- Интеллектуальные обучающие системы, включающие в себя соответствующие интеллектуальные справочные системы и имеющие возможности управлять деятельностью обучаемых, учитывая их индивидуальные особенности.

Заключение

Переход от традиционных принципов и технологий построения компьютерных систем, в которых доминируют программы, управляющие функционированием систем к принципам и технологиям, направленным на построение систем, в которых хранящиеся знания полностью управляют их деятельностью, инициируя активность различных агентов, осуществляющих обработку этих знаний, - неизбежен. Чем раньше мы станем активнее накапливать опыт в разработке систем, управляемых знаниями, и совершенствовать соответствующие технологии, тем конкурентоспособнее мы будем на следующем этапе развития информационных технологий. Опыт создания комплексной

технологии OSTIS и опыт разработки прикладных систем на ее основе создает необходимые предпосылки для начала перехода от традиционных компьютерных систем к системам, управляемым знаниями, в которых не всегда требуется умение решать интеллектуальные задачи. Процесс развития технологий проектирования систем, управляемых знаниями, должен быть перманентным, а фундаментом его должно быть активное сотрудничество науки, образования, инженерии и бизнеса.

Библиографический список

[Андреева, 2006] Андреева О.А., Боровикова О.И., Булгаков С.В. и др. Археологический портал знаний: содержательный доступ к знаниям и информационным ресурсам по археологии // Тр. 10-й национальной конференции по искусственному интеллекту с международным участием КИИ-2006 (25–28 сентября 2006 г., Обнинск). М.: Физматлит, 2006. Т. 3. С. 832–840.

[Боровикова, 2006] Боровикова О.И., Загорюлько Ю.А., Загорюлько Г.Б. и др. Разработка портала знаний по компьютерной лингвистике // Тр. 11-й национальной конференции по искусственному интеллекту с международным участием КИИ-2008 (Дубна, 2008 г.). М.: ЛЕНАНД, 2008. Т. 3. С. 380–388.

[Гаврилова и др., 2007] Гаврилова, Т.А. Интеллектуальные технологии в менеджменте: инструменты и системы: Учеб. пособие / Т. А. Гаврилова, Д.М. Муромцев; Высшая школа менеджмента СПбГУ. – СПб.: Изд-во «Высшая школа менеджмента»; Издат. дом С.-Петерб. гос ун-та, 2007. – 488 с.

[Гладун и др., 2006] Гладун А.Я., Рогушина Ю.В. Онтологический подход к поиску веб-сервисов в распределенной среде Интернет // Информатика, Минск, № 4, 2006. – С.116-127.

[Гладун и др., 2013] Гладун А.Я., Рогушина Ю.В. Репозитории онтологий как средство повторного использования знаний для распознавания информационных объектов // Онтология проектирования, № 1 (7), 2013. – С.35-50.

[Глоба, 2014] Л.С. Глоба, Р.Л. Новогрудская, Подход к построению формальной алгебраической системы порталов знаний, Онтология проектирования. - 2014. - №2(11). – ISSN 2223-9537- С.40-59

[Голенков и др, 2001] Голенков, В.В. Представление и обработка знаний в графодинамических ассоциативных машинах /В. В. Голенков [и др.] – Мн. : БГУИР, 2001.

[Ефименко, 2011] Онтологическое моделирование экономики предприятий и отраслей современной России: Часть 1. Онтологическое моделирование: подходы, модели, методы, средства, решения: препринт WP7/2011/08 (ч.1) [Текст] / И.В. Ефименко, В.Ф. Хорошевский; Нац. Исслед. ун-т «Высшая школа экономики». – М.: Изд. дом Высшей школы экономики, 2011. – 76 с.

[Джанетто и др., 2005] Джанетто, К. Управление знаниями: Руководство по разработке и внедрению корпоративной системы управления знаниями / К. Джанетто, Э. Уилер. – М.: Хорошая книга, 2005.

[Загорюлько, 2008] Загорюлько Ю.А., Боровикова О.И. Подход к построению порталов научных знаний // Автоматика. № 1, 2008, т. 44. –с. 100–110.

[Ивашенко, 2012] Ивашенко, В. П. Семантические модели и средства интеграции и отладки баз знаний / В. П. Ивашенко // Материалы международной научно-технической конференции «Открытые семантические технологии проектирования интеллектуальных систем» – Минск, 2012. –С. 193-204

[Касьянов, 2003] Касьянов, В.Н. Графы в программировании: обработка, визуализация и применение/ В. Н. Касьянов, В. А. Евстигнеев // ВHV–Санкт-Петербург, 2003.–1104 с.

[Клещев и др., 2011] Клещев, А.С. Облачная платформа для разработки и управления интеллектуальными системами / А.С. Клещев, В.В. Грибова, Д.А. Крылов, Ф.М. Москаленко, С.В. Смагин, В.А. Тимченко, М.Б. Тютюнник, Е.А. Шалфеева // материалы междунар. научн.-техн. конф. OSTIS-2011 (Минск, 10-12 февраля 2011г.). – Минск: БГУИР, 2011. С. 5-14.

[Кузнецов, 2007] Семантико-ориентированные системы на основе баз знаний / И.П. Кузнецов, А.Г. Мацкевич. – М.: Связьиздат, 2007. – 173с.

[Левыкин и др., 2014] Левыкин, М.В. Паттерны проектирования требований к информационным системам: моделирование и применение / В.М. Левыкин, М.В. Евланов, М.А. Керносов. Харьков: ООО «Компанія «СМІТ»», 2014. – 320 с.

[Лукашевич, 2011] Лукашевич, Н.В. Тезаурусы в задачах информационного поиска. – М.: Издательство Московского университета, 2011. – 512 с.

[Рогушина, 2014] Рогушина Ю.В. Знание-ориентированные средства поддержки семантического поиска в Web. – LAP LAMBERT Academic Publishing. 2014. – 214 с. – ISBN 978-3-659-56520-5.

[Рубашкин, 2012] Рубашкин, В.Ш. Онтологическая семантика. Знания. Онтологии. Онтологически ориентированные методы информационного анализа текстов. – М.: ФИЗМАТЛИТ, 2012. – 348с.

[Тузовский и др., 2005] Тузовский, А.Ф. Системы управления знаниями (методы и технологии) / А.Ф. Тузовский, С.В. Чириков, В.З. Ямпольский. – Томск: Изд-во НТЛ, 2005. – 260 с.

[Хорошевский, 2009] Хорошевский В.Ф. Пространства знаний в сети Интернет и Semantic Web (ч. 2) // Искусственный интеллект и принятие решений. 2009. №4. С. 15-36.

[Шведин, 2010] Онтология предприятия: экспириентологический подход: Технология построения онтологической модели представления / Шведин Б.Я. – М.:Ленард, 2010. – 210с.

SEMANTIC TECHNOLOGY OF COMPONENT DESIGN OF SYSTEMS, MANAGED BY KNOWLEDGES

Golenkov V.V., Guliakina N.A.

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

golenkov@bsuir.by

guliakina@bsuir.by

The article considers a result of five-year development of the Project OSTIS, oriented on the creation of Open semantic technology for intelligent systems design. The basis of that technology is the knowledge representation using semantic networks with theoretical-set interpretation. The article considers such system classes, as knowledge based systems and systems, managed by knowledges.



OSTIS-2015

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822:514

СРЕДСТВА ПОДДЕРЖКИ КОМПОНЕНТНОГО ПРОЕКТИРОВАНИЯ СИСТЕМ, УПРАВЛЯЕМЫХ ЗНАНИЯМИ

Шункевич Д.В., Давыденко И.Т., Корончик Д.Н., Жуков И.И., Паркалов А.В.

*Белорусский государственный университет информатики и радиоэлектроники,
г. Минск, Республика Беларусь*

shu.dv@tut.by

ir.davydenko@gmail.com

deniskoronchik@gmail.com

В работе рассматривается подход к проектированию систем, управляемых знаниями, ориентированный на использование совместимых многократно используемых компонентов, что существенно сокращает трудоемкость разработки таких систем. Работа по реализации данного подхода ведется в рамках открытого проекта OSTIS.

Ключевые слова: системы, управляемых знаниями; семантические сети; базы знаний; интеллектуальный решатель задач.

Введение

Как показывает анализ современных информационных технологий, что наряду с достижениями они имеют целый ряд серьезных недостатков, связанных с трудоемкостью их разработки и сопровождения [Голенков 2013]. В частности, к таким недостаткам можно отнести следующие:

- Отсутствует общее унифицированное решение проблемы семантической совместимости компьютерных систем, что порождает высокую трудоемкость создания комплексных интегрированных компьютерных систем. Наиболее остро проблема совместимости компьютерных систем проявляет себя при разработке web-ориентированных систем.

- Высока степень зависимости архитектур компьютерных систем от платформ, на которых они реализованы, что порождает высокую трудоемкость переноса компьютерных систем на новые платформы.

- Современные информационные технологии не ориентированы на широкий круг разработчиков прикладных компьютерных систем.

Для устранения указанных недостатков разрабатывается *Технология OSTIS [IMS]*, в основе которой лежит унифицированное представление любых видов знаний при помощи семантических сетей с теоретико-множественной интерпретацией.

Одной из основных задач данной технологии является обеспечение возможности быстро создавать системы, управляемые знаниями, с использованием готовых совместимых компонентов. Данная работа рассматривает типологию такого рода компонентов, их особенности и достоинства.

1. Описание Метасистемы IMS

К средствам поддержки компонентного проектирования систем, управляемых знаниями, относится сама *Метасистема IMS*, база знаний которой включает *Библиотеку многократно используемых компонентов OSTIS*, которая будет подробнее рассмотрена далее.

Все системы, построенные по *Технологии OSTIS* имеют общую унифицированную структуру базы знаний. *Метасистема IMS* также строится по *Технологии OSTIS*, и имеет следующую структуру базы знаний:

База знаний IMS

<= декомпозиция раздела:*

- {
- *Документация. Технология OSTIS*
- *Документация. IMS*
- *Контекст Технологии OSTIS и Метасистемы IMS в рамках Глобальной базы знаний*
- *История развития IMS*

- *Документация. Проект IMS*

Раздел *Документация. IMS* представляет собой описание собственно самой IMS: ее базы знаний, машины обработки знаний, пользовательского интерфейса.

Раздел *Контекст Технологии OSTIS и Метасистемы IMS в рамках Глобальной базы знаний* содержит знания, которые непосредственно не входят в описание *Технологии OSTIS*, текущего состояния *Метасистемы IMS* и процесса ее развития, но тесно связаны с этими знаниями в рамках *Глобальной базы знаний (Глобального смыслового пространства)*.

Раздел *История развития IMS* содержит историю изменений базы знаний IMS и других ее частей, то есть, по сути, прошлое системы.

Раздел *Документация. Проект IMS* содержит описание того, каким видится будущее системы, то есть описание планов по ее развитию, текущих задач, исполнителей и т.д.

Раздел *Документация. Технология OSTIS*, в свою очередь, имеет следующую структуру:

Документация. Технология OSTIS

<= декомпозиция раздела*:

- {
- *Обоснование разработки. Технология OSTIS*
- *Раздел. Унифицированные семантические сети и различные варианты их представления*
- *Раздел. Базовая модель обработки унифицированных семантических сетей*
- *Раздел. Унифицированные логико-семантические модели интеллектуальных систем*
- *Раздел. Семейство платформ интерпретации унифицированных логико-семантических моделей интеллектуальных систем*
- *Раздел. Средства проектирования унифицированных логико-семантических моделей интеллектуальных систем*
- *Раздел. Методика проектирования интеллектуальных систем по Технологии OSTIS*
- *Раздел. Библиотека OSTIS*
- }

Далее более подробно будет рассмотрено содержание Раздела *Библиотека OSTIS*, как одного из ключевых разделов *Документации OSTIS*, обеспечивающих поддержку компонентного проектирования систем, управляемых знаниями.

2. Библиотека многократно используемых компонентов OSTIS

Библиотека многократно используемых компонентов OSTIS

= Библиотека OSTIS

= многократно используемый компонент OSTIS

= многократно используемый компонент интеллектуальных систем, построенных по Технологии OSTIS

Под **многократно используемым компонентом OSTIS** понимается компонент некоторой интеллектуальной системы, который может быть использован в рамках другой интеллектуальной системы. Для этого необходимо выполнение как минимум двух условий:

- есть техническая возможность встроить компонент в другую систему, либо путем физического копирования, переноса и встраивания его в проектируемую систему, либо использования компонента, размещенного в исходной системе наподобие сервиса, то есть без явного копирования и переноса компонента. Трудоемкость встраивания зависит, в том числе, от реализации компонента;

- использование компонента в каких-либо системах, кроме материнской, является целесообразным, то есть компонентом не может быть частное решение, ориентированное на узкий круг задач. Стоит, однако, отметить, что в общем случае практически каждое решение может быть использовано в каких-либо других системах, круг которых определяется степенью общности и предметной зависимостью такого решения.

С формальной точки зрения каждый **многократно используемый компонент OSTIS** представляет собой *sc-структуру*, которая содержит все те (и только те) *sc-элементы*, которые необходимы для функционирования компонента в дочерней *sc-системе* и, соответственно, должны быть в нее скопированы при включении компонента в одну из таких систем. Конкретный состав данной *sc-структуры* зависит от типа компонента и уточняется для каждого типа отдельно. По сути, данная *sc-структура* представляет собой эталон или образец, который копируется при включении соответствующего компонента в дочернюю систему.

Каждый **многократно используемый компонент OSTIS** может быть атомарным, либо неатомарным, то есть состоять из более простых самодостаточных компонентов.

В зависимости от типа компонента в его составе, т.е. в составе соответствующей *sc-структуры*, могут дополнительно вводиться роли некоторых *sc-элементов*, если это необходимо. Например, в случае **многократно используемого sc-агента**, сам *sc-узел*, обозначающий *sc-агент*, будет являться **ключевым sc-элементом** в рамках компонента.

В каждый момент времени в текущем состоянии *sc-памяти* каждый многократно используемый компонент может представлен полностью, т.е. в памяти явно присутствуют все *sc-дуги принадлежности*, соединяющие соответствующую компоненту *sc-структуру* и все ее элементы, или представлен неявно, например, при помощи указания *ключевых sc-элементов'* данного компонента или путем задания декомпозиции данного компонента на более частные.

Каждый **многократно используемый компонент OSTIS** имеет формальную спецификацию, то есть некоторую *sc-окрестность*, характеризующую данный компонент. На основе формальной спецификации осуществляется поиск подходящего компонента в библиотеке, сравнение его с другими компонентами и т.д.

Данная спецификация включает, как минимум:

- Информацию об авторстве компонента, то есть связь компонента со знаком автора (физического лица, коллектива и т.д.) при помощи отношения *автор**;
- Информацию о типе компонента, посредством указания принадлежности компонента какому-либо классу многократно используемых компонентов;
- Описание назначения компонента, его особенностей;
- Историю изменений компонента по версиям;
- При необходимости сведения об открытости компонента и возможностях его использования в различных системах с точки зрения проприетарности;
- И др.

Не следует путать понятия **версии компонента** и **модификации компонента**. Версии отражают историю изменений компонента (как правило, какие-либо улучшения или устранения ошибок). Модификации представляют собой функционально эквивалентные, но разные варианты реализации одного и того же компонента, которые могут быть синтаксически эквивалентны (то есть быть реализованными при помощи одних и тех же языковых средств). В качестве примера синтаксически не эквивалентной модификации можно привести реализацию одного и того же *sc-агента* на одном и том же языке но с отличиями в алгоритме, в качестве синтаксически эквивалентной модификации – платформенно-зависимую и платформенно-независимую реализацию одного и того же *sc-агента*.

Библиотека многократно используемых компонентов OSTIS постоянно развивается. Появляются абсолютно новые компоненты, у существующих компонентов появляются новые модификации и новые версии.

В общем случае система *IMS* [Корончик 2014], [Давыденко 2014], [Шункевич 2014] как материнская система взаимодействует со всеми своими *дочерними sc-системами* (с системами, построенными по *Технологии OSTIS*), обеспечивая в дочерних системах автоматическое обновление версий *многократно используемых компонентов OSTIS*. Любая дочерняя система, построенная по *Технологии OSTIS*, в том числе, выполняет роль посредника между разработчиком такой системы и системой *IMS*. Разработчик имеет возможность выбрать интересующий его компонент или набор компонентов в одной из библиотек, и включить их в разрабатываемую дочернюю систему. Таким образом, можно говорить о том, что **разработчик систем, построенных по Технологии OSTIS, является конечным пользователем системы IMS**. При обеспечении такого механизма взаимодействия между системами, построенными на основе *Технологии OSTIS*, указанные системы формируют *Глобальную базу знаний*, в пределах которой различные системы могут координироваться и решать более глобальные задачи, нежели это может делать одна отдельно взятая система. В случае намеренной изоляции какой-либо из систем из такого коллектива, в частности, потери связи с системой *IMS*, она теряет возможность получать своевременные обновления используемых компонентов, а также использовать знания, накопленные в других системах для решения стоящих перед ней задач. Речь в данном случае идет не только о физической изоляции рассматриваемой системы, которая может быть легко устранена, а о рассогласовании знаний данной системы и других систем, что не позволит безболезненно интегрировать компоненты из библиотек в такую систему. Таким образом, разработчик каждой системы обязан следить за тем, чтобы его система находилась в постоянном согласовании с глобальным смысловым пространством, что позволит пользователям в полной мере использовать все возможности коллектива систем.

В некоторых случаях может оказаться, что для использования одного **многократно используемого компонента OSTIS** целесообразно или даже необходимо дополнительно использовать несколько других **многократно используемых компонентов OSTIS**. Например, может оказаться целесообразным вместе с каким либо *sc-агентом информационного поиска* использовать соответствующую команду интерфейса, которая представлена отдельным компонентом и позволит пользователю задавать вопрос для указанного *sc-агента* через интерфейс системы. В таких случаях для связи компонентов используется отношение *сопутствующий компонент**. Наличие таких связей позволяет устранить возможные проблемы неполноты знаний и навыков в дочерней системе, из-за которых какие-либо из компонентов могут не выполнять свои функции. Связки отношения *сопутствующий компонент** связывают **многократно используемые компоненты OSTIS**, которые целесообразно или

необходимо использовать в дочерней системе вместе. При этом каждая связка направляется от зависящего компонента к зависимому. Каждая такая связка может дополнительно быть снабжена *sc-комментарием* или *sc-пояснением*, отражающим суть указываемой зависимости.

Включение компонента в дочернюю *sc-систему* в самом общем случае состоит из следующих этапов:

- поиск подходящего компонента (или набора компонентов) во множестве библиотек, входящих в состав *IMS*. Для облегчения задачи поиска могут быть реализованы специализированные поисковые агенты. В любом случае, поиск и выделение компонента будет осуществляться на основе спецификации компонента. Данный этап с точки зрения пользователя не зависит от типа компонента и особенностей его реализации. Конкретные действия на следующих этапах сильно зависят от реализации и типа компонента и будут более детально описаны при рассмотрении подклассов **многократно используемых компонентов OSTIS**;
- выделение компонента (или набора компонентов) в рамках *IMS* в виде, удобном для транспортировки в дочернюю *sc-систему* (при необходимости – создание физической копии компонента);
- транспортировка выделенного компонента в дочернюю *sc-систему*;
- интеграция компонента в дочернюю *sc-систему*. Если в системе уже использовалась более старая версия компонента, то необходимо произвести либо локальное обновление, либо полную замену устаревшей версии компонента. Дальнейший процесс интеграции зависит от типа компонента, например, в случае добавления нового *sc-агента* он должен быть помечен как *активный sc-агент* и т.п.

Для обеспечения возможности встраивания **многократно используемых компонентов OSTIS** в дочернюю систему, каждая такая система обязана иметь в своем составе средства, обеспечивающие интеграцию новых компонентов в систему и, при необходимости, удаление устаревших версий этих компонентов (или автоматического локального обновления компонентов до более новой версии).

3. Классификация компонентов

Рассмотрим классы *многократно используемых компонентов OSTIS*.

3.1. Классификация компонентов по характеру их копируемости

Библиотека многократно используемых компонентов OSTIS

*<= разбиение**:

- {
- копируемый компонент OSTIS
- шаблон типового компонента OSTIS
- }

шаблон типового компонента OSTIS

= образец типового компонента OSTIS

= атомарная логическая формула, описывающая структуру аналогичных (чаще всего изоморфных) компонентов баз знаний *sc-систем*

В процессе использования **шаблона типового компонента OSTIS** при формировании баз знаний проектируемых *sc-систем* вместо *sc-переменных*, входящих в состав компонента, подставляются их значения.

В случае **копируемых компонентов OSTIS** в каждой дочерней системе создается конструкция, полностью синонимичная исходному компоненту.

На множестве **копируемых компонентов OSTIS** задано отношение *семантическое включение**, т.е. связь более детального описания с менее детальным. Это значит, что степень детализации каких-либо типовых знаний, включаемых в состав баз знаний различных проектируемых *sc-систем*, может быть различной.

3.2. Классификация компонентов по признаку атомарности

Библиотека многократно используемых компонентов OSTIS

*<= разбиение**:

- {
- атомарный многократно используемый компонент OSTIS
- неатомарный многократно используемый компонент OSTIS
- }

Под **атомарным многократно используемым компонентом OSTIS** подразумевается компонент, который в текущем состоянии библиотеки компонентов рассматривается как неделимый, то есть не содержит в своем составе других компонентов, представленных в какой-либо из библиотек компонентов в рамках *IMS*. В общем случае атомарный компонент может перейти в разряд неатомарных в случае, если будет принято решение выделить какую-то из его частей в качестве отдельного компонента. Все вышесказанное, однако, не означает, что даже в случае его платформенной независимости, атомарный компонент всегда хранится в *sc-памяти* как сформированная *sc-структура*. Например, *платформенно-независимая реализация sc-агента* всегда будет представлена набором *scr-программ*, но будет **атомарным многократно используемым компонентом OSTIS** в случае, если ни одна из этих программ не будет представлять интереса как самостоятельный компонент.

Под **неатомарным многократно используемым компонентом OSTIS** подразумевается компонент, который в текущем состоянии библиотеки компонентов содержит в своем составе более простые компоненты, представленные в каких-либо библиотеках компонентов в рамках IMS. В общем случае неатомарный компонент может перейти в разряд атомарных в случае, если будет принято решение по каким-либо причинам исключить все его части из рассмотрения в качестве отдельных компонентов.

Следует отметить, что неатомарный компонент необязательно должен складываться полностью из других компонентов, в его состав могут также входить и части, не являющиеся самостоятельными компонентами. Например, в состав реализованного на языке SCP *sc-агента*, являющегося **неатомарным многократно используемым компонентом** могут входить как *scr-программы*, которые могут являться **многократно используемыми компонентами** (а могут и не являться), а также *агентная scr-программа*, которая не имеет смысла как многократно используемый компонент.

3.3. Классификация компонентов по признаку платформенной независимости

Библиотека многократно используемых компонентов OSTIS

<= разбиение*:

- ```
{
 • платформенно-зависимый многократно
 используемый компонент OSTIS
 • платформенно-независимый многократно
 используемый компонент OSTIS
}
```

Под **платформенно-зависимым многократно используемым компонентом OSTIS** понимается компонент, частично или полностью реализованный при помощи каких-либо сторонних с точки зрения Технологии OSTIS средств. Основной недостаток платформенно-зависимых компонентов состоит в том, что их интеграция в интеллектуальные системы может сопровождаться дополнительными трудностями, зависящими от конкретных средств реализации компонента. В качестве возможного преимущества **платформенно-зависимых многократно используемых компонентов OSTIS** можно выделить их, как правило, более высокую производительность за счет реализации их на более приближенном к платформе уровне.

В общем случае **платформенно-зависимый многократно используемый компонент OSTIS** может поставляться как в виде набора исходных кодов, так и в бинарном виде, например в виде скомпилированной библиотеки.

Процесс интеграции **платформенно-зависимого многократно используемого компонента OSTIS** в дочернюю систему, разработанную по Технологии OSTIS, сильно зависит от технологий реализации

данного компонента и в каждом конкретном случае может состоять из различных этапов.

Для того чтобы **платформенно-зависимый многократно используемый компонент OSTIS** мог быть успешно встроен в дочернюю систему, необходимо выполнение следующих условий:

- в состав *sc-структуры*, соответствующей компоненту, должны входить *sc-ссылки*, содержащие сведения о местонахождении исходных текстов компонента или уже собранной его версии, то есть ссылки на внешние ресурсы или явно включенные в систему файлы компонента в виде указанных *sc-ссылок*;
- каждый **платформенно-зависимый многократно используемый компонент OSTIS** должен иметь соответствующую подробную, корректную и понятную инструкцию по его установке и внедрению в дочернюю систему;

Под **платформенно-независимым многократно используемым компонентом OSTIS** понимается компонент, который целиком и полностью представлен в SC-коде. В случае *неатомарного многократно используемого компонента* это означает, что все более простые компоненты, входящие в его состав также обязаны быть **платформенно-независимыми многократно используемыми компонентами OSTIS**.

Процесс интеграции **платформенно-зависимого многократно используемого компонента OSTIS** в дочернюю систему, разработанную по Технологии OSTIS, существенно упрощается за счет использования общей унифицированной формальной основы представления и обработки знаний.

В случае **платформенно-независимого многократно используемого компонента OSTIS** процесс интеграции конкретизируется до следующих этапов:

- формирование *sc-структуры*, явно содержащей все *sc-элементы*, входящие в состав компонента, а также все *sc-элементы*, входящие в спецификацию компонента, необходимую для его функционирования в дочерней системе. В случае **неатомарного многократно используемого компонента OSTIS** в указанную *sc-структуру* должны быть полностью включены и все более частные компоненты;
- транспортировка компонента в дочернюю систему. В худшем случае может быть осуществлена путем выгрузки всей *sc-структуры* компонента в какой-либо формат, например SCs-код, с последующим переносом файла в дочернюю систему разработчиком вручную. В общем случае, дочерняя система по команде разработчика должна

самостоятельно обращаться к родительской и осуществлять загрузку необходимых компонентов;

- интеграция нового компонента в дочернюю sc-систему либо обновление компонента с более старой версии. Если функция обновления не поддерживается, то интеграция может проводиться в два этапа – удаление старой версии компонента и добавление в систему более новой версии. В зависимости от типа компонента (*многократно используемый компонент баз знаний* или *многократно используемый компонент машин обработки знаний*) интеграция осуществляется по-разному.

### 3.4. Классификация компонентов по их месту в архитектуре системы, построенной по Технологии OSTIS

**Библиотека многократно используемых компонентов OSTIS**

=> включение\*:

- Библиотека платформ реализации sc-систем
- Библиотека многократно используемых компонентов sc-моделей баз знаний
- Библиотека многократно используемых компонентов sc-машин
- Библиотека многократно используемых компонентов sc-моделей интерфейсов
- Библиотека типовых подсистем интеллектуальных систем

**Библиотека платформ реализации sc-моделей**

= Библиотека интерпретаторов унифицированных логико-семантических моделей интеллектуальных систем

= интерпретатор унифицированных логико-семантических моделей интеллектуальных систем

= платформа реализации sc-моделей

Под **платформой реализации sc-моделей** понимается некоторая реализация платформы интерпретации указанных моделей, которая в общем случае включает в себя:

- хранилище sc-текстов (*sc-хранилище*);
- файловую память *sc-машины*;
- средства, обеспечивающие взаимодействие *sc-агентов* над общей памятью;
- базовые средства интерфейса для взаимодействия системы с внешним миром (пользователем или другими системами). Указанные средства включают в себя, как минимум, редактор, транслятор (в *sc-память* и из нее) и визуализатор для одного из базовых универсальных вариантов представления *SC-кода* (*SCg-код*, *SCs-код*, *SCn-код*), средства, позволяющие задавать системе вопросы из некоторого универсального класса (например,

запрос семантической окрестности некоторого объекта);

- *абстрактную scr-машину*, то есть интерпретатор *scr-программ*.

При необходимости, в **платформу реализации sc-моделей** могут быть заранее на платформенно-зависимом уровне включены какие-либо компоненты машин обработки знаний или баз знаний, например, с целью упрощения создания первой версии дочерней системы на основе Технологии OSTIS.

Реализация платформы может осуществляться на основе произвольного набора существующих технологий, включая аппаратную реализацию каких-либо ее частей. С точки зрения Технологии OSTIS любая **платформа реализации sc-моделей** является **платформенно-зависимым многократно используемым компонентом**.

**Библиотека многократно используемых компонентов sc-моделей баз знаний**

= **многократно используемый компонент sc-моделей баз знаний**

Каждый **многократно используемый компонент sc-моделей баз знаний** представляет собой *sc-структуру*, либо явно представленную в текущем состоянии *sc-памяти*, либо не полностью сформированную *sc-структуру*, которая при необходимости может быть полностью сформирована путем объединения своих частей, указанных при помощи какого-либо *отношения декомпозиции*, например *разбиение\**, или *отношения включения\**.

Интеграция **многократно используемого компонента sc-моделей баз знаний** в дочернюю систему сводится к склеиванию ключевых узлов по идентификаторам и устранению возможных дублирований и противоречий, которые могли возникнуть в случае, если разработчик дочерней системы вручную вносил какие-либо изменения в ее базу знаний.

К основным типам компонентов баз знаний, хранящихся в библиотеке компонентов баз знаний, относятся:

- онтологии различных предметных областей, которые могут быть самыми различными по содержанию, однако должны быть семантически совместимыми;
- базовые фрагменты теорий, соответствующие различным уровням знания пользователя, начиная от базового школьного до профессионального;
- различные *sc-окрестности* различных объектов;
- спецификации формальных языков описания различных предметных областей.

Для обеспечения семантической совместимости компонентов баз знаний, которые являются унифицированными семантическими моделями, необходимо

- согласовать семантику всех используемых ключевых узлов;
- согласовать *системные идентификаторы\** ключевых узлов, используемых в разных компонентах. После этого интеграция всех компонентов, входящих в состав библиотеки, и в любых комбинациях осуществляется автоматически, без вмешательства разработчика.

Для включения компонента в библиотеку необходимо его специфицировать по следующим критериям:

- предметная область, описание которой содержится в компоненте;
- класс (тип) компонента базы знаний;
- состав компонента;
- количественные характеристики ключевых узлов компонента;
- информация о разработчиках компонента;
- дата создания компонента;
- информация о верификации компонента;
- версия компонента;
- условия распространения компонента базы знаний;
- сопровождающая информация.

К средствам проектирования баз знаний также относятся:

- средства верификации баз знаний, включающие пополняемую библиотеку команд и *sc-агентов* верификации баз знаний;
- средства анализа качества баз знаний, позволяющие определить такие характеристики баз знаний как полнота, связность;
- средства редактирования баз знаний, решающие проблему синхронизации редактирования семантической модели базы знаний и соответствующего фрагмента ее исходного текста несколькими разработчиками;
- средства поддержки коллективной разработки баз знаний.

#### **Библиотека многократно используемых компонентов sc-машин**

= многократно используемый компонент sc-машин обработки знаний

Если *многократно используемый компонент sc-машин* является *платформенно-зависимым многократно используемым компонентом OSTIS*, то его интеграция производится в соответствии с инструкцией, как и для любого компонента такого

рода. В противном случае, процесс интеграции можно конкретизировать в зависимости от подклассов данного типа компонентов.

#### **Библиотека многократно используемых компонентов sc-машин**

<= разбиение\*:

- ```
{
• Библиотека многократно используемых sc-агентов
• Библиотека многократно используемых программ обработки sc-текстов
}
```

Библиотека многократно используемых программ обработки sc-текстов

= многократно используемая программа обработки sc-текстов

Под *многократно используемой программой обработки sc-текстов* подразумевается компонент, соответствующий программе, записанной на произвольном языке программирования, которая ориентирована именно на обработку знаний, то есть с точки зрения *Технологии OSTIS*, обработку sc-конструкций. Приоритетным в данном случае является использование *scr-программ* по причине их платформенной независимости, за исключением случаев проектирования некоторых компонентов интерфейса, когда полная платформенная независимость невозможна (например, при проектировании эффекторных и рецепторных *sc-агентов*).

Библиотека многократно используемых программ обработки sc-текстов

=> включение*:

Библиотека многократно используемых *scr-программ*

Библиотека многократно используемых sc-агентов

= многократно используемый sc-агент

Под *многократно используемым sc-агентом* подразумевается компонент, соответствующий некоторому *абстрактному sc-агенту*, который может быть использован в других системах, возможно, в составе более сложных *неатомарных абстрактных sc-агентов*. Указанный абстрактный sc-агент входит в соответствующую компоненту sc-структуру под атрибутом *ключевой sc-элемент'*. Каждый *многократно используемый sc-агент* должен содержать всю информацию, необходимую для функционирования соответствующего *sc-агента* в дочерней системе.

Таким образом, соответствующая *многократно используемому sc-агенту sc-структура* формируется следующим образом:

- 1) в нее включается *sc-узел*, обозначающий соответствующий *абстрактный sc-агент*, и вся его спецификация, то есть, как минимум, указание *ключевых sc-элементов sc-агента**, условия

иницирования и результат*, первичного условия инициирования*, *sc-описание поведения sc-агента* и класса решаемых им задач;

2) в случае, если входящий в **многократно используемый sc-агент** абстрактный *sc-агент* рассматривается как *неатомарный абстрактный sc-агент*, то **многократно используемый sc-агент** будет содержать *sc-узлы*, обозначающие все более частные *абстрактные sc-агенты*, а также все их спецификации согласно п.1. Для каждого включенного в **многократно используемый sc-агент** абстрактного *sc-агента* необходимо выполнить п.2 и п.3;

3) для каждого *атомарного абстрактного sc-агента*, знак которого вошел в **многократно используемый sc-агент** необходимо выбрать вариант его реализации (то есть элемент класса **платформенно-независимый абстрактный sc-агент** или **платформенно-зависимый абстрактный sc-агент**, связанный с исходным *атомарным абстрактным sc-агентом* связкой отношения *включение**) и включить в **многократно используемый sc-агент** *sc-узел*, обозначающий указанную реализацию, а также знаки всех программ, входящие во множество, связанное с указанной реализацией отношением *программа sc-агента**. Выбранная реализация включается в **многократно используемый sc-агент** под атрибутом *ключевой sc-элемент'*.

4) для всех *scp-программ*, знаки которых включены в **многократно используемый sc-агент**, необходимо включить в него полный текст *scp-программы*, то есть все параметры с указанием типа параметра, *scp-операторы*, входящие в ее состав, все соответствующие им операнды с указанием всех выполняемых ими ролей, связи отношения *следующий оператор** для этой операторов этой *scp-программы* и т.д.;

5) в **многократно используемый sc-агент** включаются также все связи отношений, указанных в п.1-4, связывающие уже включенные в его состав *sc-элементы*, а также сами знаки этих отношений (например, *включение**, *программа sc-агента** и т.д.);

После того, как **многократно используемый sc-агент** был скопирован в дочернюю систему, необходимо сгенерировать *sc-узел*, обозначающий конкретный *sc-агент*, работающий в данной системе, принадлежащий выбранной реализации *абстрактного sc-агента* и добавить его во множество *активных sc-агентов* при необходимости.

Также каждую *scp-программу*, попавшую в дочернюю систему при копировании **многократно используемого sc-агента**, необходимо добавить ее

во множество *корректных scp-программ* (корректность верифицируется при попадании в библиотеку компонентов в рамках *IMS*).

Библиотека многократно используемых scp-программ

= *многократно используемая scp-программа*

Под **многократно используемой scp-программой** понимается компонент, соответствующий некоторой универсальной *scp-программе*, которая может быть использована в составе сразу нескольких *sc-агентов*.

В **многократно используемую scp-программу** включается полный текст *scp-программы*, то есть все параметры с указанием типа параметра, все *scp-операторы*, входящие в ее состав, все соответствующие им операнды с указанием всех выполняемых ими ролей, связи отношения *следующий оператор** для этой операторов этой *scp-программы* и т.д. Сам *sc-узел*, обозначающий *scp-программу*, входит в соответствующий компонент под атрибутом *ключевой sc-элемент'*.

После того, как **многократно используемая scp-программа** была скопирована в дочернюю систему, необходимо добавить ее во множество *корректных scp-программ* (корректность верифицируется при попадании в библиотеку компонентов в рамках *IMS*).

Библиотека типовых подсистем интеллектуальных систем

= *многократно используемая типовая подсистема интеллектуальных систем*

Каждая **многократно используемая типовая подсистема интеллектуальных систем** в общем случае представляет собой совокупность *sc-агентов* и фрагментов баз знаний, которые объединены для решения какого-либо класса задач. Таким образом, указанная совокупность не может быть отнесена ни к *многократно используемым компонентам sc-моделей баз знаний*, ни к *многократно используемым компонентам sc-машин*.

Библиотека многократно используемых компонентов sc-моделей интерфейсов

= *многократно используемый компонент sc-моделей интерфейсов*

Каждый **многократно используемый компонент sc-моделей интерфейсов** может быть отнесен также к классу *многократно используемый компонент sc-моделей баз знаний* либо *многократно используемый компонент sc-машин*, однако такие компоненты обладают своей спецификой и поэтому выделяются в отдельный класс. **Многократно используемые компоненты sc-моделей интерфейсов** отвечают за взаимодействие интеллектуальной системы с внешней средой, включая также файлы *sc-машин*.

4. Средства автоматизации проектирования систем, управляемых знаниями

Средства автоматизации проектирования систем, управляемых знаниями. можно разделить на средства, входящие в состав *Метасистемы IMS* и средства, которые содержатся в самой *дочерней sc-системе* и представляют собой *многократно используемую типовую подсистему интеллектуальных систем*.

В рамках интеллектуальной *Метасистемы IMS* выделяется подсистема для поддержки проектирования различных компонентов и различных классов интеллектуальных систем.

В рамках рассматриваемой подсистемы необходимо выделить *sc-агент сборки многократно используемого компонента*.

Задачей *sc-агента сборки многократно используемого компонента* является формирование *sc-структуры*, соответствующей выбранному *многократно используемому компоненту*, то есть явную генерацию всех *sc-дуг принадлежности* из указанной *sc-структуры* в ее элементы. После завершения работы данного агента рассматриваемая *sc-структура* становится полностью сформированным множеством, а после копирования компонента в дочернюю систему с целью экономии памяти сгенерированные дуги могут снова быть удалены до следующего аналогичного запроса. Данный агент декомпозируется на более частные абстрактные *sc-агенты*.

sc-агент сборки многократно используемого компонента

<= декомпозиция абстрактного sc-агента:*

- ```
{
• sc-агент сборки многократно используемого компонента sc-моделей баз знаний
• sc-агент сборки многократно используемого sc-агента
• sc-агент сборки многократно используемой scr-программы
}
```

В свою очередь, в составе каждой дочерней *sc-системы* выделяется подсистема поддержки проектирования этой системы. В ее состав входит *sc-агент импорта многократно используемого компонента в дочернюю систему*

##### *sc-агент импорта многократно используемого компонента в дочернюю систему*

*<= декомпозиция абстрактного sc-агента\*:*

- ```
{
• sc-агент импорта многократно используемого компонента sc-моделей баз знаний в дочернюю систему
```

- *sc-агент импорта многократно используемого sc-агента в дочернюю систему*
- *sc-агент импорта многократно используемой scr-программы в дочернюю систему*

Заключение

В работе рассмотрена типология многократно используемых компонентов, используемых в рамках открытой семантической технологии проектирования интеллектуальных систем.

Библиографический список

[Голенков, 2013] Голенков, В.В., Гулякина Н.А. Открытый проект, направленный на создание технологии компонентного проектирования интеллектуальных систем./В.В. Голенков, Н.А. Гулякина// Открытые семантические технологии проектирования интеллектуальных систем (OSTIS-2013): материалы III междунар. науч.-техн.конф. – Минск: БГУИР, 2013 – с.55-78

[IMS] Документация. Технология OSTIS. [электронный ресурс]. – Режим доступа: <http://ims.ostis.net/>

[Корончик 2014] Корончик, Д.Н. Пользовательский интерфейс интеллектуальной метасистемы поддержки проектирования интеллектуальных систем./ Д.Н. Корончик// Открытые семантические технологии интеллектуальных систем (Ostis-2014): Материалы IV международной науч.-тех. конф. – Минск: БГУИР, 2014 – с.79-82

[Давыденко 2014] Гракова Н.В., Давыденко И.Т., Русецкий К.В. База знаний интеллектуальной метасистемы поддержки проектирования интеллектуальных систем./ Н.В. Гракова, И.Т., Давыденко, К.В. Русецкий// Открытые семантические технологии интеллектуальных систем (Ostis-2014): Материалы IV международной науч.-тех. конф. -- Минск: БГУИР,2014. – с.83-92

[Шункевич 2014] Шункевич, Д.В. Машина обработки знаний интеллектуальной метасистемы поддержки проектирования интеллектуальных систем. /Д.В. Шункевич// Открытые семантические технологии интеллектуальных систем (Ostis-2014): Материалы IV международной науч.-тех. конф -- Минск: БГУИР,2014. – с.93-96

SUPPORT TOOLS KNOWLEDGE-BASED SYSTEMS COMPONENT DESIGN

Shunkevich D.V., Davydenko I.T., Koronchik D.N., Zukov I.I., Parkalov A.V.

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

shu.dv@tut.by

ir.davydenko@gmail.com

deniskoronchik@gmail.com

The paper shows a manage knowledge systems design approach, oriented on using compatible reusable components, which help to reduce laboriousness in systems development. The work is carried out within the framework of OSTIS open project.

Key words: manage knowledge systems, semantic networks, knowledge bases, intelligent task resolvent.

Introduction

Analisation of modern information technology demonstrates a number of serios drawbacks associated with laboriousness of their development and maintenance. In particular, there are the following disadvantages:

- There is no common unified problem solution of the computer systems semantic compatibility. It creates difficalty of laboriosness in comprehensive integrated computer systems design, which is particularly acute in development of web-oriented systems.
- The computer system's architecture is platform-dependent and hardly transfered.
- Modern information technology is not focused on a wide range of applied computer systems development engineeres.

To eliminate these shortcomings the OSTIS technology is developed, based on unified knolege representation using the sematic networks whith set-theoretic interpretation. One of the main goal of this technology is to insure knowledge based systems production applying the ready-made compossible components.

Main Part

Metasystem IMS is a supporting tools for manage knowledge systems component design, which knowledge base includes *Library of reusable components OSTIS*, an it will be further showed.

All systems based on *OSTIS Technology*, have the general structure of knowledge base. *Metasystem IMS* is also based on *OSTIS Technology* have the following structure of knowledge base:

Knowledge base IMS

<= section decomposition*:

- {
- *OSTIS Technology documentation*
- *IMS documentation*
- *OSTIS Technology and IMS Metasystem context inside of Global knowledge base.*
- *Historyof IMS evolution*
- *Documentation. IMS Project*
- }

Section *IMS Documentation* is a description of IMS itself: it's knowledge base, knowledge elaboration machine and human-system interface.

Section *OSTIS Technology and IMS Metasystem context inside of Global knowledge base* contain knowledge, which are not included in OSTIS Technology description, but are closely related with knowledge of *Global knowledge base*.

Section *History of IMS evolution* contain the history of IMS knowledge base and it's parts changes.

Section *Documentation. IMS Project* contains the description of its future state.

Conclusion

In the paper described the typology of reusable components, which are used in open semantic technology of intelligent systems design.



УДК 004.822:514

РЕАЛИЗАЦИЯ ПЛАТФОРМЫ ДЛЯ WEB-ОРИЕНТИРОВАННЫХ СИСТЕМ, УПРАВЛЯЕМЫХ ЗНАНИЯМИ

Корончик Д. Н.

* *Белорусский государственный университет информатики и радиоэлектроники,
г. Минск, Республика Беларусь*

denis.koronchik@gmail.com

В работе приводится описание технической реализации платформы для web-ориентированных систем, управляемых знаниями. Описаны основные принципы, которые лежат в основе платформы: ориентация на коллективную разработку базы знаний, возможность отображения (редактирования) базы знаний на различных внешних языках и их отображение в режиме реального времени.

Ключевые слова: клиент-сервер, компоненты пользовательского интерфейса, платформа.

Введение

В рамках *Проекта OSTIS* [Голенков, 2013] ведется разработка платформы, которая позволяет реализовывать web-ориентированные системы, управляемые знаниями. Познакомиться с описанием предыдущей версии можно по ссылке [Корончик, 2014]. Основные положения, которые лежат в основе указанной платформы следующие:

- база знаний (БЗ) системы редактируется множеством разработчиков, что можно даже делать во время эксплуатации системы;
- разграничение доступа к различным частям базы знаний;
- возможность отображать (редактировать) части базы знаний с использованием различных внешних языков (SCg-код [Голенков и др, 2001], SCn-код, чертежи, ЕЯ и т. д.);
- изменения в базе знаний в режиме реального времени должны отображаться у всех пользователей системы.

Архитектура

Архитектура разработанной платформы показана на рисунке 1. Используется клиент-серверная архитектура, где клиентская реализована с использованием web-технологий. Использование web-технологий в реализации клиентской части позволяет использовать приложение в любом (достаточно новом) браузере. Как видно из рисунка 1, платформа состоит из следующих частей:

- http сервер – серверное приложение, которое выполняет следующие функции:

- обработка http запросов (их список можно посмотреть в [sc-web api, 2014]);
- перенаправления запросов из клиентской части в sc-память и событий из sc-памяти в клиентскую часть;
- авторизация и уровни доступа пользователей;
- sc-память – это хранилище текстов записанных с помощью SC-кода [Корончик, 2013];
- машина обработки знаний – набор sc-агентов, которые обеспечивают функционирование системы;
- компоненты ядра пользовательского интерфейса (ПИ) – это основа клиентской части, которая обеспечивает интеграцию компонентов ПИ и их взаимодействие с базой знаний;
- компоненты ПИ – обеспечивают ввод и вывод информации представленной с помощью различных внешних языков.

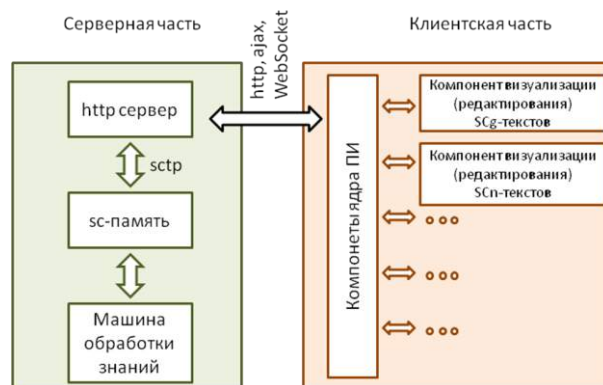


Рисунок 1 – Архитектура программной платформы для реализации web-ориентированных систем основанных на знаниях

Реализация **http сервера** основана на использовании библиотеки tornado [tornado, 2014]. Предыдущая версия http сервера была реализована с помощью библиотеки django. Однако для того, чтобы появилась возможность реализации sc-агентов ПИ на клиентской части, необходимо было реализовать возможность инициировать функции в JavaScript по событиям из sc-памяти. С этой целью было принято решение использовать стандарт WebSocket [websocket, 2014]. В связи с этим была использована именно библиотека tornado.

При старте приложения в браузере, устанавливается WebSocket соединение между сервером и клиентской частью. На стороне клиента реализована библиотека, которая позволяет, используя бинарный протокол sctp [sctp, 2014], работать с sc-памятью, для этого в http сервере имеется небольшой модуль, который преобразует запросы из WebSocket в TCP/IP Socket и обратно (параллельно проверяя наличие необходимого уровня доступа на выполнение той или иной команды). Авторизация на сервере реализована с помощью google (пользователи имеющие акант google могут авторизоваться на сайте). Каждому пользователю устанавливается роль (по умолчанию это Гость с минимальными правами на чтение и редактирование). Управление ролями пользователя может выполнять администратор через панель администратора (как и во всех современных сайтах).

Чтобы обеспечить возможность одновременной работы множества пользователей с системой была пересмотрена и значительно улучшена реализация **sc-памяти** [Корончик, 2013]. Основные изменения, которые были внесены в реализацию sc-памяти:

- переработан механизм синхронизации при обработке команд из разных потоков. Раньше выполнение команды одного потока блокировало выполнение команд из других потоков, что приводило к большому падению производительности при увеличении числа потоков (клиентов). Новый механизм делит каждый сегмент на N частей. При попытке заблокировать элемент со смещением X в сегменте, он блокирует X % N часть (% - операция получения остатка от деления). Таким образом, при обращении к элементам в разных сегментах или разных частях сегмента, выполняемые в разных потоках команды не блокируют друг друга;

- уменьшено количество потребляемой оперативной памяти, за счет уменьшения размера ячейки sc-памяти с 56 до 44 байт, что дает экономию в 21%. Это было достигнуто за счет удаления временных меток из sc-элемента. Теперь команда удаления сразу удаляет элементы, чтобы остальные команды не могли использовать удаляемые элементы, так как они блокируются. Если на удаляемый sc-элемент указывает итератор, то элемент просто помечается как необходимый к удалению (он становится недоступным для инициированных после этого команд) и как только

количество ссылающихся на него итераторов становится равным нулю, элемент удаляется;

- для каждого sc-элемента были добавлены поля, которые обозначают уровень доступа к sc-элементу. Выделено 16 уровней доступа на чтение и 16 уровней доступа на запись (соответственно для хранения используется 1 байт). Правило работы с ними очень простое – каждой программе (sc-агенту, пользователю), которая использует sc-память, ставится в соответствие её уровень доступа. Эта программа имеет право изменять (читать) sc-элемент, если её уровень доступа на изменение (чтение) больше, либо равен уровню доступа у sc-элемента. Кроме того, если уровень доступа позволяет изменять sc-элемент, то программа имеет право изменить его на любое другое значение (как в меньшую, так и в большую сторону).

Машина обработки знаний не претерпела больших изменений. Она была лишь адаптирована под изменения, сделанные в реализации sc-памяти. Сама по себе МОЗ на текущий момент представляет собой набор библиотек расширений, которые загружаются при старте sc-памяти и регистрируют в ней обработчики на события (sc-агенты).

Компоненты ядра пользовательского интерфейса реализованы на клиентской части приложения с использованием JavaScript. По сути, они отвечают за весь базовый функционал клиентской части приложения и обеспечивают корректную работу компонентов. Вкратце опишем функции, которые выполняют компоненты ядра ПИ:

- отображение и инициирование команд ПИ, которые располагаются в главном меню [Корончик2, 2013];

- логика работы с аргументами команд. В качестве аргумента команды может быть указан любой элемент на странице, у которого установлен атрибут *sc_addr* (адрес sc-элемента в памяти);

- навигация по истории диалога пользователя с системой;

- переключение между режимами идентификации (русский, английский). Ядро автоматически запрашивает идентификаторы на необходимом языке и меняет их для всех элементов на странице с атрибутом *sc_addr*. Если же идентификатор для указанного языка не найден, то устанавливается системный идентификатор sc-элемента;

- обеспечивает взаимодействие между компонентами пользовательского интерфейса. Именно ядро принимает решение, какой компонент использовать для отображения информации в том или ином виде. Пользователь лишь указывает внешний язык, на котором он хочет увидеть ответ на свой запрос. Взаимодействие между компонентом и ядром ПИ осуществляется через песочницу - component sandbox. Компонент в своей работе может работать лишь со своей песочницей, не используя функции ядра напрямую. Это позволяет значительно сократить временные затраты на поддержку большого числа

компонентов, когда происходят изменения в ядре ПИ;

- обеспечивает поиск по идентификатору. При введении пользователем 3-х или более символов в строку поиска, ядро делает запрос на сервер, где происходит поиск всех идентификаторов, которые содержат указанную подстроку символов. На сервере имеется sc-агент, который реагирует на появление новых идентификаторов в системе, и добавляет их в базу ключ значений redis (в которой их ищет сервер по запросу клиента);
- отображение всплывающих подсказок. Для каждого элемента на странице с атрибутом `sc_addr`, при наведении, формируется всплывающая подсказка. Эта подсказка описана в базе знаний для разных языков (рисунок 2)

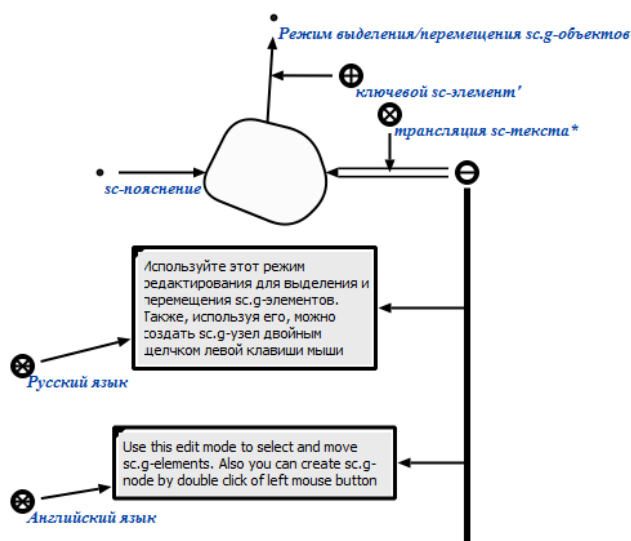


Рисунок 2 – пример указания пояснений для элемента управления в редакторе sc.g-текстов

Компоненты пользовательского интерфейса реализуются также с использованием JavaScript. Раньше мы рассматривали лишь один вид компонентов – компоненты, которые позволяют просматривать (редактировать) содержимое sc-ссылок (файлы). Но для того чтобы изменения в базе знаний отображались у пользователей в режиме реального времени этого было недостаточно. Чтобы решить данную проблему были добавлены компоненты, которые позволяют просматривать (редактировать) структуры напрямую из базы знаний.

При регистрации в ядре, компоненту ставится в соответствие некоторое его описание, где указано, может ли он отображать (редактировать) содержимое sc-ссылок и/или структур. В JavaScript коде это выглядит следующим образом:

```
SCgComponent = {
  ext_lang: 'scg_code',
  formats: ['format_scg_json'],
  struct_support: true,
  factory: function(sandbox) {
    return new scgWindow(sandbox);
  }
};
```

Рассмотрим поля объекта *SCgComponent* подробнее:

- *ext_lang* – в этом поле указывается системный идентификатор sc-узла, который обозначает внешний язык с которым работает компонент;
- *fomats* – в этом поле указывается список системных идентификаторов sc-узлов, которые обозначают форматы файлов (sc-ссылок) поддерживаемые компонентом. Если данное поле пусто, то компонент не может отображать (редактировать) sc-ссылки;
- *sturct_support* – наличие этого поля со значением true, указывает на то, что данный компонент умеет отображать (редактировать) структуры напрямую из памяти;
- *factory* – это функция, которая вызывается, когда ядру ПИ необходимо создать экземпляр данного компонента. В качестве параметра передается экземпляр объекта `component sandbox`, через который будет осуществляться взаимодействие между компонентом и ядром. Возвращает функция экземпляр созданного компонента.

В момент создания экземпляра компонента доступна следующая информация:

- *sandbox.container* – это строка, которая содержит id элемента на странице, куда должен быть встроен компонент;
- *sandbox.canEdit()* – функция, которая возвращает true, если требуется создать редактор, а не просмотрщик;
- *sandbox.is_struct* – поле, которое равно true, если необходимо просматривать (редактировать) структуры, а не sc-ссылки;
- *sandbox.addr* – это адрес структуры или sc-ссылки с которой работает компонент.

В момент создания компонент должен подписаться на необходимые ему события (установить в соответствующие поля `sandbox` функции обработчики):

- *sandbox.eventDataAppend* – это событие, которое вызывается, когда необходимо отобразить содержимое sc-ссылки. В качестве аргумента функция обработчика принимает данные (содержимое sc-ссылки);
- *sandbox.getObjectsToTranslate* – данное событие вызывается ядром, когда происходит переключение между режимами идентификации. Обработчик данного события – это функция, которая возвращает список всех sc-адресов отображаемых (редактируемых) элементов;
- *sandbox.eventApplyTranslation* – это событие, которое происходит, когда необходимые идентификаторы для нового режима идентификации найдены. Обработчик данного события принимает в качестве параметра объект, полями которого являются адреса sc-элементов, а значениями – новые идентификаторы. Если же для

какого-то элемента нет поля, то идентификатор для него не найден;

- `sandbox.eventStructUpdate` – это событие инициируется, когда изменяется отображаемая структура (появляются или удаляются выходящие дуги из sc-узла обозначающего эту структуру).

Таким образом, если нам необходимо реализовать компонент, который отображает содержимое sc-ссылок, необходимо:

- описать формат данных в базе знаний;
- зарегистрировать компонент для отображения нового формата;
- реализовать функцию `sandbox.eventDataAppend`, которая на вход примет данные из sc-ссылки. Эти данные уже необходимо обрабатывать в компоненте и как-то визуализировать.

Заключение

В настоящее время исходные коды серверной и клиентской частей можно найти по ссылке [sc-web, 2014] там же во вкладке “Issues” находится список текущих задач. Инструкцию по установке можно найти там же на wiki (страница “version dev”), данная инструкция постоянно обновляется. Исходные коды реализации sc-памяти можно найти по ссылке [sc-memory, 2014]. Текущие задачи по ней находятся также во вкладке “Issues”.

Хотелось бы отметить, что еще не все задачи решены, но уже на текущем этапе можно делать системы, управляемые знаниями, с возможностью коллективной разработки базы знаний и градацией разработчиков и пользователей по уровням доступа и не только через web-интерфейсы. При этом одну и ту же информацию пользователи смогут редактировать одновременно с использованием различных внешних представлений, которые синхронизируются в режиме реального времени (альтернативы этому найти в интернете не удалось). Сейчас эти наработки используются в метасистеме IMS OSTIS [ims ostis, 214].

В ближайшее время планируются следующие доработки в платформе:

- улучшение sctp протокола за счет добавления более сложных запросов (создать целую конструкцию по шаблону, создать N узлов и т. д.), что позволит значительно ускорить взаимодействие клиентской части с sc-памятью;
- создание полноценной библиотеки компонентов ПИ с возможностью установки в новую систему по сети;
- разработка распределенной версии sc-памяти.

Библиографический список

- [Голенков и др, 2001] Представление и обработка знаний в графодинамических ассоциативных машинах /В. В. Голенков, [и др]; – Мн. : БГУИР, 2001
- [Голенков, 2013] Открытый проект, направленный на создание технологии компонентного проектирования

интеллектуальных систем / В. В. Голенков, Н. А. Гулякина // Материалы международной научно-технической конференции «Открытые семантические технологии проектирования интеллектуальных систем» - Минск, 2013

[Корончик, 2013] Реализация хранилища унифицированных семантических сетей / Д. Н. Корончик // Материалы международной научно-технической конференции «Открытые семантические технологии проектирования интеллектуальных систем» - Минск, 2013

[Корончик, 2014] Пользовательский интерфейс интеллектуальной метасистемы поддержки проектирования интеллектуальных систем / Д. Н. Корончик // Материалы международной научно-технической конференции «Открытые семантические технологии проектирования интеллектуальных систем» - Минск, 2014

[Корончик2, 2013] Унифицированные семантические модели пользовательских интерфейсов интеллектуальных систем и технология их проектирования / Д. Н. Корончик // Материалы международной научно-технической конференции «Открытые семантические технологии проектирования интеллектуальных систем» - Минск, 2013

[ims ostis, 2014] Сайт метасистемы IMS OSTIS [Электронный ресурс]. – 2014 – Режим доступа: <http://ims.ostis.net/> – Дата доступа: 02.11.2014

[sctp, 2014] Протокол sctp [Электронный ресурс]. – 2014 – Режим доступа: <https://github.com/deniskoronchik/sc-machine/wiki/sctp> – Дата доступа: 02.11.2014

[sc-memory, 2014] Исходные коды реализации sc-памяти [Электронный ресурс]. – 2014 – Режим доступа: <https://github.com/deniskoronchik/sc-machine> – Дата доступа: 05.11.2014

[sc-web, 2014] Исходные коды реализации web-платформы [Электронный ресурс]. – 2014 – Режим доступа: <https://github.com/deniskoronchik/sc-web> – Дата доступа: 05.11.2014

[sc-web api, 2014] Rest api платформы [Электронный ресурс]. – 2014 – Режим доступа: <https://github.com/deniskoronchik/sc-web/wiki/Api> – Дата доступа: 03.11.2014

[tornado, 2014] Tornado Web Server [Электронный ресурс]. – 2014 – Режим доступа: <http://www.tornadoweb.org/en/stable/> – Дата доступа: 05.11.2014

[websocket, 2014] WebSocket [Электронный ресурс]. – 2014 – Режим доступа: <https://www.websocket.org/> – Дата доступа: 05.11.2014

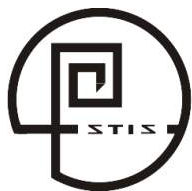
IMPLEMENTATION OF WEB-PLATFORM FOR SYSTEMS BASED ON KNOWLEDGES

Koronchik D. N

**Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus*

denis.koronchik@gmail.com

This article describes technical aspects web-platform for systems based on knowledges. Described platform allows developers to edit knowledge base of system by using different languages and instruments with real-time synchronization. Another words if one developer describes solar system in any formal language, and another developer draw it in special tool, they will have synchronized their views of data by system.



OSTIS-2015

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822:514

МЕТОДИКА КОМПОНЕНТНОГО ПРОЕКТИРОВАНИЯ СИСТЕМ, УПРАВЛЯЕМЫХ ЗНАНИЯМИ

Шункевич Д.В., Давыденко И.Т., Корончик Д.Н., Губаревич А.В., Борискин А.С.

*Белорусский государственный университет информатики и радиоэлектроники,
г. Минск, Республика Беларусь*

shunkevichdv@gmail.com

ir.davydenko@gmail.com

deniskoronchik@gmail.com

stasia@tut.by

coloss_000@mail.ru

В данной работе рассматривается методика проектирования систем, управляемых знаниями, по технологии OSTIS. Указанная технология является открытой технологией, ориентированной на разработку систем, управляемых знаниями, самого различного назначения. В работе подробно рассматривается процесс разработки систем на примере справочной системы по геометрии, описаны процессы расширения базы знаний, машины обработки знаний и пользовательского интерфейса системы.

Ключевые слова: системы, управляемых знаниями; семантические сети; базы знаний; интеллектуальный решатель задач.

Введение

В настоящее время особенно актуальной становится проблема отсутствия хорошо продуманной методики конструктивного использования опыта завершённых разработок компьютерных систем, что порождает высокую степень дублирования разработок различных компонентов этих систем. [Борисов 2013] Многократная повторная разработка уже имеющихся технических решений обусловлена либо тем, что известные технические решения плохо интегрируются в разрабатываемую систему, либо тем, что эти технические решения трудно найти. [Голенков 2012] Данная проблема актуальна как в целом в сфере разработки компьютерных систем, так и в сфере разработки систем, основанных на знаниях, поскольку в системах такого рода степень согласованности различных видов знаний влияет на возможность системы решать нетривиальные задачи. Одной из основных задач Технологии OSTIS [IMS] является решение проблемы совместимости различных компонентов и разработка средств компонентного проектирования систем, управляемых знаниями.

Для решения проблемы совместимости различных видов знаний предложены

унифицированные семантические сети с теоретико-множественной интерпретацией.

При проектировании систем управляемых знаниями применяется методика компонентного проектирования, которая основывается на постоянно расширяемых библиотеках многократно используемых компонентов (типовых технических решений). В качестве примера использования такой методики в данной работе мы рассмотрим подробно процесс проектирования справочной системы по геометрии.

1. Подсистема IMS для поддержки компонентного проектирования дочерних sc-систем

В рамках интеллектуальной метасистемы IMS, ориентированной на консультационное обслуживание и поддержку разработчиков систем по Технологии OSTIS, а также накопление библиотек многократно используемых компонентов OSTIS выделяется подсистема для поддержки проектирования различных компонентов и различных классов интеллектуальных систем. Основными ее задачами являются:

- Обеспечение возможности поиска нужного компонента или набора компонентов в *Библиотеке*

многократно используемых компонентов Технологии OSTIS;

- Формирование, при необходимости, *sc-структур*, соответствующих выбранным многократно используемым компонентам;
- Обеспечение возможности транспортировки выбранных многократно используемых компонентов в *дочернюю sc-систему*;
- В случае развертывания стартовой версии *дочерней sc-системы* – обеспечение возможности установки первой базовой версии *sc-системы*, которая впоследствии может наращиваться дополнительными компонентами.

2. Этапы проектирования дочерних sc-систем

При создании стартовой версии *дочерней sc-системы* по *Технологии OSTIS* можно выделить четыре основных этапа:

- Выбор и установка платформы реализации *дочерней sc-системы*;
- Установка *Ядра sc-моделей баз знаний*, то есть набора базовых *многократно используемых компонентов sc-моделей баз знаний*, необходимых для работы даже первого прототипа *sc-системы*;
- Установка *Ядра sc-машин*, то есть набора базовых *многократно используемых компонентов sc-машин*, необходимых для работы даже первого прототипа *sc-системы*;
- Установка *Ядра sc-моделей интерфейсов*, то есть набора базовых *многократно используемых компонентов sc-моделей интерфейсов*, необходимых для работы даже первого прототипа *sc-системы*;
- Установка ядра подсистемы поддержки проектирования дочерней системы в составе проектируемой дочерней *sc-системы*.

После того, как собрана самая базовая конфигурация *дочерней sc-системы*, можно начинать постепенное ее наполнение либо с использованием *Библиотеки многократно используемых компонентов Технологии OSTIS*, либо путем создания своих собственных компонентов, которые, в свою очередь, могут в дальнейшем быть включены в состав указанной библиотеки.

3. Выбор и установка платформы реализации дочерней sc-системы

3.1. Выбор и установка sc-хранилища

В качестве платформы для разработки справочной системы по геометрии в данной работе выбрана *Web-ориентированная платформа реализации sc-систем*, основанная на специальном формате кодирования *sc-текстов*, входящая в состав *Библиотеки платформ реализации sc-систем [IMS]*. Согласно документации по данному платформенно-зависимому многократно используемому компоненту *OSTIS [sc-machine]*

данный вариант реализации платформы ориентирован на операционные системы семейства Linux и сопровождается скриптами, позволяющими собрать и установить выбранную платформу на нужном сервере [Структура *sc-машины*].

Данный вариант реализации платформы включает в себя собственно *sc-хранилище*, а также набор трансляторов.

3.2. Установка *scp-машины*

После того, как на сервер была установлена выбранная реализация *sc-памяти*, разработчик может установить также совместимую с выбранной реализацией платформы реализацию абстрактной *scp-машины*. На семантическом уровне совместимость задается при помощи отношения совместимый компонент*. Таким образом, для выбранной реализации *sc-памяти* подойдет реализация абстрактной *scp-машины* на базе *Web-ориентированной платформы реализации sc-систем*. Исходные коды и инструкцию по установке данного платформенно-зависимого многократно используемого компонента *OSTIS* согласно его описанию в Библиотеке платформ реализации *sc-моделей* можно найти по адресу [sc-machine]. В данном варианте компонента для реализации *sc-агентов* используются языки C и C++.

Для корректной работы какого-либо варианта реализации абстрактной *scp-машины* необходимо наличие в системе соответствующего многократно используемого компонента *sc-моделей баз знаний*, описывающего Предметную область программ базового языка программирования, ориентированного на обработку *sc-моделей баз знаний*.

После того, как были установлены необходимые части платформы реализации *sc-моделей*, можно приступать к установке необходимых платформенно-независимых многократно используемых компонентов *OSTIS*.

4. Установка ядра базы знаний дочерней sc-системы

Ядро sc-моделей баз знаний рассматривается как один *неатомарный многократно используемый компонент OSTIS* и содержит в себе все фрагменты базы знаний, необходимые для корректной работы платформы реализации *sc-моделей интеллектуальных систем*, а также базовых *sc-агентов*, использование которых целесообразно в любой *дочерней sc-системе*.

Для понятности кратко перечислим фрагменты базы знаний, входящие в *Ядро sc-моделей баз знаний*:

- Раздел. Предметная область семантических окрестностей;
- Раздел. Предметная область ситуативно-событийных моделей изменения состояний баз знаний;

- Раздел. Предметная область моделей действий, направленных на изменение состояния баз знаний;
- Раздел. Предметная область чисел;
- Раздел. Предметная область сообщений, принимаемых и передаваемых интеллектуальной системой;
- Раздел. SC-код
 - Раздел. Предметная область sc-элементов
 - Раздел. Предметная область sc-множеств
 - Раздел. Предметная область sc-отношений
 - Раздел. Предметная область библиографических источников
 - Раздел. Предметная область sc-структур
 - Раздел. Предметная область sc-ссылок
 - Раздел. Предметная область внешних sc-идентификаторов
 - Раздел. Предметная область физических лиц и коллективов
- Раздел. SCs-код
- Раздел. SCn-код
- Раздел. SCg-код
- Базовый набор команд информационного поиска
 - Команды запроса структуры
 - Запрос всех надклассов
 - Запрос декомпозиции
 - Запрос всех подклассов
 - Команды базовых запросов
 - Запрос позитивных, константных выходящих дуг с отношениями
 - Запрос позитивных, константных входящих дуг с отношениями
 - Запрос позитивных, константных выходящих дуг
 - Запрос позитивных, константных входящих дуг
 - Команды запроса семантических окрестностей
 - Запрос полной семантической окрестности
 - Команды запроса идентификаторов
 - Запрос всех идентификаторов

5. Установка ядра машины обработки знаний дочерней sc-системы

Ядро машины обработки знаний *sc-систем* (*sc-машин*) включает в себя минимальный набор предметно независимых *sc-агентов* информационного поиска, необходимый для навигации по базе знаний дочерней *sc-системы*.

В текущем варианте указанное ядро включает в себя:

- *sc-агент* поиска всех выходящих константных позитивных стационарных *sc-дуг* принадлежности;
- *sc-агент* поиска всех выходящих константных позитивных стационарных *sc-дуг* принадлежности с их ролевыми отношениями;
- *sc-агент* поиска всех входящих константных позитивных стационарных *sc-дуг* принадлежности;
- *sc-агент* поиска всех выходящих константных позитивных стационарных *sc-дуг*

- принадлежности с их ролевыми отношениями;
- *sc-агент* поиска связей декомпозиции для заданного *sc-элемента*;
- *sc-агент* поиска всех известных сущностей, являющихся частными по отношению к заданной;
- *sc-агент* поиска всех известных сущностей, являющихся общими по отношению к заданной;
- *sc-агент* поиска всех идентификаторов заданного *sc-элемента*;
- *sc-агент* поиска полной семантической окрестности заданного элемента;
- *sc-агент* поиска связей заданного отношения, компонентом которых является заданный *sc-элемент*;
- *sc-агент* поиска всех конструкций, изоморфных заданному образцу;

Данный набор *sc-агентов* может быть реализован как на платформенно-независимом уровне, так и на уровне платформы. В нашем случае реализация всех перечисленных *sc-агентов* входит в состав выбранной платформы реализации *sc-моделей*, что, с одной стороны, упрощает их установку в дочернюю *sc-систему*, но с другой сильно осложняет процесс внесения изменений в состав ядра.

Чтобы обеспечить возможность задания более сложных поисковых запросов в систему необходимо добавить дополнительные *sc-агенты* из Библиотеки многократно используемых *sc-агентов* *OSTIS*, о чем будет сказано ниже.

Для корректной работы данных *sc-агентов* дополнительно необходимо наличие в базе знаний определенного набора сопутствующих компонентов*.

6. Установка ядра пользовательского интерфейса дочерней sc-системы

Ядро *sc-моделей* пользовательских интерфейсов, как и собственно платформа реализации *sc-моделей*, представляет собой платформенно-зависимый многократно используемый компонент *OSTIS*, который должен быть совместим с выбранным вариантом платформы. Подходящий вариант реализации ядра пользовательских интерфейсов, описанный в Библиотеке многократно используемых компонентов *sc-моделей* интерфейсов может быть найден по адресу [sc-web]. Согласно описанию, выбранный компонент содержит в себе, помимо собственно ядра, следующий набор компонентов:

- Визуализатор *sc-конструкций* в *SCg-коде*;
- Редактор *sc-конструкций* в *SCg-коде*;
- Визуализатор *sc-конструкций* в *SCn-коде*;
- Набор компонентов для отображения *sc-ссылок* различных форматов (PNG, GIF, PDF, HTML, TXT и др.).

Также в выбранный ранее вариант платформы реализации *sc-моделей* по умолчанию входят

трансляторы из *sc-памяти* в формат, необходимый для SCg и SCn визуализации, а также транслятор из SCg-редактора в *sc-память*. Таким образом, установленных средств достаточно для базовой визуализации и редактирования *sc-конструкций* посредством web-интерфейса.

7. Установка ядра подсистемы поддержки проектирования дочерней sc-системы

Подсистема поддержки проектирования дочерней *sc-системы* в свою очередь состоит из

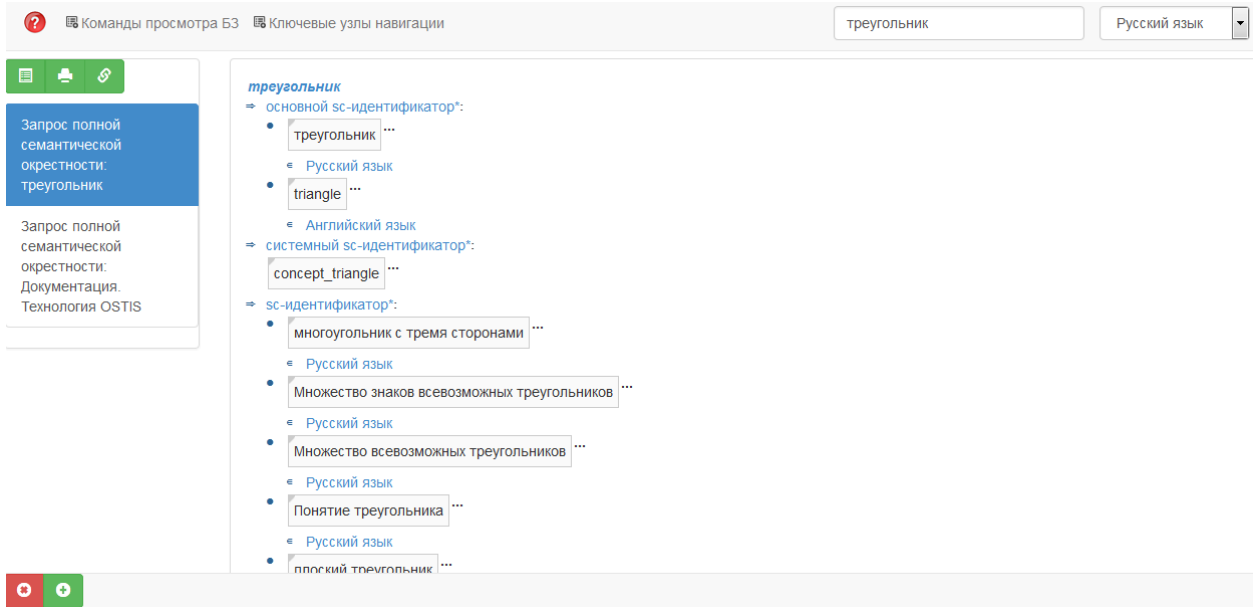


Рисунок 1 – Отображение фрагмента базы знаний в SCn-коде

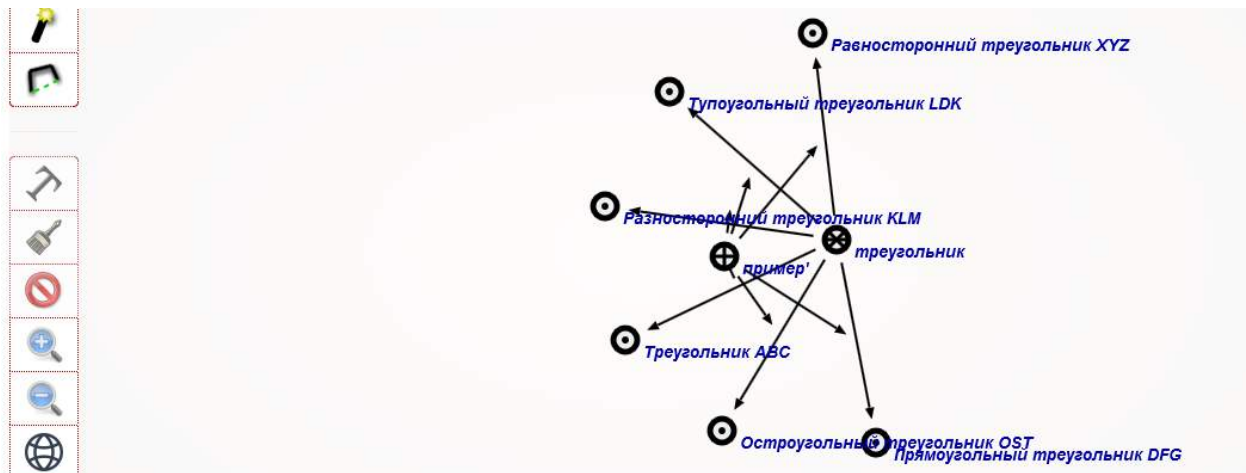


Рисунок 2 – Отображение фрагмента базы знаний в SCg-коде

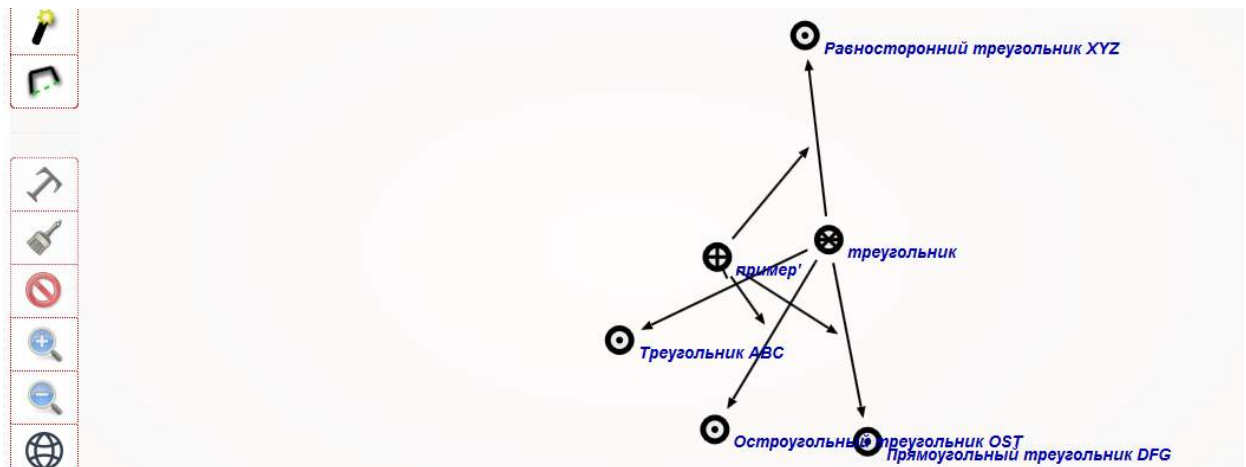


Рисунок 3 – Результат редактирования фрагмента базы знаний в SCg-коде

двух взаимосвязанных подсистем, одна из которых входит в состав самой *IMS*, а другая является многократно используемой типовой подсистемой интеллектуальных систем, и входит в состав минимального набора компонентов, необходимых для полноценного функционирования дочерней *sc-системы* в рамках *Глобальной базы знаний*. [Шункевич, 2015]

Ядро данной типовой подсистемы включает в себя необходимые средства интеграции многократно используемых компонентов *OSTIS* в дочернюю *sc-систему* и средства управления проектированием дочерней *sc-системы*.

8. Расширение базы знаний дочерней *sc-системы*

Помимо базовых многократно используемых компонентов баз знаний, входящих в состав ядра базы знаний, каждая дочерняя система может быть дополнена другими компонентами из *Библиотеки многократно используемых компонентов sc-моделей баз знаний*. Общие принципы проектирования баз знаний по *Технологии OSTIS* рассмотрены в [Давыденко, 2013]

8.1. Построение системы предметных областей и их онтологий, входящих в состав базы знаний проектируемой *sc-системы*

Первым и важнейшим этапом проектирования базы знаний является уточнение структуры описываемой предметной области или нескольких взаимосвязанных предметных областей. Уточнение такой структуры – это, прежде всего, уточнение класса исследуемых объектов, уточнение предмета исследования, уточнение всего семейства ключевых узлов семантической сети, представляющей предметную область. В рамках предметной области возможно выделение частных предметных областей на основе выделения подмножества из семейства классов исследуемых объектов.

При описании структуры предметной области используются ключевые узлы, входящие в состав *Предметной области предметных областей*, входящей в *Библиотеку многократно используемых компонентов баз знаний* в виде компонента, следовательно, данный компонент необходимо добавить в состав базы знаний системы по геометрии.

Пример структуры *Предметной области Геометрии Евклида*:

Предметная область Геометрии Евклида

∈ предметная область

=> частная предметная область*:

- *Предметная область геометрических точек*
- *Предметная область линий*
- *Предметная область планарных фигур*
=> частная предметная область*:
 - *Предметная область*

прямолинейных фигур

- *Предметная область планарных углов*
- *Предметная область многоугольников*
=> частная предметная область*:
 - *Предметная область треугольников*
 - *Предметная область четырехугольников*
- *Предметная область вписанных планарных фигур*
- *Предметная область кругов и окружностей*
- *Предметная область геометрических поверхностей*
- *Предметная область геометрических тел*
=> частная предметная область*:
 - *Предметная область многогранников и их поверхностей*
 - *Предметная область непланарных углов*
 - *Предметная область тел вращения и их поверхностей*
- *Предметная область конгруэнтности геометрических фигур*

Рассмотрение базы знаний с позиции ее соотношения с предметной областью позволяет рассматривать исследуемые объекты на различных уровнях детализации, которые отражаются в различных типах онтологий, описывающих определенное направление описания свойств объекта в рамках рассматриваемой предметной области. К таким типам таких онтологий относятся:

- *структурная спецификация предметной области* – описание связей рассматриваемой предметной области с другими предметными областями и ролей всех понятий, входящих в состав данной предметной области;
- *терминологическая онтология* – описание терминов и их синонимов ключевых понятий рассматриваемой предметной области, близких между собой терминов, этимологии терминов и правила построения идентификаторов экземпляров понятий;
- *теоретико-множественная онтология* – описание теоретико-множественных связей между понятиями рассматриваемой предметной области;
- *логическая онтология* – описание всех высказываний рассматриваемой предметной области;
- *логическая система понятий и их определений* – это структура, являющаяся надстройкой над логической онтологией, включающая описание системы определений понятий заданной предметной области с указанием набора понятий, через которые определяется каждое определяемое понятие рассматриваемой предметной области;
- *логическая система утверждений и их доказательств* – это структура, являющаяся

надстройкой над логической онтологией, включающая описание системы утверждений рассматриваемой предметной области с указанием набора утверждений, через которые доказывается каждое утверждение;

- *онтология задач и решений задач* – описание конкретных задач, рассматриваемых в заданной предметной области, и их решений;

- *онтология классов задач и способов решения задач* – описание классов задач, рассматриваемых в заданной предметной области, и способов их решений. Данная структура является надстройкой над *онтологией задач и решений задач*.

Для описания указанной структуры предметной области используются ключевые *sc-элементы*, входящие в состав *многократно используемого компонента sc-моделей баз знаний Предметная область онтологий*.

Таким образом, для всех разделов базы знаний, описывающих предметную область и ее спецификацию, имеет смысл задать типовую структуру. В качестве примера рассмотрим *Предметную область четырехугольников и ее онтологии*.

Раздел. Предметная область четырехугольников

∈ предметная область и ее онтологии

≤ декомпозиция раздела:*

- {
- *Предметная область четырехугольников*
- *Структурная спецификация предметной области четырехугольников*
- *Терминологическая онтология предметной области четырехугольников*
- *Теоретико-множественная онтология предметной области четырехугольников*
- *Логическая онтология предметной области четырехугольников*
- *Логическая иерархия понятий предметной области четырехугольников*
- *Логическая иерархия высказываний о предметной области четырехугольников*
- *Онтология задач и решений задач предметной области четырехугольников*
- *Онтология классов задач и способов решения задач предметной области четырехугольников*
- }

Далее рассмотрим подробнее фрагменты указанных выше разделов спецификации предметной области.

8.2. Разработка фрагментов используемых предметных областей

Любая предметная область также может содержать примеры конкретных объектов исследования, т.е. экземпляры классов исследования.

Ниже приведен пример отображения в системе семантической окрестности объекта Параллелограмм ABCD.



Рисунок 4 – Семантическая окрестность Параллелограмма ABCD

8.3. Разработка структурных спецификаций для каждой используемой предметной области

Рассмотрим фрагмент *Структурной спецификации предметной области четырехугольников*:

Предметная область четырехугольников

⇒ sc-онтология:*

Структурная спецификация предметной области четырехугольников

=[

Предметная область четырехугольников

≤ частная предметная область:*

Предметная область многоугольников

≡ максимальный класс объектов исследования': четырехугольник

≡ не максимальный класс объектов исследования':

- *параллелограмм*
- *прямоугольник*
- *ромб*
- *квадрат*
- *трапеция*
- *равнобедренная трапеция*
- *прямоугольная трапеция*

≡ объект исследования':

- *Квадрат ABCD*

- *Четырехугольник KLMN*
 - *Трапеция BMNO*
- ⇒ исследуемое отношение':
- *внутренний угол**
 - *сторона**
 - *высота**
 - *площадь**
 - *периметр**
 - *средняя линия**
-]

Наличие указанного фрагмента в базе знаний позволяет задавать следующие вопросы sc-системе:

- Что является объектом исследования данной предметной области?
- Какие отношения исследуются в данной предметной области?

8.4. Разработка терминологических онтологий для каждой используемой предметной области

Рассмотрим фрагмент *Терминологической онтологии предметной области четырехугольников*:

Предметная область четырехугольников

⇒ sc-онтология*:

Терминологическая онтология предметной области четырехугольников

= [

четырёхугольник

= многоугольник с четырьмя сторонами

= многоугольник с четырьмя углами 0

<= правила идентификации элементов*:

{

- [Четырёхугольник < суффикс >]
- [Чет-к < суффикс >]
- [Четырёхугольник (<идентификатор точки>; <идентификатор точки>; <идентификатор точки>; <идентификатор точки>)]

}

прямоугольник

= четырёхугольник с прямыми внутренними углами

<= правила идентификации элементов*:

{

- [Прямоугольник < суффикс >]
- [Прям-к < суффикс >]
- [Прямоугольник (<идентификатор точки>; <идентификатор точки>; <идентификатор точки>; <идентификатор точки>)]

}

ромб

= четырёхугольник с равными сторонами

= четырёхугольник с конгруэнтными сторонами

<= правила идентификации элементов*:

{

- [Ромб < суффикс >]
- [Ромб (<идентификатор точки>; <идентификатор точки>; <идентификатор точки>; <идентификатор точки>)]

}

квадрат

= ромб с прямыми углами

= прямоугольник с равными сторонами

= прямоугольник ∩ ромб

<= правила идентификации элементов*:

{

- [Квадрат < суффикс >]
- [Квадрат (<идентификатор точки>; <идентификатор точки>; <идентификатор точки>; <идентификатор точки>)]

}

трапеция

<= правила идентификации элементов*:

{

- [Трапеция < суффикс >]
- [Трапеция (<идентификатор точки>; <идентификатор точки>; <идентификатор точки>; <идентификатор точки>)]

}

равнобедренная трапеция

= трапеция с равными сторонами

прямоугольная трапеция

= трапеция с прямыми углами при боковой стороне

]

Наличие указанного фрагмента в базе знаний позволяет задавать следующие вопросы sc-системе:

- Какие синонимичные термины известны для заданного термина?
- По какому правилу идентифицируются элементы заданного множества?

8.5. Разработка теоретико-множественных онтологий для каждой используемой предметной области

Рассмотрим Фрагмент *Теоретико-множественной онтологии предметной области четырехугольников*:

Предметная область четырехугольников

⇒ sc-онтология*:

Теоретико-множественная онтология предметной области четырехугольников

= [

четырёхугольник

⊃ трапеция

трапеция

⊃ параллелограмм

⊃ равнобедренная трапеция

▷ *прямоугольная трапеция*

параллелограмм

▷ *прямоугольник*

▷ *ромб*

квадрат

⊂ *прямоугольник*

⊂ *ромб*

средняя линия*

∈ *бинарное отношение*

=> *область определения**:

трапеция ∪ *отрезок*

=> *первый домен**:

трапеция

=> *второй домен**:

отрезок

]

Наличие указанного фрагмента в базе знаний позволяет задавать следующие вопросы *sc*-системе:

- Как классифицируется заданное понятие?
- Какие надклассы известны для заданного понятия?
- Являются ли два заданных понятия пересекающимися?
- Какова область определения заданного отношения?

8.6. Разработка логических онтологий для каждой используемой предметной области

Рассмотрим фрагмент *Логической онтологии предметной области четырехугольников*

Предметная область четырехугольников

=> *sc-онтология**:

Логическая онтология предметной области четырехугольников

=[

четырёхугольник

∈ *ключевой sc-элемент**:

Опр. (четырёхугольник)

∈ *sc-определение*

<= *трансляция sc-текста**:

...

∃ *пример**:

[**четырёхугольник** – это многоугольник с четырьмя сторонами.]

∈ *Русский язык*

...

∈ *sc-утверждение*

<= *трансляция sc-текста**:

...

∃ *пример**:

[Сумма мер внутренних углов четырёхугольника равна 360 угловых градусов.]

∈ *Русский язык*

...

∈ *sc-утверждение*

<= *трансляция sc-текста**:

...

∃ *пример**:

[Сумма квадратов длин диагоналей четырёхугольника равна удвоенной сумме длин квадратов отрезков, соединяющих середины его противоположных сторон.]

∈ *Русский язык*

...

∈ *sc-утверждение*

<= *трансляция sc-текста**:

...

∃ *пример**:

[Если около четырёхугольника можно описать окружность, то сумма мер его любых противоположных углов равна 180 угловых градусов.]

∈ *Русский язык*

]

Наличие указанного фрагмента в базе знаний позволяет задавать следующие вопросы *sc*-системе:

- Какие высказывания известны в рамках заданной предметной области?
- Какие высказывания являются аксиомами в рамках заданной предметной области?
- Как определяется/поясняется то или иное понятие?

8.7. Разработка логической системы понятий и их определений для каждой используемой предметной области

Рассмотрим фрагмент *Логической системы понятий и их определений предметной области четырехугольников*

Логическая онтология предметной области четырехугольников

=> *метазнание**:

Логическая система понятий и их определений предметной области четырехугольников

=[

четырёхугольник

∈ *ключевой sc-элемент**:

...

∈ *sc-определение*

<= *трансляция sc-текста**:

...

∃ *пример**:

[**четырёхугольник** – это многоугольник с четырьмя сторонами.]

∈ *Русский язык*

<= *используемые константы**:

{

• *многоугольник*

• *сторона**

}

трапеция

∈ *ключевой sc-элемент**:

...

```

∈ sc-определение
<= трансляция sc-текста*:
...
∃ пример':
[трапеция – это четырехугольник,
у которого две противоположные
стороны параллельны.]
∈ Русский язык
<= используемые константы*:
{
• четырехугольник
• сторона*
• параллельность*
• противоположий*
}
]

```

Наличие указанного фрагмента в базе знаний позволяет задавать следующие вопросы sc-системе:

- Через какие понятия определяется заданное понятие?
- Какие понятия определяются на основе заданного понятия?

8.8. Разработка логической системы утверждений для каждой используемой предметной области

В разрабатываемой системе по геометрии уместно привести типовые доказательства для некоторых теорем и типовые решения для некоторых примеров задач.

Для обеспечения такой возможности необходимо добавить в систему описание *Предметной области вопросов и задач*, содержащей все необходимые ключевые узлы, то есть *Раздел. Предметная область вопросов и задач*.

Рассмотрим фрагмент *Логической системы утверждений и их доказательств предметной области четырехугольников*

Логическая онтология предметной области четырехугольников

=> метазнание*:

Логическая система высказываний и их доказательств предметной области четырехугольников

=[

параллелограмм

∈ ключевой sc-элемент':

Утв.(параллелограмм; диагональ; четырехугольник)*

∈ sc-утверждение

<= трансляция sc-текста*:

...

∃ пример':

[Если диагонали четырехугольника пересекаются и точкой пересечения делятся пополам, то этот четырехугольник - параллелограмм]

∈ Русский язык
∃ главный ключевой sc-элемент':

...

=> основное доказательство*:

Док-во. Утв.(параллелограмм; диагональ; четырехугольник)*

<= используемые утверждения*:

{

• *Утв.(вертикальные углы*; конгруэнтность*)*

• *Утв.(треугольник; конгруэнтность*; сторона*; внутренний угол*)*

• *Опр.(внутренние накрест лежащие углы*)*

• *Опр.(параллелограмм)*

}

]

Теперь, используя упомянутые ключевые узлы, добавим в систему пример доказательства теоремы о равенстве вертикальных углов:

```

...
= основное доказательство*:
  stat_quadilateral_parallelogram_diagonal_point_intersect_middle
= декомпозиция sc-действия*:
...
  ∃ rrel_1:
    ...
    ∃ ...
    ∃ ...
    ∃ ...
    ∃ ...
    ∃ ...
    ∃ ...
    ∃ ...
    ∃ ...
    ∃ ...
    ∃ ...
  ∈ successfully_completed_action

```

Рисунок 5 – Пример отображения sc-текста общей структуры доказательства, представленного в SCn-коде

```

...
= результат*:
...
= трансляция sc-текста*:
...
  ∃ [Прямая AD параллельна прямой BC. ...]
    ∈ Русский язык
= последовательность действий*:
...
= последовательность действий*:
...
  ∃ rrel_2:
    stat_parallelism_congruence_straight_line_internal_cross_lying_angles
  ∃ rrel_1:
    ...
  ∈ ...
  = декомпозиция sc-действия*:
    ...
  ∈ successfully_completed_action
  ∈ sc-действие применения логического утверждения

```

Рисунок 6 – Пример отображения sc-текста шага доказательства

Внесенное в систему доказательство можно просмотреть при помощи базовых средств навигации, как показано на рисунках выше, однако это не всегда удобно, поэтому имеет смысл использовать специальный *sc-агент поиска доказательства заданного утверждения*, о чем будет сказано ниже.

8.9. Разработка онтологий задач и решений для каждой используемой предметной области

Аналогичным образом можно внести в систему пример решения, например, следующей задачи: «В параллелограмме ABCD $\sin(C) = 5/7$. AD = 7. Найдите высоту, опущенную на сторону АВ».

Ниже приведет фрагмент *Онтологии задач и решений задач предметной области четырехугольников*.

Предметная область четырехугольников

=> *sc-онтология**:

Онтология задач и решений задач предметной области четырехугольников

=[

параллелограмм

∈ *ключевой sc-элемент**:

Задача. Нахождение высоты параллелограмма по длине стороны и синусу внутреннего угла

∈ *sc-задача*

<= *трансляция sc-текста**:

...

∃ *пример**:

[В параллелограмме ABCD $\sin(C) = 5/7$. AD = 7. Найдите высоту, опущенную на сторону АВ]

∈ *Русский язык*

=> *решение**:

Решение. Задача. Нахождение высоты параллелограмма по длине стороны и синусу внутреннего угла

<= *используемые утверждения**:

{

• *Утв.(параллелограмм; противоположащий*; внутренний угол*; конгруэнтность*)*

• *Опр.(синус*)*

}

]

Наличие указанного фрагмента в базе знаний позволяет задавать следующие вопросы *sc-системе*:

- Примеры каких задач известны в рамках заданной предметной области?
- Как решается заданной задача?
- С помощью каких утверждений решается заданная задача?

Формальная запись условия этой задачи на языке SCg представлена на рисунке 8.

Предлагаемый вариант решения данной задачи состоит из следующих шагов:

- $\sin(A) = \sin(C) = \frac{5}{7}$;
- $\sin(A) = \frac{5}{7} = \frac{DH}{AD} = \frac{DH}{7}$. DH = 5.

Решение по шагам в формальном представлено на рисунке 9.

В системе решение можно просмотреть по шагам при помощи базовых *sc-агентов информационного поиска*, однако, как и в случае с доказательствами, это не всегда удобно, поэтому имеет смысл добавить в систему *sc-агент поиска решения задачи*, позволяющий просмотреть все решение сразу.

8.10. Разработка онтологий классов задач и способов решения задач для каждой используемой предметной области

Рассмотрим фрагмент *Онтологии задач и решений задач предметной области четырехугольников*.

Онтология задач и решений задач предметной области четырехугольников

=> *метазнание**:

Онтология классов задач и способов решения задач предметной области четырехугольников

=[

Задача. Нахождение высоты параллелограмма по длине стороны и синусу внутреннего угла

∈ *sc-задача поиска заданной величины*

∈ *sc-задача на параллелограммы*

∈ *sc-задача на четырехугольники*

∈ *sc-задача планиметрии*

]

Наличие указанного фрагмента в базе знаний позволяет задавать следующие вопросы *sc-системе*:

- Какие классы задач известны для заданной предметной области?
- Какие способы решения класса задач известны для заданной предметной области?

Таким образом, база знаний *дочерней sc-системы* может быть пополнена как за счет использования компонентов из *Библиотеки многократно используемых компонентов sc-моделей баз знаний*, так и за счет добавления разработчиком в систему предметных знаний из рассматриваемой области.

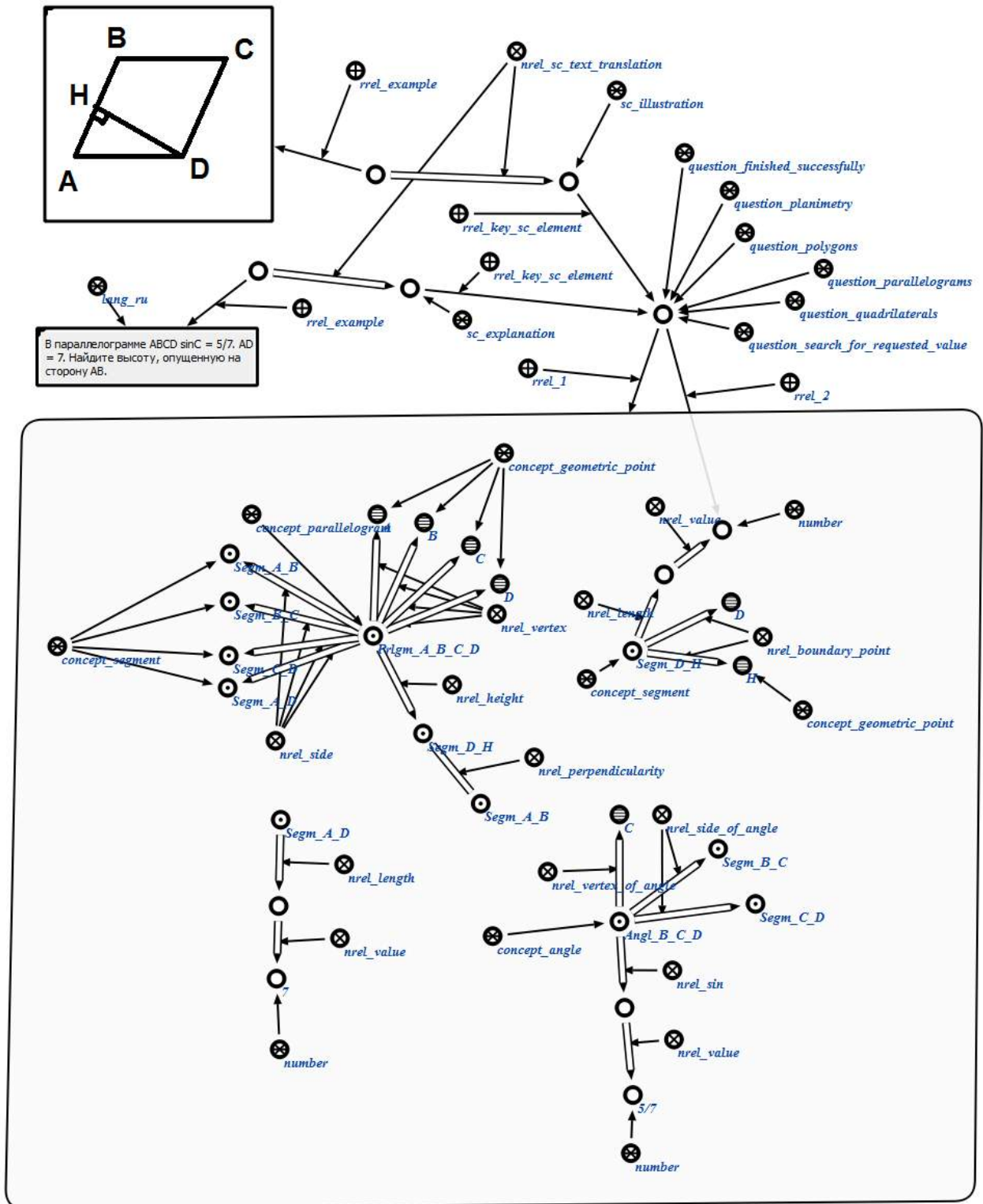


Рисунок 8 – Условие задачи, представленное в SCg-коде

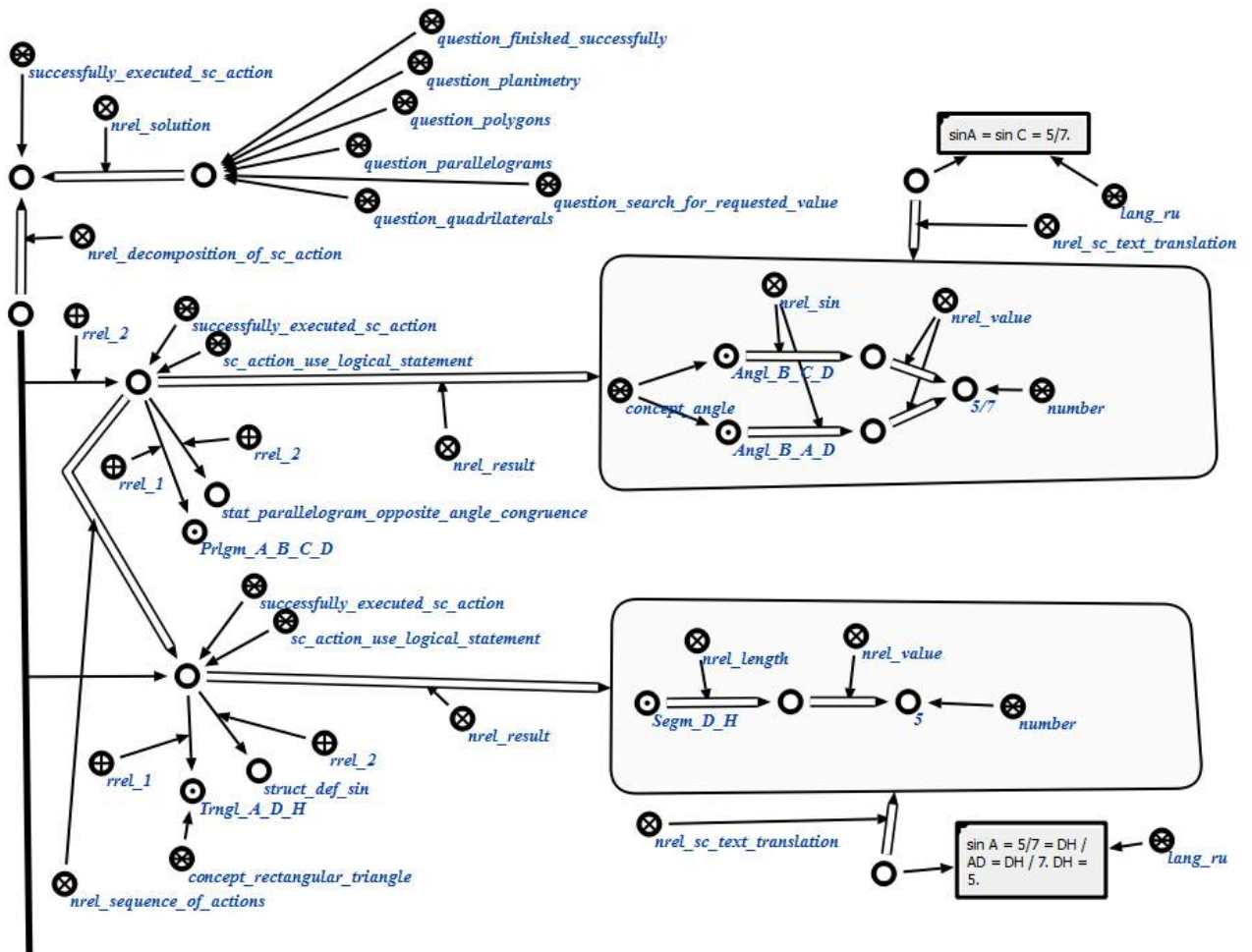


Рисунок 9 – Решение задачи, представленное в SCg-коде

9. Расширение машины обработки знаний дочерней sc-системы

Машина обработки знаний любой системы, управляемой знаниями, определяет ее функциональные возможности и во многом обеспечивает удобство работы с ее контентом. Общие принципы проектирования машин обработки знаний по *Технологии OSTIS* рассмотрены в [Шункевич, 2013].

9.1. Расширение машины обработки знаний с использованием многократно используемых компонентов OSTIS

Как было сказано ранее, в справочной системе по геометрии, например, представлены типовые доказательства некоторых теорем и решения некоторых задач. Для удобства навигации по знаниям такого рода разработаны специальные sc-агенты и описаны соответствующие им компоненты в *Библиотеке многократно используемых компонентов sc-машин*:

- Компонент библиотеки. sc-агент поиска доказательства для заданного утверждения;
- Компонент библиотеки. sc-агент поиска решения заданной задач;

Для указанных компонентов *сопутствующими компонентами** являются:

- Компонент библиотеки. Команда поиска доказательства для заданного утверждения;
- Компонент библиотеки. Команда поиска решения заданной задач;

Также *сопутствующим компонентом** для указанных является *Раздел. Предметная область вопросов и задач*, но данный компонент уже был внедрен в систему ранее.

После того, как в систему были добавлены указанные компоненты, конечный пользователь системы получил возможность увидеть полный *sc-текст доказательства* выбранного утверждения или решения выбранной задачи.

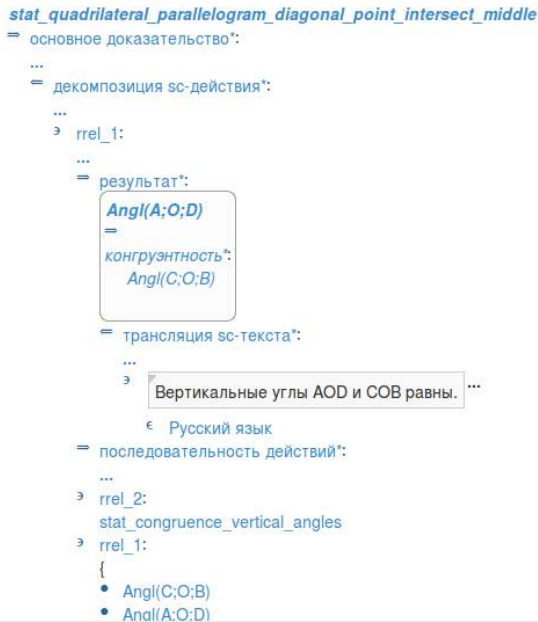


Рисунок 10 – Фрагмент результата работы *sc-агента поиска доказательства*



Рисунок 11 – Фрагмент результата работы *sc-агента поиска решения задачи*

9.2. Расширение машины обработки знаний на основе собственных разработок

В процессе разработки конкретной дочерней системы по *Технологии OSTIS* может оказаться, что необходимый компонент такой системы, хотя и может использоваться в целом ряде систем, но пока еще не реализован другими разработчиками или реализован, но по каким-либо причинам пока еще не вошел в состав *Библиотеки многократно используемых компонентов OSTIS*. В таком случае разработчик *дочерней sc-системы* может самостоятельно разработать нужный компонент и, поскольку *Технология OSTIS* является открытой технологией, попробовать внедрить его в состав соответствующей *библиотеки многократно используемых компонентов*.

Одним из важнейших достоинств интеллектуальной справочной системы по геометрии, является то, что она строит ответы на заданные пользователем вопросы (задачи) не только путем поиска и локализации (выделения) этих ответов в рамках текущего состояния базы знаний, но и путем генерации этих ответов, если их нет в текущем состоянии базы знаний. Интеллектуальная справочная система по геометрии, например, способна решать различные вычислительные задачи и задачи на доказательство. Для этих целей был разработан интеллектуальный решатель задач, который, однако, разрабатывался без привязки к конкретной предметной области и может войти в состав *Библиотеки типовых подсистем интеллектуальных систем*.

В процессе решения конкретной задачи решателем были выделены **6 ключевых этапов**:

- этап задания условия геометрической задачи;
- этап работы поисковых операций
 - а) поиск нахождения значения указанной величины;
 - б) поиск нахождения доказательства для указанного утверждения;
- этап аналогии;
- этап применения стратегий решения задач:
 - а) «итерационный обход»;
 - б) «обход тезаурусом»;
- этап применения правил логического вывода:
 - а) применение имплицативных высказываний;
 - б) применение утверждений с эквивалентными частями;
- этап оптимизации сгенерированных знаний и сборки мусора.

9.2.1. Этап задания условия геометрической задачи

Выбранная нами модель представления знаний в виде семантических сетей предполагает, что все

ключевые понятия базы знаний и связи между ними представлены в виде узлов и дуг, то есть принимают вид связного графа. Решатель опирается на граф условия и параметры запроса. Граф условия – условие задачи, представленное на языке семантических сетей (SC). Параметры запроса – узлы графа условия, к которым обращается пользователь при постановке задачи. Предположим, что нам известны длины одного из катетов и гипотенузы, и необходимо найти длину второго катета. Параметром запроса станет узел, содержащий длину неизвестного катета.

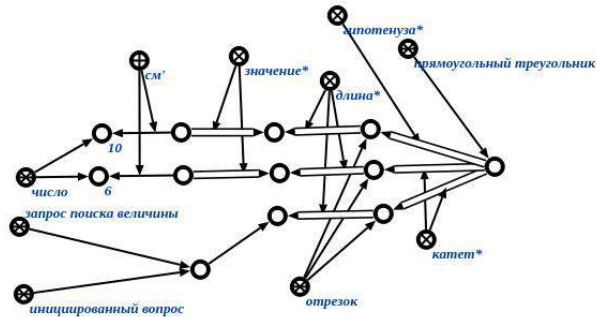


Рисунок 12 - Граф условия

9.2.2. Этап работы поисковых операций

Вне зависимости от типа задачи всегда имеется вероятность того, что данная задача уже была решена системой ранее или системе уже откуда-либо известен ответ на поставленный вопрос. Соответственно, первым шагом к решению поставленной задачи станет использование поисковой операции, соответствующей классу решаемой задачи.

А) Класс задач, ориентированный на поиск значения неизвестной величины, решается путём вывода значения искомой величины с пошаговым выводом действий в виде применённых в ходе решения утверждений и результатов их применения. В случае, если граф условия соответствует одному из шаблонов решённых задач, то применение операции вывода решения для заданной задачи позволит решателю с минимальными затратами времени предоставить пользователю ответ.

Б) Класс задач, ориентированный на поиск доказательства, принимает в качестве параметра запроса не узел неизвестной величины, а *с-структура*, содержащее произвольное высказывание. В ходе решения задачи оно будет либо доказано, вследствие применения вспомогательных логических утверждений, либо опровергнуто, что будет свидетельствовать о его невыполнимости в текущем состоянии базы знаний. Если доказательство высказывания ранее было вынесено в качестве типового, то результат выполнения операции поиска доказательства для заданного утверждения предоставит пользователю ответ в виде декомпозиции шагов доказательства с указанием последовательности применения утверждений.

9.2.3. Этап вывода по аналогии

При решении задачи мы часто сталкиваемся с тем, что одну величину можно выразить через другую. Идеальным случаем является наличие между объектами бинарного неориентированного отношения, типа *конгруэнтность** или *подобие**, позволяющего на основании знаний об одном объекте делать достоверные выводы о другом. Так как основной характеристикой разрабатываемых систем является их способность к осуществлению различного рода логического вывода, то это подразумевает под собой применение системой не только стандартных алгоритмов, но и других логических схем, таких как аналогия, что позволяет сократить процесс поиска решения в разы. Пример применения аналогии для рассмотренной ранее задачи.

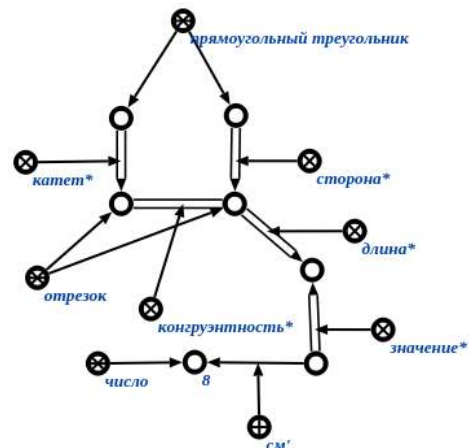


Рисунок 13 — Аналогия

9.2.4. Этап применения стратегий решения задач

На данном этапе осуществляется выбор между различными стратегиями решения задач, и, при необходимости, параллельный запуск различных стратегий. На данный момент интеллектуальный решатель реализует комбинированную стратегию, рассмотренную в работе [Заливако 2012].

Вначале рассматривается некоторый объект, для которого осуществляется поиск всех классов объектов, которым он принадлежит. Далее для каждого класса осуществляется поиск утверждений, справедливых для данного класса объектов. При рассмотрении каждого утверждения осуществляется попытка применить его в рамках некоторой семантической окрестности рассматриваемого объекта.

Если ни одно из утверждений класса применить не удастся, то стратегия продолжает свою работу и переходит к другим объектам. Когда применения одного утверждения оказывается недостаточно для получения ответа, то стратегия запускается заново, с узла запроса, и поиск решения продолжается.

Были разработаны два возможных варианта обхода стратегии:

а) «итерационный обход» - попытавшись применить по разу каждое из утверждений, мы на основании того, был ли выявлен какой-то прогресс в ходе решения, то есть, сгенерированы ли какие-либо новые знания в семантической окрестности задачи, определяем, стоит ли производить ещё один обход.

б) «обход тезаурусом» - по мере единственного обхода мы применяем каждое из утверждений по разу и формируем словарь: в него войдут все утверждения, отсечённые на этапе оптимизации сгенерированных знаний по причине нехватки известных параметров. После успешного применения логического высказывания, мы поочерёдно применяем каждое из утверждений, находящихся в словаре, в надежде получить желаемый результат, сокращая таким образом количество утверждений, рассматриваемых на каждом шаге обхода.

Когда при «итерационном обходе» прогресс в решении остановится, а при «обходе тезаурусом» все логические утверждения окажутся пройденными, можно утверждать о том, что известных параметров для нахождения величины было недостаточно (класс задач, ориентированный на поиск неизвестной величины) или заданное утверждение не является истинным в текущем состоянии базы знаний (класс задач, ориентированный на поиск доказательства).

Для выигрыша во времени решатель пользуется специальным набором утверждений, связка которых соединена с классами объектов через отношение *основные утверждения**. Необходимость введения такой связки ещё и тем, что не все логические утверждения имеют прикладной характер: часть из них используется для дидактических целей, например при обучении, а не для применения в рамках задач.

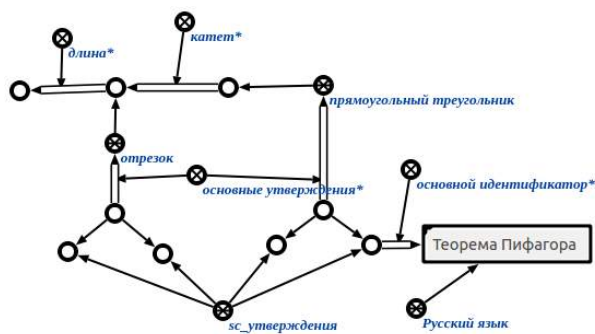


Рисунок 14 – Стратегия

9.2.5. Этап применения правил логического вывода

На данном этапе происходит попытка применения утверждения, полученного на предыдущем шаге, с целью генерации в системе новых знаний. Если такое применение справедливо (например, посылка истинна) и имеет смысл (в результате применения будут сгенерированы новые знания), то осуществляется генерация новых знаний на основе одного из правил логического вывода. Если в данном контексте вывод на основе данного утверждения невозможен или нецелесообразен,

решение возвращается на предыдущий этап. В случае успешного применения утверждения происходит переход к следующему этапу решения.

На данный момент решатель поддерживает возможность применения двух типов утверждений:

а) *применение имплицитных высказываний* - граф условия соотносится с левой частью и, в случае успеха, дополняется *sc-конструкцией*, состоящей из *sc-констант*, содержащейся в правой части импликация.

б) *применение утверждений с эквивалентными частями* - сначала для графа условия находится часть логического утверждения, изоморфная ему, после чего к нему достраиваются новые знания, в соответствии с оставшимися частями.

Граф условия после применения теоремы Пифагора изображен на рисунке 15.

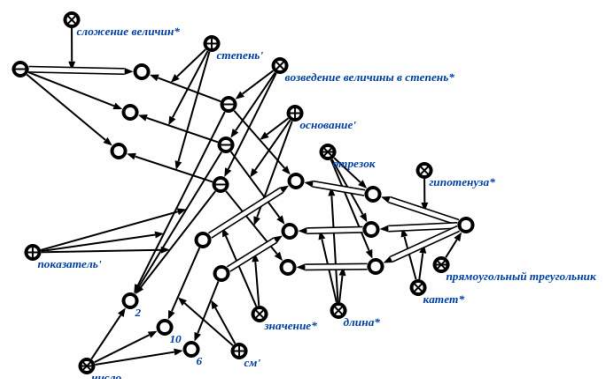


Рисунок 15 – Логика

9.2.6. Этап оптимизации сгенерированных знаний и сборки мусора

На данном этапе происходит интерпретация математических отношений, сгенерированных в процессе решения на предыдущем этапе, то есть попытка вычисления недостающих значений компонентов связок на основе имеющихся значений.

В существующей версии решателя возможна как обработка арифметических действий (сумма, умножение, возведение в степень, взятие корня и логарифма), так и вычисление тригонометрических выражений (синуса, косинуса, тангенса и обратных им функций). Вместе они представляют собой мощный математический аппарат, позволяющий решать обширный спектр задач, находящихся на стыке предметных областей.

Если вычислить все недостающие значения не представляется возможным, то все знания, сгенерированные на предыдущем этапе, уничтожаются и решение переходит на этап применения стратегий. В таком случае применение логического вывода для рассматриваемого на предыдущем шаге утверждения считается нецелесообразным. В конечном итоге происходит удаление конструкций, ставших ненужными и по каким-либо причинам не удаленных на предыдущих этапах решения.

Если все этапы решения выполнены успешно, то решение возвращается ко второму этапу, и в результате выполнения поисковой операции пользователь видит полученный ответ. В противном случае, процесс повторяется еще раз, то есть, мы возвращаемся к этапу стратегии.

Такой вид примет граф условия в конце решения задачи:

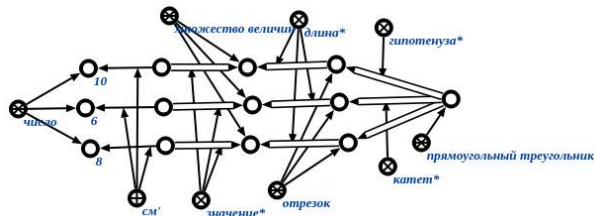


Рисунок 16 – Финальный граф

В будущем, после тестирования и полной отладки решателя, планируется внедрить его в Библиотеку типовых подсистем интеллектуальных систем.

Добавление компонента в Библиотеку многократно используемых компонентов OSTIS или обновление уже существующего может осуществляться двумя способами:

- В случае платформенно-зависимого многократно используемого компонента OSTIS логично использовать стандартные системы контроля версий и с их помощью с учетом прав доступа вносить предложения по изменению компонента.
- В случае платформенно-независимого многократно используемого компонента OSTIS предложение по добавлению или изменению компонента может формироваться прямо в *sc-памяти* средствами *SC-кода* и рассматриваться администратором базы знаний с учетом методики проектирования баз знаний (в общем случае любой платформенно-независимый многократно используемый компонент OSTIS рассматривается как часть базы знаний).

10. Расширение возможностей пользовательского интерфейса дочерней *sc-системы*

В конкретных прикладных системах, разработанных по Технологии OSTIS помимо Ядра *sc-моделей пользовательских интерфейсов* и средств работы с базовыми языками представления *SC-кода* может оказаться целесообразным использование в системе дополнительных средств отображения или редактирования знаний какого-либо конкретного вида. Подробно методика и средства проектирования пользовательских интерфейсов по Технологии OSTIS рассмотрена в [Корончик, 2013]

Для рассматриваемой в данной работе справочной системы по геометрии был разработан

специализированный редактор геометрических чертежей, совместимый с выбранным вариантом реализации ядра пользовательских интерфейсов. Данный компонент поставляется в виде набора исходных кодов [github.com ясько], и может быть встроен в реализацию ядра по инструкции приведенной в описании данной реализации. [github.com sc-web]. После встраивания редактора в систему пользователь получит возможность создавать геометрические чертежи из доступных элементов, сохранять в файл и загружать из файла созданный чертеж.

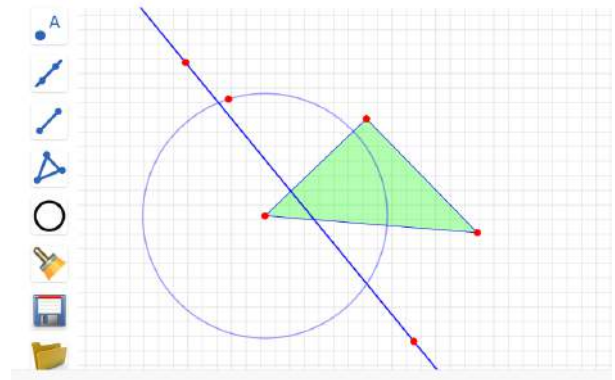


Рисунок 17 – Пример использования редактора геометрических чертежей

Заключение

В работе рассмотрена методика проектирования систем, управляемых знаниями, по Технологии OSTIS на примере справочной системы по геометрии. Описаны основные этапы проектирования, указаны возможные способы расширения функционала дочерней *sc-системы*.

При этом рассмотренные процессы развития базы знаний, машины обработки знаний и пользовательского интерфейса в дочерней *sc-системе* могут выполняться параллельно и в достаточно большой степени независимо друг от друга.

Библиографический список

- [Борисов, 2014] Борисов, А.Н. Построение интеллектуальных систем, основанных на знаниях, с повторным использованием компонентов. Открытые семантические технологии проектирования интеллектуальных систем (OSTIS-2014): материалы IV Междунар.научн.-техн.конф./Д.В. Шункевич// Мн.: БГУИР, 2014, -- С.97-103
- [Голенков, 2012] Голенков, В.В., Гулякина Н.А. Графодинамические модели параллельной обработки знаний: принципы построения, реализации и проектирования./В.В. Голенков, Н.А., Гулякина// Открытые семантические технологии проектирования интеллектуальных систем (OSTIS-2012): материалы Междунар. научн.-техн.конф. Мн.: БГУИР, 2012 – С.23-52
- [IMS] Документация. Технология OSTIS. [электронный ресурс]. – Режим доступа: <http://ims.ostis.net/>
- [sc-machine] sc-machine. [электронный ресурс]. – Режим доступа: <https://github.com/deniskoronchik/sc-machine>
- [Структура sc-машинны] Структура sc-машинны. [электронный ресурс]. – Режим доступа: <https://github.com/deniskoronchik/sc-machine/wiki>

[scp-machine] scp-machine. [электронный ресурс]. – Режим доступа: <https://github.com/ShunkevichDV/scp-machine>

[sc-web] sc-web. [электронный ресурс]. – Режим доступа: <https://github.com/deniskoronchik/sc-web>

[Шункевич, 2015] Шункевич, Д.В. Средства поддержки компонентного проектирования систем, управляемых знаниями. Открытые семантические технологии проектирования интеллектуальных систем (OSTIS-2015): материалы V Междунар.научн.-техн.конф./Д.В. Шункевич [и др.]// Мн.: БГУИР, 2015

[Давыденко, 2013] Давыденко, И.Т. Технология компонентного проектирования баз знаний на основе унифицированных семантических сетей. Открытые семантические технологии проектирования интеллектуальных систем (OSTIS-2013): материалы III Междунар.научн.-техн.конф./И.Т. Давыденко// Мн.: БГУИР, 2013 – С.185-190

[Шункевич, 2013] Шункевич, Д.В. Модели и средства компонентного проектирования машин обработки знаний на основе семантических сетей. Открытые семантические технологии проектирования интеллектуальных систем (OSTIS-2013): материалы III Междунар.научн.-техн.конф./Д.В. Шункевич// Мн.: БГУИР, 2013.

[Заливако, 2012] Заливако, С.С., Шункевич, Д.В. Семантическая технология компонентного проектирования интеллектуальных решателей задач. Открытые семантические технологии проектирования интеллектуальных систем (OSTIS-2012): материалы II Междунар.научн.-техн.конф. /Д.Н. Корончик// Мн.: БГУИР, 2012. – С. 297-315

[Корончик, 2013] Корончик, Д.Н. Реализация хранилища унифицированных семантических сетей. Открытые семантические технологии проектирования интеллектуальных систем (OSTIS-2013): материалы III Междунар.научн.-техн.конф. /Д.Н. Корончик// Мн.: БГУИР, 2013 – С.125-129

THE METHODOLOGY OF KNOWLEDGE BASED SYSTEM COMPONENT DESIGN

Shunkevich D.V., Davydenko I.T., Koronchik D.N., Hubarevich N.U., Boriskin A.S.

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

shunkevichdv@gmail.com

ir.davydenko@gmail.com

deniskoronchik@gmail.com

stasia@tut.by

coloss_000@mail.ru

In this paper the principles of knowledge-based systems design, based on OSTIS technology, is considered. Indicated technology is an open technology, focused on knowledge-based system design of various intelligent systems. In paper is considered the process of systems design by the example of reference system on the geometry, the processes of knowledge base increase, knowledge processing machine and human-system interface are taken up.

Key words: systems, knowledge-based, semantic networks, knowledge base, intelligent task resolving.

Introduction

Currently, the main issue of the day is absence of well thought-out principles structural using of the eventual computer systems design experience. It is the main granary of reduplication during the development of different system's components. Multiple recurring development of existing technical solutions is stipulate

for the well-known technical tasks are bad integrated in developing system or it is hard to find them. And the main goal of the OSTIS technology is the issue of components compatibility and the development of knowledge-based component design tools.

To solve the problem of knowledge compatibility the semantic networks with set-theoretic interpretation are used. And as an example the process of reference system on the geometry would be described.

1. IMS subsystem for supporting sc-subsystems component design

In the context of intelligent metasystem IMS, which is directed on consulting services and supporting system developers who use the OSTIS technology, and accumulation libraries of reusable components of OSTIS single out a support system for components and categories of intelligent system design. It's main tasks are:

- Search possibility of necessary component or set of components in Library of reusable components of OSTIS technology;
- Formation, if it is necessary, sc-structures, comply with the selected reusable components;
- Possibility of transportation of selected reusable components in sc-subsystem;
- In the case of deployment basic version of child sc-system – possibility to install the first based version of sc-system, which can be increased by the added components.

2. The stages of sc-subsystem design

During creating starting version sc-subsystem on OSTIS Technology the four main steps can be marked:

- Selection and installation platform implementation of the sc-subsystem.
- Installation the Core of knowledge base sc-models, the set of based reusable components of knowledge base sc-models, needed for the work of the first sc-system prototype.
- Installation the Core of sc-machine, the set of based reusable components of sc-machine, needed for the work of the first sc-system prototype.
- Installation the Core of interface sc-models, the set of based reusable components of interface sc-models, needed for the work of the first sc-system prototype.
- Installation the core of subsystem for supporting the filial system design in the composition of estimated sc-subsystem.

Conclusion

The paper considers the principles of knowledge-based systems design, based on the OSTIS technology, by the example of the reference system on the geometry.



OSTIS-2015

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822:514

МОДЕЛИ И АЛГОРИТМЫ ИНТЕГРАЦИИ ЗНАНИЙ НА ОСНОВЕ ОДНОРОДНЫХ СЕМАНТИЧЕСКИХ СЕТЕЙ

Ивашенко В.П.

*Белорусский государственный университет информатики и радиоэлектроники,
г. Минск, Республика Беларусь*

ivashenko@bsuir.by

Рассматриваются модели и алгоритмы интеграции знаний, представленных в унифицированном виде однородными семантическими сетями, имеющими теоретико-множественную интерпретацию, учитывающие неполноту, нечёткость, неопределённость представленных знаний и их изменение со временем

Ключевые слова: представление знаний, онтология, семантическая сеть, интеграция, нечёткие множества.

ВВЕДЕНИЕ

В системах, основанных на знаниях [Гаврилова и др., 2000], одной из важных задач является задача интеграции знаний, заключающаяся в выявлении фрагментов семантической сети, представляющих одни и те же знания, и их слиянии, что позволяет исключить избыточные элементы семантической сети, способные затормозить до предела работу системы, или сократить их количество, увеличить число ассоциаций для конкретных понятий, уточнить структуру модели предметной области. Задача интеграции необходима для обеспечения производительности и развития интеллектуальных систем в условиях неполноты знаний [Нариньяни, 2000], которая в свою очередь неизбежна в открытых, обучающихся интеллектуальных системах.

Интеграция знаний заключается в погружении новых знаний в состав знаний, уже известных интеллектуальной системе, являясь необходимым этапом понимания информации, которая либо поступает в интеллектуальную систему извне, либо генерируется (порождается) в процессе обработки информации. Процесс понимания в интеллектуальной системе заключается не только и не столько в помещении добавляемых новых знаний в память, но – в выявлении синонимов, связывании и отождествлении поступающих фрагментов с уже известными. Таким образом, интеграция знаний является основой функционирования любой интеллектуальной системы и во многом определяет качество интеллектуальной системы, будучи последним и основным этапом понимания поступающих (приобретаемых) знаний.

Решение задачи интеграции связано с рядом трудностей на разных уровнях рассмотрения интеллектуальной системы:

- на уровне языков и моделей представления знаний;
- на уровне моделей, методов и алгоритмов, применяемых для решения задачи интеграции;
- на уровне архитектуры программных средств, которыми осуществляется интеграция знаний.

На уровне языков и моделей представления трудности обусловлены следующими недостатками:

- необходимость выбора среди нескольких моделей представления знаний, между которыми нет однозначного предпочтения;
- неоднородность моделей представления знаний, приводящая к тому, что в базе появляются синонимичные структуры разных типов, требующие от пользователя их согласования;
- ограниченные возможности представления знаний в некоторых языках и моделях;
- отсутствие разделения понятий и терминов (исключения – DOGMA и Gellish),
- отсутствие поддержки монотонного расширения базы знаний (исключения – OWL, OWL2 и языки, построенные на основе классических логических моделей).

На уровне архитектуры программных средств присутствуют следующие недостатки:

- ограничения на расширение базы знаний;
- в силу зависимости от сторонних средств, ограниченность иерархической таксономической структурой онтологии возможности интеграции знаний или отсутствие таких возможностей, необходимость выбора средств интеграции пользователем;

- отсутствие общих стандартов совместимости разработанных фрагментов баз знаний;
- ограниченность средств поиска и каталогизации разработанных фрагментов баз знаний;
- ограниченность или отсутствие развитых средств взаимодействия с внешней средой.

К языкам представления знаний в разных системах на сегодняшний день относятся: Conceptual Graph [Sowa et al., 2008], Frame-logic [Michael Kifer et al., 1995], Knowledge Interchange Format [Genesereth et al., 1992], Integrated DEFinition for Ontology Description Capture Method (IDEF5) [IDEF5, 1994], Common Algebraic Specification Language [CoFI:CASL-Summary, 2004], Concept maps/UML [Novak et al., 2008] [ISO24707], RDF/RDFS [W3C:RDFS, 2004], DARPA Agent Markup Language [DAML, 2006], CycL [CycL, 2002], Ontology Inference Layer [W3C:DAML+OIL, 2001], RDF/OWL-Lite, RDF/OWL-DL, RDF/OWL Full [W3C:OWL, 2004], Topic Maps [ISO13250], XTM/LTM/CML/GML [XTM, 2001], Common Logic [ISO24707], Developing Ontology-Grounded Methods and Applications [Mustafa Jarrar et al., 2008], Formal English [Martin Ph., 2002], Gellish [Van Renssen, 2005], RDF/Rule Interchange Format [W3C:RIF, 2010], Open Biomedical Ontologies [Smith et al., 2007], RDF/OWL2 [W3C:OWL2, 2009], RDF/OWL2 EL, RDF/OWL2 RL и др.

Для поиска в базах знаний и онтологиях [Хорошевский, 2008] используются такие языки, как: RDQL, squish, SPARQL [W3C:SPARQL, 2008], KQML, DMX, Datalog, TMQL, ERROL, RuleML, RQL, OQL, TQL, VERSA, DQL и др.

Среди средств, которые могут рассматриваться в качестве основы для разработки баз знаний, можно выделить: оболочки экспертных систем (CLIPS (FuzzyCLIPS, DYNACLIPS, WxCLIPS) [CLIPS, 1991], SOAR, OPS83, RT-EXPERT, MIKE, BABYLON, WindExS, ES; ACQUARE, Easy Reasoner, ECLIPSE, EXSYS Professional, SIMER+MIR, AT ТЕХНОЛОГИЯ, CAKE v2.0) [Гаврилова и др., 2000]; инструментальные пакеты для разработки экспертных систем (G2, ART, KEE, Knowledge KRAFT); системы, ориентированные на обработку онтологий [Sowa et al., 2008] – Protégé, WebOnto, OntoEdit, WebODE, OilEd, OntoLingua. К существующим на настоящий момент средствам интеграции онтологий можно отнести: Optima, Prompt, Ontolingua, Chimaera [McGuinness et al., 2000], ONION [Gangemi et al., 1996], COMA++ [Aumueller et al., 2005] и др.

Достоинствами приведённых средств являются: поддержка представления знаний различного вида различными моделями представления знаний в рамках одной системы; наличие средств визуального проектирования баз знаний; наличие средств верификации базы знаний, включая проверку на непротиворечивость; возможность монотонного расширения базы знаний, наличие средств интеграции баз знаний; наличие средств

поддержки обмена данными с внешней средой, включая средства обмена данными в реальном времени.

Для преодоления трудностей семантической интеграции [Doan and Halevy, 2005], [Кудрявцев, 2008] (отображения онтологий (ontology mapping)) и интеграции знаний (knowledge integration) в базах знаний используются следующие подходы:

- сравнение и выравнивание онтологий (ontology matching & alignment),
- интеграция онтологий (ontology merging),
- семантическое сравнение (semantic matching),
- семантическая унификация (semantic unification).

Для интеграции онтологий можно выделить методы:

- структурно-синтаксические:
 - анализ внутренней структуры
 - экстенционально-статистические
 - анализ внешней структуры (метаструктурный анализ)
 - терминологические (лексические)
- логико-семантические

К наиболее развитым подходам и методам интеграции можно отнести ONION (ONtology composition [Mitra, 2001]), формальный концептуальный анализ (FCA [Ganter, 1999], [Stumme, 2001]) и варианты его развития для нечётких (FOGA [IEEE, 2006]) и неопределённых множеств, методы использующие элементы семантического анализа – СТХMATCH [Bouquet, 2003] и S-match. Однако все эти методы, несмотря на использование некоторыми из них нечётких и неопределённых множеств, плохо приспособлены или не приспособлены к интеграции знаний в условиях наличия НЕ-факторов [Нариньяни, 2000]. Перечисленные методы, и методы, основанные на мерах близости (CUPID [Maldavan, 2001] и т.п.), не обеспечивают достаточной формальной строгости и непротиворечивости онтологий или баз знаний, получаемых в результате.

Поэтому возникает противоречие: существует необходимость точного решения задачи интеграции, позволяющего исключить привлечение эксперта для получения качественного результата, при отсутствии моделей, методов и алгоритмов интеграции, которые бы обеспечивали точность этого решения в условиях необходимости интеграции неполных, нечётких и изменяющихся со временем знаний (при наличии НЕ-факторов).

Следует отметить, что интеграции знаний рассматривается как развитие задачи интеграции данных, учитывающих их семантические аспекты. На уровне интеграции знаний основные трудности интеграции вызваны наличием различных моделей представления знаний и их неоднородностью. Для их преодоления строятся обёртки, оболочки (медиаторы), отображения одних моделей

представления на другие. Как правило, среди моделей представления существует модель глобального (универсального) представления, на которую можно отобразить все другие. Это позволяет достаточно просто свести количество отображений к необходимому минимуму. При наличии модели глобального представления существует два подхода, ведущих своё происхождение от интеграции данных и основывающихся на выборе базовых моделей представления: выбор глобальной или локальных моделей представления. Предпочтительнее выбор модели глобального представления в качестве базовой, так как отображение в локальные модели будет зависеть только от числа медиаторов (посредников) отображения, в ином случае – не все данные могут быть отображены в глобальную модель, а только те, которые представлены и представимы в локальных моделях. Если глобальная (универсальная) модель позволяет представлять семантически различающиеся знания единственным образом, то можно говорить об универсальной модели унифицированного представления знаний. В работе предлагается подход, в котором используется отображение знаний на глобальную модель унифицированного семантического представления знаний (рисунок 1).

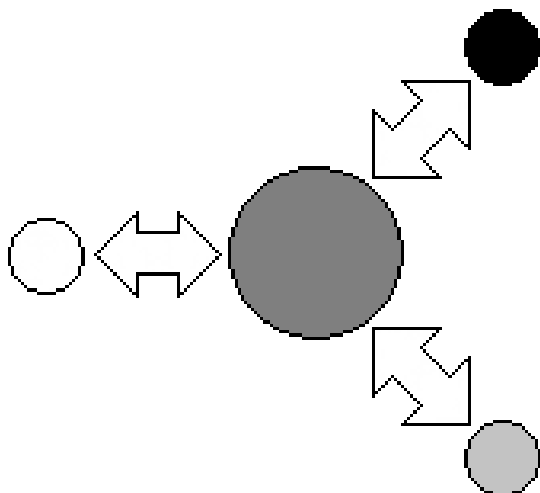


Рисунок 1 – Отображение (трансляция) знаний на модель унифицированного представления знаний

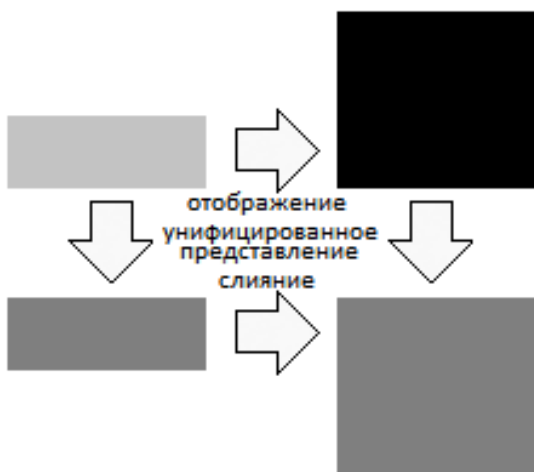


Рисунок 2 – Схема интеграции знаний

В этом случае интеграция знаний может быть сведена к следующей схеме (рисунок 2): 1) перейти к унифицированному представлению знаний путём трансляции на языки модели унифицированного представления знаний, в результате имеющиеся отображения одних знаний на другие можно заменить отношением слияния (склеивания) элементов семантической сети; 2) осуществить слияние синонимов, т.е. элементов дублирующегося представления знаний.

В результате интеграция знаний в языках модели унифицированного семантического представления знаний сводится к слиянию элементов семантической сети (рисунок 3).

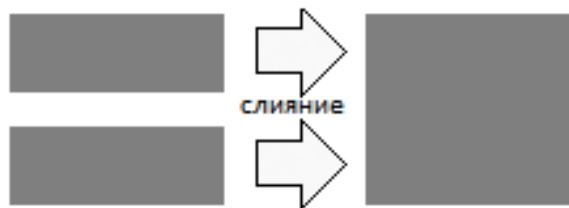


Рисунок 3 – Интеграция знаний в модели унифицированного семантического представления знаний

1. Модель унифицированного семантического представления знаний

Семантическая модель интеграции использует унифицированное представление знаний и обеспечивает интеграцию sc-моделей баз знаний и их фрагментов [Ивашенко, 2012]. Унифицированное представление знаний обеспечивается моделью унифицированного семантического представления знаний, которая является частным случаем такой модели представления знаний, как семантические сети и задаётся семейством совместимых sc-языков, использующих унифицированный способ семантического кодирования Semantic Computer code (SC-код) [Голенков и др, 2001] и поддерживающих представление знаний различного вида [Ивашенко, 2003], [Ивашенко, 2004], [Ивашенко, 2011a]. Особенности SC-кода являются: простой алфавит, содержащий узлы и дуги, простой синтаксис, базовая теоретико-множественная интерпретация. Семантика sc-языка задаётся на основе модели ситуативных множеств, являющейся развитием моделей L-нечётких множеств и неоднородных нечётких множеств, предложенных Дж. Гогеном и А. Кофманом [Кофман, 1982].

Модель ситуативных (событийных, нестационарных [Ивашенко, 2012]) множеств может быть задана следующей шестёркой компонентов:

$$\langle Universe, [0;1], Events, r, h, sM \rangle \quad (1)$$

где *Universe* – универсальное множество объектов предметной области, *Events* – множество элементарных событий, $r \subseteq Events \times Events$ – отношение доступности (следования во времени)

событий, $h \in (2^{\text{Events}})^{\text{Universe}}$ – функция, задающая множество событий существования каждого элемента универсального множества, sM – семейство пар множеств событий существования ситуативного множества и соответствий (нечёткой) ситуативной принадлежности элементов универсального множества ситуативному множеству, отображающих элементы ситуативных множеств, множества событий и соответствующие им наборы степеней нечёткой принадлежности высших порядков на множество степеней нечёткой принадлежности $[0;1]$.

В соответствии с введённой моделью ситуативных множеств для описания динамических предметных областей, используются понятия нестационарной принадлежности и непринадлежности.

Выбор семантических сетей в качестве основы для представления знаний обусловлен их качественным отличием от строковых языков представления знаний в аспекте ассоциативности. Семантические сети позволяют качественно увеличить связность концептов и абстракций, обеспечивая экспоненциальный рост количества $\dot{V}(\langle p, q \rangle)$ различных связанных фрагментов от числа обозначений p и их связей q , являющихся готовыми ответами на вопросы пользователя, в отличие от полиномиального роста их количества в случае представления знаний в виде строк, при котором в худшем случае поиск требует просмотра всех символов в строке:

$$\frac{p * (p + 1)}{2}. \quad (2)$$

$$\dot{V}(\langle p, q \rangle) \geq \frac{(\lceil (q - p) / (2 * p) \rceil + p - 1)!}{(\lceil (q - p) / (2 * p) \rceil - 1)! * (p!)}. \quad (3)$$

Модель унифицированного семантического представления знаний задаётся следующими компонентами.

$$\{SCLanguages, R_{SC}, F_{SC}\} \\ F_{SC} = A_{SC} \cup I_{SC} \cup N_{SC} \cup S_{SC} \cup E_{SC} \cup K_{SC} \quad (4)$$

$SCLanguage$ – множество sc-языков, R_{SC} – отношения на множестве sc-языков (sc-подъязыка и трансляции), F_{SC} – функции sc-языков, A_{SC} – алфавитные функции sc-языков, I_{SC} – функции отношений инцидентности, N_{SC} – синтаксические предикаты sc-языков, S_{SC} – семантика sc-языков, E_{SC} – ключевые элементы sc-языков, K_{SC} – спецификация sc-языков и их ключевых элементов (отображение на множество онтологий sc-языков).

Semantic Code ориентирован на универсальность, т.е. на представление любых видов знаний. Semantic Code является средством

унификации представляемых знаний. На множестве sc-языков (языков, представленных в SC-коде) определены отношения sc-подъязыка и трансляции. Подъязык, являющийся пересечением выделенного семейства совместимых специализированных sc-языков, рассматривается как интегрированный sc-язык представления знаний (SCK). Основным принципом построения sc-языков является представление понятий, соответствующих основным классам объектов, описываемых sc-языком, и отношений между этими объектами ключевыми узлами такого sc-языка: каждому sc-языку однозначно сопоставляется конечное множество ключевых узлов (элементов) этого языка. Каждый ключевой узел задаёт ограничения на собственную семантическую окрестность в информационных конструкциях (текстах) этого языка. Множество ключевых элементов интегрированного sc-языка представления знаний является объединением множеств ключевых элементов остальных sc-языков выделенного семейства. Характеристиками sc-языка являются: мощность множества ключевых узлов sc-языка; семейство множеств собственных семантических окрестностей ключевых элементов языка; наличие функциональных зависимостей между собственными окрестностями множеств ключевых элементов sc-языка; соотношение алгоритмических сложностей поиска или вычисления элементов собственных семантических окрестностей на основании существующих зависимостей. Построена семантическая онтология существующих языков и моделей представления знаний, в которой указано место унифицированной модели представления знаний и её соотношение с существующими моделями представления знаний.

Введенные в sc-языки ключевые элементы поддерживают соответствующее семантически эквивалентное представление концептов OWL 2 QL и OWL 2 EL [W3C:OWL2, 2009], а также поддерживают представление множеств, мультимножеств, ситуативных множеств, отношений, включая ролевые отношения – строгие подмножества отношения принадлежности, простых, целых и рациональных чисел, логических формул. Остальные концепты могут быть определены с помощью логического sc-языка.

Модель унифицированного семантического представления знаний поддерживает представление знаний в условиях влияния различных НЕ-факторов: неполноты, неопределённости, нечёткости и пр. Далее приведены примеры представления нечётких и изменяющихся во времени знаний.

2. Интеграция знаний

Модель унифицированного семантического представления знаний, позволяет представлять знания различного вида, в т.ч. необходимые для решения задач интеграции, включая отношение совпадения. На основе модели унифицированного

семантического представления знаний разработана модель спецификации знаний, которая позволяет рассматривать различные типы фрагментов баз знаний [Ивашенко, 2009а] и их спецификации, задающие требования к фрагменту базы знаний (семантической сети) и позволяющие осуществить верификацию и оценить его качество. Модель спецификации знаний задаётся множеством формальных моделей онтологий фрагментов баз знаний Z , множеством морфизмов, соответствий между ними, и множеством отношений между парами формальных моделей и множеством соответствий и морфизмов.

$$\left\langle Z \cup 2^{\{\omega(z) \mid z \in Z\}^2}, 2^\Omega \right\rangle; \quad (5)$$

$$\Omega = \bigcup_{\{x\} \cup \{y\} \in Z} (\{x\} \times \{y\}) \times 2^{\omega(x) \times \omega(y)},$$

где 2^Ω – множество отношений модели спецификации; ω – функция элементов онтологической модели. Онтологическая модель каждого фрагмента базы знаний из множества Z задаётся тройкой $\langle G, R, O \rangle$, где G – непустое конечное множество обозначений спецификации фрагмента; R – ориентированное конечное множество отношений на обозначениях фрагмента; O – ориентированное конечное множество функций интерпретации обозначений.

$$\omega(z_i) = G_i \cup \{r \mid r = R_{ij}\} \cup \{o \mid o = O_{ij}\} \cup \{k \mid k \in R_{ij}\} \cup \{p \mid p \in O_{ij}\} \cup \{a \mid \langle a, v \rangle \in O_{ij}\}. \quad (6)$$

В рамках модели спецификации знаний построены онтологии sc-языков, используемые для описания типов фрагментов баз знаний и их интеграции. Среди требований качества по логико-семантическим признакам выделены требования полноты, непротиворечивости и неизбыточности. По методологическим признакам выделены требования к конструкциям предметно-специализированных языков, к конструкциям, описывающим определяющие признаки типа фрагмента базы знаний (семантической сети), и к конструкциям, описывающим отображения элементов и отношения интеграции фрагментов баз знаний. Для вычисления мер в соответствии с выделенным языком спецификации и записываемыми в его текстах требованиями используются функции вида

$$\left\{ L \times 2^\Phi \left. \vphantom{L \times 2^\Phi} \right| L \in SCLanguages \right\}, \quad (7)$$

где Ω – объединение отношений модели спецификации; ζ – отображение множества текстов языка спецификации на множество их онтологических моделей; Φ – упорядоченное множество оценок (например $\Phi = [0, 1]$).

Справедлива коммутативная диаграмма, описывающая взаимосвязь между фрагментами баз знаний и их моделями в моделях интеграции и

спецификации знаний, при заданных отображениях полного представления g :

$$\begin{array}{ccc} z_1 & \xrightarrow{h \in R_{NMH}((z_1, z_2))} & z_2 \\ \downarrow g & & \downarrow g \\ j_1 & \xrightarrow{f \in R_F((j_1, j_2))} & j_2 \end{array}; z_1, z_2 \in Z; j_1, j_2 \in J, \quad (8)$$

где J – множество фрагментов баз знаний, представленных в модели унифицированного семантического представления знаний, R_F – отношение слияния, R_{NMH} – отношение гомоморфного отображения.

Путём интеграции осуществляется переход от некоторого исходного фрагмента семантической сети к требуемому оптимизированному фрагменту базы знаний, имеющим более высокое качество [Ивашенко, 2011а].

В задаче интеграции двух фрагментов баз знаний в качестве исходных данных используются два фрагмента семантической сети и дополнительная метаинформация о свойствах обозначений, принадлежащих этим фрагментам. Результатом решения этой задачи является нахождение интегрированного фрагмента семантической сети, такого, что каждое обозначение из исходных интегрируемых фрагментов, имеет единственное представление в виде соответствующего обозначения интегрированном фрагменте [Ивашенко, 2009а]. Будем говорить, что осуществляется слияние двух обозначений исходных фрагментов тогда и только тогда, когда каждому из обоих соответствует в интегрированном фрагменте единственный знак. При интеграции в качестве дополнительной метаинформации может использоваться информация о внешних обозначениях (идентификаторах) понятий (обозначений) или информации, заданная базовой или ключевой рефлексивной семантикой обозначений из этих фрагментов. В условиях неполноты информации можно выделить два типа стратегий слияния обозначений: безопасные (выполняемые однозначно и непротиворечиво) и небезопасные. Безопасная стратегия включается в любую стратегию слияния.

При интеграции двух фрагментов между множеством их элементов (обозначений) выявляется некоторое множество пар потенциально синонимичных элементов (обозначений). Методика точного решения этой задачи включает как решение этой задачи вручную, путём получения соответствующего ответа от разработчика, так и автоматизированное решение на основе известной базовой теоретико-множественной семантики элементов и ключевых узлов sc-языка. Исходя только из базовой теоретико-множественной интерпретации, уже в некоторых фрагментах базы знаний можно легко установить факт наличия или отсутствия синонимии. Зная множество таких пар и множество всех возможных пар всех элементов в заданном фрагменте базы знаний, легко вычислить

множество и число пар потенциально синонимичных элементов такой базы знаний.

Имея информацию о потенциально синонимичных обозначениях, можно оценить множество и количество всевозможных структурно различимых вариантов слияния потенциально синонимичных элементов (обозначений) в элементы результирующего фрагмента. Это множество обозначим величиной $I(G)$, где G – симметричный ориентированный граф, множество рёбер $E(G)$ которого является множеством всех пар потенциально синонимичных обозначений исходных баз знаний, а множество вершин $V(G)$ – множеством всех обозначений исходных фрагментов

$$E(G) \subseteq (V(G))^2. \quad (9)$$

Для того, чтобы определить множество $I(G)$, используем специальную операцию

$$A \overset{\text{def}}{\cup} B = \bigcup_{(P,Q) \in A \times B} \{P \cup Q\}, \quad (10)$$

которая является ассоциативной и коммутативной.

Чтобы задать множество $I(G)$, введём семейство всевозможных множеств рёбер ориентированных графов $C(G)$ на множестве вершин графа G , каждая компонента связности которых является полным подграфом [Ивашенко, 2009b], тогда

$$I(G) = \left(\bigcup_{e_{ij} \in E(G)} \{\emptyset, \{e_{ij}\}\} \right) \cap C(G). \quad (11)$$

Число Q элементов множества $I(G)$ можно рассматривать в виде критерия качества: чем меньше это число, тем выше качество и наоборот. Однако, уже для небольших фрагментов баз знаний, это число может оказаться большим и трудным для расчёта, поэтому на практике более целесообразно использовать или логарифм этого числа или логарифм его оценки, или связанные с ними величины (число пар потенциальных синонимов).

$$Q = |I(G)|. \quad (12)$$

В работе [Ивашенко, 2009b] для числа Q установлены соотношения для верхней и нижней границы.

Интеграция онтологий и фрагментов баз знаний осуществляется через слияние обозначений. Рассмотрим m объектов области значений семантической интерпретации некоторого знака. Тогда, на множестве этих объектов, число неоднозначных и однозначных постоянных семантических интерпретаций (семантик) знака равно $2^{2^m} - 1$.

Чтобы осуществить слияние обозначений (воплощений знака) необходимо, чтобы пересечение областей значения их семантик было непустым. В частности – теоретико-множественных семантик.

Если пересечение областей значения семантик знаков является пустым, то слияние таких

обозначений невозможно и такие обозначения различны.

Тогда как для выявления различных обозначений достаточно одного контр-примера, для выявления подлежащих слиянию знаков необходимо не только установить, что пересечение областей значения их семантик не является пустым, но и доказать, что эти знаки совпадают. Для этой цели используются различные утверждения о совпадении в виде аксиом и теорем.

Опишем свойства этого отношения совпадения. Если два обозначения совпадают, то не существует множества, которому одновременно принадлежит один из них и не принадлежит другой. Если два обозначения множеств совпадают, то множества равны. Выявить неравенство множеств просто: достаточно найти элемент, который принадлежит одному множеству и не принадлежит другому. Процедура выявления равенства множеств зависит от их свойств. Если множества имеют конечное, не очень большое число принадлежностей или непринадлежностей элементов, что характерно для понятий, то перечислив все принадлежности или все непринадлежности, число которых совпадает соответственно с его мощностью или количеством непринадлежностей, и убедившись, что среди них нет непринадлежностей элементов, принадлежащих другому множеству, можно заключить, что множества равны. Если же множества бесконечны или число принадлежностей, равно как и непринадлежностей у них, очень большое, то равенство множеств можно доказать только через утверждения об их свойствах.

Прежде, чем проводить слияние обозначений двух фрагментов семантической сети, следует провести отображение этих фрагментов (по аналогии с отображением онтологий).

Для отображения в некоторых работах [Maltese et al., 2010] используются следующие отношения между онтологическими понятиями $\{\equiv, \supseteq, \subseteq, \perp, \sqcap\}$ (таблица 1), соответственно – совпадения, обобщения, частности, строгого исключения и строгого пересечения. Например, эти отношения в проекции на теоретико-множественные соответствуют отношениям равенства множеств ($=$), надмножества (\supseteq), подмножества (\subseteq), пустого пересечения множеств ($S1 \cap S2 = \emptyset$) и непустого пересечения множеств ($S1 \cap S2 \neq \emptyset$).

Таблица 1.

Обозначение		Следствие	Пояснение
справа	слева		
\sqcap	\sqcap	\neq	строго пересекаются
\supseteq	\subseteq	$-$	включение
\sqsupset	\sqsubset	\neq	строгое включение
\perp	\perp	$-$	не пересекаются
\equiv	\equiv	\equiv	совпадение

Таблица 2.

Обозначение		Следствие	Пояснение
справа	слева		
—	—	—	соотношение не определено
≠	≠	≠	не совпадают (различны)
⊃	⊃	—	пересекаются (связность)
⊄	⊄	≠	исключение
⊈	⊈	≠	исключение пересекающихся
⊉	⊉	≠	симметричное исключение
⊋	⊋	≠	строго пересекаются
⊇	⊆	—	включение
⊇	⊇	—	включение пересекающихся
⊃	⊂	≠	строгое включение
⊃	⊃	≠	строгое включение пересекающихся
⊆	⊆	—	равенство
=	=	—	равенство пересекающихся
⊥	⊥	—	не пересекаются
≡	≡	≡	совпадение
≡	≡	≡	совпадение пересекающихся (непустых)

Для решения задачи интеграции неполных и нечётких знаний в предлагается расширенный набор отношений (таблица 2, таблица 3) {не уточнены*, различие*, связность*, исключение* (исключаемое_, исключающее_), исключение пересекающихся* (исключаемое пересекающееся_, исключающее пересекающееся_), симметричное исключение*, строгое пересечение*, совпадение*} и др. Отношение потенциальной синонимии можно выразить, как $возможная\ синонимия^* = (не\ уточнены^* \cup связность^* \cup (исключение^* \cup исключение\ пересекающихся^* \cup различие^*))$.

Следующий фрагмент (рисунок 4) позволяет в рамках модели унифицированного семантического представления знаний и модели спецификации знаний задать правило отображения и слияния знаков баз знаний при использовании методов лексико-терминологического анализа (анализа идентификаторов) [Ивашенко, 2011b].

Таблица 3.

Обозначение		Следствие	Пояснение
справа	слева		
≠	≠	≠	неинъективность (строгая неравномощность)
≠	≠	≠	неинъективность пересекающихся
≠	≠	—	инъекция (неравномощность)
≠	≠	—	инъекция пересекающихся
>	<	—	строгая инъекция
Δ	∇	—	строгая инъекция пересекающихся
≡	≡	—	биекция (равномощность)
≡	≡	—	равномощность пересекающихся

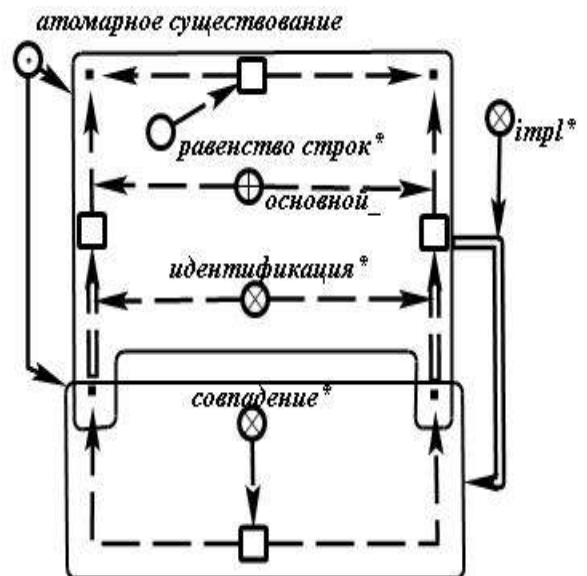


Рисунок 4 – Правило совпадения элементов с равными основными идентификаторами

Важным признаком совпадения обозначений является их принадлежность одной связке под однозначным ролевым отношением или наличие принадлежности им общего элемента под обратным однозначным ролевым отношением.

Все эти отношения можно однозначно и непротиворечиво установить в результате соответствующего базовой теоретико-множественной семантике структурного анализа, включая экстенциональный, за полиномиальное время. Более глубокий логико-семантический анализ может потребовать больших временных затрат.

Для анализа на логико-семантическом уровне важно выделять утверждения, которые приводят к совпадению знаков интегрируемых фрагментов, в результате чего можно произвести их слияние. Такими высказываниями являются высказывания о свойствах логических высказываний о единственности, высказывания о свойствах множеств без кратных вхождений элементов (канторовских множеств), высказывания об отношениях без кратных связей. Ниже приведён пример (рисунок 5), описывающий свойство логических утверждений о единственности и сводящий задачу анализа таких утверждений к задаче выявления связей отношений совпадения. Таким образом, при наличии средств логического вывода для решения задачи интеграции логико-семантическими методами достаточно описать свойства соответствующих ключевых узлов, выражающих количественные ограничения (например, единственность) или описывающих отсутствие кратных связей, через понятия совпадения sc-элементов (знаков текстов sc-языка).

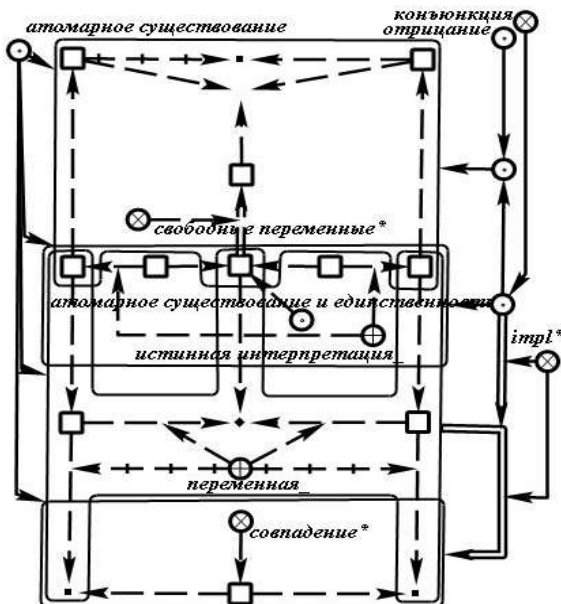


Рисунок 5 – Высказывание о слиянии значений переменных в утверждениях единственности

Алгоритм процедуры поиска множества решений задачи интеграции приведён на рисунке 7 [Ивашенко, 2011b]. Этот алгоритм использует

алгоритм сопоставления онтологий (рисунки 8, 9). Ниже (рисунок 6) приведён алгоритм выявления пар потенциальных синонимов, использующий вышеописанные правила экстенционального метода.

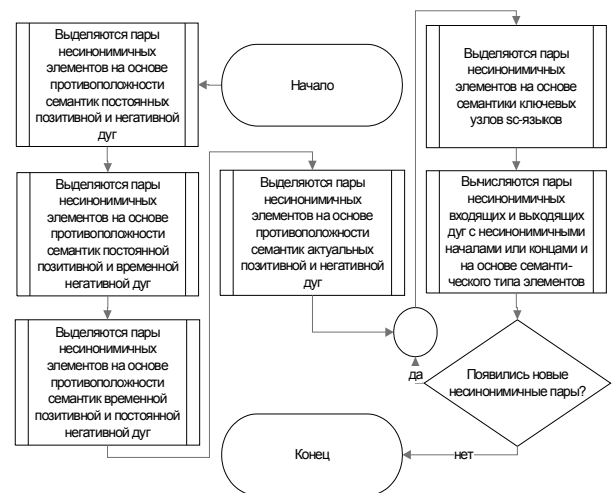


Рисунок 6 – Алгоритм вычисления множества пар различающихся (несинонимичных) элементов

При интеграции тезаурусов необходимо построить связную иерархию определяемых понятий. Для каждого понятия необходимо указать является оно определяемым или нет. Для каждого понятия необходимо указать множества понятий, на основе которых оно может быть однозначно определено, а на основе каких – нет. Для построения такой иерархии применим алгоритм, подобный обобщённому алгоритму интеграции. При поиске выполнимых интерпретаций схем обход возможных предикативных выражений осуществляется по всевозможным предварённым конъюнктивным нормальным формам, атомарные подформулы которых соответствуют алфавитным меткам, связкам инцидентности и равенству значения переменной константе и задают всевозможные размеченные привязанные с помощью констант подграфы в графе онтологии. Для сокращения перебираемых вариантов используются эвристики.

Интеграция тезаурусов (баз знаний, содержащих определение) может быть осуществлена на основе отсеивания наборов понятий, которые не могут относиться как определяющие и определяемые, в результате анализа по следующим схемам (фильтрация неоднозначно определяющих множеств понятий для каждого понятия).

Схема n-значности ($|v| = n$) свойства α задана следующим выражением.

$$\lambda(\alpha, x, v, z) = \forall w (\alpha(\langle x, w, z \rangle) \sim (w \in v)) \quad (13)$$

Схема необходимости.

$$\neg \exists U \exists V \exists Y (\neg \beta(\langle X, U, Y \rangle) \wedge \gamma(\langle Y, V, Z \rangle)) \quad (14)$$

Схема достаточности.

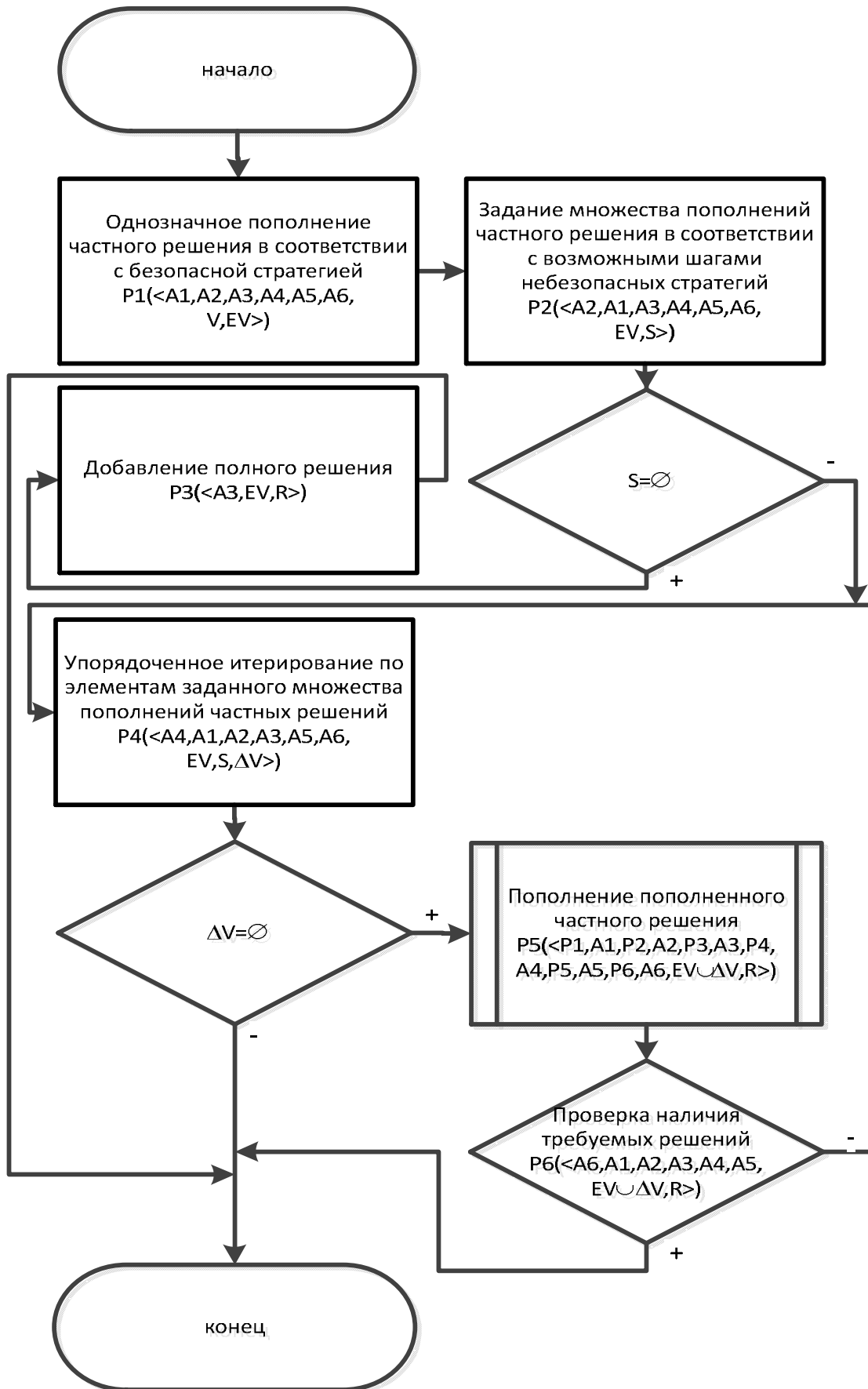


Рисунок 7 – Алгоритм процедуры поиска множества решений задачи интеграции

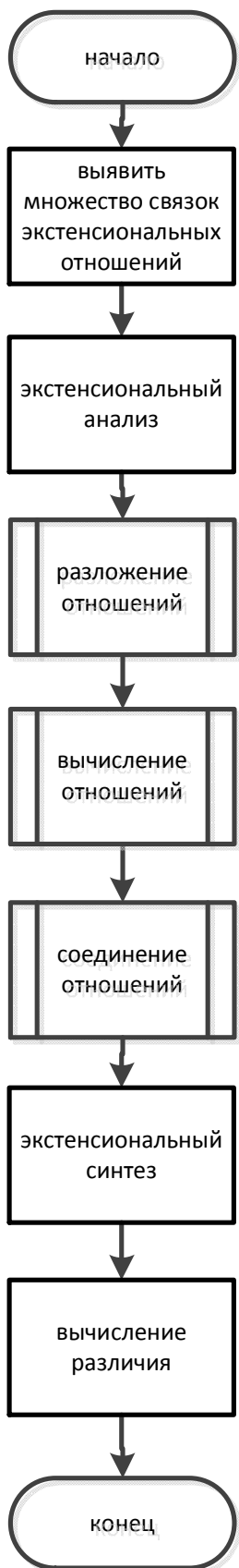


Рисунок 8 – Алгоритм процедуры сопоставления онтологий на основе экстенционального анализа троек sc-узлов

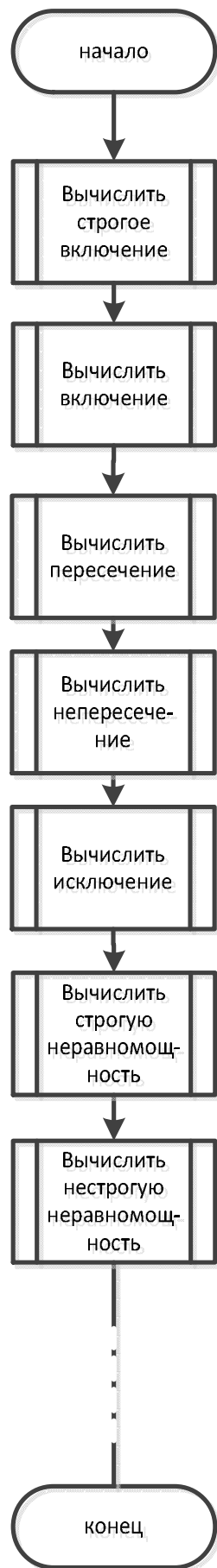


Рисунок 9 – Алгоритм процедуры вычисления отношений сопоставления онтологий на основе экстенционального анализа

Задача

В декартовой плоскости p даны точки A и C и окружность O с центром A и радиусом δ ($= 9600$).

$A \in p$

$C \in p$

$O \subset p$

Ученик провёл прямую a из точки C , касающуюся окружности O в точке H , отложил на касательной a от точки C отрезок BC длиной α ($=5600$) и построил треугольник ABC , периметр которого γ , вычисленный учеником, оказался целым числом.

$|AH| = \delta$

$\delta = 9600$

$H \in O$

$H \in a$

$AH \perp a$

$C \in a$

$BC \subset a$

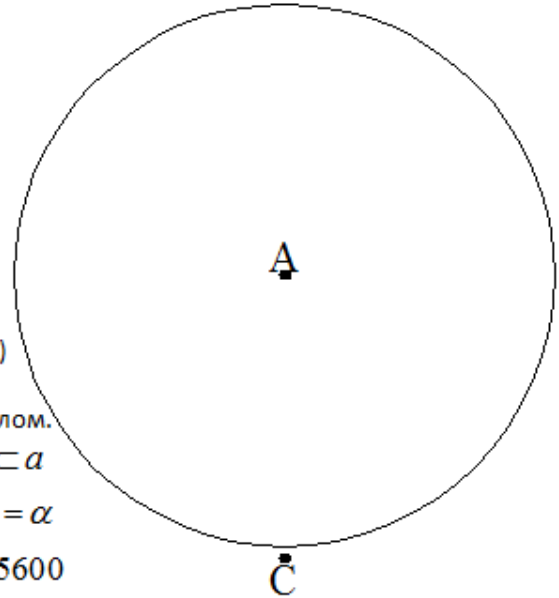
$|BC| = \alpha$

$\alpha = 5600$

ABC

$|\partial ABC| = \gamma$

$\gamma \in \mathbb{Z}$



Постройте этот треугольник в декартовой плоскости p и вычислите его периметр, если известно, что в этой декартовой плоскости точка A имеет координаты $\langle 0, 9126 \rangle$, а точка $C - \langle 0, -784 \rangle$.

$A \stackrel{p}{\cong} \langle 0, 9126 \rangle$

$C \stackrel{p}{\cong} \langle 0, -784 \rangle$

Рисунок 10 – Условие задачи на построение треугольника

$$\neg \exists U \exists V \exists Y (\beta(\langle X, U, Y \rangle) \wedge \neg \gamma(\langle Y, V, Z \rangle)) \quad (15)$$

Схема прямой однозначности.

$$\exists U \exists V \exists Y (\lambda(\langle \beta, X, \{U\}, Y \rangle) \wedge \neg \lambda(\langle \gamma, Y, \{V\}, Z \rangle)) \quad (16)$$

Схема обратной однозначности.

$$\exists U \exists V \exists Y (\lambda(\langle \gamma, Y, \{V\}, Z \rangle) \wedge \neg \lambda(\langle \beta, X, \{U\}, Y \rangle)) \quad (17)$$

Здесь U, V, X, Y, Z являются множествами элементов: X – определяющий набор понятий; Z – определяемый набор понятий; U, V, Y – элементы окрестности X и Z ; α, β и γ – предикативные выражения определяющих и определяемых свойств.

Для работы с исходными текстами онтологий фрагментов баз знаний была разработана модель гипертекстового представления знаний, которая ориентирована на массовую разработку фрагментов баз знаний. Были реализованы соответствующие программные средства. В соответствии с технологией решение задачи интеграции сводится к следующей схеме: 1) данные и знания представляются на некотором языке, удобном для

диалога с пользователем или взаимодействия с внешней средой, например на языке SCn в гипертекстовой модели представления знаний (рисунок 11); 2) затем они транслируются на язык модели унифицированного представления знаний (рисунок 12), где осуществляется их интеграции в соответствии с алгоритмом поиска решений задачи интеграции знаний; 3) при поиске решения на основе анализа сходства и различия формируется сопоставление элементов интегрируемых фрагментов семантической сети, выявляются отображения и соответствия слияния на основе чего формируется результат решения задачи интеграции знаний.

Рассмотрим решение задачи интеграции на примере анализа решения планиметрической задачи на построение треугольника. В этой задаче неустранимая неполнота знаний связана с естественным взаимодействием субъектов в процессе обучения. Условие задачи приведено на рисунке 10. Для ввода в базу знаний это условие транслируется на язык SCn (рисунок 13). Затем осуществляется трансляция на язык модели унифицированного семантического представления знаний (рисунок 16).

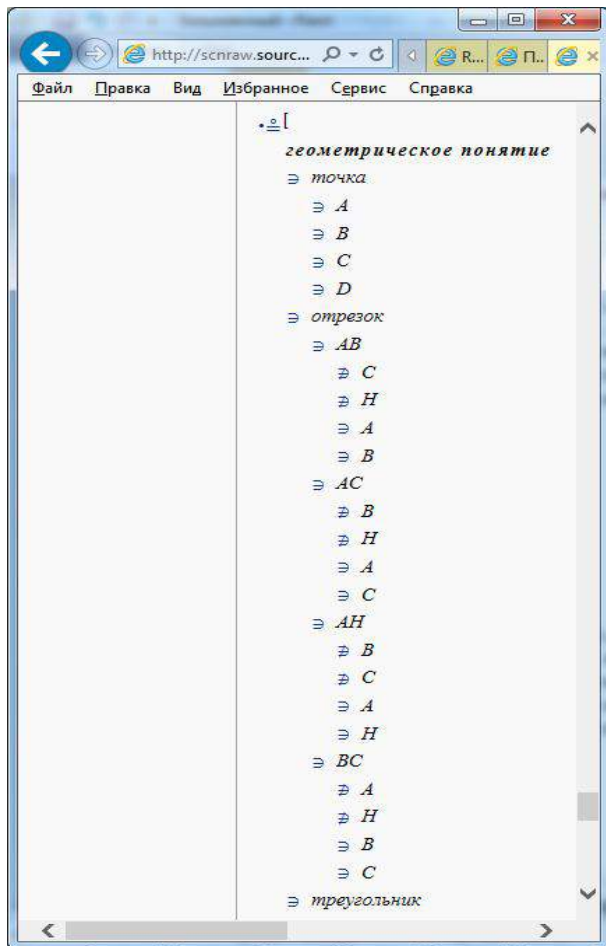


Рисунок 11 – Гипертекстовое представление знаний

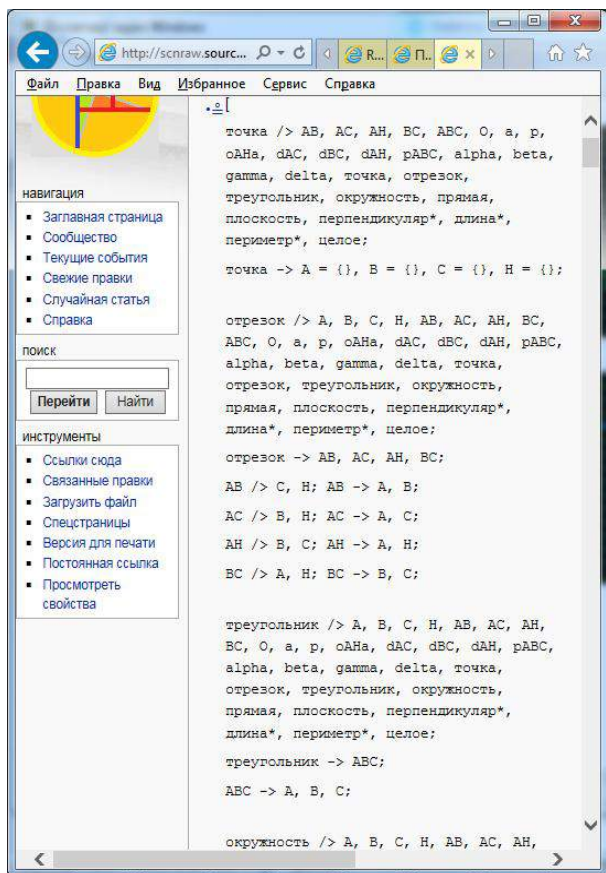


Рисунок 12 – Представление на языке SCs

Пример с треугольником

•.⌈	
<i>геометрическое понятие</i>	⊃ <i>перпендикуляр*</i>
⊃ <i>точка</i>	⊃ <i>oAHa</i>
⊃ <i>A</i>	⊃ <i>AH</i>
⊃ <i>B</i>	⊃ <i>a</i>
⊃ <i>C</i>	⊃ <i>длина*</i>
⊃ <i>D</i>	⊃ <i>dAC</i>
⊃ <i>отрезок</i>	⊃ <i>AC</i>
⊃ <i>AB</i>	⊃ <i>alpha</i>
⊃ <i>C</i>	⊃ <i>dBC</i>
⊃ <i>H</i>	⊃ <i>BC</i>
⊃ <i>A</i>	⊃ <i>beta</i>
⊃ <i>B</i>	⊃ <i>dAH</i>
⊃ <i>AC</i>	⊃ <i>AH</i>
⊃ <i>B</i>	⊃ <i>delta</i>
⊃ <i>H</i>	⊃ <i>периметр*</i>
⊃ <i>A</i>	⊃ <i>pABC</i>
⊃ <i>C</i>	⊃ <i>ABC</i>
⊃ <i>AH</i>	⊃ <i>gamma</i>
⊃ <i>B</i>	⊃ <i>целое</i>
⊃ <i>C</i>	⊃ <i>alpha</i>
⊃ <i>A</i>	⊃ <i>beta</i>
⊃ <i>H</i>	⊃ <i>gamma</i>
⊃ <i>BC</i>	⊃ <i>delta</i>
⊃ <i>A</i>	⊃ <i>A</i>
⊃ <i>H</i>	⊃ <i>B</i>
⊃ <i>B</i>	⊃ <i>C</i>
⊃ <i>C</i>	⊃ <i>D</i>
⊃ <i>треугольник</i>	⊃ <i>AB</i>
⊃ <i>ABC</i>	⊃ <i>AC</i>
⊃ <i>A</i>	⊃ <i>AH</i>
⊃ <i>B</i>	⊃ <i>BC</i>
⊃ <i>C</i>	⊃ <i>ABC</i>
⊃ <i>окружность</i>	⊃ <i>O</i>
⊃ <i>O</i>	⊃ <i>a</i>
⊃ <i>A</i>	⊃ <i>p</i>
⊃ <i>B</i>	⊃ <i>oAHa</i>
⊃ <i>C</i>	⊃ <i>dAC</i>
⊃ <i>H</i>	⊃ <i>dBC</i>
⊃ <i>прямая</i>	⊃ <i>dAH</i>
⊃ <i>a</i>	⊃ <i>pABC</i>
⊃ <i>A</i>	⊃ <i>точка</i>
⊃ <i>B</i>	⊃ <i>отрезок</i>
⊃ <i>C</i>	⊃ <i>треугольник</i>
⊃ <i>H</i>	⊃ <i>окружность</i>
⊃ <i>плоскость</i>	⊃ <i>прямая</i>
⊃ <i>p</i>	⊃ <i>плоскость</i>
⊃ <i>A</i>	⊃ <i>перпендикуляр*</i>
⊃ <i>B</i>	⊃ <i>длина*</i>
⊃ <i>C</i>	⊃ <i>периметр*</i>
⊃ <i>H</i>	⊃ <i>целое</i>

Рисунок 13 – Условие задачи в гипертекстовом представлении

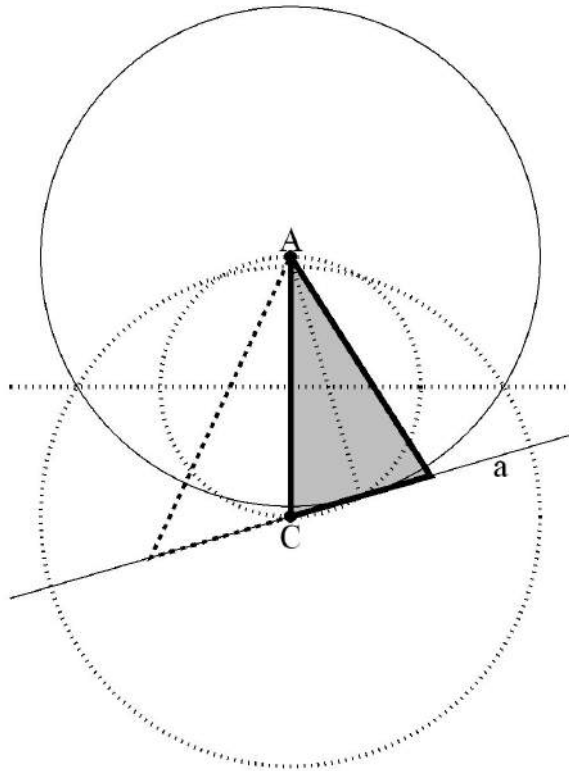


Рисунок 14 – Два треугольника с вершиной A и с равными основаниями на одной касательной a

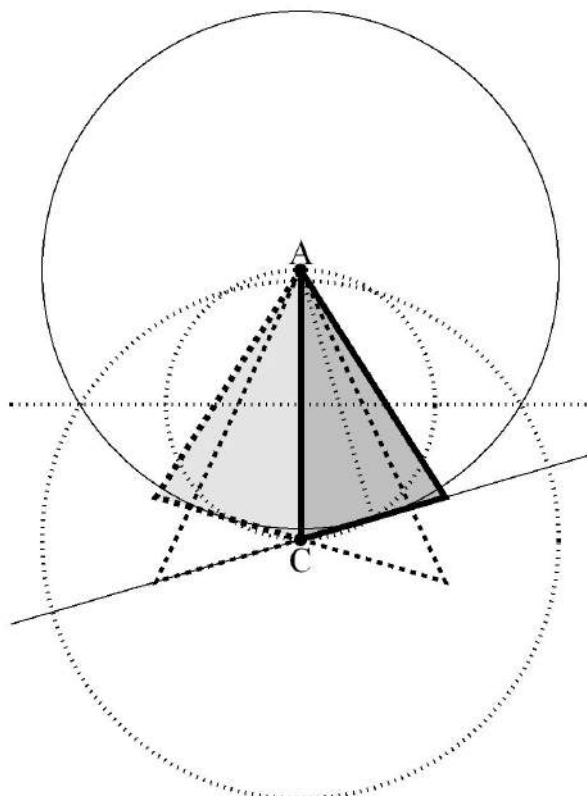


Рисунок 15 – Четыре треугольника с основаниями равной длины на двух касательных к окружности и вершиной в точке A

точка /> AB, AC, AH, BC, ABC, O, a, p, oAHa, dAC, dBC, dAH, pABC, alpha, beta, gamma, delta, точка, отрезок, треугольник, окружность, прямая, плоскость, перпендикуляр*, длина*, периметр*, целое;

точка -> A = {}, B = {}, C = {}, H = {};

отрезок /> A, B, C, H, AB, AC, AH, BC, ABC, O, a, p, oAHa, dAC, dBC, dAH, pABC, alpha, beta, gamma, delta, точка, отрезок, треугольник, окружность, прямая, плоскость, перпендикуляр*, длина*, периметр*, целое;

отрезок -> AB, AC, AH, BC;

AB /> C, H; AB -> A, B;

AC /> B, H; AC -> A, C;

AH /> B, C; AH -> A, H;

BC /> A, H; BC -> B, C;

треугольник /> A, B, C, H, AB, AC, AH, BC, O, a, p, oAHa, dAC, dBC, dAH, pABC, alpha, beta, gamma, delta, точка, отрезок, треугольник, окружность, прямая, плоскость, перпендикуляр*, длина*, периметр*, целое;

треугольник -> ABC;

ABC -> A, B, C;

окружность /> A, B, C, H, AB, AC, AH,

Рисунок 16 – Результат трансляции условия задачи на язык SCs

В соответствии с алгоритмом поиска множества решений задачи интеграции вызывается процедура однозначного пополнения решения, которая применяет правила слияния по равенству основных идентификаторов и которая осуществляет вызов алгоритма сопоставления элементов интегрируемых фрагментов. В результате из исходных данных (рисунок 17) будет получено сопоставление (рисунок 18), которое способно обеспечить выявление различия всех сопоставляемых узлов семантической сети, тогда как существующие модели не могут обеспечить этого (рисунок 19). Далее, при решении задачи, на построение будет сформирована следующая семантическая сеть, описывающая четыре варианта построения треугольника (рисунки 15, 20). В силу неполноты информации, неизвестно какой именно вариант треугольника дан в условии. Процедура однозначного пополнения решения или процедура добавления полного решения (рисунок 7) в

результате анализа противоречий может отбросить два варианта, например, если была предположена возможность слияния с треугольниками этих двух вариантов на ранних этапах поиска (рисунки 14, 21, 22, 23). В результате останутся два варианта построения треугольника (рисунок 24). Два варианта гипотетического слияния треугольников и будут результатом решения соответствующей задачи

интеграции (рисунки 25, 26). При неоднократном повторении решения этой задачи на построения разные результаты построения могут быть интегрированы, т.е. сколько бы раз конкретно первый или второй треугольник не был бы построен их количество, и количество их обозначений не изменится. В памяти системы будет только два обозначения треугольников (рисунок 27).

	A	B	C	H	AB	AC	AH	BC	ABC	a	O	P	alpha=5600 beta=10000	gamma	delta=9600	dAC	dAH	dBC	pABC	oAHa	точка	отрезок	треугольник	прямая	окружность	плоскость	целое	длина*	периметр*	перпендикуляр*	геометрическое понятие
A	III				m	m	m		m	m	m										m										
B		III			m	m	m		m	m	m											m									
C			III		m	m	m		m	m	m											m									
H				III			m			m	m											m									
AB					III																	m									
AC						III										m						m									
AH							III										m					m									
BC								III										m				m									
ABC					U	U		U	III												m		m								
a								U		III											m		m								
O											III															m					
P								U	U	U	III															m					
alpha												III						m									m				
beta													#	III		m											m				
gamma															III				m								m				
delta													#	#	III	m											m				
dAC																III												m			
dAH																	III											m			
dBC																		III										m			
pABC																			III									m			
oAHa																					III										m
точка																	+	+	+	+	+	+	III								m
отрезок																	+	+	+	+	+	+	+	III							m
треугольник																	+	+	+	+	+	+	+	+	III						m
прямая																	+	+	+	+	+	+	+	+	+	III					m
окружность																	+	+	+	+	+	+	+	+	+	+	+	III			m
плоскость																	+	+	+	+	+	+	+	+	+	+	+	+	+	+	m
целое																	+	+	+	+	+	+	+	+	+	+	+	+	+	+	m
длина*																	+	+	+	+	+	+	+	+	+	+	+	+	+	+	m
периметр*																	+	+	+	+	+	+	+	+	+	+	+	+	+	+	m
перпендик*																	+	+	+	+	+	+	+	+	+	+	+	+	+	+	m
геометрич.																	+	+	+	+	+	+	+	+	+	+	+	+	+	+	III

Рисунок 17 – Исходное сопоставление обозначений в условии задачи

	A	B	C	H	AB	AC	AH	BC	ABC	a	O	p	alpha=5600	beta=10000	gamma	delta=9600	dAC	dAH	dBC	pABC	oAHa	точка	отрезок	треугольник	прямая	окружность	плоскость	целое	длина*	периметр*	перпендикуляр*	геометрическое понятие		
A	III				m	m	m	m	m	m	m	m					m	m	m	m	m	m	m	m	m	m	m	m	m	m	m	m		
B	#	III			m	m	m	m	m	m	m	m					m	m	m	m	m	m	m	m	m	m	m	m	m	m	m	m		
C	#	#	III		m	m	m	m	m	m	m	m					m	m	m	m	m	m	m	m	m	m	m	m	m	m	m	m		
H	#	#	#	III			m			m	m	m					m	m	m	m	m	m	m	m	m	m	m	m	m	m	m	m		
AB	#	#	#	#	III												m	m	m	m	m	m	m	m	m	m	m	m	m	m	m	m		
AC	#	#	#	#	□	III											m	m	m	m	m	m	m	m	m	m	m	m	m	m	m	m		
AH	#	#	#	#	#	#	III										m	m	m	m	m	m	m	m	m	m	m	m	m	m	m	m		
BC	#	#	#	#	□	□	⊂	III									m	m	m	m	m	m	m	m	m	m	m	m	m	m	m	m		
ABC	#	#	#	#	□	□	#	□	III								m	m	m	m	m	m	m	m	m	m	m	m	m	m	m	m		
a	#	#	#	#	□	□	□	□	#	III							m	m	m	m	m	m	m	m	m	m	m	m	m	m	m	m		
O	#	#	#	#	#	#	#	#	#	#	III						m	m	m	m	m	m	m	m	m	m	m	m	m	m	m	m		
p	#	#	#	#	□	□	□	□	□	□	□	III					m	m	m	m	m	m	m	m	m	m	m	m	m	m	m	m		
alpha	#	#	#	#	#	#	#	#	#	#	#	#	III				m	m	m	m	m	m	m	m	m	m	m	m	m	m	m	m		
beta	#	#	#	#	#	#	#	#	#	#	#	#	#	#	III		m	m	m	m	m	m	m	m	m	m	m	m	m	m	m	m		
gamma	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	III		m	m	m	m	m	m	m	m	m	m	m	m	m	m	m	m	
delta	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	III		m	m	m	m	m	m	m	m	m	m	m	m	m	m	m	
dAC	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	III		m	m	m	m	m	m	m	m	m	m	m	m	m	m	
dAH	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	III		m	m	m	m	m	m	m	m	m	m	m	m	m	
dBC	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	III		m	m	m	m	m	m	m	m	m	m	m	m	
pABC	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	III		m	m	m	m	m	m	m	m	m	m	m	m
oAHa	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	III		m	m	m	m	m	m	m	m	m	m	
точка	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#		
отрезок	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#		
треугольник	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#		
прямая	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#		
окружность	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#		
плоскость	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#		
целое	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#		
длина*	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#		
периметр*	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#		
перпендик.*	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#		
геометрич. понятие	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#	#		

Рисунок 18 – Результат сопоставления, полученный по алгоритму процедуры сопоставления онтологий, каждая пара обозначений различна

	A	B	C	H	AB	AC	AH	BC	ABC	a	O	p	alpha=5600	beta=10000	gamma	delta=9600	dAC	dAH	dBC	pABC	oAHa	точка	отрезок	треугольник	прямая	окружность	плоскость	целое	Длина*	периметр*	перпендикуляр*	геометрическое понятие		
A	III				m	m	m		m	m												m												
B		III			m	m	m	m	m	m													m											
C			III			m	m	m	m	m													m											
H				III			m			m	m												m											
AB					III																		m											
AC						III											m						m											
AH							III											m				m												
BC								III											m				m											
ABC					U	U	U	U	III													m			m									
a								U		III												m		m										
O											III															m								
p					U	U	U	U	U	U	U	III															m							
alpha													III						m									m						
beta													#	III			m											m						
gamma															III						m							m						
delta													#	#	III			m										m						
dAC																		III													m			
dAH																			III													m		
dBC																				III												m		
pABC																					III											m		
oAHa																						III										m		
точка																							III										m	
отрезок																								U	III								m	
треугольник																								U	U	III							m	
прямая																								U	U	U	III						m	
окружность																								U	U	U	U	III					m	
плоскость																								U	U	U	U	U	III				m	
целое																								U	U	U	U	U	U	III			m	
Длина*																								U	U	U	U	U	U	U	III		m	
периметр*																								U	U	U	U	U	U	U	U	III		m
перпендик.*																								U	U	U	U	U	U	U	U	U	III	m
геометрич.																								U	U	U	U	U	U	U	U	U	U	III

Рисунок 19 – Результат сопоставления традиционными подходами

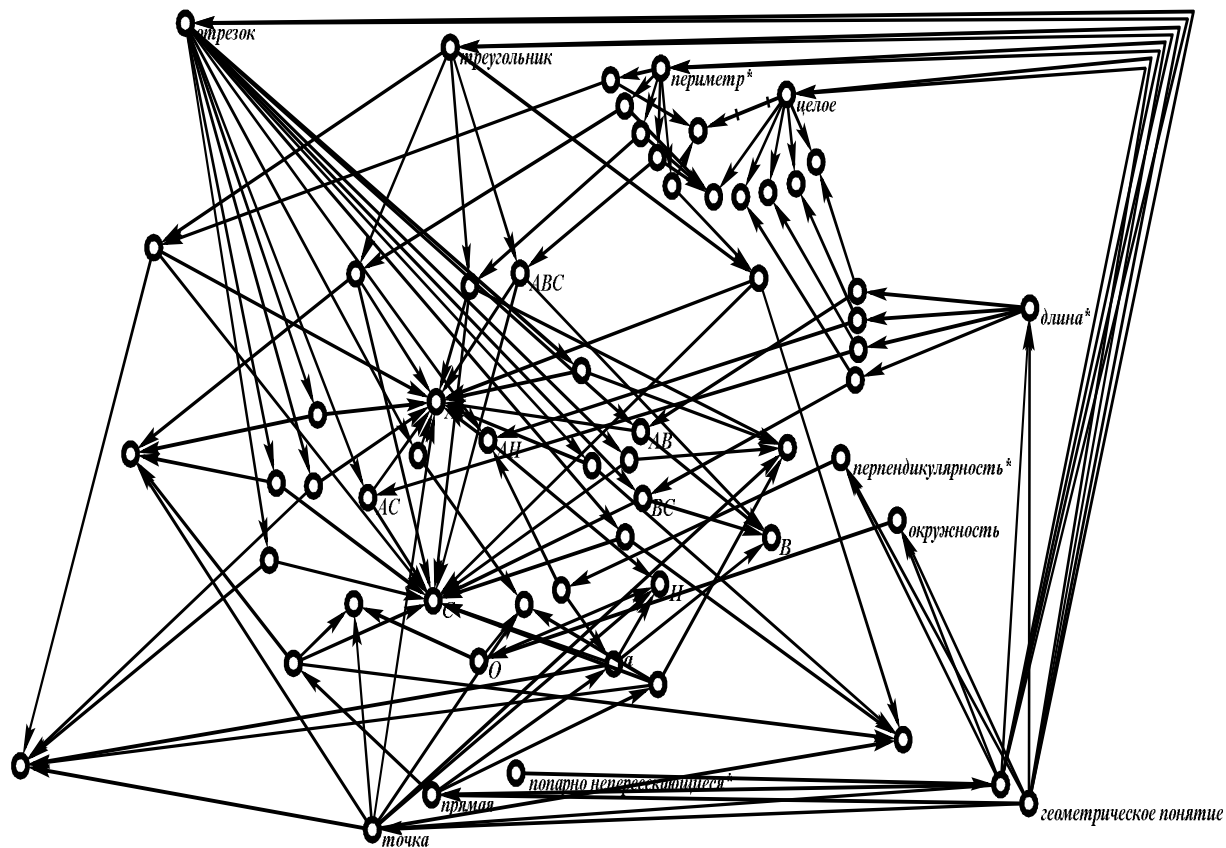


Рисунок 20 – Четыре варианта построения треугольника

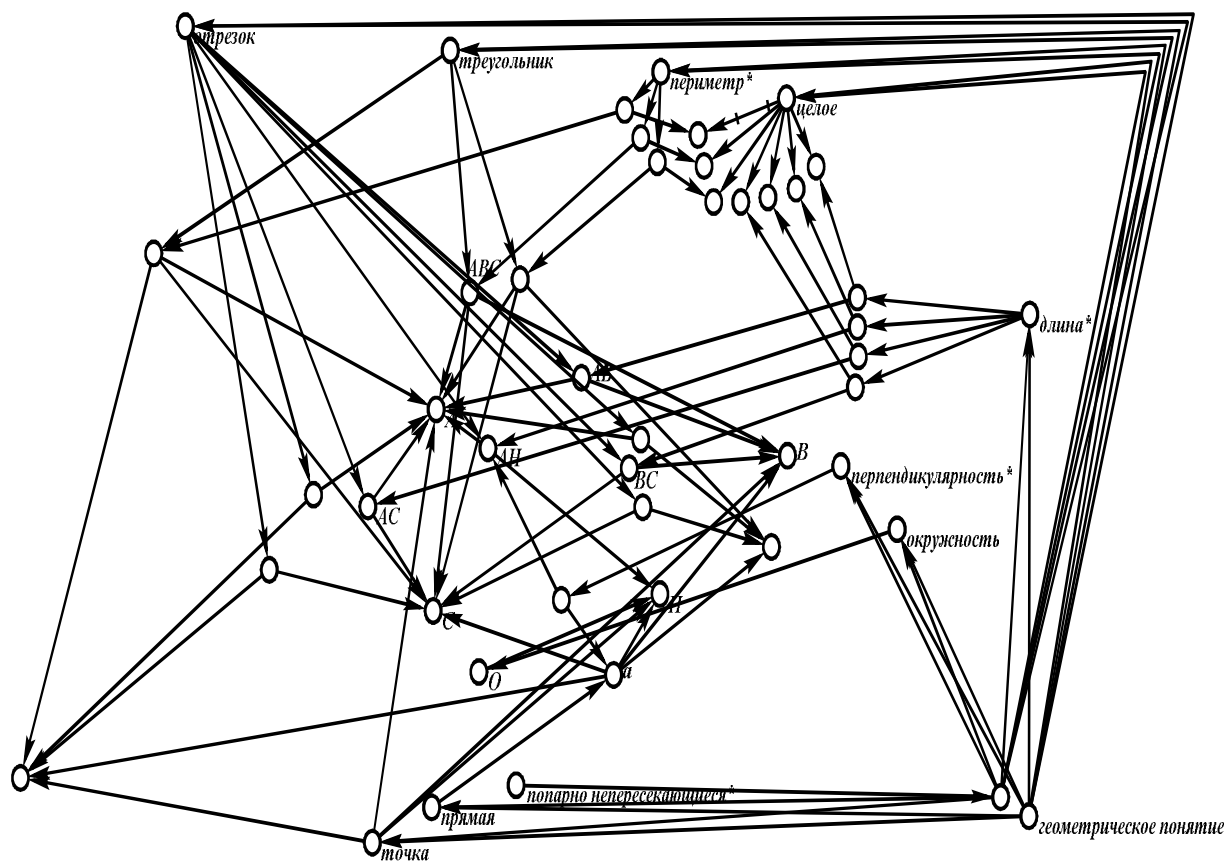


Рисунок 21 – Два варианта построения треугольника на одной касательной

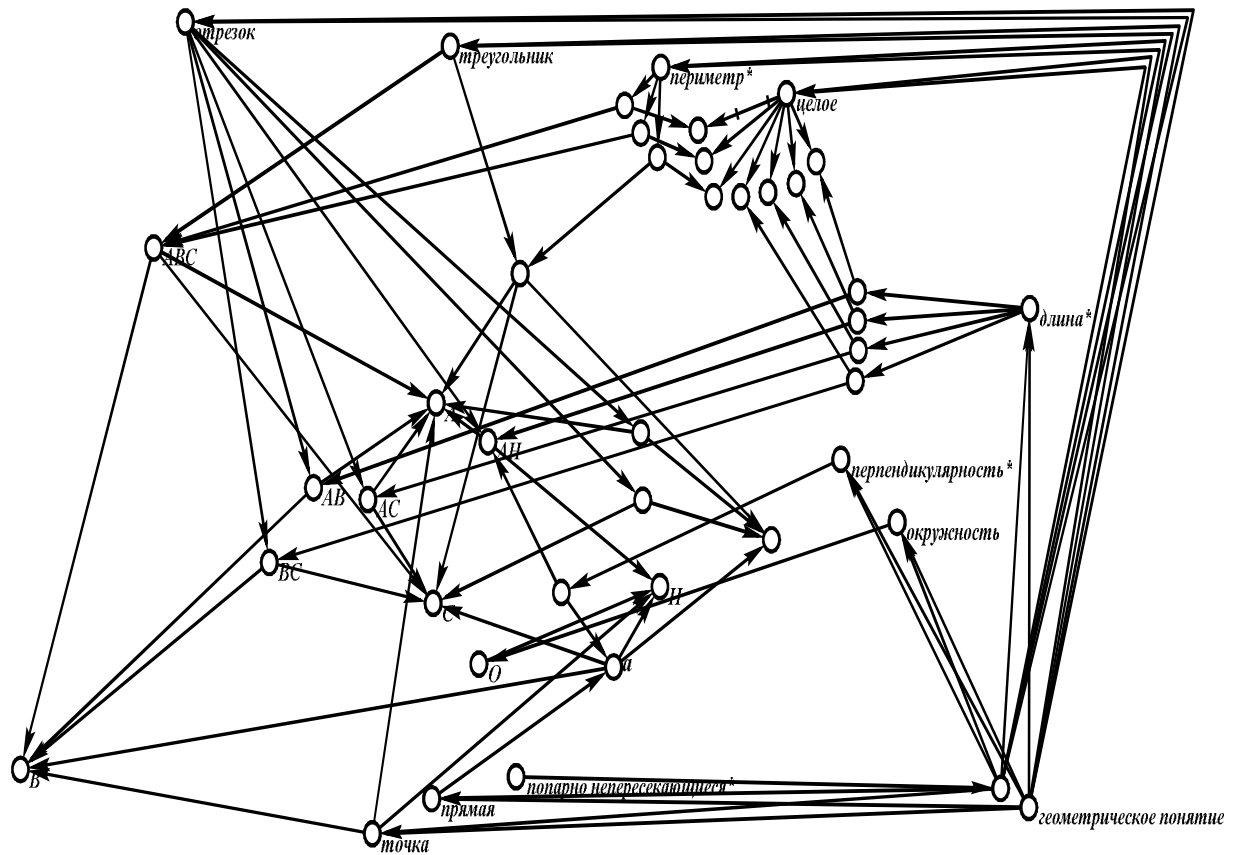


Рисунок 22 – Предполагаемое слияние с неравносторонним треугольником

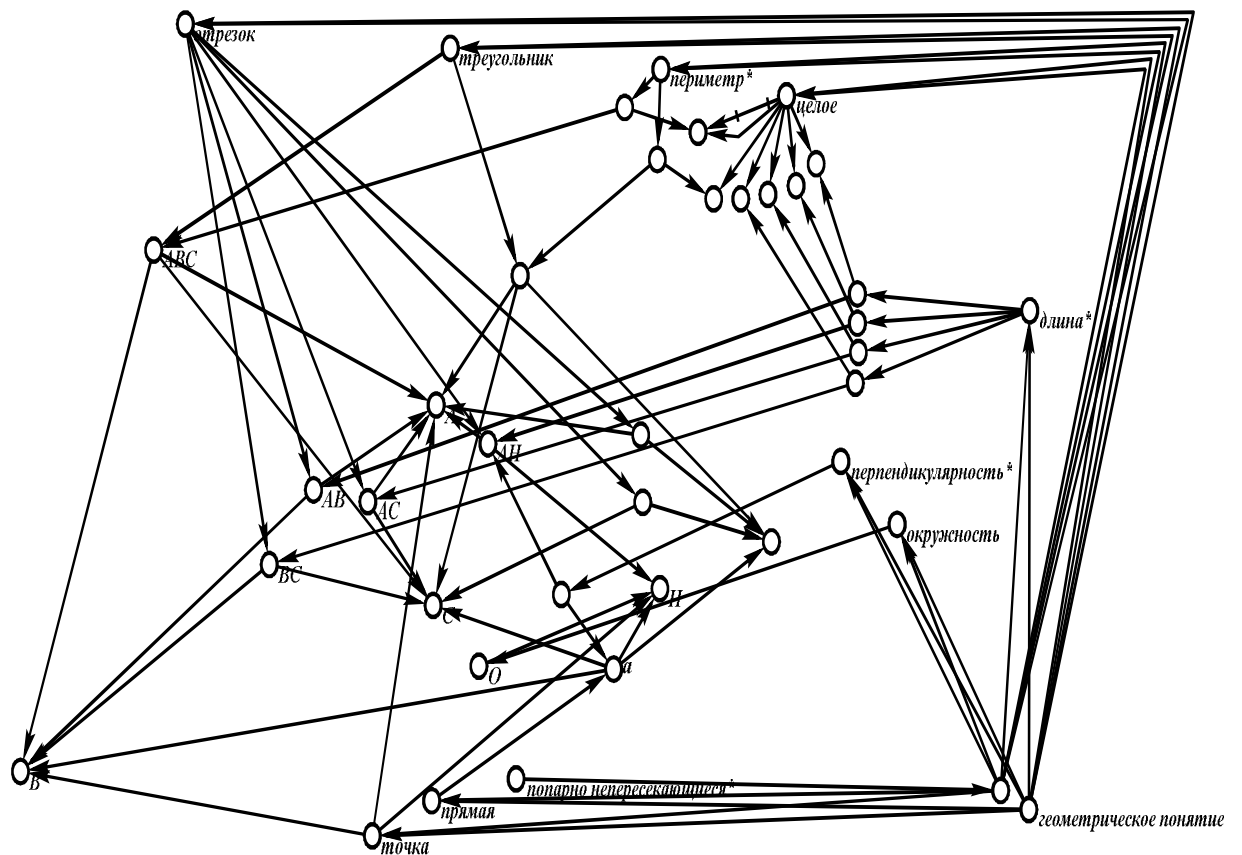


Рисунок 23 – Противоречие свойств периметра, полученное в результате слияния с неравносторонним треугольником

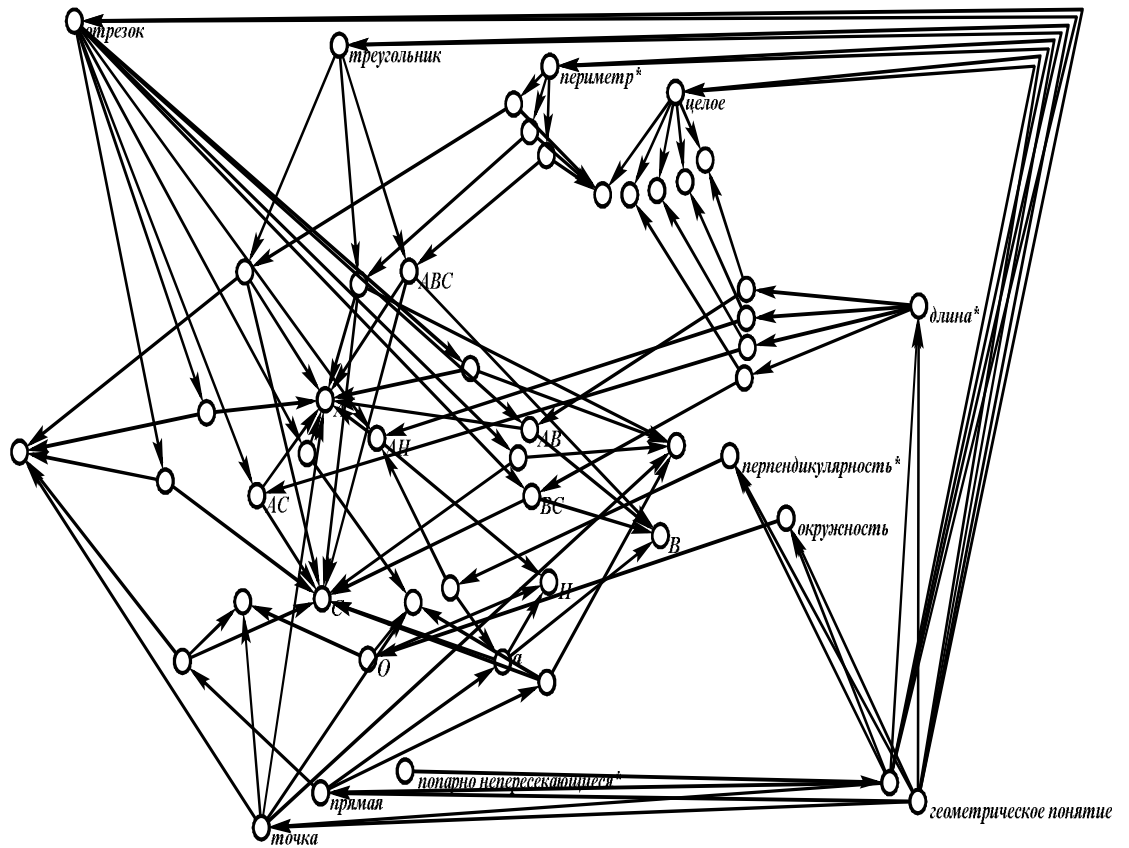


Рисунок 24 – Два допустимых варианта слияния

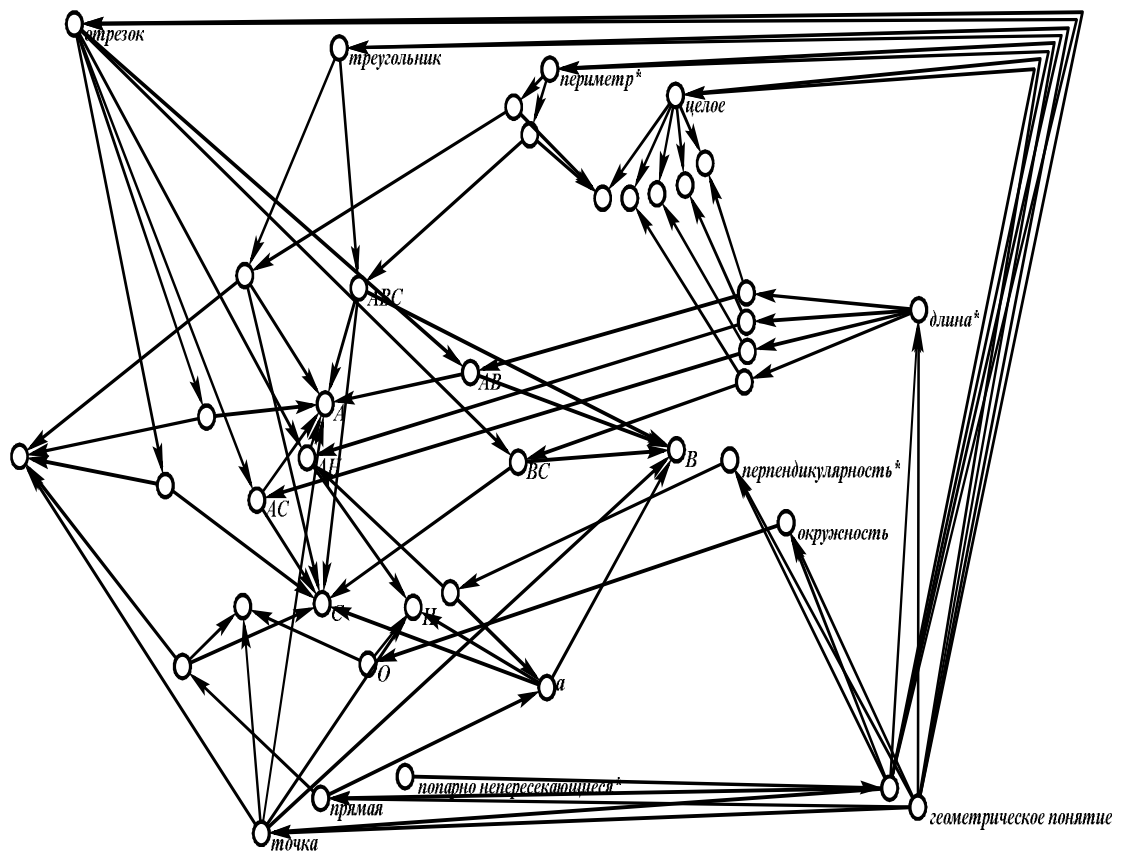


Рисунок 25 – Слияние с одним из вариантов

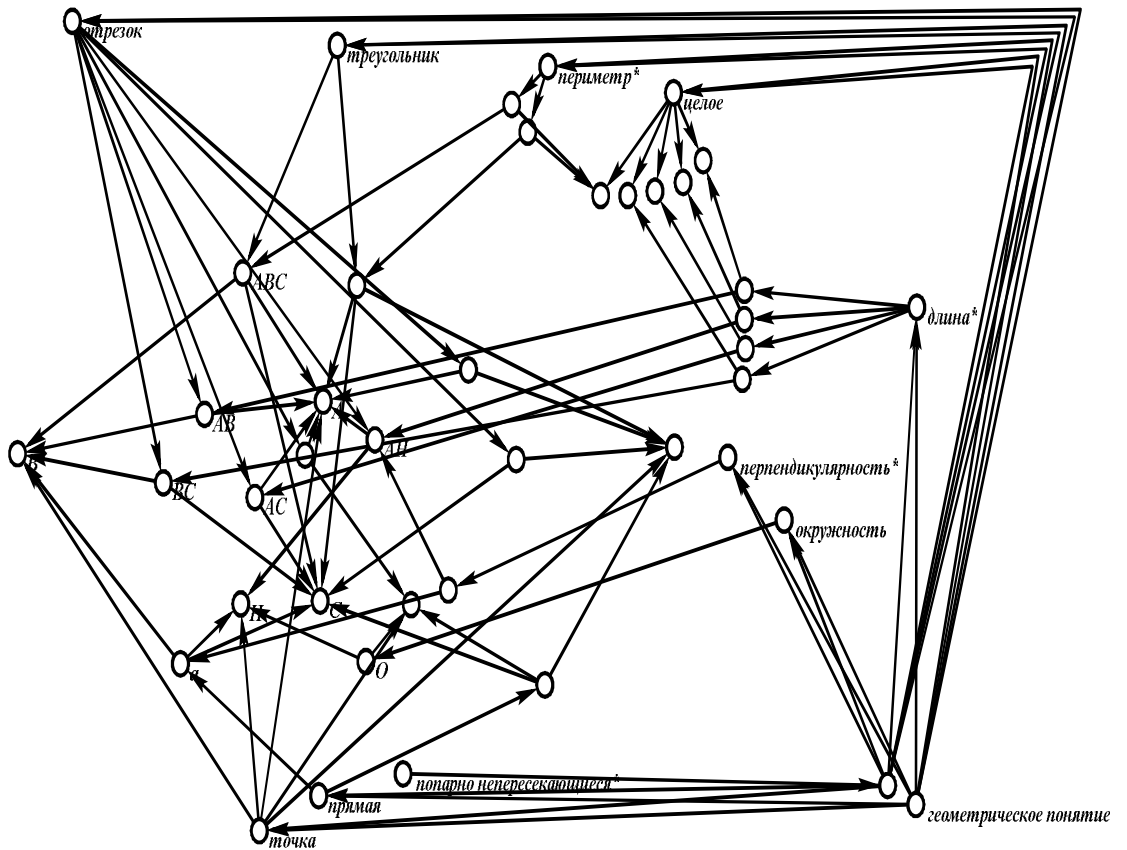


Рисунок 26 – Слияние со вторым из вариантов

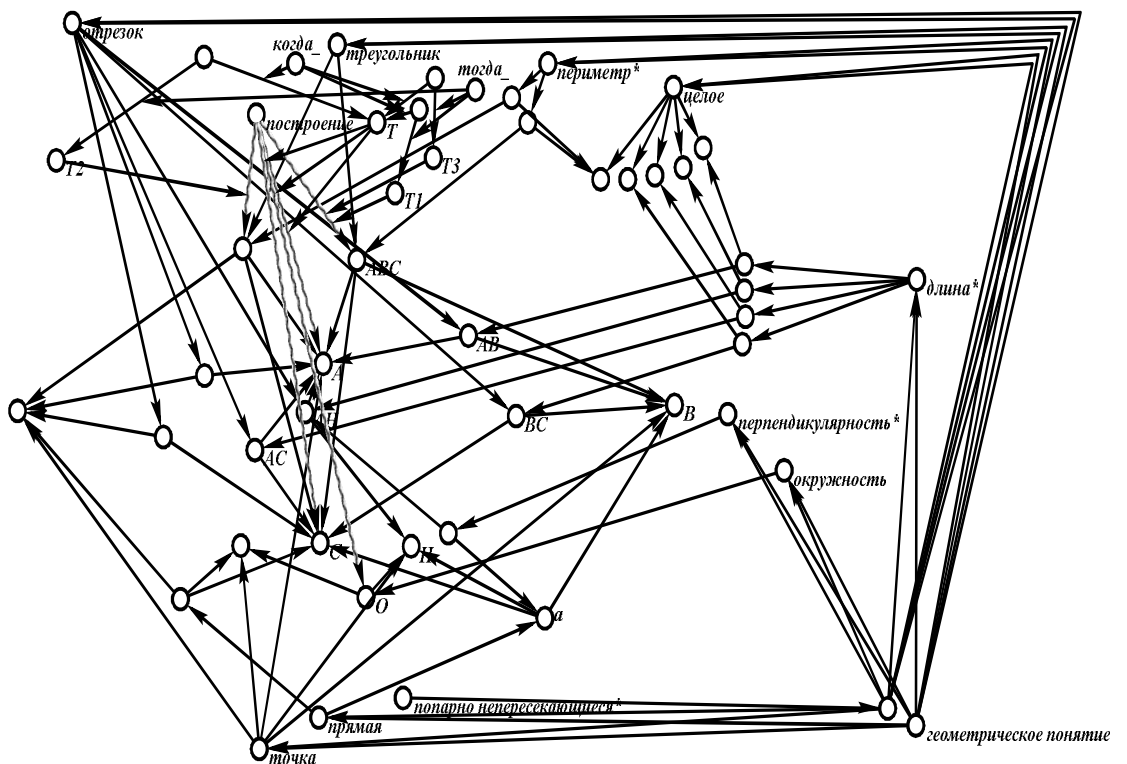


Рисунок 27 – Слияние трёх построений двух равных треугольников

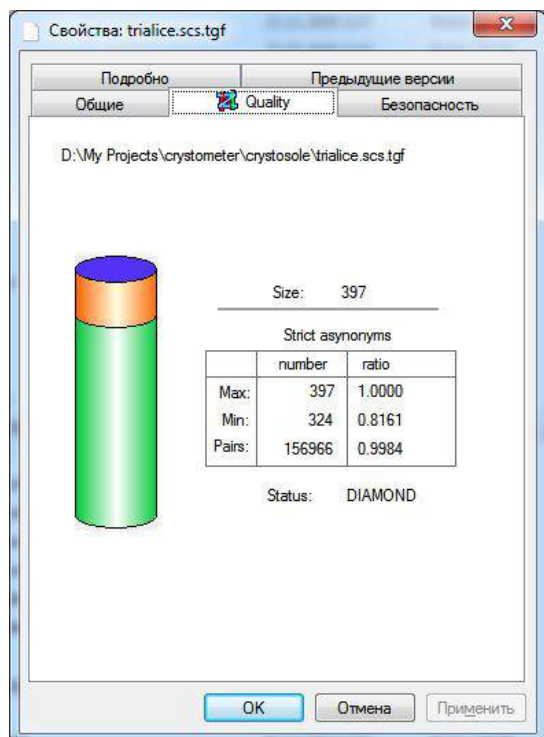
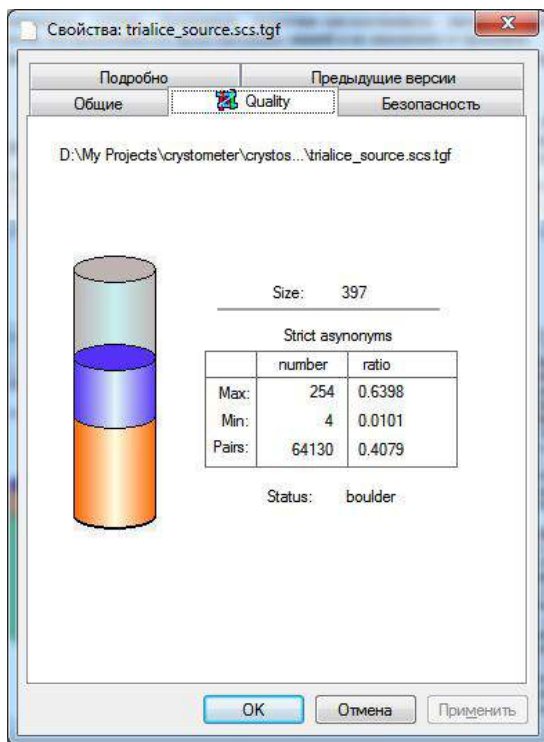


Рисунок 28 – Число пар различных обозначений

Алгоритм вычисления множества пар различающихся обозначений позволяет оценить качество фрагментов баз знаний. Так для фрагмента, который соответствует рисунку 17, количество пар различных обозначений составляет 64130, а для фрагмента на рисунке 18 – 156966 (рисунок 28).

ЗАКЛЮЧЕНИЕ

Рассмотренные модели и алгоритмы применяются для решения задач интеграции знаний в рамках технологии OSTIS [OSTIS, 2014]. Сравнение

результатов работы алгоритмов с системой ASMOV [Jean-Mary et al., 2007] показало преимущество предлагаемых решений от 1,45 раз по показателю точности и от 1,54 раз по показателю полноты. Реализация разработанных алгоритмов проведена средствами C/C++ и средствами процедурного языка обработки знаний SCP (Semantic Code Programming).

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

[Гаврилова и др., 2000] Гаврилова, Т.А. Базы знаний интеллектуальных систем / Т. А. Гаврилова, В.Ф. Хорошевский. – СПб.: Питер, 2000.

[Голенков и др., 2001] Голенков, В.В. Представление и обработка знаний в графодинамических ассоциативных машинах / В. В. Голенков[и др.]. – Мн.: БГУИР, 2001.

[Гулякина и др., 2004] Гулякина Н.А. Ивашенко В.П. Интеграция знаний в информационных системах. / Н.А. Гулякина, В.П. Ивашенко // Доклады БГУИР. – 2004. – №6. – С. 113-119.

[Ивашенко, 2003] Представление нейронных сетей и систем продукции в однородных семантических сетях. / В.П. Ивашенко // Известия Белорусской инженерной академии. – 2003. – №1(15)/1. – с.184-188.

[Ивашенко, 2004] Применение однородных семантических сетей для представления знаний о нестационарных предметных областях. / В.П. Ивашенко // Известия Белорусской инженерной академии. – 2004. – №1(17)/3. – с.77-80.

[Ивашенко, 2009а] Ивашенко В.П. Семантические модели баз знаний / В.П. Ивашенко Информационные системы и технологии (IST'2009): материалы V Междунар. конф.-форума в 2-х ч. Ч.2 – Минск: А.Н.Вараксин, 2009.- с.125-128.

[Ивашенко, 2009б] Ивашенко В.П. Алгоритмы верификации и интеграции баз знаний. Вестник Брестского государственного технического университета, БрГТУ, 2009, №5.

[Ивашенко, 2011а] Ивашенко В.П. Семантическая технология компонентного проектирования баз знаний. Материалы Международной научн.-техн. Конференции OSTIS,2011:Минск, Республика Беларусь, БГУИР 10-12 февраля 2011.

[Ивашенко, 2011б] Ивашенко В.П. Алгоритмы операций отладки и интеграции баз знаний. Дистанционное обучение – образовательная среда XXI века: материалы VII Международной научн.-метод. конференции (Минск, 1-2 декабря 2011г.). – Минск: БГУИР, 2011, сс.227-229.

[Ивашенко, 2012] Ивашенко В.П. Семантическая модели интеграции и отладки баз знаний. Материалы Международной научн.-техн. Конференции OSTIS,2012:Минск, Республика Беларусь, БГУИР 16-18 февраля 2012.

[Колб, 2012] Колб, Д. Г. Web-ориентированная реализация семантических моделей интеллектуальных систем / Д. Г. Колб // Материалы международной научно-технической конференции «Открытые семантические технологии проектирования интеллектуальных систем» – Минск, 2012

[Кофман, 1982] Кофман А. Введение в теорию нечетких множеств. М.: Радио и связь, 1982. — 432 с.

[Кудрявцев, 2008] Кудрявцев Д. В. Практические методы отображения и интеграции онтологий. Семинар Знания и онтологии *Elsewhere*, КИИ-2008, Дубна, 2008.

[Нариньяни, 2000] Нариньяни А. С. НЕ-факторы: неточность и недоопределенность — различие и взаимосвязь // Изв. РАН. Сер. Теория и системы управления. 2000. № 5. С. 44-56.

[Тэрано, 1993] Прикладные нечеткие системы: Пер. с япон. / К. Асаи, Д. Ватада, С. Иван и др.; Под ред. Т. Терано, К. Асаи, М. Сугэно. – М.: Мир, 1993.

[Хорошевский, 2008] Хорошевский, В.Ф. Пространства знаний в сети Интернет и Semantic Web (Часть 1) / В. Ф. Хорошевский // Искусственный интеллект и принятие решений. – 2008. - № 1. - С.80-97.

[Allen, 1991] Allen, J.F.: Time and time again: the many ways to represent time. International Journal of Intelligent Systems 6 (1991) 341-355.

[Allen, 1994] Allen, J. F. Ferguson, G. Actions and Events in Interval Temporal Logic. J. Log. Comput., 1994, 4, 531-579.

[Aumuller et al., 2005] D. Aumuller, H. Do, S. Massmann, E. Rahm. 2005. Schema and ontology matching with COMA++. Proc. of

the 2005 International Conference on Management of Data, pp. 906-908.

[**Bouquet, 2003**] P. Bouquet, L.Serafini and S. Zanobini, M. Benerecetti. An algorithm for semantic coordination. Proc. of Semantic Integration Workshop, colocated with the 2nd Int'l Semantic Web Conference (ISWC2003), 20-23 October 2003, Sanibel Island, Florida, US.

[**DAML, 2006**] DAML.org: The DARPA Agent Markup Language Homepage, Электронный ресурс. Режим доступа: <http://www.daml.org/>.

[**Doan and Halevy, 2005**] A. H. Doan, A. Y. Halevy. Semantic integration research in the database community: A brief survey. AI magazine, 26(1), 2005

[**CLIPS, 1991**] CLIPS Version 5.1 User's Guide, NASA Lyndon B. Johnson Space Center, Software Technology Branch, Houston, TX, 1991.

[**CoFI:CASL-Summary, 2004**] CoFI Language Design Group. Casl summary. In Casl Reference Manual, LNCS 2960 (IFIP Series), part I. Springer, 2004.

[**CycL, 2002**] The Syntax of CycL. Электронный ресурс. Режим доступа: <http://www.cyc.com/cycdoc/ref/cycl-syntax.html>.

[**Ganter, 1999**] B. Ganter, R. Wille: Formal Concept Analysis – Mathematical Foundations. Springer, Heidelberg 1999.

[**Gangemi et al., 1996**] A. Gangemi, G. Steve and F. Giacomelli, 1996. "ONIONS: An Ontological Methodology for Taxonomic Knowledge Integration", ECAI-96 Workshop on Ontological Engineering, Budapest, August 13th.

[**Genesereth and Fikes, 1992**] Michael R. Genesereth and Richard E. Fikes. Knowledge interchange format version 3.0 reference manual. Logic Group First Draft January 1992 Report Logic-92-1 Current Version June 1992.

[**Giunchiglia et al., 2006**] F. Giunchiglia, M. Marchese and I. Zaihrayeu (2006). Encoding classifications into lightweight ontologies. University of Trento Technical Report # DIT-06-016, March 2006.

[**Jean-Mary et al., 2007**] Jean-Mary Y., Kabuka, M. ASMOV: Ontology Alignment with Semantic Validation. Joint SWDB-ODDIS Workshop, September 2007, Vienna, Austria, 15-20.

[**IEEE, 2006**] IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, Automatic Fuzzy Ontology Generation for Semantic Web, VOL. 18, NO. 6, JUNE 2006.

[**IDEF5, 1994**] Information Integration for Concurrent Engineering (IICE). IDEF5 Method Report. - Knowledge Based Systems, Inc. , 1408 University Drive East College Station, Texas, USA. September 21, 1994.

[**ISO13250**] ISO/IEC 13250:2003 Topic maps – representation and interchange of knowledge, with an emphasis on the findability of information.

[**ISO24707**] ISO/IEC 24707:2007 – Information technology – Common Logic (CL): a framework for a family of logic-based languages.

[**Maldavan, 2001**] J. Madhavan, P. A. Bernstein, E. Rahm. Generic Schema Matching with Cupid. In. Proc. of the 27th Conference on Very Large Databases, 2001.

[**Maltese et al., 2010**] V. Maltese, F. Giunchiglia, A. Autayeu: Save up to 99% of your time in mapping validation In Proceedings of ODBASE, 2010.

[**Martin Ph., 2002**] Martin Ph. (2002). Knowledge representation in CGLF, CGIF, KIF, Frame-CG and Formalized-English. Proceedings of ICCS 2002, 10th International Conference on Conceptual Structures (Springer, LNAI 2393, pp. 77-91), Borovets, Bulgaria, July 15-19, 2002.

[**McGuinness et al., 2000**] D. L. McGuinness, R. Fikes, J. Rice, and S. Wilder. The chimaera ontology environment. In Proceedings of AAAI, pages 1123-1124, 2000.

[**Mitra, 2001**] Prasenjit Mitra, Gio Wiederhold and Stefan Decker: A Scalable Framework for Interoperation of Information Sources. 1st International Semantic Web Working Symposium (SWWS '01), Stanford University, Stanford, CA, July 29-Aug 1, 2001, Jul.

[**Michael Kifer et al., 1995**] Michael Kifer, Georg Lausen, James Wu: Logical Foundations of Object-Oriented and Frame-Based Languages. Journal of ACM 42(4): 741-843 May (1995).

[**Mustafa Jarrar et al., 2008**] Mustafa Jarrar and Robert Meersman: "Ontology Engineering -The DOGMA Approach". Book Chapter (Chapter 3). In Advances in Web Semantics I. Volume LNCS 4891, Springer. 2008.

[**Nagy et al., 2010**] Miklos Nagy and Maria Vargas-Vera. Towards an Automatic Semantic Data Integration: Multi-Agent Framework Approach. Sematic Web. Gang Wu (ed), Chapter 7, pp.

107-134; In-Tech Education and Publishing KG; 2010, ISBN 978-953-7619-54-1.

[**Novak et al., 2008**] Joseph D. Novak & Alberto J. Canas, The Theory Underlying Concept Maps and How to Construct and Use Them. Florida Institute for Human and Machine Cognition. Pensacola FL, 32502. Technical Report IHMC CmapTools 2006-01 Rev 2008-01.

[**OSTIS, 2014**] Открытая семантическая технология проектирования интеллектуальных систем [Электронный ресурс]. – 2014. - Режим доступа: <http://ostis.net>. – Дата доступа: 2.11.2014

[**Smith et al., 2007**] Smith, B.; Ashburner, M.; Rosse, C.; Bard, J.; Bug, W.; Ceusters, W.; Goldberg, L. J.; Eilbeck, K. et al. (2007). "The OBO Foundry: Coordinated evolution of ontologies to support biomedical data integration". Nature Biotechnology 25 (11): 1251-1255. doi:10.1038/nbt1346.

[**Sowa et al., 2008**] Sowa, J. Conceptual Graphs/ John F. Sowa, F. van Harmelen, V. Lifschitz, B. Porter// eds., Handbook of Knowledge Representation, Elsevier, 2008, pp. 213-237.

[**Stumme, 2001**] Gerd Stumme and Alexander Maedche. FCA-merge: bottom-up merging of ontologies. In In Proceedings of 17th IJCAI, pages 225{230, Seattle (WA), USA, 2001.

[**W3C:DAML+OIL, 2001**] DAML+OIL (March 2001) Reference Description: W3C Note 18 December 2001. Dan Connolly, Frank van Harmelen, Ian Horrocks, Deborah L. McGuinness, Peter F. Patel-Schneider, Lynn Andrea Stein and Lucent Technologies, Inc. Электронный ресурс. Режим доступа: <http://www.w3.org/TR/daml+oil-reference>.

[**W3C:RDFS, 2004**] RDF Vocabulary Description Language 1.0: RDF Schema: W3C Recommendation 10 February 2004 Электронный ресурс. / eds. Dan Brickley, R.V. Guha. Режим доступа: <http://www.w3.org/TR/rdf-schema/>, свободный.

[**W3C:RIF, 2010**] RIF Overview: W3C Working Group Note 22 June 2010. Электронный ресурс. / eds. Michael Kifer, Harold Boley. Режим доступа: <http://www.w3.org/TR/2010/NOTE-rif-overview-20100622/>.

[**W3C:OWL, 2004**] OWL Web Ontology Language. Overview: W3C Recommendation 10 February 2004 Электронный ресурс. /eds. Deborah L. McGuinness, Frank van Harmelen. Режим доступа: <http://www.w3.org/TR/owl-features/>.

[**W3C:OWL2, 2009**] OWL 2 Web Ontology Language Document Overview: W3C Recommendation 27 October 2009. Электронный ресурс. / eds. W3C OWL Working Group. Режим доступа: <http://www.w3.org/TR/owl2-overview/>.

[**W3C:SPARQL, 2008**] SPARQL Query Language for RDF: W3C Recommendation 15 January 2008. Электронный ресурс. / eds. Eric Prud'hommeaux, Andy Seaborne. Режим доступа: <http://www.w3.org/TR/rdf-sparql-query/>.

[**XTM, 2001**] XML Topic Maps (XTM) 1.0: TopicMaps.Org Specification. Members of the TopicMaps.Org Authoring Group. Электронный ресурс. / eds. Steve Pepper, Graham Moore. Режим доступа: <http://www.topicmaps.org/xtm/>.

[**Van Renssen, 2005**] Van Renssen, Andries (2005). Gellish: A Generic Extensible Ontological Language. Delft University Press. ISBN90-407-2597-4. <http://repository.tudelft.nl/view/ir/uuid%3Ade26132b-6f03-41b9-b882-c74b7e34a07d/>.

MODELS AND ALGORITHMS OF INTEGRATION OF KNOWLEDGE BASED ON HOMOGENEOUS SEMANTIC NETWORKS

Ivashenko V.P.

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

ivashenko@bsuir.by

The work concerns models and algorithms of integration of knowledge with unified representation in the form of homogeneous semantic networks. These networks have model (set-theoretic) semantic that describes incompleteness, fuzziness, uncertainty of represented knowledge in dynamic conditions and changes over time.



OSTIS-2015

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822:514+004.272:43+004.272:32

ПРЕДСТАВЛЕНИЕ СЕМАНТИЧЕСКИХ СЕТЕЙ И АЛГОРИТМЫ ИХ ОРГАНИЗАЦИИ И СЕМАНТИЧЕСКОЙ ОБРАБОТКИ НА ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМАХ С МАССОВЫМ ПАРАЛЛЕЛИЗМОМ

Ивашенко В.П. *, Вереник Н.Л. *, Гирель А.И. *, Сейткулов Е.Н. **, Татур М.М. *

* *Белорусский государственный университет информатики и радиоэлектроники,
г. Минск, Республика Беларусь*

ivashenko@bsuir.by

nick.verenik@gmail.com

tatur@i-proc.com

** *Евразийский национальный университет имени Л.Н. Гумилева,
г. Астана, Казахстан*

seitkulov_y@enu.kz

В работе приводится описание способов и некоторых алгоритмов организации хранения и представления семантических сетей в архитектуре проблемно-ориентированного семантического процессора. Рассмотрена возможность применения разработанных алгоритмов для решения прикладных задач.

Ключевые слова: массовый параллелизм; семантическая сеть; семантическая обработка информации; каноническая разметка графа; интеграция знаний.

Введение

К системам искусственного интеллекта, или интеллектуальным системам, обычно относят системы, способные решать задачи на семантическом уровне, связанные с анализом и обработкой знаний в некоторой заданной предметной области. Примерами могут служить различные системы поддержки принятия решений, экспертные и поисковые системы, направленные на широкий спектр прикладных задач, таких как: управление взаимоотношениями с клиентами (CRM-системы), планирование ресурсов предприятий (ERP-системы), аналитический мониторинг контента, массовое обслуживание (контроль информационных потоков) в сетях и др. В настоящее время при построении сложных информационных систем акцент проблемы все больше смещается от физического аспекта, связанного с хранением и передачей данных, СУБД, к семантическому. В основе любой «интеллектуальной» модели обработки информации можно выделить ряд актуальных базовых задач, таких как представление знаний (например, в виде семантической сети), семантический анализ информации, ассоциативный поиск информации по

некоторому ключу, модификация семантической сети в соответствии с результатом обработки информации.

Под семантической сетью понимается графовая структура, вершины и дуги которой в соответствии с определенными правилами наделены некоторой смысловой нагрузкой. Аппарат семантических сетей обеспечивает не только визуализацию структуры информации, но, в первую очередь, позволяет разрабатывать формальные методы и алгоритмы анализа и модификации знаний. Характерной чертой решаемых задач является их высокая структурная и динамическая сложность, общее решение таких задач зачастую сводится к классу трансвычислительных: уже при относительно небольшом числе вершин графа (узлов семантической сети) время решения методом перебора превышает возможности любых теоретически мыслимых вычислительных систем. Однако с учетом специфики частных задач, а также в результате формулировки задачи на языке теории графов, позволяющих обеспечить высокую степень ассоциативности, можно получить более простые методы решения, часто практически реализуемые.

Информационный взрыв (в частности, переход к многоуровневой иерархической структуре знаний, а

также активное использование метаинформации) привел к резкому увеличению объемов хранимых и обрабатываемых знаний, значительно усложнил их структуру, и, следовательно, объем и структуру семантических сетей. Полученные в результате семантические сети обладают сложной нерегулярной структурой, а используемые алгоритмы обработки информации различаются от системы к системе, используя большое количество эвристик, построенных на особенностях конкретной прикладной системы. Все это обусловило проблему создания эффективной аппаратной платформы с параллельной архитектурой, ориентированной на данный класс задач [Guo et al., 2009], [Allemang et al., 2008], [Kitano et al., 1992], [Chung et al., 1995], [Голенков и др., 2012].

Известны попытки реализовать алгоритмы семантической обработки с применением серийных аппаратных платформ с параллельной архитектурой, таких как графические ускорители (GPU) [Brodkorb et al., 2013], [Navarro et al., 2014], кластеры и суперЭВМ [Каляев и др., 2012], [Kitano et al., 1992], [Chung et al., 1995], [Jan et al., 2012]. В случаях, когда использование GPU не ведёт к кардинальному повышению производительности по сравнению с персональными компьютерами, а использование мощных кластеров или суперЭВМ на практике зачастую невозможно по экономическим, тактико-техническим, массогабаритным и прочим характеристикам, в качестве альтернативы могут быть разработаны и применены специализированные вычислительные системы, ориентированные на решение задач обработки знаний. Однако на сегодняшний день подобные комплексы представлены лишь в виде единичных лабораторных образцов, дорогих и сложных в применении, а общедоступных систем такого класса на рынке серийных устройств не существует.

Своеобразной «золотой серединой» между универсальными системами и спецпроцессорами, ориентированными на обработку знаний, являются проблемно-ориентированные процессоры [Tatur et al., 2010]. Основная идея в их использовании состоит в том, чтобы обеспечить унификацию процессора в «некоторых» рамках, сохранив при этом достаточный уровень производительности. В то же время достигаемые технические характеристики и издержки на обеспечение универсальности оригинальной архитектуры должны быть конкурентными по сравнению со спецпроцессорами и серийными параллельными процессорами (многоядерными CPU, GPU, DSP).

В статье приводится описание разработанной оригинальной архитектуры ассоциативного процессора (обозначим ее как SNP, Semantic Network Processor) и процесса ее интеграции в существующую интеллектуальную систему (ИС). Основной задачей, решаемой архитектурой, является та самая унификация процесса решения интеллектуальных задач, сохранив при этом некоторую независимость от аппаратной

платформы. Так, на текущий момент, была разработана программная реализация процессора (на основе key-value БД RocksDB) и реализация на основе GPU (технология CUDA), с учетом выполнения как на одном графическом ускорителе, так и кластере GPU. В дальнейшем планируется реализовать архитектуру в виде проблемно-ориентированного процессора (на базе FPGA), что позволит сильно сократить стоимость одной ячейки памяти.

В результате, заказчик получает возможность построить интеллектуальную систему на основе SNP (в идеале используя банк готовых базовых решений) и выбрать подходящую под технические требования аппаратную платформу. При изменении технического задания (возросшем объеме базы знаний, изменении требований ко времени решений и т.д.) смена аппаратной платформы не должна повлиять на программную реализацию самой системы.

В системах, основанных на знаниях [Гаврилова и др., 2000], одной из важных задач является задача интеграции знаний, заключающаяся в выявлении фрагментов семантической сети, представляющих одни и те же знания, и их слиянии, что позволяет исключить избыточные элементы семантической сети, способные затормозить до предела работу системы, или сократить их количество, увеличить число ассоциаций для конкретных понятий, уточнить структуру модели предметной области. Задача интеграции необходима для обеспечения производительности и развития интеллектуальных систем в условиях неполноты знаний [Нариньяни, 2000], которая в свою очередь неизбежна в открытых, обучающихся интеллектуальных системах. Для решения задачи в условиях неполноты предложен ряд моделей и подходов [Ивашенко, 2007, 2009b, 2011b, 2012a, 2013b], [Кудрявцев, 2008], [Кофман, 1982], [Тэрано, 1993], [Doan and Havelly, 2005], [Giunchiglia et al., 2006], [Jean-Mary et al., 2007], [IDEF5, 1994], [Stumme, 2001]. Однако даже в этих условиях некоторые фрагменты знаний могут быть представлены полностью и так же важно выявить дублирование таких фрагментов с целью устранения избыточных знаний. Для выявления такого дублирования предлагается подход на основе установления сходства элементов дублирующихся фрагментов, использующий каноническую разметку графа семантической сети. Каноническая разметка графа семантической сети также может быть применена для решения частного случая задачи поиска по образцу, когда искомым граф изоморфен образцу.

1. Проблемно-ориентированный семантический процессор

1.1. Архитектура

Ассоциативная вычислительная система (ВС) есть не что иное, как одна из разновидностей параллельных вычислительных систем SIMD класса

(рисунок 1), в которых n процессорных элементов ПЭ представляют собой простые устройства, как правило, последовательной поразрядной обработки. При этом каждое слово (ячейка) ассоциативной памяти имеет свое собственное устройство обработки данных (сумматор). Операция осуществляется одновременно всеми n ПЭ. Все или часть элементарных последовательных ПЭ могут синхронно выполнять операции над всеми ячейками или над выбранным множеством слов ассоциативной памяти.

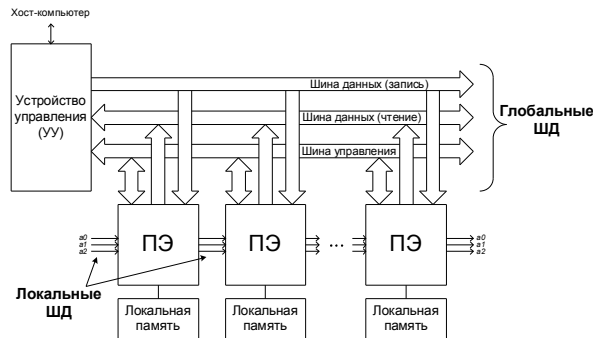


Рисунок 1 – Структурная ассоциативного процессора

Время обработки N m -разрядных слов в ассоциативной ВС определяется выражением [Цилькер и др., 2007]:

$$T = mt \left(\frac{N}{n} + K \right),$$

где t – время цикла ассоциативной памяти; n – число ячеек ассоциативной системы; K – коэффициент сложности выполнения элементарной операции (количество последовательных шагов, каждый из которых связан с доступом к памяти). Таким образом, время обработки T является константой и в большей мере зависит от величины $\frac{N}{n}$, т.е. количества ячеек памяти, приходящихся на

один ПЭ. После фиксации величины $\frac{N}{n}$ (выбора конфигурации нитей GPU или настройка схемы ПЭ на FPGA [Verenik et al., 2014]) время обработки остается неизменным вне зависимости от общего

объема обрабатываемой памяти.

SNP представляет собой классическую SIMD архитектуру, в которой каждый ПЭ содержит схему характерную для ассоциативной памяти (рисунок 2). На схеме SNP представлен одним устройством управления (УУ), обеспечивающим последовательное выполнение команд программы, которые транслируются множеству процессорных элементов, каждый из которых обрабатывает свои данные. Коммуникация с внешней средой осуществляется посредством глобальных шин данных (ШД):

- ШД записи – шина, по которой исполняемая команда параллельно транслируется на все ПЭ;
- ШД чтения – шина, по которой выполняется последовательное чтение данных с процессорных элементов (ШД с временным разделением);
- шина управления (ШУ) – множество управляющих сигналов.

Каждый ПЭ представляет собой ассоциативное запоминающее устройство (АЗУ) и включает в себя (см. рисунок 2):

- запоминающий массив для хранения N m -разрядных слов;
- регистр ассоциативного признака, куда помещается код искомой информации (признак поиска). Разрядность регистра k равна длине слова m ;
- схемы совпадения, используемые для параллельного сравнения каждого бита всех хранимых слов с соответствующим битом признака поиска и выработки сигналов совпадения;
- регистр совпадений, где каждой ячейке запоминающего массива соответствует один разряд, в который заносится единица, если все разряды соответствующей ячейки совпали с одноименными разрядами признака поиска;
- комбинационную схему, которая на основании анализа содержимого регистра совпадений формирует сигналы, извещающие о том, что искомая информация:
 - $a0$ – не найдена;
 - $a1$ – содержится в одной ячейке;

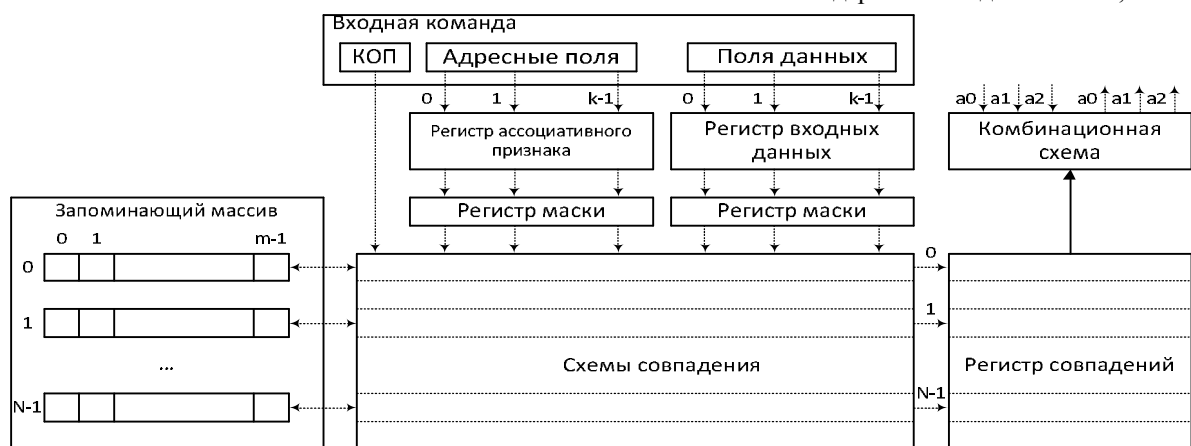


Рисунок 2 – Структурная схема ПЭ

- a_2 – содержится более чем в одной ячейке.
- регистр входных данных, куда помещаются данные для записи в память (запись производится в ячейки, подошедшие согласно признаку поиска). Разрядность регистра также равна k ;
- регистры маски, позволяющие запретить сравнение определенных битов при поиске и изменения состояния определенных битов при записи;

Формируемые на выходе ПЭ сигналы a_0, a_1, a_2 поступают на вход последующего ПЭ, который, в свою очередь, учитывает их при формировании своих выходных сигналов. Тем самым, линейка ПЭ образует единую ассоциативную память, результирующие сигналы которой будут сформированы на выходе последнего ПЭ. Процессор является хорошо масштабируемым, позволяя наращивать объем данных системы простым подключением ПЭ к последнему в массиве. Сохраняется основное преимущество ассоциативной памяти – увеличение количества ПЭ приводит к увеличению общего объема памяти без увеличения результирующего времени обработки данных. А в случае использования процессора в качестве вычислительного ядра интеллектуальной системы из-за огромных размеров баз знаний эта особенность может оказаться решающей.

1.2. Представление команд и данных

Система команд SNP представлена всего одной командой (рисунки 2, 3).



Рисунок 3 – Формат команды SNP

Команды хранятся отдельно от данных, более подробное описание работы команды см. [Verenik et al., 2014].

Данными является классический двудольный граф регулярной структуры. Формат хранения данных определяется системный программистом, подробный пример см. в [Вереник и др., 2013]. Порядок преобразования семантической сети в «регулярный» граф схематично проиллюстрирован на рисунках 4 и 5 [Вереник и др., 2012].

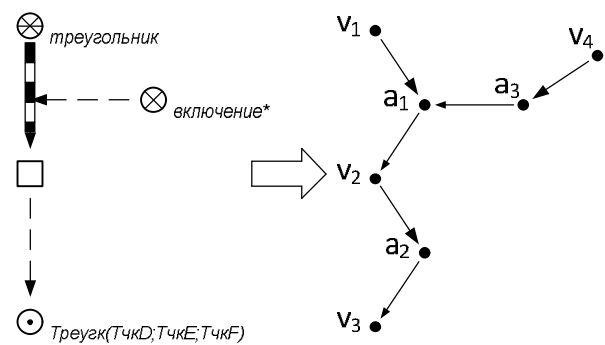


Рисунок 4 – Преобразование исходного графа семантической сети к оргграфу

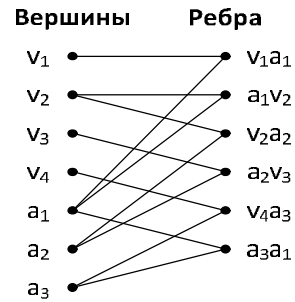


Рисунок 5 – «Регулярный» граф

2. Представление и каноническая разметка семантических сетей

Для решения задачи канонической разметки [Babai, 1980], [Hartke, 2009] графа семантической сети в работах [Ивашенко, 2007, 2013b] использовалось представление семантической сети в виде ориентированного псевдографа, основанное на преобразовании в соответствии со следующей схемой.

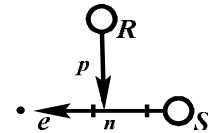


Рисунок 6 – Структура из пяти элементов, семантическая сеть

Фрагмент семантической сети, конструкция, структура из пяти элементов, изображённая на рисунке 6 в этом случае представляется в виде, изображённом на рисунке 7. В отличие от представления графа в SNP начало ребра семантической сети связывается с ним двумя встречными дугами, что может позволить обеспечить временную сложность поиска начала ребра меньшую, чем линейная.

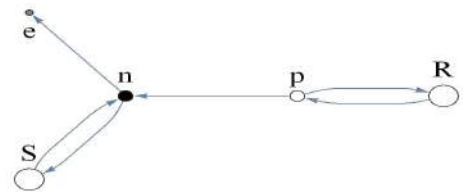


Рисунок 7 – Структура из пяти элементов, размеченный оргграф

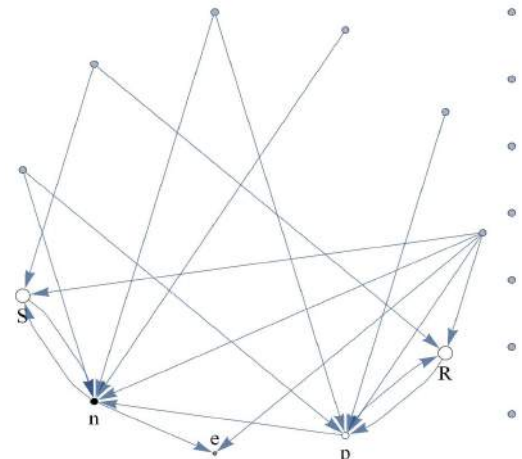


Рисунок 8 – Структура из пяти элементов и их типы, оргграф

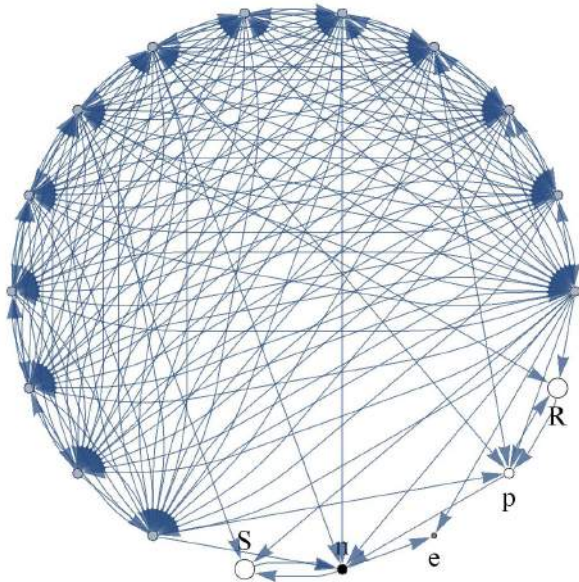


Рисунок 9 – Структура из пяти элементов, связный орграф

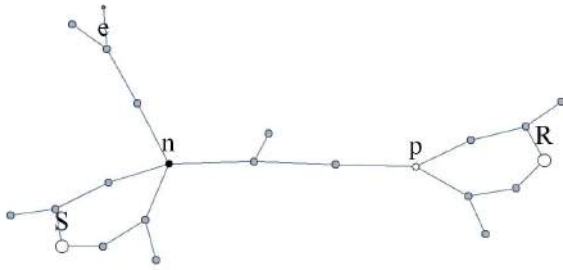


Рисунок 10 – Структура из пяти элементов, граф

Такое представление является структурно неоднородным, так как необходимо дополнительно хранить информацию о типах элементов семантической сети в неграфовом виде, например в виде меток. Для представления информации о типах элементов в графовом виде алгоритмы канонической разметки используют дополнительные вершины, кодирующие информацию о типах. Всего используется тринадцать вершин, позволяющих закодировать 2^{13} типов (рисунок 8). Для отделения и отличия этих вершин от вершин, представляющих элементы семантической сети, они связываются дугами в полный ориентированный граф (орграф), из которого вычтена упорядочивающая их цепочка, позволяя получить связный орграф (рисунок 9). При необходимости перейти от ориентированного псевдографа к неориентированному графу каждая дуга ориентированного псевдографа может быть заменена тремя новыми дугами и тремя новыми вершинами (рисунок 10).

При реализации алгоритмов канонической разметки была разработана библиотека алгоритмов преобразования графовых структур для разных представлений, включая матрицу смежности, форматы graph6, sparse6 и другие. В том числе поддерживается версия формата Transfer Graph Format (TGF), в котором хранится исходная семантическая сеть. Так же существуют конверторы

в формат TGF и из него в формат SCs, который поддерживает представление конструкций языков модели унифицированного семантического представления знаний [Голенков и др., 2001], [Гулякина и др., 2004], [Ивашенко, 2003, 2004, 2009а, 2011а, 2013а, 2013б], [OSTIS, 2014]. Алгоритм преобразования семантической сети в графовое представление полагается на следующие условия:

- информация о типе и неграфовом содержимом элементов семантической сети таким, как системный идентификатор и содержимое, хранится локально, совместно с элементом семантической сети, то есть не требуется буферизировать эту информацию, чтобы обеспечить её связность;
- информация о содержимом элемента может быть указана независимо от команды генерации этого элемента;
- большинство алгоритмов имеют интерфейс генерации узловых элементов семантической сети и дуг с явным указанием инцидентных элементов;
- возможны нетривиальные структуры на семантической сети, например, когда каждая из двух дуг является концом другой.

```

T ←
Z ← T
Y ← ∅
пока Y ⊂ Z
  X ← Z
  Z ← Y
  Y ← ∅
пока ∅ ⊂ X
  для ∀x∀t(x ∈ X) ∧ (t = type(⟨T, x⟩))
    если x ∈ edges(T), то
      если map(beg(⟨T, x⟩)) ∧
         map(end(⟨T, x⟩)), то
        b ← map(beg(⟨T, x⟩))
        e ← map(end(⟨T, x⟩))
        map(⟨x, gen(⟨x, t, b, e⟩)⟩)
      иначе если map(beg(⟨T, x⟩)) ∧
         (end(⟨T, x⟩) = x), то
        b ← map(beg(⟨T, x⟩))
        map(⟨x, gen(⟨x, t, b, x⟩)⟩)
      иначе Y ← Y ∪ {x}
    иначе map(⟨x, gen(⟨x, t⟩)⟩)
  если map(x), то
    если ∃i(i = identifier(x)), то
      identifier(⟨map(x), i⟩)
    если ∃c(c = content(x)), то
      content(⟨map(x), c⟩)
Y ← ∅
пока Y ⊂ Z
  X ← Z
  Z ← Y
  Y ← ∅
пока ∅ ⊂ X
  для ∀x∀t(x ∈ X) ∧ (t = type(⟨T, x⟩))

```

```

если  $map(beg(\langle T, x \rangle)) \wedge$ 
 $map(end(\langle T, x \rangle))$ , то
 $b \leftarrow map(beg(\langle T, x \rangle))$ 
 $e \leftarrow map(end(\langle T, x \rangle))$ 
если  $map(x)$ , то
 $set\_end(\langle map(x), e \rangle)$ 
иначе  $map(\langle x, gen(\langle x, t, b, e \rangle) \rangle)$ 
если  $\exists i (i = identifier(x))$ , то
 $identifier(\langle map(x), i \rangle)$ 
если  $\exists c (c = content(x))$ , то
 $content(\langle map(x), c \rangle)$ 
иначе если  $map(beg(\langle T, x \rangle))$ , то
 $b \leftarrow map(beg(\langle T, x \rangle))$ 
 $map(\langle x, gen(\langle x, t, b \rangle) \rangle)$ 
 $Y \leftarrow Y \cup \{x\}$ 

```

Здесь T – текст с элементами семантической сети, множество рёбер и вершин, x – элемент, вершина или ребро семантической сети, $type(\langle T, x \rangle)$ – тип элемента семантической сети, множество меток, в тексте T , $edges(T)$ – множество рёбер в тексте T , $map(\langle x, \alpha \rangle)$ – отобразить x на α , $map(x)$ – получить образ x , $beg(\langle T, x \rangle)$ – начало ребра в тексте T , $end(\langle T, x \rangle)$ – конец ребра в тексте T , $set_end(\langle \alpha, e \rangle)$ – установить для элемента x конец e , $gen(\langle x, t \rangle)$ – сгенерировать элемент x с типом t , $gen(\langle x, t, b, e \rangle)$ – сгенерировать элемент x с типом t с началом b и концом e , $gen(\langle x, t, b \rangle)$ – сгенерировать элемент x с типом t с началом b , $identifier(x)$ – идентификатор элемента x , $identifier(\langle \alpha, i \rangle)$ – установить элементу α идентификатор i , $content(x)$ – содержимое элемента x , $content(\langle \alpha, c \rangle)$ – установить элементу α содержимое c .

Для решения задачи канонической разметки используется алгоритм Л. Бабаи [Babai, 1980].

```

 $\langle V, E \rangle \leftarrow$ 
 $n \leftarrow |V|$ 
 $r \leftarrow \lceil 3 * \ln n / \ln 2 \rceil$ 
 $f \leftarrow true$ 
для  $(i \leftarrow 1; i < n; i++)$ 
 $p(i) \leftarrow i$ 
 $d1 \leftarrow 2 * \sum_{j=1}^r E(i, j)$ 
 $d2 \leftarrow 2 * (n - n^2) * \sum_{j=1}^r (E(i, j) * E(j, i))$ 
 $d3 \leftarrow 2 * n^2 * \sum_{j=1}^r (E(i, j) + E(j, i))$ 
 $d4 \leftarrow (n^2 + (n + 1)^2) * E(i, i)$ 
 $d(i) \leftarrow d1 + d2 + d3 - d4$ 
 $b \leftarrow sort(\langle d, 1, n \rangle)$ 
для  $(i \leftarrow 1; i < r; i++)$ 
если  $d(b(i)) = d(b(i + 1))$ , то  $f \leftarrow false$ 

```

```

если  $f$ , то для  $(i \leftarrow r + 1; i \leq n; i++)$ 
 $k \leftarrow b(i)$ 
 $q(i) \leftarrow$ 
 $\sum_{j=1}^r (E(b(j), k) + (2 + E(b(j), k)) * E(k, b(j))) * 8^j$ 
если  $f$ , то  $p \leftarrow sort(\langle q, r + 1, n \rangle)$ 
если  $f$ , то для  $(i \leftarrow r + 1; i < n; i++)$ 
если  $q(p(i)) = q(p(i + 1))$ , то  $f \leftarrow false$ 
 $\leftarrow \langle f, b \circ p \rangle$ 
Здесь  $V$  – множество вершин ориентированного псевдографа,  $E$  – матрица (функция) смежности его вершин,  $sort(\langle \alpha, \beta, \gamma \rangle)$  – возвращает отображение, сортирующее элементы  $\alpha$  от индекса  $\beta$  до индекса  $\gamma$ ,  $\alpha \circ \beta$  – композиция отображений  $\alpha$  и  $\beta$ .

```

Если с помощью него не удаётся получить разметку, то используется следующий рекурсивный алгоритм (*canonize*).

```

 $\langle V, E, m, p \rangle \leftarrow$ 
 $n \leftarrow |V|$ 
 $\langle m, p \rangle \leftarrow subclassify(\langle V, E, m, p \rangle)$ 
для  $(i \leftarrow 1; i < n; i++)$ 
если  $m(i) > i + 1$ , то
 $k \leftarrow m(i)$ 
 $s \leftarrow m$ 
 $s(i) \leftarrow i + 1$ 
 $\langle s, pi \rangle \leftarrow canonize(\langle V, E, s, p \rangle)$ 
для  $(j \leftarrow i + 1; j < k; j++)$ 
 $b \leftarrow p$ 
 $swap(\langle b, i, j \rangle)$ 
 $s \leftarrow m$ 
 $s(i) \leftarrow i + 1$ 
 $\langle s, pj \rangle \leftarrow canonize(\langle V, E, s, b \rangle)$ 
если  $less(\langle E, pi, pj \rangle)$ , то
 $swap(\langle p, j, --, --, k \rangle)$ 
иначе если  $less(\langle E, pj, pi \rangle)$ , то
 $swap(\langle p, i, j \rangle)$ 
 $j \leftarrow i + 1$ 
иначе  $j++$ 
 $\leftarrow \langle m, order(\langle V, E, m, p \rangle) \rangle$ 
Здесь  $canonize(\langle V, E, s, \alpha \rangle)$  – рекурсивный
вызов,  $less(\langle E, pi, pj \rangle)$  – сравнение матрицы  $E$  в
перестановке  $pi$  с этой же матрицей в перестановке
 $pj$ ,  $swap(\langle \alpha, \beta, \gamma \rangle)$  – обмен значений отображения
 $\alpha$  по индексам  $\beta$  и  $\gamma$ .

```

При реализации этого рекурсивного алгоритма на параллельной системе, возможен выигрыш, так как рекурсивные вызовы в алгоритме не зависят по данным, однако для реализации требуется наличие эффективных средств поддержки параллелизма независимых ветвей, разделяющих общие данные по чтению.

Рекурсивный алгоритм использует два алгоритма. Первый (*subclassify*) основан на подразбиении множества вершин исходного графа на основе анализа их степеней в подграфах,

образованных вершинами множеств разбиения. В некоторых случаях он позволяет получить полное решение за однократное применение.

```

⟨V, E, m, p⟩ ←
n ← |V|
для (k ← 1; k ≤ n; k ← i)
  q ← k
  для (i ← 1; i < m(k); i++)
    если (E(p(i), p(i)) > 0), то
      swap(⟨p, q++, i⟩)
  пока (k < q) m(k++) ← q
  s(i) ← 1
пока s ≠ m
  s ← m
  для (ilb ← 1; ilb ≤ n; ilb ← u)
    u ← m(ilb)
    для (jlb ← ilb; jlb ≤ n; jlb ← m(jlb))
      для (i ← ilb; i < ulb; i++)
        q ← p(i)
        d1 ← ∑_{j=ilb}^{m(jlb)} (E(q, p(j)) + E(p(j), q))
        d2 ← ∑_{j=ilb}^{m(jlb)} (E(q, p(j)) * E(p(j), q))
        d3 ← ∑_{i=jlb}^{m(jlb)} E(q, p(j))
        d(i) ← n2 * d1 + (n - n2) * d2 + d3
        b ← sort(⟨d, ilb, u⟩)
        p ← p ∘ b
      для (i ← u - 1; i ≥ ilb; i--)
        m(i) ← u
        если d(b(i-1)) > d(b(i)), то u ← i
← ⟨m, p⟩

```

По теоретическим оценкам временная сложность этого алгоритма в случае параллельной реализации, если не учитывать временные затраты на доступ к памяти [Ивашенко, 2012b], может быть улучшена с $O(n^3)$ до $O((n^3 * \ln^3 p) / p^2)$, где n – количество вершин, а p – количество процессорных элементов.

Второй алгоритм (*order*) сортирует вершины во множествах разбиения в результате анализа матрицы смежности.

```

⟨V, E, m, p⟩ ←
n ← |V|
для (i ← 1; i < n; i++)
  если m(i) > i + 1, то
    для (k ← i + 1; k < n; k ← u)
      u ← m(k)
      q ← k
      для (j ← k; j < u; j++)
        если
          (E(p(i), p(j)) * E(p(j), p(i)) > 0),
          то swap(⟨p, q++, j⟩)
      пока (k < q) m(k++) ← q
      для (j ← k; j < u; j++)
        если (E(p(i), p(j)) > 0), то
          swap(⟨p, q++, j⟩)

```

```

пока (k < q) m(k++) ← q
для (j ← k; j < u; j++)
  если (E(p(j), p(i)) > 0), то
    swap(⟨p, q++, j⟩)
пока (k < q) m(k++) ← q
m(i) ← i + 1

```

← p

По теоретическим оценкам временная сложность второго алгоритма (*order*) в случае параллельной реализации, если не учитывать временные затраты на доступ к памяти, может быть улучшена с $O(n^2)$ до $O((n^2 * \ln p) / p)$. Однако,

для достижения наиболее эффективных результатов в архитектуре вычислительной системы, которая будет исполнять этот алгоритм, должны быть предусмотрены эффективные средства исполнения операций редукции и параллельной (битонической) сортировки. В случае, когда такой поддержки нет, выигрыша может не быть или он будет незначительным.

Заключение

Рассмотрены основные принципы организации архитектуры проблемно-ориентированного процессора, предназначенного для решения задач на графах, рассмотрена задача канонической разметки семантической сети и алгоритмы её решения, приведена теоретическая оценка результатов параллельной реализации этой задачи.

Библиографический список

- [Вереник и др., 2012] Разработка проблемно-ориентированных процессоров семантической обработки информации / Н. Л. Вереник, Е. Н. Сейткулов, М. М. Татур // Электроника-инфо. – 2012. – №8. – С. 95–98.
- [Вереник и др., 2013] Имитационная модель векторного процессора на примере задачи поиска пути в графе / Н.Л. Вереник, Е. Н. Сейткулов, М. М. Татур // Искусственный интеллект. – 2013. – №4. – с.89–100.
- [Гаврилова и др., 2000] Гаврилова, Т.А. Базы знаний интеллектуальных систем / Т. А. Гаврилова, В.Ф. Хорошевский. – СПб.: Питер, 2000.
- [Голенков и др., 2001] Голенков, В.В. Представление и обработка знаний в графодинамических ассоциативных машинах / В. В. Голенков [и др.] – Мн. : БГУИР, 2001.
- [Гулякина и др., 2004] Гулякина Н.А. Ивашенко В.П. Интеграция знаний в информационных системах. / Н.А. Гулякина, В.П. Ивашенко // Доклады БГУИР. – 2004. – №6. – С. 113-119.
- [Голенков и др., 2012] Графодинамические модели параллельной обработки знаний: принципы построения, реализации и проектирования. / В. А. Голенков, Н. А. Гулякина // Открытые семантические технологии проектирования интеллектуальных систем: материалы II Междунар. науч.-техн. конф. (Минск, 16-18 февраля 2012 г.) / редкол. : В.В. Голенков (отв. ред.) [и др.] – Минск, БГУИР, 2012 – С. 23–52.
- [Ивашенко, 2003] Представление нейронных сетей и систем продукций в однородных семантических сетях. / В.П. Ивашенко // Известия Белорусской инженерной академии. – 2003. – №1(15)/1. – с.184-188.
- [Ивашенко, 2004] Применение однородных семантических сетей для представления знаний о нестационарных предметных областях. / В.П. Ивашенко // Известия Белорусской инженерной академии. – 2004. – №1(17)/3. – с.77-80.
- [Ивашенко, 2007] Разработка баз знаний семантических интеллектуальных систем / В.П. Ивашенко // Дистанционное обучение – образовательная среда XXI века : материалы VI

Междунар. науч.-метод. конф., Минск, 22–23 нояб. 2007 г. / Беларус. гос. ун-т информатики и радиоэлектроники. – Минск, 2007. – С. 180–182

[Ивашенко, 2009а] Ивашенко В.П. Семантические модели баз знаний / В.П. Ивашенко Информационные системы и технологии (IST'2009): материалы V Междунар. конф.-форума в 2-х ч. Ч.2 – Минск: А.Н.Вараксин, 2009.- с.125-128.

[Ивашенко, 2009б] Ивашенко В.П. Алгоритмы верификации и интеграции баз знаний. Вестник Брестского государственного технического университета, БрГТУ, 2009, №5.

[Ивашенко, 2011а] Ивашенко В.П. Семантическая технология компонентного проектирования баз знаний. Материалы Международной научн.-техн. Конференции OSTIS,2011:Минск, Республика Беларусь, БГУИР 10-12 февраля 2011.

[Ивашенко, 2011б] Ивашенко В.П. Алгоритмы операций отладки и интеграции баз знаний. Дистанционное обучение – образовательная среда XXI века: материалы VII Международной научн.-метод. конференции (Минск, 1-2 декабря 2011г.). – Минск: БГУИР, 2011, сс.227-229.

[Ивашенко, 2012а] Ивашенко В.П. Семантическая модели интеграции и отладки баз знаний. Материалы Международной научн.-техн. Конференции OSTIS, 2012:Минск, Республика Беларусь, БГУИР 16-18 февраля 2012.

[Ивашенко, 2012б] Ивашенко, В.П. Алгоритмы полилогарифмической временной и логарифмической пространственной сложности для системы динамического распределения линейной адресуемой памяти с однородным доступом к данным / В.П. Ивашенко // Карповские научные чтения : сб. науч. ст. / Беларус. гос. ун-т; редкол. : А.И. Головня (отв. ред.) [и др.]. – Минск, 2012. – Вып. 6, ч. 1. – С. 196–201.

[Ивашенко, 2013а] Ивашенко В.П. Унифицированное представление и интеграция знаний. Материалы Международной научн.-техн. Конференции OSTIS, 2013:Минск, Республика Беларусь, БГУИР 20-23 февраля 2013.

[Ивашенко, 2013б] Ивашенко В.П. Интеграция на основе унифицированного представления знаний / В. П. Ивашенко // Электроника инфо. – 2013. – № 10. – С. 37–46.

[Каляев и др., 2012] Высокопроизводительные реконфигурируемые вычислительные системы нового поколения / И.А. Каляев, А. И. Дордопуло, И.И. Левин, Е.А. Семерников // Вычислительные методы и программирование новые вычислительные технологии. – 2011. – Т. 12. – №2. – С. 82–89.

[Кофман, 1982] Кофман А. Введение в теорию нечетких множеств. М.: Радио и связь, 1982. — 432 с.

[Кудрявцев, 2008] Кудрявцев Д. В. Практические методы отображения и интеграции онтологий. Семинар Знания и онтологии *Elsewhere*, КИИ-2008, Дубна, 2008.

[Нариньяни, 2000] Нариньяни А. С. НЕ-факторы: неточность и недоопределенность — различие и взаимосвязь // Изв. РАН. Сер. Теория и системы управления. 2000. № 5. С. 44-56.

[Тэрано, 1993] Прикладные нечеткие системы: Пер. с япон. / К. Асаи, Д. Ватада, С. Иваи и др.; Под ред. Т. Тэрано, К. Асаи, М. Сугэно. – М.: Мир, 1993.

[Цилькер и др., 2007] Цилькер Б.Я., Орлов С.А. Организация ЭВМ и систем: Учебник для вузов. – СПб.: Питер, 2007. – 668 с.

[Allemang et al., 2008] Dean Allemang, James Hendler. Semantic Web for the Working Ontologist: Effective Modeling in RDFS and OWL. Elsevier. – 2008. ISBN: 978-0-12-373556-0.

[Babai, 1980] Babai, L. Random graph isomorphism / L. Babai, P. Erdős, M. Stanley // SIAM J. on Computing. – 1980. – Vol. 9. – P. 628–635.

[Brodkorb et al., 2013] André R. Brodtkorb, Trond R. Hagen, Martin L. Sætra. Graphics processing unit (GPU) programming strategies and trends in GPU computing. *Journal of Parallel and Distributed Computing*, 73(1):4–13, January 2013.

[Chung et al., 1995] Minhwa Chung, Dan Moldovan. Parallel natural language processing on a semantic network array processor. *IEEE Transactions on Knowledge and Data Engineering*, 7(3):391–405, June 1995.

[Doan and Halevy, 2005] A. H. Doan, A. Y. Halevy. Semantic integration research in the database community: A brief survey. *AI magazine*, 26(1), 2005

[Giunchiglia et al., 2006] F. Giunchiglia, M. Marchese and I. Zaihrayeu (2006). Encoding classifications into lightweight ontologies. University of Trento Technical Report # DIT-06-016, March 2006.

[Guo et al., 2009] Qinglin Guo, Ming Zhang. Question answering based on pervasive agent ontology and Semantic Web.

Knowledge-Based Systems. – August 2009. – Vol. 22. – Issue 6. – P. 443–448.

[Hartke, 2009] Hartke, S.G. McKay's canonical graph labeling algorithm / S.G. Hartke, A.J. Radcliffe // *Communicating Mathematics*. – 2009. – Vol. 479. – P. 99–111.

[IDEF5, 1994] Information Integration for Concurrent Engineering (ICE). IDEF5 Method Report. - Knowledge Based Systems, Inc. , 1408 University Drive East College Station, Texas, USA. September 21. 1994.

[Jan et al., 2012] Yahya Jan, Lech Józwiak. Scalable communication architectures for massively parallel hardware multi-processors *Journal of Parallel and Distributed Computing*. – November 2012. – Vol. 72. – Issue 11. – P. 1450–1463.

[Jean-Mary et al., 2007] Jean-Mary Y., Kabuka, M. ASMOV: Ontology Alignment with Semantic Validation. Joint SWDB-ODBIS Workshop, September 2007, Vienna, Austria, 15-20.

[Kitano et al., 1992] Semantic network array processor as a massively parallel computing platform for high performance and large-scale natural language processing. In *Proc. Int'l Conf. on Computational Linguistics (COLING '92)*, volume 2, pages 813–819, 1992.

[Navarro et al., 2014] Cristobal A. Navarro, Nancy Hitschfeld-Kahler, Luis Mateu. A survey on parallel computing and its applications in data-parallel problems using GPU architectures. *Communications in Computational Physics*, 15(2):285–329, February 2014.

[OSTIS, 2013] Открытая семантическая технология проектирования интеллектуальных систем [Электронный ресурс]. – 2010. - Режим доступа: <http://ostis.net>. – Дата доступа: 2.11.2014

[Stumme, 2001] Gerd Stumme and Alexander Maedche. FCA-merge: bottom-up merging of ontologies. In *Proceedings of 17th IJCAI*, pages 225 {230, Seattle (WA), USA, 2001.

[Tatur et al., 2010] Tatur M., Adzinet D., Lukashevich M., Bairak S. Synthesis and Analysis of Classifiers Based on Generalized Model of Identification. *Advances in intelligent and soft computing*. – 2010. – Vol. 71. – P. 529–536.

[Verenik et al., 2014] Verenik N.L., Seitkulov Y.N., Girel A.I., Tatur M.M. Some regularities and objective limitations of implementing semantic processing algorithms on computing systems with massive parallelism. *Eurasian Journal of Mathematical and computer Applications*, 2(2):92–101, 2014.

SEMANTIC NETWORKS REPRESENTATION AND ALGORITHMS FOR THEIR ORGANIZATION AND SEMANTIC PROCESSING ON MASSIVELY PARALLEL COMPUTERS

Ivashenko V.P. *, Verenik N.L. *, Girel A.I. *, Seitkulov Y.N. **, Tatur M.M. *

*Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

ivashenko@bsuir.by

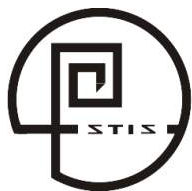
nick.verenik@gmail.com

tatur@i-proc.com

** L.N. Gumilyov Eurasian national university, Astana, Kazakhstan

seitkulov_y@enu.kz

In the article description of semantic networks representation for processing on special parallel architecture and semantic graph canonical labeling are considered. Some estimations for time complexity canonical labeling algorithms and potential benefits of parallel processing are given.



OSTIS-2015

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822:514

КОМПОНЕНТНАЯ АРХИТЕКТУРА ИНТЕЛЛЕКТУАЛЬНОЙ СИСТЕМЫ КОНСУЛЬТАЦИОННОГО ОБСЛУЖИВАНИЯ И ОБУЧЕНИЯ РАЗРАБОТЧИКОВ ПРОГРАММ

Пивоварчик О.В.*

* *Барановичский государственный университет, г. Барановичи, Республика Беларусь*

pivovarchyk@tut.by

В данной статье обоснована необходимость создания интеллектуальных компьютерных средств обучения в области программирования, имеющих открытую многокомпонентную архитектуру. В качестве компьютерного средства такого вида предложена интеллектуальная система консультационного обслуживания и обучения. В статье представлена общая компонентная архитектура системы и семантические модели каждого ее компонента.

Ключевые слова: интеллектуальная система; консультационная система; обучающая система; разработка программ; база знаний.

Введение

Множество существующих методов и средств разработки программ, ориентированных на обработку баз знаний, а также постоянное их расширение и совершенствование, требуют от разработчиков не только глубоких знаний и умений, но и способностей быстро изучать новые технологии. Эффективными помощниками в изучении являются интеллектуальные компьютерные средства обучения (далее – КСО). В настоящее время в области технологий разработки программ существует множество интеллектуальных КСО. К числу наиболее упоминаемых в современных работах относятся интеллектуальные обучающие системы (далее – ИОС) и интеллектуальные учебные среды (далее - ИУС). ИОС основаны на интеграции технологий компьютерного обучения и экспертных систем и предназначены для освоения методов решения слабо структурированных задач [Попов и др., 1996]. ИУС предназначены для формирования навыков разработки программ.

Основными направлениями развития ИОС и ИУС являются использование технологий адаптации и многоагентных технологий, компонентно-ориентированная разработка, веб-ориентация. Примерами систем в области разработки программного обеспечения являются: ELM-ART, ACT-R Programming Tutor, GREATERP, LISP-TUTOR, FLEX, InterBook, QuizGuide (обеспечивают изучение функциональных языков);

VC Prolog Tutor (обеспечивает изучение языков логического программирования); J-LATTE, CITS, JITS, BITS (обеспечивают изучение объектно-ориентированных языков).

Актуальным является развитие специализированных инструментов, поддерживающих полный жизненный цикл создания и сопровождения ИОС и ИУС. Следует отметить, что в настоящее время разработано немного таких инструментов, примерами которых являются: МОНАП, АТ-ТЕХНОЛОГИЯ, KBS-HYPERBOOK.

Основным недостатком существующих ИОС и ИУС является архитектура. Большинство существующих систем создано как автономные программные продукты, которые не могут быть использованы в качестве компонентов других систем. Необходимо использовать либо целую систему, либо ничего. Небольшое число систем поддерживает компонентно-ориентированную архитектуру способную интегрироваться с другими системами. Однако, их интеграция возможна при условии использования одинаковых технологий и только при проектировании одной командой разработчиков, т.е. они не являются открытыми системами. Отсутствие возможности использования отдельно компонентов различных систем не позволяет создавать открытые распределенные архитектуры, усложняет процесс расширения функциональности.

Вторым недостатком является монофункциональность большинства отдельно

взятых систем. Например, QuizGuide позволяет проводить тестирование пользователя, InterBook предоставляет справочную информацию, ELM-ART проводит курс обучения языку программирования. Для обеспечения всестороннего индивидуального обслуживания пользователя, повышения удобства использования и эффективности решения задач необходимо объединение функциональности систем. Это возможно обеспечить путем создания одной многофункциональной системы, либо интеграцией систем или их компонентов.

Третьим недостатком является ограниченная функциональность при взаимодействии со средами программирования. Большинство существующих систем обеспечивает предоставление справочной информации по запросу и не имеет возможности проводить обучение и консультирование на основании информации о пользователе. Лишь единицы обеспечивают консультирование пользователя. Например, система FLEX [Стефанюк и др., 2004], [Стефанюк, 2007]. С другой стороны, лишь очень малое число ИУС включает компоненты среды программирования.

Решением данных проблем является разработка обучающих средств, имеющих многокомпонентную архитектуру. Технология разработки должна позволять повторно использовать компоненты, интегрировать их с другими компонентами, построенными с помощью как данной, так и других технологий. Также она должна быть открытой, чтобы обеспечить возможность использования компонентов различными командами разработчиков.

В данной статье представлена компонентная архитектура интеллектуальной системы консультационного обслуживания и обучения разработчиков программ, а также, общие принципы построения ее компонентов. Система проектируется на базе *Технологии OSTIS* (Open Semantic Technology for Intelligent Systems) [OSTIS], [IMS]. Формальные модели интеллектуальных систем, проектируемых на базе OSTIS, строятся на основе семантических сетей, что создает необходимые условия для обеспечения семантической совместимости интеллектуальных систем и их компонентов [Голенков и др., 2013].

1. Компонентная архитектура

Интеллектуальная система, спроектированная по *Технологии OSTIS*, представляет собой интеграцию *многократно используемых компонентов баз знаний, многократно используемых компонентов машин обработки знаний и многократно используемых компонентов интерфейсов*. Архитектура дочерней интеллектуальной системы включает:

- Многократно используемые компоненты, обеспечивающие базовые функции интеллектуальной системы – ядро базы знаний, ядро машины обработки знаний (набор базовых

sc-агентов, обеспечивающих редактирование базы знаний, навигацию по базе знаний, базовые информационно-поисковые операции), ядро интерфейсов.

- Многократно используемые компоненты, расширяющие базовую функциональность до требований, предъявляемых к интеллектуальной системе. Необходимые компоненты могут существовать в *Библиотеке многократно используемых компонентов OSTIS*, а также могут разрабатываться самостоятельно.

Описание базовых компонентов представлено в работах [Гракова и др., 2014], [Шункевич, 2014], [Корончик, 2014]. Задачей данной работы является описание компонентов, требуемых для функционирования интеллектуальной системы консультационного обслуживания и обучения разработчиков программ.

На основе анализа работ [Стефанюк, 2002], [Brusilovsky, 2012], [Рыбина, 2008], а также существующих систем были выделены основные структурные компоненты ИОС: эксперт по предметной области (или модель предметной области), модель обучаемого, педагог (или модель учебного материала), интерфейс. На основании обобщенной структурной модели и требуемой функциональности был определен набор необходимых компонентов для создания интеллектуальной системы консультационного обслуживания и обучения:

- *многократно используемые компоненты баз знаний*, описывающие предметную область;
- *многократно используемые компоненты баз знаний*, описывающие учебный материал;
- *многократно используемые компоненты баз знаний*, описывающие модель пользователя интеллектуальной системы;
- *многократно используемые компоненты баз знаний*, описывающие стратегии управления обучением и консультированием;
- *многократно используемые компоненты машин обработки знаний*, обеспечивающие обработку модели пользователя;
- *многократно используемые компоненты машин обработки знаний*, обеспечивающие управление обучением и консультированием;
- *многократно используемые компоненты машин обработки знаний*, обеспечивающие мониторинг и анализ деятельности пользователя;
- *многократно используемые компоненты интерфейсов*, обеспечивающие взаимодействие интеллектуальной системы с пользователем, преподавателем, экспертом;
- *многократно используемые компоненты интерфейсов*, обеспечивающие взаимодействие интеллектуальной системы с другими

интеллектуальными системами, другими платформами.

Каждый компонент может включаться в любую другую интеллектуальную систему.

2. Многократно используемые компоненты баз знаний

Многократно используемые компоненты баз знаний, описывающие предметную область, являются *неатомарными многократно используемыми компонентами*.

Предметная область может быть декомпозирована на подобласти, каждая из которых представляется в виде отдельного компонента. Например, предметная область разработка программного обеспечения включает множество используемых концептуальных, методических и технических средств. На основании анализа литературы была выделена следующая структура предметной области:

- теория программ
- инструменты:
 - языки программирования;
 - языки представления знаний;
 - среды разработки программ;
 - библиотеки типовых многократно используемых компонентов;
- методы и методики разработки программного обеспечения;
- управление разработкой программного обеспечения;
- оценка качества программного обеспечения;
- методы и методики обучения разработке программ.

Для каждой подобласти необходимо разработать несколько типов компонентов баз знаний. Для обучающих систем основными типами компонентов являются семантически совместимые онтологии предметных областей и/или онтологии задач, решаемых в предметной области. Были выделены следующие онтологии, которые необходимы для функционирования системы:

- *Терминологическая онтология* – спецификация терминов (понятий и отношений), используемых для описания предметной области.
- *Теоретико-множественная онтология* – онтология, содержащая теоретико-множественные связи (отношения) между понятиями и отношениями, определенными *терминологической онтологией*. К отношениям относятся: *включение**, *разбиение**, *объединение**, *пересечение**, *разность множеств**, *область определения**, *домен**, *функция**.

• *Информационная онтология* – спецификация структуры знаний, используемых для хранения информации о терминах. Для описания каждого ключевого понятия предметной области используются отношения: *определение**, *пояснение**, *детальное пояснение**, *комментарий**, *пример**, *обучающая информация**.

• *Статическая онтология* – описывает понятия, их атрибуты и отношения между ними. Используется для описания неизменяемых в некоторый период времени понятий.

• *Динамическая онтология* – используется для описания понятий, изменяемых во времени. Для описания понятий необходимо использовать процедурное представление знаний. Примерами общих терминов, включаемых в данную категорию, являются состояния и переходы из одного состояния в другое.

• *Эталонная модель предметной области* – онтология, используемая для моделирования уровня знаний и уровня умений пользователя (для обучения) или интересов пользователя (для консультирования и предоставления справочной информации).

Компонент эталонная модель предметной области формируется с целью использования обучающими и консультирующими системами. В эталонную модель включаются *sc-элементы*, которые отражают суть предметной области с точки зрения определенного объекта (обучающей системы, обучающего курса и др.). Она изоморфна целям (или цели) объекта.

Для систем, целью которых является обучение пользователя, онтология представляет собой максимальный уровень знаний и/или умений, который можно достигнуть. Знания и умения представляются в виде иерархии. При изучении предметной области модель знаний и умений пользователя структурируется в соответствии с данной иерархией. Предметная область считается изученной, если пользователь достиг самого высокого уровня иерархии. Знания входят во множество *sc-элементов знаний предметной области*. Умения входят во множество *sc-элементов умений предметной области*.

Для систем, целью которых является предоставление информации, онтология представляет собой множество возможных интересов в предметной области. Интересы также могут представляться в виде иерархии. Считается, что пользователю интересна предметная область, если уровень его интересов совпадает с самым высоким уровнем иерархии. Чем более сильный уровень интересов пользователя в предметной области, тем менее подробную информацию ему необходимо представлять о понятиях и тем больше понятий из данной предметной области ему необходимо выводить при запросе к разным

предметным областям. Интересы входят во множество *sc-элементов интересов предметной области*. Множества *sc-элементов знаний предметной области* и *sc-элементов интересов предметной области* могут пересекаться и быть равными.

Множественно используемые компоненты баз знаний, описывающие предметную область, разрабатываются экспертом и хранятся в Библиотеке *множественно используемых компонентов Технологии OSTIS*. При формировании обучающей системы некоторые компоненты уже могут существовать в библиотеке. В таком случае они просто интегрируются в интеллектуальную систему.

Множественно используемые компоненты баз знаний, описывающие учебный материал, могут быть *неатомарными множественно используемыми компонентами* или *атомарными множественно используемыми компонентами*. Компонент представляет собой *sc-структуру*, которая описывает некоторый учебный объект (обучающий курс, система тестирования, тренажер и др).

Рассмотрим формальную спецификацию компонента обучающий курс. Она включает следующие характеристики:

- Наименование курса на различных естественных языках. Знак компонента связан с *sc-ссылкой*, содержимым которой является наименование, отношением *наименование**.
- Информация об авторе. Знак компонента связан со знаком автора отношением *автор**. Автором может быть физическое лицо, коллектив, другая интеллектуальная система.
- Информация о цели или множестве целей компонента. Знак компонента связан с элементом множества *целей* отношением *цель обучающего курса**.
- Множество ключевых понятий предметной области, которые доступны для изучения в рамках *обучающего курса*. Знак компонента связан с множеством *sc-элементов* отношением *ключевые sc-элементы обучающего курса**.
- Множество других обучающих курсов, разделов, тестов и др., которые должны быть изучены, выполнены до начала изучения данного объекта. Связь компонента со знаком предварительного условия при помощи отношения *предварительное условие**. Отношение является строгим, его выполнение обязательно.
- Информация о последовательности предоставления информации для изучения. Единицей информации является элемент множества *раздел обучающего курса*. Разделы могут составлять иерархию. Самый низкий уровень иерархии является минимальной единицей учебного материала. Разбиение разделов по уровням иерархии осуществляется

посредством отношения *подраздел**. При выводе учебной информации по заданному разделу также включаются разделы более низких уровней иерархии. Последовательность изучения разделов определяется отношением *приоритет**. Оглавление обучающего курса формируется на основании отношений *подраздел** и *приоритет**. Знак раздела может быть связан со знаками других разделов или обучающих курсов отношением *предварительное условие**. Фрагмент разбиения обучающего курса на разделы представлен на рисунке 1.

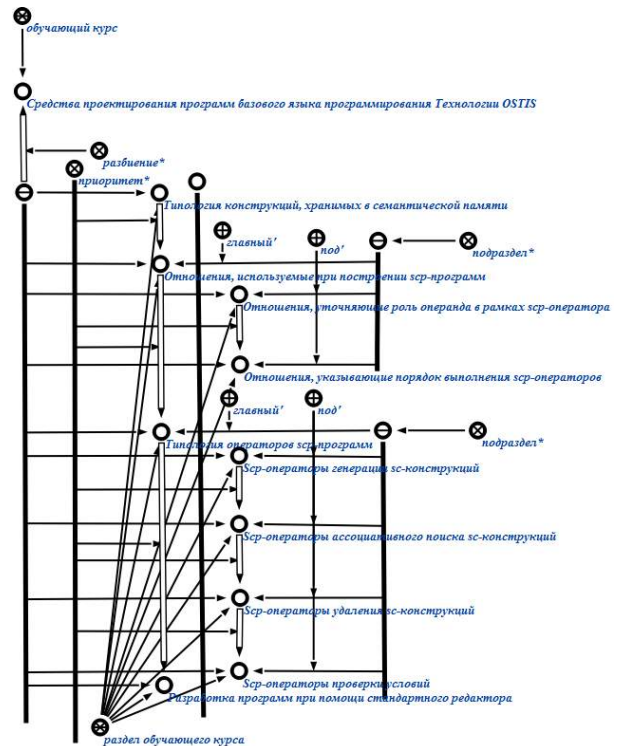


Рисунок 1 – Фрагмент разбиения обучающего курса на разделы

- Спецификация *раздела обучающего курса* включает:
 - Наименование раздела. Связь посредством отношения *наименование**.
 - Множество ключевых понятий предметной области, которые доступны для изучения в рамках *раздела обучающего курса*. Связь посредством отношения *множество ключевых элементов раздела**. Множество ключевых элементов раздела является подмножеством ключевых элементов обучающего курса. Учебный материал формируется из содержимого *sc-элементов*, связанных со знаком ключевого понятия отношениями: *определение**, *пояснение**, *детальное пояснение**, *комментарий**, *пример**, *обучающая информация**. Выбор связки для вывода учебной информации осуществляется на основании модели пользователя.

- Множество тестов для определения уровня усвоения раздела пользователем. Связь посредством отношения *тест раздела**.

Обучающий курс может быть декомпозирован на множество более простых курсов (подкурсов). Для курсов, которые разбиваются на подкурсы, наличие информации о разделах не обязательно. Для курсов самого низкого уровня иерархии наличие информации о разделах обязательно. Курс считается изученным, если достигнуты его цели, а также цели всех подкурсов.

Многokrратно используемые компоненты баз знаний, описывающие модель пользователя интеллектуальной системы, являются *неатомарными многократно используемыми компонентами*. Для представления пользователя используется компонент, представленный в виде *sc-окрестности*, характеризующей его формальную спецификацию, а также используются компоненты в виде *статической онтологии* и *динамической онтологии*. Пользователем интеллектуальной системы может быть физическое лицо, коллектив, другая интеллектуальная система.

Рассмотрим *sc-окрестность* объекта физическое лицо. Она включает:

- Идентификационные данные.
- Общую информацию: персональные данные (фамилия, имя, отчество), возраст, пол и др.
- Информацию о психологических характеристиках.
- Информацию об индивидуальных особенностях, связанных с динамическими аспектами деятельности.
- Информацию об индивидуальных особенностях, связанных с процессами познания и мышления.
- Информацию о способностях.
- Лингвистические характеристики.
- Информацию о целях. Цели могут быть иницированными, т.е. подлежащими достижению в текущий период времени, и неиницированными. Иницированные цели отмечаются атрибутом *иницированный'*.
- Представление уровня знаний, умений или интересов. Для представления используется оверлейная модель, которая описывает знания, умения и интересы как подмножество *эталонной модели предметной области*. Уровень знаний, умений и интересов оценивается по заданной абсолютной или вероятностной шкале. Оценка уровня знаний может определяться в ходе изучения учебного материала, тестирования пользователя или решения задач. Оценка уровня умений определяется в ходе тестирования пользователя или в ходе решения задач. Оценка степени интереса определяется из количества запросов к данному *sc-элементу*.

- Информацию о совершенных ошибках.
- Историю взаимодействия пользователя с интеллектуальными системами.

Для представления формальной спецификации в *sc-коде* используется следующий набор *sc-элементов*: *полное имя**, *имя'*, *фамилия'*, *отчество'*, *возраст**, *пол*, *эмоциональный тип*, *тип темперамента*, *когнитивная способность*, *способность**, *лингвистические характеристики**, *родной язык'*, *знакомый язык'*, *предпочитаемый язык'*, *цель взаимодействия**, *уровень знаний**, *уровень умений**, *интересы**, *история пользователя**.

Фрагмент формальной спецификации, характеризующей компонент физического лица, представлен на рисунке 2.

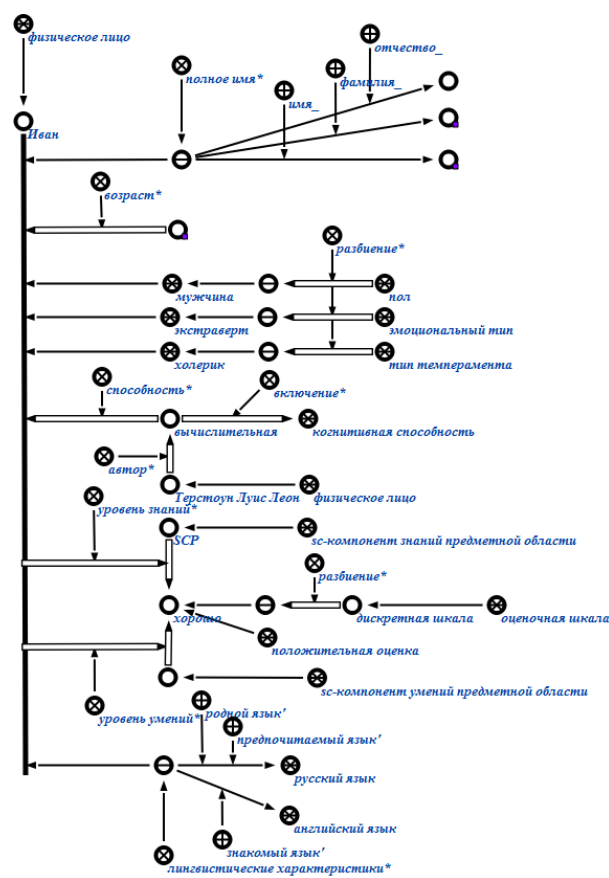


Рисунок 2 – Фрагмент формальной спецификации компонента

Одна часть из представленных характеристик не изменяется в различные периоды времени (статические характеристики), другая часть может изменяться (динамические характеристики). Изменяемые характеристики могут быть зависимы от процесса взаимодействия с интеллектуальной системой и не зависимы. Для описания данных характеристик используются компоненты: *статическая онтология* и *динамическая онтология*.

Многokrратно используемые компоненты баз знаний, описывающие стратегии управления обучением и консультированием, являются *неатомарными многократно используемыми компонентами* или *атомарными многократно*

используемыми компонентами. Компонент представляет собой шаблоны нескольких взаимосвязанных *sc-структур*, которые описывают последовательность обучения и применение обучающих воздействий к пользователю. *Sc-структуры* создаются в *sc-памяти* в процессе взаимодействия пользователя и интеллектуальной системы.

Центральным элементом *sc-структур* является пользователь, который помечен как активный. Активным может быть только зарегистрированный пользователь. Каждый зарегистрированный пользователь связан с ней отношением *пользователь**. Включение знака пользователя во множество пользователей интеллектуальной системы подразумевает наличие знаний о нем. Знания могут быть различного рода в зависимости от вида объекта, которым является пользователь. Например, если пользователем является физическое лицо, то обязательно наличие персональной информации, индивидуальных характеристик, квалификации.

Первой формируемой *sc-структурой* является сеанс пользователя. Он включает знак интеллектуальной системы, знак активного пользователя, *sc-ссылки*, содержащие дату и время начала и окончания сеанса работы, знак последовательности действий пользователя в рамках сеанса. Роли *sc-ссылок* задаются атрибутивными отношениями *время начала'*, *время окончания'*. *Sc-структура* существует только в момент работы пользователя с системой, после завершения сеанса она становится элементом коротежа отношения *история пользователя**.

Второй формируемой *sc-структурой* является сценарий обучения, который определяет стратегию или множество стратегий для обучения пользователя. Он связывает знак активного пользователя, знак обучающего курса, статический сценарий и/или динамический сценарий обучения под атрибутами *статический'*, *динамический'*. Статический сценарий обучения формируется в виде *sc-конструкции*, которая включает последовательность пар «обучающее воздействие – элементы раздела обучающего курса». Динамический сценарий – в виде *scr-программы*. *Scr-программа* формирует последовательность пар в процессе обучения.

Формальная спецификация компонента задается следующими множествами и отношениями: *пользователь**, *активный пользователь**, *сеанс пользователя**, *время начала'*, *время окончания'*, *последовательность*, *начало'*, *окончание'*, *следующий'*, *доступный режим взаимодействия**, *сценарий обучения**, *обучающее воздействие*, *задача*, *действие*, *способы обратной связи**, *ознакомление с предметной областью**, *решение задач**, *активные задачи**, *шаблоны решений**, *объяснение решения задачи**, *тестирование**.

3. Многократно используемые компоненты машин обработки знаний

Многократно используемые компоненты машин обработки знаний, обеспечивающие обработку модели пользователя, являются *платформенно-независимым многократно используемыми компонентами*, *неатомарными многократно используемыми компонентами*. Компоненты включают множество *абстрактных sc-агентов*, обеспечивающих формирование, модификацию, анализ модели пользователя, а также предоставление знаний о пользователе другим компонентам. В соответствии с функциональными требованиями были выделены следующие *sc-агенты*:

- *sc-агент формирования (первичного заполнения) модели пользователя;*
- *sc-агент обучения модели пользователя;*
- *sc-агент определения уровня знаний пользователя;*
- *sc-агент определения уровня умений пользователя;*
- *sc-агент определения интересов пользователя;*
- *sc-агент определения мотивации пользователя;*
- *sc-агент запроса уровня знаний пользователя;*
- *sc-агент запроса уровня умений пользователя;*
- *sc-агент запроса интересов пользователя.*

Первичное формирование модели пользователя производится при регистрации. Затем изменение модели производится интеллектуальной системой. По запросу корректировку модели может выполнять преподаватель или сам пользователь.

Многократно используемые компоненты машин обработки знаний, обеспечивающие управление обучением и консультированием, являются *платформенно-независимым многократно используемыми компонентами*, *неатомарными многократно используемыми компонентами*. Компоненты включают множество *абстрактных sc-агентов*, обеспечивающих управление обучением и консультированием. Основной функцией компонентов является управление прохождением сценария обучения и формирование сеанса пользователя. В соответствии с функциональными требованиями были выделены следующие *sc-агенты*:

- *sc-агент запуска сеанса пользователя;*
- *sc-агент выбора режима взаимодействия;*
- *sc-агент формирования сценария обучения;*
- *sc-агент прохождения обучающего курса;*

- *sc-агент ознакомления с ключевыми понятиями;*
- *sc-агент тестирования;*
- *sc-агент решения задачи;*
- *sc-агент применения обучающих воздействий;*
- *sc-агент объяснения решения задачи;*
- *sc-агент контроля правильности разработанных программ;*
- *sc-агент вывода комментариев;*
- *sc-агент локализации ошибок;*
- *sc-агент рекомендации обучающих курсов;*
- *sc-агент определения уровня сложности обучающего курса;*
- *sc-агент корректировки сценария обучения;*
- *sc-агент завершения сеанса пользователя.*

Множественно используемые компоненты машин обработки знаний, обеспечивающие мониторинг и анализ деятельности пользователя, являются платформенно-независимыми множественно используемыми компонентами, неатомарными множественно используемыми компонентами. Компоненты включают множество абстрактных *sc-агентов*, осуществляющих мониторинг и анализ действий пользователя.

Мониторинг и анализ осуществляются на предмет соответствия действия или множества действий предыдущим ошибкам данного пользователя, типичным ошибкам пользователей, а также на предмет возникновения некоторых критических ситуаций. Множество типичных ошибок и шаблоны критических ситуаций оформляются в виде отдельных компонентов баз знаний. В случае возникновения таких ситуаций корректировка действий пользователя осуществляется компонентом, обеспечивающим управление обучением и консультированием. Наличие их расширяет функциональность системы, повышает комфорт работы пользователя, однако, не обязательно. В соответствии с функциональными требованиями были выделены следующие *sc-агенты*:

- *sc-агент сбора данных о действиях пользователя;*
- *sc-агент запроса ошибок пользователя;*
- *sc-агент запроса типичных ошибок;*
- *sc-агент запроса критических ситуаций;*
- *sc-агент оценки действий пользователя.*

4. Множественно используемые компоненты интерфейсов

Множественно используемые компоненты интерфейсов, обеспечивающие взаимодействие интеллектуальной системы с пользователем,

преподавателем, экспертом или с другими интеллектуальными системами, другими платформами, могут быть платформенно-независимым множественно используемыми компонентами или платформенно-зависимым множественно используемыми компонентами.

Такие компоненты обладают своей спецификой и их реализация зависит от требований конкретной интеллектуальной системы. Некоторые из требуемых компонентов могут существовать в Библиотеке множественно используемых компонентов Технологии OSTIS. В таком случае они просто интегрируются в интеллектуальную систему.

Заключение

В данной статье обоснована необходимость создания компонентно-ориентированных открытых ИОС и ИУС в области разработки программ, ориентированных на обработку баз знаний, что позволит расширить круг разработчиков. А также, показана необходимость их интеграции со средами разработки программ, что обеспечит не только дополнительное удобство разработчику, но и поспособствует росту его квалификации.

В качестве КСО, удовлетворяющего вышеперечисленным требованиям, предлагается интеллектуальная система консультационного обслуживания и обучения, построенная на базе Технологии OSTIS. В работе представлена компонентная архитектура системы. А также, семантические модели множественно используемых компонентов баз знаний, машин обработки знаний и интерфейсов.

Библиографический список

- [Попов и др., 1996] Попов Э.В., И.Б. Фоминых, Е.Б. Кисель, М.Д. Шапот. Статические и динамические экспертные системы, учебное пособие. - М.: «Финансы и статистика», 1996. -318 с.
- [Стефанюк, 2004] Стефанюк, В. Л. Локальная организация интеллектуальных систем / В. Л. Стефанюк. - М.: ФИЗМАТЛИТ, 2004. - 328 с.
- [Стефанюк и др., 2007] Стефанюк, В. Л. Сотрудничающий компьютер : монография / В. Л. Стефанюк, А. В. Жожикашвили. - М.: Наука, 2007. - 274 с.
- [OSTIS] Открытая семантическая технология компонентного проектирования интеллектуальных систем [Электронный ресурс]. - Электронные данные. - Режим доступа: <http://ostis.net/mediawiki/index.php/>.
- [IMS] Открытая семантическая технология компонентного проектирования интеллектуальных систем [Электронный ресурс]. - Электронные данные. - Режим доступа: <http://ims.ostis.net/>.
- [Голенков и др., 2013] Голенков, В. В. Открытый проект, направленный на создание технологии компонентного проектирования интеллектуальных систем / В.В. Голенков, Н. А. Гулякина // Материалы международной научно-технической конференции «Открытые семантические технологии проектирования интеллектуальных систем» - Минск : БГУИР, 2013. - С. 55-78.
- [Гракова и др., 2014] Гракова, Н. В. База знаний интеллектуальной метасистемы поддержки Проектирования интеллектуальных систем / Н. В. Гракова, И. Т. Давыденко, К. В. Русецкий // Материалы международной научно-технической конференции «Открытые семантические технологии проектирования интеллектуальных систем». - Минск: БГУИР, 2014. - с. 83 - 92.

[Шункевич, 2014] Шункевич, Д. В. Машина обработки знаний интеллектуальной метасистемы поддержки проектирования интеллектуальных систем / Д. В. Шункевич // Материалы международной научно-технической конференции «Открытые семантические технологии проектирования интеллектуальных систем». – Минск: БГУИР, 2014. – с. 93 – 96

[Корончик, 2014] Корончик, Д. Н. Пользовательский интерфейс интеллектуальной метасистемы поддержки проектирования интеллектуальных систем / Д. Н. Корончик // Материалы международной научно-технической конференции «Открытые семантические технологии проектирования интеллектуальных систем». – Минск: БГУИР, 2014. – с. 79 – 82.

[Стефанюк, 2002] Стефанюк, В. Л. Учить или учиться? / В. Л. Стефанюк // Новости искусственного интеллекта. – 2002. – №5. – с. 13-24.

[Brusilovsky, 2012] Brusilovsky, P. Adaptive Hypermedia for Education and Training / P. Brusilovsky // Adaptive Technologies for Training and Education. – Cambridge : Cambridge University Press, 2012. – p. 46-68.

[Рыбина, 2008] Рыбина, Г.В. Обучающие интегрированные экспертные системы: некоторые итоги и перспективы / Г.В. Рыбина // Искусственный интеллект и принятие решений. – 2008. – №1. – с. 22-46.

A COMPONENT-BASED ARCHITECTURE OF INTELLIGENT SYSTEMS OF CONSULTATION PROVIDING AND TRAINING OF SOFTWARE DEVELOPERS

Pivovarchyk O.

* *Baranovich State University, Baranovich,*
Belarus
pivovarchyk@tut.by

The given article explains the necessity of creation of intelligent computer means of training, having open multicomponent architecture. An intelligent system of consultation providing and training is offered as a computer means of this kind. The article presents general component architecture of intelligent system and semantic models of each component.

Introduction

Continuous progress of methods and tools for software development requires deep knowledge, skills and ability from programmers and quickly learn new technologies. Computer training tools are effective helpers in learning new technologies. Currently, there are many computer training tools in developing software for the processing of knowledge bases. The main disadvantages of the existing systems are architecture, monofunctionality, limited functionality when interacting with development environments. Solving problems is the development of learning tools with multi-component architecture.

The article presents a component-based architecture of intelligent systems of consultation providing and training of software developers. The system is designed on the basis of Technology OSTIS.

Main Part

An intelligent system designed on the basis of Technology OSTIS consists of reusable components of knowledge bases, reusable components of knowledge processing machines and reusable components of

interfaces. The architecture of the child's intellectual system includes:

- Reusable components that provide the basic functions of an intelligent system (the core of the knowledge base, the core of the knowledge processing machine, the core of interfaces).
- Reusable components that extend the basic functionality of an intelligent system. The necessary components can exist in *The Library of OSTIS reusable components* or can be developed.

Components that extend the basic functionality of an intelligent system of consultation providing and training of software developers are:

- reusable components of knowledge bases that describe the knowledge domain;
- reusable components of knowledge bases that describe an educational material;
- reusable components of knowledge bases that describe the user model of an intelligent system;
- reusable components of knowledge bases that describe strategies of training and consulting management;
- reusable components of knowledge processing machines that provide processing of an user model;
- reusable components of knowledge processing machines that provide training and consulting management;
- reusable components of knowledge processing machines that provide monitoring and analysis of user activity;
- reusable components of interfaces for communication an intelligent system with users, experts, teachers;
- reusable components of interfaces for communication an intelligent system with the others systems.

Each component may be included in any other intelligent system.

Conclusion

This article describes the necessity of the creation of component-oriented open Intelligent Training Systems and Intelligent Educational Environments that will expand the range of developers. It shows the necessity of their integration with development environments that will not only provide additional convenience to the programmers, but also will contribute to professional growth.

The component-based architecture of intelligent systems of consultation providing and training are proposed in this article. The semantic models of reusable components of knowledge bases, knowledge processing machines and interfaces are presented.



OSTIS-2015

(Open Semantic Technologies for Intelligent Systems)

УДК 004.8

ПРИНЦИПЫ ПОСТРОЕНИЯ КЛИНИЧЕСКОЙ СИСТЕМЫ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ НА ОСНОВЕ ТЕХНОЛОГИИ OSTIS

Каешко А.И.*, Маргунов Е.А.**

* *Белорусский государственный университет информатики и радиоэлектроники,
г. Минск, Республика Беларусь*

ondister@gmail.com

** *Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

margunov@tut.by

Рассмотрены основные цели клинических систем поддержки принятия решений и их принципиальная структура. Обозначены их основные ограничения и недостатки. Предложены подходы к созданию клинической системы поддержки принятия решений с использованием баз знаний на основе технологии OSTIS.

Ключевые слова: медицинская информационная система; система поддержки принятия решений; онтологии

ВВЕДЕНИЕ

Бурное развитие информатики и удешевление аппаратного обеспечения в конце 20 века явились факторами, обеспечившими создание различных информационных систем. Медицина стала одной из предметных областей, в которой создание и внедрение автоматизированных информационных систем всегда было приоритетной задачей в развитии предметной области [eHealth, 2013]. Начавшаяся в 70-х годах 20 столетия [VistA, 2014], информатизация медицины на каждом этапе своего развития использует новейшие достижения современной науки техники, в том числе и информатики, как для аппаратных, так и программных средств. Стремление медицинских провайдеров к повышению качества и снижению стоимости услуг привело к созданию большого количества разнообразных медицинских информационных систем (МИС). В настоящее время даже их классификация вызывает трудности, так как любая МИС это сложная, комплексная информационная система для автоматизации деятельности, а медицина является широкой и крайне сложно поддающейся формализации предметной областью. Одним из вариантов МИС является клиническая информационная система, состоящая из системы электронных медицинских карт, системы поддержки принятия решений и системы телемедицины [Walderhaug, 2005].

Именно клинические информационные системы обеспечивают накопление и управление информацией о состоянии здоровья пациентов. Клинические МИС должны состоять как минимум из системы управления медицинским документооборотом и системы поддержки принятия решений. Последняя при этом должна являться ключевой, так как именно она обеспечивает достижение конечных целей МИС: улучшение качества оказываемых услуг и уменьшение их стоимости. Эти цели должны достигаться выбором оптимальной тактики ведения пациента, то есть диагностики, лечения и реабилитации. Конечно, важная роль в достижении конечных целей отводится подсистемам прогнозирования и планирования, которые позволяют подготовить материальную базу клиники к оптимальному оказанию услуг, однако, как показывает практика, удешевление оказания услуг достигается именно оптимизацией деятельности медицинского персонала [Kobelt, 1999]. Система управления медицинским документооборотом должна обеспечивать систему поддержки принятия решений (СППР) необходимой информацией.

Системы поддержки принятия решений в клинической медицине можно разделить на два класса: клинических системы принятия решений (СПР) и системы поддержки диагностики (СПД) [Rodríguez-González, 2012]. Системы поддержки диагностики являются аппаратно-программными комплексами и достаточно хорошо развиты. Их

развитие тесно связано с достижениями медицинской физики и химии, компьютерной техники, и, несомненно, с такими направлениями искусственного интеллекта как машинное зрение и машинное обучение. Именно системы поддержки диагностики явились первыми медицинскими системами с искусственным интеллектом (система диагностики инфекций крови MYCIN).

Однако, несмотря на то, что подобные системы в состоянии определить изменение объективных показателей человеческого организма, они не в состоянии установить точный клинический диагноз, так как это требует анализа не только объективных данных, но и данных анамнеза жизни, болезни пациента, и проведения диагностического поиска, который недостижим для современных СПД.

Системы принятия решений в составе клинических медицинских информационных систем гораздо менее распространены. Такие системы должны самостоятельно выбрать оптимальную тактику ведения пациента, предоставив при этом обоснования такого выбора для оператора системы, то есть они не должны быть «черным ящиком», так как ответственность за принятие решения несет оператор медицинской системы – медицинский работник. Система при этом одновременно является и «учителем» для оператора, позволяя минимизировать его ошибки. Таким образом, клинические системы принятия решений являются верхним звеном эволюции медицинских информационных систем.

Создание клинической системы принятия решений является трудноразрешимой проблемой. Если в системах поддержки диагностики возможно применение математических методов и алгоритмов, то клиническая система принятия решений для корректной работы должна обладать знаниями из различных предметных областей медицины, базовыми (нормальная и патологическая анатомии, нормальная и патологическая физиологии, гистология) и клиническими (терапия, хирургия и т.д.). То есть обладать тем же уровнем знаний, что и врач. Но стоит понимать, что обладая только априорными знаниями, такая система будет, по сути, только «врачом-стажером». Для достижения уровня «врача-специалиста» ей необходимы собственные апостериорные знания, то есть необходимы как минимум две базы знаний.

Создание априорной базы знаний не вызывает особых методологических трудностей, хотя и потребует значительных временных затрат. Но создание апостериорной базы знаний требует стандартизации и метаописания всей накапливаемой медицинской информации.

На первый взгляд может показаться, что стандартизация электронного представления медицинских документов не так уж сложна. Достаточно взять утвержденные «бумажные» формы медицинских документов, составить списки их полей, их описание и способы реализации. Но

здесь существует две основные проблемы. Во-первых, форм медицинского учета довольно много и они могут различаться в различных системах здравоохранения. Во-вторых, число форм и их полей всегда конечно, и необходимость добавления или удаления новых полей будет приводить к необходимости изменения стандартов и, следовательно, к изменению архитектуры МИС. Отклонение же от стандартов при построении МИС будет приводить к нарушению интероперабельности между информационными системами. Стоит помнить о том, что стандарты никак не описывают способ и техническую реализацию хранения информации.

С метаописанием хранимых данных дело обстоит гораздо сложнее: для работы системы принятия решений данные, сохраняемые системой документооборота, должны подвергаться машинной обработке, то есть в оптимальном случае, должны помещаться в базу знаний. Сама база знаний должна использовать систему метаданных, контролируемые словари и онтологии для получения новых знаний на основе уже имеющихся. Однако стандарты, использующие систему метаданных, словари и онтологии в настоящее время отсутствуют в странах СНГ. Из переведенных с английского языка в Республике Беларусь широко используется только международная классификация болезней 10-го пересмотра (МКБ-10). В Казахстане и в Российской Федерации идет работа по переводу стандарта обмена медицинской информацией HL7 (Health Level 7) и онтологии клинических терминов SNOMED CT. При этом Эстония уже перешла на МИС на основе стандарта HL7. В Российской Федерации была принята Концепция создания единой государственной информационной системы в сфере здравоохранения (утверждена приказом Министерства здравоохранения и социального развития Российской Федерации от 28 апреля 2011 № 364). В Республике Казахстан принята к реализации Концепции развития электронного здравоохранения на 2013-2020 годы, которая была разработана с учетом ключевых приоритетов отрасли, отраженных в Стратегии «Казахстан-2050» и в государственной программе «Информационный Казахстан-2020» и подразумевает использование в качестве базовых стандартов HL7.

Без гибких стандартов МИС, которые не только будут описывать, что и как хранить, но стандартизировать технологии построения МИС, без контролируемых словарей, классификаций, медицинских онтологий, переведенных на русский язык, которые будут учитывать особенности организации национального здравоохранения, невозможно создать полноценную систему принятия решений. А без нее нельзя говорить о полноценной клинической медицинской информационной системе вообще.

1. Системы поддержки принятия решений.

Многообразие входных параметров и сложность формализации предметной области требуют использования СППР, которые позволяют снять со специалистов нагрузку по составлению и поддержке базы знаний, обобщать и классифицировать накопленную информацию, применять правила или алгоритмы выбора пути решения конкретной медицинской задачи. В зависимости от объемов исходных данных, специфичности и изолированности признаков, наличия или отсутствия общепринятых методик диагностики, используются различные подходы при построении таких систем:

- применение нейронных сетей для обобщения и классификации данных;
- применение нечеткой логики;
- использование эвристик и формальных алгоритмов, экспертные системы;
- системы, основанные на базах знаний.

Искусственные нейронные сети (ИНС) применяются в клинических СППР (в основном, СПД) для отнесения представленных признаков к одной из категорий, выделенных на этапе обучения. Нейронные сети обобщают различные входные данные в контексте предыдущей истории обучения на ограниченной эталонной выборке данных и предоставляют клинически значимый выход, к примеру, вероятность определенной патологии или класс биомедицинского объекта. Ввиду значительной вариативности входных данных, ИНС хорошо зарекомендовали себя в анализе образцов крови и мочи пациентов с диабетом, туберкулезом, лейкозией, а также в задачах распознавания снимков тканей и рентгенограмм.

Недостатками нейронных сетей являются необходимость в обучающей выборке, собственно процессе обучения системы, ограниченность предметной области конкретной системы.

Недостатками нейронных сетей являются необходимость в обучающей выборке, собственно процессе обучения системы, небольшое число результирующих классов (предметная область конкретного классификатора не может быть широкой). Нейронная сеть представляет собой "серый" ящик: во-первых, топология сети здесь задается исходя из эвристических соображений и, во-вторых, в натренированных сетях со сложной топологией веса сотен и тысяч межнейронных связей не поддаются анализу и интерпретации человеком. Нейронные сети позволяют обобщить множество входных параметров, классифицировать вектор признаков на основании ограниченного набора альтернатив, для которых производилось обучение нейронной сети, а также (в зависимости от конкретного вида нейронной сети) оценить вероятность отнесения конкретного вектора признаков к тому или иному классу. Достоинством и

недостатком нейронной сети является получение результата без выделения решающих правил: с одной стороны, это позволяет принимать решения только на основании прецедентов, без глубокого анализа, с другой стороны, невозможно выделить решающее правило системы при получении ответа. Тем более это критично при условии, что практически невозможно достичь стопроцентной эффективности метода [Amato, 2013].

Системы, основанные на нечеткой логике, используют общую идею моделирования процесса постановки диагноза или классификации объекта специалистом. При этом существует определенная последовательность или набор признаков, необходимых для правильной классификации, а сами признаки можно разделить на несколько категорий. Так, например, Holzmann и другие рассматривают три категории признаков: ассоциативные, неассоциативные и исключающие, каждый признак может иметь собственный вес и отражает вклад в общую меру принадлежности признака тому или иному классу [Holzmann, 1988]. Модель принятия решений в данном случае является иерархической, первичная классификация выборки предоставляет набор промежуточных состояний, которые также могут иметь свой набор признаков. Последовательные переходы от одного состояния к другому определяются процедурой диагностики и в результате сводятся к одному из результирующих состояний.

Применение нечеткой модели позволяет использовать некантованные значения шкалы состояния того или иного признака, опираться при оценке на субъективную информацию о состоянии, полученную непосредственно от пациента. При этом необходимо подчеркнуть, что ввиду нелинейности такой модели принятия решений, результат в задачах диагностики в значительной степени зависит от текущего состояния пациента. Важным преимуществом модели по сравнению с ИНС является возможность проследить последовательность действий, приводящих к тому или иному результату, скорректировать процесс поиска решения без полного переобучения системы.

Системы, основанные на формальных алгоритмах диагностики, используют заранее известный набор критериев и четкие правила их распознавания, полученные от специалиста в прикладной области. Но здесь проявляется теперь общеизвестный парадокс - чем выше квалификация специалиста, тем менее он способен объяснить свои рассуждения. Поэтому данный подход применим лишь в частных случаях, однако он имеет и ряд достоинств:

- для каждого критерия можно предоставить исчерпывающую информацию по его оценке, включая фото, видео, статистические данные, возможные отклонения;
- поскольку процедура строго формализована, результат диагностики в меньшей

степени требует высокой квалификации проводящего оценку персонала;

- процедура диагностики в большей степени специфична для конкретного случая, позволяет оценить прогресс заболевания и предоставить при необходимости конкретные рекомендации после каждого шага диагностики.

В качестве иллюстрации можно рассмотреть автоматизированную ИС на базе Гентских критериев диагностики синдрома Марфана [Рудой, 2012]. Подобные системы могли бы широко использоваться для автоматизации рабочего места как врача общей практики, так и врача-специалиста, если бы существовал стандарт реализации алгоритмов в виде программных компонентов.

Несмотря на то, что развитие клинических СППР началось в 1970-х годах, к настоящему времени эта область научных разработок находится в самом начале своего развития. Однако появление суперкомпьютера IBM Watson, прошедшего стадию клинической апробации, и объявление IBM о начале разработки мобильных приложений для врачей и пациентов начинает революцию в этом сегменте. База знаний «медицинской части» Watson включает в себя как априорные знания (3 469 книг, 69 руководств, 247 460 статей журналов), так и апостериорные знания (605 000 обоснований диагнозов, 2 млн. страниц историй болезней, 25 000 учебных случаев и 14 700 часов настройки точности принятия решения) [Forbes, 2013]. Эти две базы знаний используют свыше 30 медицинских классификаций, онтологий, а также более 10 мапингов между ними и внешними данными, для поддержки принятия решений в онкологии (первоначально тактики лечения рака легких). Точность предлагаемой тактики лечения пациента достигает 90%. При этом решение основывается на «доказательной медицине». Учитывая тот факт, что основным реселлером Watson на рынке является медицинская страховая компания WellPoint, становится понятна основная роль этой системы – удешевление медицинских услуг путем повышения их эффективности.

Российская МИС Socmedica [Socmedica, 2014] так же претендует на революционность благодаря своей Объединенной Базе Медицинских Знаний (УМКВ). Это семантическая модель представления знаний, где используются боковые связи между признаками, весовые мультивариантные отношения между концептами, принцип нечёткой логики и сочетание нескольких уровней семантических систем, что в совокупности позволяет представлять сложные слабо формализованные медицинские знания [УМКВ, 2014].

Компанией Socmedica были разработаны новые медицинские классификаторы, что было продиктовано отсутствием единообразно построенных классификаторов медицинских терминов, понятий, критериев и нозологий. Эти классификаторы структурированы и активно

наполняются знаниями, между ними формируются определенные связи [Бледжянц, 2013].

Для наполнения классификаторов и структурирования связей между ними была разработана система модерации, наполнение осуществляется по принципам краудсорсинга, что решает проблему доверия к знаниям семантической сети. Система распределяет задачи среди экспертов – врачей и биологов, а из поступающих фрагментов информации формирует семантическую сеть. Информация в систему вводится не только в текстовом виде на разных языках, но и в виде логических связей с той или иной вероятностью. За счет этого система может усреднять мнения неограниченного количества врачей, создавая «коллективный разум».

Сегодня общее количество медицинских признаков (терминов без учета синонимов), представленных в УМКВ, составляет более 850 тысяч. И более двух миллионов родовидовых отношений, формирующих структуру классификаторов. Это соизмеримо с клинической классификацией медицинских терминов SNOMED CT. Система классификаторов УМКВ является динамической и может постоянно обновляться за счет добавления новых элементов и формирования медицинских признаков. Также создатели системы анонсировали возможность анализа электронных медицинских карт любой структуры для создания апостериорных баз знаний.

СППР, построенная на такой базе знаний сможет полноценно решать задачи прогнозирования рисков заболеваний, ранней диагностики и планирования лечения, то есть определять тактику ведения пациента. При этом создатели системы не ограничивают ее применение какой-либо одной клинической областью.

2. Недостатки и ограничения современных клинических систем поддержки принятия решений, на основе баз знаний.

Применение баз знаний решает основную проблему СППР – их узкую направленность на решение конкретных, как правило, диагностических задач. Наличие и наполнение апостериорных баз знаний добавляет системе возможность обучения.

Осознавая преимущества и возможности СППР на основе баз знаний, необходимо учитывать ограничения их применения, в первую очередь это скорость работы системы. Также неясно, как будет осуществляться диалог СППР-врач-пациент и непонятно, стоит ли предоставлять право использования СППР пациенту непосредственно, например, предоставлять онлайн консультации без участия врача. В отдельных случаях, согласно статистике, точность постановки диагноза, например, Watson, на треть выше, чем у опытных врачей [Forbes, 2013], что оставляет открытым

вопрос, как же поступать в случаях несогласия врача-эксперта с предложениями СППР по тактике ведения пациента.

С учетом развития технологической и законодательной базы, можно определить основную роль СППР в осуществлении всеобъемлющей информационной поддержки при постановке клинического диагноза и выборе тактики лечения пациента. При этом уровень интеллектуальности системы будет определяться ее способностью к синтезу новых знаний на основе априорной и апостериорной баз знаний.

Если ограничения СППР носят временный характер, и являются препятствием их применения, то недостатки препятствуют развитию таких систем.

Основной недостаток существующих СППР состоит в их полной обособленности от систем медицинского документооборота. Для наполнения апостериорной базы знаний им необходим конвертор электронных медицинских записей в семантическую сеть. Компании-разработчики СППР создают такие компоненты, кроме того, используются компоненты экспорта линейных медицинских текстов в базу знаний. И если проект Watson использует данные электронных медицинских карт, основанные на определенных стандартах представления медицинских данных (например, HL7), которые используют онтологии, классификаторы и контролируемые словари, то система Socmedica выбирает данные из различных баз (и линейного текста), и адаптирует их под собственную систему метаданных.

В обоих случаях для работы СППР используется семантическая сеть, полученная путем предварительной обработки данных. При такой архитектуре в первую очередь появляется проблема согласованности систем метаданных, справочников, классификаций и онтологий внешней МИС и СППР в условиях постоянного их изменения. При этом адаптируются, как правило, разработчики СППР, так как они предоставляют продукт разработчикам и пользователям сторонних МИС. Таким образом, такой подход не решает проблему стандартизации представления медицинских данных, а усугубляет ее.

Если рассматривать процесс разработки любой СППР, то можно заметить ряд серьезных недостатков современных информационных технологий. К числу таких недостатков, в частности, относятся:

- Отсутствие общего унифицированного решения проблемы семантической совместимости компьютерных систем, что порождает высокую трудоемкость создания комплексных интегрированных компьютерных систем.

- Высокая степень зависимости архитектур компьютерных систем от платформ, на которых они реализованы и в результате большая трудоемкость

переноса компьютерных систем на новые платформы.

- Отсутствие хорошо продуманной методики конструктивного использования опыта завершенных разработок компьютерных систем, что порождает высокую степень дублирования разработок различных компонентов этих систем.

- Отсутствие хорошо продуманной унифицированной методики обновления, изменения и развития разработанных компьютерных систем в процессе их сопровождения, что порождает высокую трудоемкость такого процесса [Голенков 2013].

Таким образом, главной технической проблемой является трудоемкость наполнения базы знаний. Из-за отсутствия единого стандарта проектирования и закрытости большинства существующих систем их развитие происходит слишком медленно, а возможность интеграции баз знаний различных СППР отсутствует. На сегодняшний день существует множество подходов реализаций СППР основанных на базах знаний, однако общей чертой клинических СППР является отсутствие единой технологии их создания.

3. Подход для создания клинической системы поддержки принятия решений на основе технологии OSTIS

Для устранения основных недостатков современных СППР, основанных на базах знаний, необходимо использовать технологию, которая позволит создать единую систему включающую систему документооборота и СППР на основе единой базы знаний. Создание системы документооборота подразумевает введение электронной медицинской карты пациента, которая будет содержать в себе всю доступную на текущий момент информацию о пациенте, включая записи обследований у конкретных специалистов, результаты анализов (как графические, так и текстовые) и данные о здоровье предоставленные самим пациентом или его доверенными лицами (например, родителями). Обобщенная модель такой системы представлена на рисунке 1:

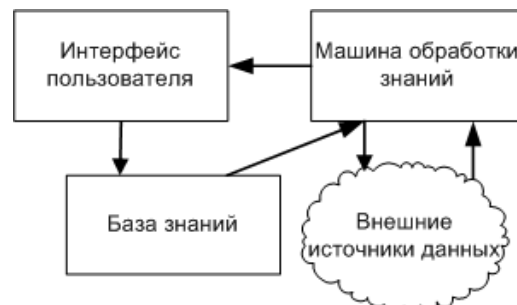


Рисунок 1 – Модель клинической МИС на основе единой БЗ

Ключевая роль в представленной модели отводится базе знаний. Она состоит из двух частей: априорной и апостериорной. Однако наполнение ее апостериорной части не обеспечивается работой

конверторов и трансляторов данных из баз данных и линейных текстов, а осуществляется напрямую врачебным персоналом при заполнении электронных медицинских карт. Для этой цели интерфейс пользователя предоставляет доступ к базе знаний.

Главное требование к пользовательскому интерфейсу в контексте данной системы это возможность его простой адаптации под конкретную группу пользователей (пациент, врачебный персонал, средний и младший медицинский персонал, регистраторы), а так же под любые формы медицинского учета. Эта гибкость достигается за счет того, что пользовательский интерфейс сам по себе является частью базы знаний.

За процесс принятия решений отвечает машина обработки знаний. База знаний также содержит ссылки на внешние источники данных. Стоит отметить, что эти данные служат источником информации только для человека и в машинной обработке не участвуют.

С учетом описанных в предыдущем разделе недостатков существующих СППР основанных на базах знаний для создания такой системы мы предлагаем использовать открытую семантическую технологию проектирования интеллектуальных систем (OSTIS) [Голенков, 2013]. Использование инструментов Semantic Web, несмотря на свою популярность, в данном случае имеет ряд недостатков. В первую очередь они связаны с тем, что инструменты Semantic Web изначально были нацелены на машиноориентированное описание информационных ресурсов в web-пространстве без учета комплексного решения проблем семантического представления с точки зрения теории искусственного интеллекта.

В противоположность инструментам Semantic Web инструменты проекта OSTIS имеют строгую теоретико-множественную трактовку и не привязаны к конкретной прикладной области. Что обеспечивает более компактное и формально точное представление информации [Голенков, 2011].

Это определяется рядом свойств, которые позволяют говорить о языковых средствах проекта OSTIS, как наиболее предпочтительном средстве интеграции знаний из различных источников:

- как и в языках Semantic Web в технологии OSTIS отдается предпочтение бинарным отношениям, однако существует возможность представления отношений любой арности;
- отношения представляются в виде узлов семантической сети, что позволяет характеризовать их свойства;
- экземпляры отношений выделяются как отдельные узлы семантической сети, что дает возможность характеризовать каждый экземпляр отношения уникальным образом;

- в алфавите ключевых узлов и дуг имеются элементы для описания нечетких, негативных и нестационарных объектов.

Помимо той предметной области, на которой специализируется СППР она должна содержать в базе знаний также общие знания наличие которых обусловлено необходимостью не только уметь решать задачи в заданной предметной области, но и эффективно взаимодействовать как с конечными пользователями, так и с разработчиками.

Таким образом, база знаний любой интеллектуальной системы, в том числе и нашей разрабатываемой клинической СППР представляет собой результат интеграции фактически нескольких баз знаний, каждая из которых описывает свою предметную область. К числу таких баз знаний относятся:

- база знаний, описывающая основную предметную область, в которой "специализируется" данная интеллектуальная система;
- система баз знаний, описывающих внешние языки и/или воспринимаемые образы внешней среды;
- база знаний, описывающая пользовательский интерфейс;
- семейство баз знаний, описывающих пользователей как субъектов, взаимодействующих с системой (как партнеров диалога);
- база знаний, описывающая процесс (в том числе историю) взаимодействия системы с внешней средой (в частности, с пользователями).

Технология OSTIS позволяет использоваться уже готовые разработки таких "общих" баз знаний в качестве ip-компонентов, что значительно ускорит процесс разработки нашей информационной системы.

Несмотря на то, что технология OSTIS в данный момент находится в процессе разработки, ее использование уже на данном этапе позволит решить ряд общих проблем разработки информационных систем и обеспечит определенные преимущества:

- использование совместимых компонентов (ip-компонентов) значительно ускорит разработку системы;
- технология OSTIS продолжает и развивает идею Semantic Web о представлении данные в виде семантической сети, что позволяет хранить в базе знаний любые данные, будь то текстовая информация о пациентах или графические результаты инструментальной диагностики;
- открытость и универсальность технологии позволит упростить интеграцию данных, повысить масштабируемость системы;
- проект планируется развивать так же за счет краудсорсинга, что привлечет новые идеи и позволит пополнять библиотеку ip-компонентов с помощью сторонних разработчиков.

С использованием технологии OSTIS на первом этапе будет создана система медицинского документооборота. В ее основе будет находиться априорная база знаний с системой метаданных, а так же с данными, импортированными из формата RDF свободных словарей и медицинских классификаций, например МКБ-10 и связями между ними. Помимо этого в базе знаний будут находиться ссылки на понятия из проекта DBPedia, направленного на извлечение структурированной информации из данных хранящихся в RDF-формате, созданных в рамках проекта Википедия.

Онтологии, построенные с использованием модели RDF, имеют богатые возможности описания знаний, но обладают весьма ограниченными средствами вывода следствий из имеющихся знаний. В RDF все знания должны храниться явно в виде триплетов (фактов, аксиом).

Таким образом, нужно осуществить трансляцию RDF данных в SC-код поддерживаемый технологией OSTIS. Уже разработан и используется транслятор позволяющий осуществлять трансляцию RDF графов в SC-конструкции [Каешко, 2014]. Транслятор обладает высокой скоростью работы, обеспечивает полную семантическую и логическую эквивалентность исходных RDF-графов и получаемых SC-конструкций и является полностью автоматизированным.

С наличием данного транслятора можно говорить о реально существующей совместимости технологий Semantic Web и OSTIS на уровне линейных форматов представления знаний.

Наполнение априорной базы знаний для системы медицинского документооборота будет производиться как автоматизированным способом (трансляция данных из существующих баз знаний), так и вручную.

Пользовательский интерфейс, справочная информация, описание стандартов описания информации будет являться частью априорной базы знаний.

Апостериорная база знаний будет состоять из клинической информации, накапливаемой в процессе работы МИС. В ее основе будет находиться концепция шаблонов полей и документов с использованием классификаторов, справочников и системы метаданных априорной базы знаний. Принцип работы с такой системой будет напоминать «бумажные» формы, которые так привыкли заполнять врачи, но с удобными средствами автозаполнения и подсказок. Однако ввод линейного текста без метаописания будет практически недоступен.

Наполнение такой базы знаний позволит перейти ко второму этапу – созданию СППР на основе базы знаний, то есть созданию машины обработки знаний для принятия клинических решений.

ЗАКЛЮЧЕНИЕ

Несмотря на начавшиеся еще в 70-х годах прошлого века работы по созданию клинических СППР, в настоящее время не существует технологии для их создания и работы в широких областях медицины. Существуют отличные примеры решения частных диагностических и клинических задач посредством применения технологий машинного обучения, нечеткой логики, линейных алгоритмов. Созданная на основе баз знаний система Watson демонстрирует великолепные результаты в онкологии и в немедицинских предметных областях, что позволяет сделать вывод о том, что область применения данной системы зависит только от состава базы знаний.

Российский проект Socmedica демонстрирует схожие идеи, однако, оба проекта позиционируют себя как компоненты для работы с внешними медицинскими данными. Более того, оба проекта используют значительное число разнообразных технологий для своей работы.

Представленная в статье модель клинической МИС позволит решить проблемы стандартизации и представления медицинской информации, а так же в дальнейшем, по мере накопления знаний в апостериорной базе знаний позволит создать СППР, которая будет сама являться частью базы знаний.

Использование технологий проекта OSTIS позволит решить ряд общих проблем создания СППР и перейти к новому виду клинической информационной системы, основанной на знаниях, в которой не будет существовать искусственных разграничений между системой документооборота и СППР.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- [eHealth, 2013] eHealth и имена интернет-доменов в области здравоохранения [Электронный ресурс]. – 2014. – Режим доступа: http://apps.who.int/gb/ebwha/pdf_files/WHA66/A66_26-ru.pdf. – Дата доступа: 30.12.2014
- [VistA, 2014] Veterans Health Information Systems and Technology Architecture [Электронный ресурс]. – 2014. – Режим доступа: 2014<http://www.ehealth.va.gov/VistA.asp>. – Дата доступа: 30.12.2014
- [Walderhaug, 2005] Ståle Walderhaug, Marius Mikalsen MDA Support for Military Medical Crisis Information Systems (MMCIS) Joint OMG/HL7 Workshop on Interoperability among Healthcare Services, Washington DC, October 25-27, 2005
- [Kobelt, 1999] Kobelt G. Методы фармакоэкономического анализа: минимизация затрат // Клиническая фармакология и терапия. - 1999. - №2. - С. 50-51.
- [Rodríguez-González, 2012] Alejandro Rodríguez-González and others. Analysis of a Multilevel Diagnosis Decision Support System and Its Implications: A Case Study Computational and Mathematical Methods in Medicine Volume 2012 (2012), Article ID 367345, 9 pages
- [Amato, 2013] Filippo Amato and others. Artificial neural networks in medical diagnosis. Journal of applied biomedicine. 11: 47-58, 2013
- [Рудой, 2012] А.С.Рудой, А.Г.Горустович, А.Н.Полторан. Автоматизированная информационная система диагностики синдрома Марфана. Военная медицина: научно-практический рецензируемый журнал. - 2012. - N 4. - С. 86-91
- [Holzmann, 1988] Carlos A.Holzmann, Claudio A.Perez & Eduardo Rosselot. A fuzzy model for medical diagnosis/ Medical Progress through Technology 13: 171-178 (1988).

[Forbes, 2013] IBM's Watson Gets Its First Piece Of Business In Healthcare [Электронный ресурс]. – 2013. – Режим доступа: <http://www.forbes.com/sites/bruceupbin/2013/02/08/ibms-watson-gets-its-first-piece-of-business-in-healthcare/>. – Дата доступа: 21.12.2014

[UMKB, 2014] Объединенная база медицинских знаний [Электронный ресурс]. – 2014. – Режим доступа: <http://umkb.ru/>. – Дата доступа: 21.12.2014

[Socmedica, 2014] Экспертные системы для клинической практики [Электронный ресурс]. – 2014. – Режим доступа: <http://www.socmedica.com/>. – Дата доступа: 21.12.2014

[Бледжянц, 2013] Г.А. Бледжянц и др. Будущее информатизации здравоохранения: когнитивные системы. Здравоохранение: научно-практический рецензируемый журнал. – 2013. – N 8.

[Голенков, 2013] Открытый проект, направленный на создание технологии компонентного проектирования интеллектуальных систем. – В кн Междунар. научн.-техн. конф. «Открытые семантические технологии проектирования интеллектуальных систем» (OSTIS-2013). Материалы конф. [Минск, 21-23 февр. 2013 г.]. – Минск: БГУИР, 2013, с. 55-77.

[Каешко, 2013] А.И.Каешко, Д.Г.Колб. Принципы интеграции содержимого RDF-хранилищ в проект OSTIS – В кн Междунар. научн.-техн. конф. «Открытые семантические технологии проектирования интеллектуальных систем» (OSTIS-2014). Материалы конф. [Минск, 20-22 февр. 2014 г.]. – Минск: БГУИР, 2013, с. 55-77.

[Клешев, 2011] Клешев А.С., Черняховская М.Ю., Москаленко Ф.М. Модель онтологии предметной области «Медицинская диагностика». Часть 1. Неформальное описание и определение базовых терминов / Журнал НТИ - Серия 2. #12, 2005. - [http://www.iacr.dvo.ru/is/publications/Article1-Moskalenko-IACP\(NTI-2005\).rtf](http://www.iacr.dvo.ru/is/publications/Article1-Moskalenko-IACP(NTI-2005).rtf)

[Янковская, 1994] Тестовые распознающие медицинские экспертные системы с элементами когнитивной графики / Янковская А.Е. // Компьютерная хроника. – 1994. –С. 61-83.

[Кобринский, 2009] Кобринский Б.А. Нечеткий образный ряд в клинической медицине / Б.А. Кобринский // Интегрированные модели и мягкие вычисления в искусственном интеллекте: Сб. науч. тр. V-й Международной научно-практ. конф. Т.1. – М.: Физматлит, 2009. – С.121-127.

PRINCIPLES OF CONSTRUCTION CLINICAL DECISION SUPPORT SYSTEM BASED ON OSTIS TECHNOLOGY

Kayeshko A.I. *, Marhunou Y.A. **

** Belorussian state university of informatics
and radioelectronics, Minsk, Republic of Belarus*

ondister@gmail.com

*** Belorussian state university of informatics
and radioelectronics, Minsk, Republic of Belarus*

marhunou@tut.by

In the article authors reviewed main goals of clinical decision support systems and their basic structure. Marked their main limitations and drawbacks. The approaches to the creation of a clinical decision support system using knowledge bases based on OSTIS technology was proposed.

INTRODUCTION

Despite the rapid evolution of informational technologies and dynamical designing of health information system which was started at the end of 20th century, there was no universal intelligent clinical decision support system (DSS) created. Different

standards of storing and transferring the medical data, the lack of common technologies and complexity of the problem raise a lot of issues for developers.

The main idea of the article is a creation of health information system based on knowledge base, which will allow accumulating necessary amount of knowledge to build an intelligent clinical expert system.

MAIN PART

There are several ways how a DSS system could be implemented: artificial neural networks; using of fuzzy logic; expert system (formal and heuristic algorithms) and systems based on knowledge bases. Although, all of them have their advantages and disadvantages, systems based on knowledge look more preferable in the context of DSS systems mostly because of their flexibility and scalability.

The main drawback of the existing DSS is their complete isolation from the medical workflow systems. To fill a posteriori knowledge they need to convert electronic medical records to the semantic network. Therefore, the DSS developers have to create such components and, in addition, use nonlinear medical texts to export components in the knowledge base.

To address the main drawbacks of modern DSS based on knowledge bases, it is necessary to use the technology, which will create a document management system (DMS) and decision support system based on a common knowledge base.

CONCLUSION

The model of clinical health information system, presented in this paper solves the problems of standardization and reporting of health information. Based on the OSTIS technology and model the decision support system will be created, which itself will be a part of a knowledge base. That will let us go to a new kind of clinical information system based on the knowledge that will allow to get rid of artificial distinction between the DMS and DSS.



OSTIS-2015

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822:514

ПРОЕКТИРОВАНИЕ БАЗЫ ЗНАНИЙ ИНТЕЛЛЕКТУАЛЬНОЙ СПРАВОЧНОЙ СИСТЕМЫ ПО АЛГЕБРЕ

Шарипбай А.А., Омарбекова А.С., Нургазинова Г.Ш.

*Евразийский национальный университет им. Л.Н. Гумилёва,
г. Астана, Республика Казахстан*

sharalt@mail.ru

nurgasinova@mail.ru

omarbekova@mail.ru

В работе приводится описание проектирования базы знаний интеллектуальной справочной системы по алгебре, разрабатываемой на основе открытой семантической технологии компонентного проектирования интеллектуальных систем. Описывается начальный этап проектирования, на примере, частной предметной области чисел.

Ключевые слова: база знаний, интеллектуальная система, алгебра, числовые модели.

Введение

В работе рассматривается проектирование интеллектуальной обучающей системы по алгебре, которая разрабатывается на основе комплексной открытой технологии проектирования интеллектуальных систем OSTIS (Open Semantic Technology for Intelligent Systems) совместно с сотрудниками кафедры «Интеллектуальные информационные технологии» Белорусского государственного университета информатики и радиоэлектроники (г. Минск), под руководством д.т.н., профессора В.В. Голенкова.

Целью работы является разработка интеллектуальной обучающей системы по алгебре, основанной на технологии проектирования интеллектуальных систем OSTIS. В соответствии с технологией для достижения указанной цели необходимо решить следующие задачи:

- разработать базу знаний интеллектуальной обучающей системы по алгебре;
- разработать поисковую машину обработки знаний интеллектуальной обучающей системы по алгебре;
- разработать интеллектуальный решатель задач интеллектуальной обучающей системы по алгебре;
- разработать пользовательский интерфейс интеллектуальной обучающей системы по алгебре [Голенков и др., 2013].

В рамках данного доклада говорится о

разработке базы знаний интеллектуальной обучающей системы, вернее о начальном этапе проектирования.

1. Постановка задачи и выбор технологии

На сегодняшний день бесспорным остается факт востребованности мощных компьютерных средств обучения и контроля знаний, особенно, в период повсеместного внедрения в систему образования Республики Казахстан стандартов e-learning.

Наряду с весьма впечатляющими достижениями в рамках внедрения электронного обучения в нашей стране необходимо приложить огромные усилия для создания квалифицированного учебного контента, отвечающего возможности организации эффективного поиска необходимых знаний; возможностям интеллектуального анализа обработки результатов контроля знаний, а также создания систем, поддерживающих процесс обучения на всех его этапах. В общем случае контент – любое информационно значимое наполнение информационного ресурса – тексты, графика, мультимедиа и т. д. Под учебным контентом будем понимать знания некоторой предметной области, представленные в электронном виде [Кабак, 2008].

На современном этапе, когда объемы информации стремительно возрастают, появляется необходимость в создании таких средств поддержки электронных учебников, которые позволили

пользователю не только просматривать интересующую его информацию путем навигации по гипермедийным структурам, но и задавать различные более сложные вопросы [Голенков и др., 2010].

Таким образом, актуальны вопросы создания квалифицированного учебного контента, формализованного самыми современными методами искусственного интеллекта, с целью дальнейшего более эффективного и полноценного использования в учебном процессе.

В работе мы будем говорить о создании учебного контента для интеллектуальных средств обучения и контроля знаний, на основе семантического их представления.

В современных условиях создание такого контента связано с самыми последними достижениями в области искусственного интеллекта, в частности, со способами формализации знаний, т.е. использованием современных методов их представления и организации. Поэтому, принимаясь за разработку интеллектуальных средств обучения и контроля знаний, особое внимание нужно уделить представлению знаний. Именно выбранные методы представления знаний определяют правила структуризации и систематизации учебного материала, реализации навигационно-поисковых алгоритмов, организации управления обучением.

В настоящее время с целью преодоления недостатков и использования достоинств гипертекстовой представления информации, развиваются графовые и сетевые представления знаний, предполагающие структурирование и систематизацию учебных знаний в виде семантических сетей. Достоинство семантических сетей - обеспечение наглядности отображения объектов, связей и отношений. Графическое представление связей между понятиями позволяет пользователям эффективнее усваивать и запоминать информационную структуру предметной области. Один из способов применения семантических сетей для представления учебных знаний в виде s-конструкций предложен в рамках комплексной открытой технологии проектирования интеллектуальных систем OSTIS.

Основной отличительной особенностью и преимуществом этой модели и языка описания учебного материала является то, что они построены на теоретико-множественной основе и ориентированы на семантическое описание и обработку информации. Еще одним важным достоинством предлагаемых средств представления знаний является то, что в основу реализации механизмов переработки знаний изначально заложена возможность параллельной обработки информации [Кабак, 2008].

В настоящее время посвящено большое количество работ исследованию методов и форм представления знаний в интеллектуальных

обучающих системах. В целях последующего выбора наиболее подходящей методики необходимо проанализировать основные существующие зарубежные методы и модели представления знаний в интеллектуальных системах. В качестве основы изучены предлагаемые в рамках технологии OSTIS унифицированные семантические сети с базовой теоретико-множественной интерпретацией. В основе данной технологии предложен подход, направленный на создание массовой технологии быстрого проектирования интеллектуальных систем. Данный подход включает в себя ориентацию на семантическое представление знаний, унификацию моделей интеллектуальных систем, модульное проектирование на основе библиотек типовых многократно используемых компонентов интеллектуальных систем, поэтапное эволюционное проектирование на основе быстрого прототипирования, совместимость инструментальных средств проектирования с проектируемыми системами, включение в состав технологии проектирования интеллектуальных систем комплексной интеллектуальной help-системы, а также создание подсистем самотестирования [Колб, 2012].

2. Проектирование базы знаний

В нашей работе по проектированию интеллектуальной справочной системы по алгебре мы опираемся на комплексную открытую технологию проектирования интеллектуальных систем OSTIS, ориентированную на семантическую модель представления знаний [OSTIS, 2010]. Из всего курса алгебры рассмотрим предметную область чисел, в которой рассмотрим теории натуральных, целых, рациональных, действительных и комплексных чисел. В соответствии с технологией одной из главных задач, с которой начинается разработка интеллектуальных обучающих систем, является разработка базы знаний интеллектуальной справочной системы.

Выбранная предметная область чисел является статичной предметной областью, хорошо описанной в различных источниках.

Согласно [Голенков и др., 2011] семантическую структуру нашей базы знаний, описывающую некоторую предметную область, в нашем случае - предметную область чисел, можно рассматривать как иерархическую систему предметных областей различного специального вида, надстраиваемых над заданной предметной областью. Поэтому для более четкого и детального структурирования и наполнения базы знаний предметной области чисел мы условно разделим ее на несколько предметных областей, включающих одна другую (рисунок 1).

1, 2, ...	Предметная область натуральных чисел
0, 1, -1, ...	Предметная область целых чисел
$1, -1, \frac{1}{2}, \frac{2}{3}, 0,12, \dots$	Предметная область рациональных чисел
$1, -1, \frac{1}{2}, \frac{2}{3}, 0,12, \pi, \sqrt{2}, \dots$	Предметная область действительных чисел
$-1, \frac{1}{2}, 0,12, 3i + 2, e^{i\pi/3}, \dots$	Предметная область комплексных чисел

Рисунок 1 - Структура предметной области чисел

Пусть нам дан раздел предметной области «натуральные числа», объектами исследования которой являются числа. Это предметная область, которая включается в состав (является подобластью) предметной области чисел и объектами исследования которой являются натуральные числа. Опираясь на труды профессора Голенкова В.В. [Голенков и др., 2011] базу знаний, в частности по числовым моделям, которая является описанием, например, предметной области натуральных чисел условно разделим на следующие части:

- Структурная спецификация
- Предметная область теоретико-множественных связей между понятиями предметной области
=Теоретико-множественная онтология предметной области натуральных чисел
- Предметная область терминов предметной области натуральных чисел
=Терминологическая онтология предметной области натуральных чисел
- Предметная область определения понятий предметной области натуральных чисел
=Логическая система определения понятий предметной области
- Предметная область логических формул и высказываний о предметной области натуральных чисел
=Логическая онтология предметной области натуральных чисел
- Предметная область доказательств высказываний о предметной области натуральных чисел
=Логическая система доказательств о предметной области натуральных чисел
- Предметная область задач и решений задач предметной области натуральных чисел
=Сборник задач с решениями в предметной области натуральных чисел
- Предметная область классов задач и способов решения задач предметной области натуральных чисел
=Онтология задач в предметной области натуральных чисел.

Задав каждую из этих предметных областей, мы получим описание предметной области натуральных чисел. Далее мы описываем аналогичным образом предметные области целых чисел, рациональных чисел, вещественных чисел и комплексных чисел, которые будут надстраиваться над предметной областью натуральных чисел. В итоге получаем четко структурированную модель базы знаний предметной области чисел,

представленную в виде семантической сети. Для кодирования указанной семантической сети, описывающей различные виды знаний указанной предметной области, используется универсальный абстрактный язык семантических сетей SC-код (Semantic Computer Code).

Рассматривая предметную область как совокупность некоторых более частных предметных областей, представляющих собой набор ключевых понятий и отношений между ними [Давыденко и др., 2011], в предметной области числовых моделей выделим следующую их иерархию:

- Теория чисел
- Теория натуральных чисел
- Теория целых чисел
- Теория рациональных чисел
- Теория действительных чисел
- Теория комплексных чисел

После уточнения частных теорий, для каждой из них необходимо определить ключевые понятия, которые описывают рассматриваемую теорию. К примеру, для теории натуральных чисел выделены следующие ключевые понятия и узлы, являющимися отношениями:

Таблица 1 - Структурная спецификация предметной области натуральных чисел

Максимальный класс объектов:	натуральное число
Подклассы:	четное число, нечетное число, простое число, составное число
Отношения:	сложение, умножение, возведение в степень, НОК (наименьшее общее кратное), НОД (наибольший общий делитель), факториал числа, сравнение чисел (больше, меньше, равно), натуральный ряд, десятичное представление числа и др.

В соответствии с аналогичной работой по построению ИСС по геометрии [Давыденко и др., 2011], после выделения ключевых узлов, строится теоретико-множественная онтология понятий. Для этого используются такие теоретико-множественные отношения, как включение, разбиение, пересечение, объединение, принадлежность и др.

На рисунке 2 и 3 приведены фрагмент статьи SCn-кода и фрагмент на SCg-коде соответственно, описывающие Понятие натурального числа.

натуральное число

= натуральные числа

= множество натуральных чисел

= множество N

= понятие натурального числа

= класс натуральных чисел

= natural number

⊂ целое число

⊂ положительное целое число

- Разбиение по признаку четности и нечетности:

- четное число
- нечетное число

- Разбиение на:

- простое число
- составное число
- 1

Рисунок 2 – Фрагмент статьи SCn-кода, описывающий теоретико-множественные отношения понятия натурального числа

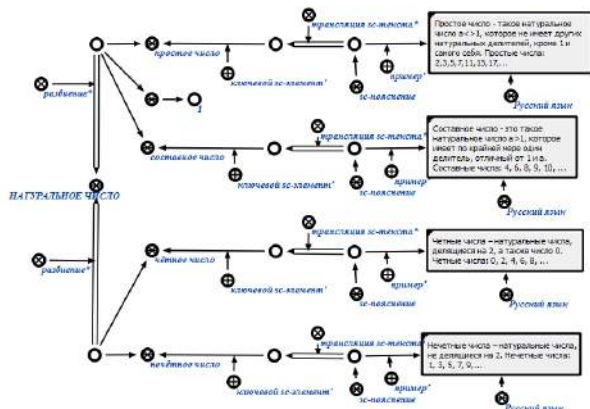


Рисунок 3 – Фрагмент статьи SCg-кода, описывающий теоретико-множественные отношения Понятия натурального числа

Далее онтология дополняется определениями, пояснениями для всех вводимых ключевых понятий. После определяются константы – понятия, посредством которого определяется ключевое понятие. Опираясь на данную информацию, строится логико-иерархическая система понятий предметной области чисел.

После построения логико-иерархической системы понятий база знаний пополняется утверждениями, описывающими свойства понятий, а также их доказательствами. Все утверждения строятся в логико-иерархическую систему утверждений на основе утверждений, входящих в их доказательство. В базе знаний также содержится описание различных классов задач и способов их решений.

Заключение

В основу проектирования интеллектуальных средств обучения должна быть положена строгая модель представления и организации учебного контента, которая будет четко согласована с методами автоматизации процессов его создания и обработки.

В результате данной работы положено начало проектированию и разработке интеллектуальной справочной системы по алгебре с использованием технологии компонентного проектирования интеллектуальных систем, в частности проектирования базы знаний на примере, частной предметной области чисел. Далее предполагается расширение базы знаний путем добавления в нее новых видов знаний. В перспективе предполагается разработка новых поисковых операций, операций интеллектуального решателя задач, компонентов пользовательского интерфейса.

Библиографический список

[Голенков и др., 2013] Голенков В.В., Гулякина Н.А. Открытый проект, направленный на создание технологии компонентного проектирования интеллектуальных систем // Материалы международной научно-технической конференции «Открытые семантические технологии проектирования интеллектуальных систем» (Минск, 21-23 февраля 2013г.) Минск: БГУИР, 2013. – с. 55-78.

[Кабак, 2008] Кабак Е.В. Методы и средства автоматизации разработки компьютерных обучающих программных модулей: дис. ... канд. техн. наук: 05.13.11/ Е.В. Кабак. – Минск, 2008. – 168 л.

[Голенков и др., 2010] Голенков В.В., Гулякина Н.А. Применение технологий искусственного интеллекта в обучении. Сборник научных статей: материалы Международной научной конференции «Четвёртые чтения, посвященные 70-летию со дня рождения В.А. Карпова» (19 – 20 марта 2010 г.). В 2-х частях. Минск, РИВШ, 2010, Ч.1. - Минск: РИВШ, 2010.

[Колб, 2012] Колб Д.Г. Модели, методы и средства разработки сайтов с переносимым сложноструктурированным контентом на базе семантических сетей: дис. ... канд. техн. наук: 05.13.11/ Д.Г. Колб. – Минск, 2012. – 133 л.

[OSTIS, 2010] Проект OSTIS [Электронный ресурс]. Минск, 2010. – Электронный адрес: ims.ostis.net

[Голенков и др., 2011] Голенков В.В., Гулякина Н.А. Принципы построения массовой семантической технологии компонентного проектирования интеллектуальных систем. // Материалы международной научно-технической конференции «Открытые семантические технологии проектирования интеллектуальных систем» (г. Минск, 10-12 февраля 2011) Минск: БГУИР, 2011. – с. 21-58.

[Давыденко и др., 2011] Давыденко И.Т. и др. Интеллектуальная справочная система по геометрии. // Материалы международной научно-технической конференции «Открытые семантические технологии проектирования интеллектуальных систем» (г. Минск, 10-12 февраля 2011) Минск: БГУИР, 2011.

KNOWLEDGE BASE DESIGN INTELLECTUAL REFERENCE SYSTEMS IN ALGEBRA

Sharipbay A.A., Omarbekova A.S., Nurgazinova G. Sh.

Eurasian national university of L.N. Gumilov, Astana, Republic of Kazakhstan

sharalt@mail.ru, nurgasinova@mail.ru
omarbekova@mail.ru

The paper describes the design of a knowledge base of intellectual reference system for algebra, projected on the basis of an open semantic technology component of designing intelligent systems. Describes the initial design phase, for example, private domain "numerical models."

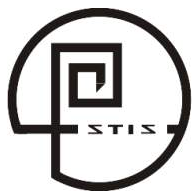
Introduction

In this paper the design of an intelligent tutoring system for algebra, which is developed on the basis of a comprehensive open technology design of intelligent systems OSTIS.

The aim is to develop intelligent tutoring system for algebra, based on the technology of intelligent systems OSTIS.

Conclusion

As a result of this work started work on the design and development of intelligent help system for algebra using the technology component of intelligent systems, in particular the design knowledge base as an example, the private domain "numerical models".



УДК 004.8

ПУТИ РАЗВИТИЯ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА В КАЗАХСТАНЕ

Шарипбай А.А.

*Евразийский национальный университет им. Л.Н. Гумилева,
г. Астана, Казахстан*

Sharalt@mail.ru

В докладе приводятся основные направления исследования по искусственному интеллекту в мире. Дается информация о исследованиях, проводимых в Казахстане. Обсуждаются проблемы и перспективы их развития.

Ключевые слова: искусственный интеллект; пути развития; речевые технологии.

Введение

Сначала уточним главное понятие, вынесенное в заголовок темы. Общепринято, что интеллект – это способность субъекта (производителя действия или носителя состояния) воспринимать (понимать), приспособиться (обучаться) и принимать решение (управлять). Если субъектом является человек, то речь идет о естественном интеллекте. Когда субъектом является искусственное устройство, то говорим об искусственном интеллекте. Искусственный интеллект – направление в области компьютерной науки, позволяющее обучить искусственных устройств самостоятельно войти в контакт с окружающим миром, понять этот мир, приспособиться к нему, а затем и управлять им.

Научный аспект искусственного интеллекта заключается в объяснении работы естественного интеллекта и имитация решения задач человеком с помощью искусственных устройств, а его прикладной аспект – решение компьютером сложных задач, не имеющих явного алгоритмического решения, порой с нечёткими целями или в условиях неопределенности.

К основным направлениям исследования по искусственному интеллекту, проводимые в разных странах и научных центрах можно отнести следующие [1-4]:

- автоматическое доказательство теорем: автоматическое доказательство конструктивных утверждений на основе заданных аксиом (актов);
- эвристическое программирование: разработка стратегии действий на основе теоретически не обоснованных правил, позволяющих сократить количество переборov в пространстве поиска;

- экспертные системы: представление знаний и моделирование действий конкретных экспертов;
- поведенческие системы: многослойное моделирование поведений объектов (роботов) в реальной физической среде;
- интеллектуальный поиск: представление знаний в виде семантических сетей и осуществление оптимального поиска;
- системы принятия решений: применения теории принятия решений и методов динамического планирования в реальном масштабе времени;
- обработка знаний: методы представления, хранения, извлечения, анализа и моделирования знаний;
- моделирование мышления: поиск моделей и методов мышления.

1. Искусственный интеллект в Казахстане

1.1. Направления исследования по искусственному интеллекту в Казахстане

Начиная с 70-х годов XX века в разных научных подразделениях проводились исследования по эвристическому программированию, распознаванию образов, экспертной системе, автоматизации синтеза и верификации программ, системы принятия решений и др. Были практические результаты, которые применялись в медицинских учреждениях по диагностированию болезни, химической промышленности по синтезу и идентификации химических веществ, Агро промышленности по составлению земельных кадастров, финансовых учреждения по прогнозированию курса валют и

ценных бумаг, образовательных и научных учреждениях по распознаванию речи, обработке естественных языков и др. [5-7].

В 2012 году в г. Астане на базе ЕНУ им. Л.Н. Гумилева был создан НИИ «Искусственный интеллект» (<http://www.e-zerde.kz/zerde/index.php/ru/>) с целью формирования необходимых финансовых, материальных, научно-методологических, нормативно-технических, социально-экономических условий для осуществления своей деятельности за счет средств, полученных от выполнения научно-исследовательских, опытно-конструкторских работ и других интеллектуальных услуг в области искусственного интеллекта..

Миссией института является превращение искусственного интеллекта в одного из приоритетных и прорывных направлений науки, техники и экономики страны.

Основные направления научных исследований НИИ «Искусственный интеллект»:

- Компьютерная логика: исследование в области компьютерной логики, создание языков спецификации для формализации описания и верификации программных систем;
- Компьютерное моделирование: исследование в области компьютерного моделирования идентификации, разработка и реализация алгоритмов анализа и диагностики различных объектов и процессов;
- Интеллектуальные информационные системы: исследование в области интеллектуальных информационных систем, создание и сопровождение базы знаний и систем поддержки принятия решений в различных отраслях экономики и управления;
- Речевые технологии: исследование в области речевых технологий, проектирование, создание и сопровождение систем распознавания и синтеза речи, разработка речевых интерфейсов;
- Интеллектуальные робототехнические системы: исследование в области робототехнических систем и мехатроники, проектирование и создание интеллектуальных роботов с микропроцессорным и микроконтроллерным управлением;
- Интеллектуальные обучающие системы: исследование и разработка государственных стандартов в области электронного обучения (e-learning), создание, экспертиза и сопровождение интеллектуальных электронных учебных изданий (e-manual) и систем управления учебным процессом (LMS - Learning Management System), а также проведение курсов повышения квалификации в области ИТК и инженерии знаний;
- Программная инженерия: исследование в области программной инженерии и разработка государственных стандартов в области информационных технологий и информационной

безопасности, экспертиза и администрирование программных систем, а также проведение курсов повышения квалификации для менеджеров проекта и бизнес аналитиков.

Перед НИИ «Искусственный интеллект» были поставлены следующие задачи:

- проведение фундаментальных и прикладных исследований в области искусственного интеллекта в соответствии с договорами уполномоченных органов страны и других отечественных и зарубежных субъектов;
- привлечение студентов, магистрантов и докторантов к реальной научной и инновационной деятельности в области искусственного интеллекта в рамках заключенных договоров;
- аккумуляция в университете талантливых молодых специалистов и вовлечение зарубежных специалистов в области искусственного интеллекта путем создания дополнительных рабочих мест за счет собственных средств;
- развитие и повышение эффективности научного, методического, нормативного и аппаратно-программного обеспечения учебного процесса на основе выполнения теоретических и практических работ в области искусственного интеллекта;
- участие в подготовке и повышения квалификации специалистов (бакалавров, магистров и докторов философии) в области искусственного интеллекта.

1.2. НИИ «Искусственный интеллект»

В 2014 году НИИ «Искусственный интеллект» проводил научные исследования по 6 проектам: «Создание акустического корпуса казахского языка и уточнение его фонетического строя, представление казахских фонем в международном фонетическом алфавите», «Методология, алгоритмы и программы генерации электронных учебных изданий», «Научные, методологические, технологические и методические основы перевода казахской письменности на латиницу», «Автоматизация распознавания и порождения письменной и устной речи казахского языка», «Разработать компьютерную модель течения расплава и распределения температурного поля по линии непрерывного литья и прокатки при производстве металлической катанки», «Разработка методики предотвращения накипи в теплообменных системах электростанции с помощью магнитных полей» (рисунки 1).

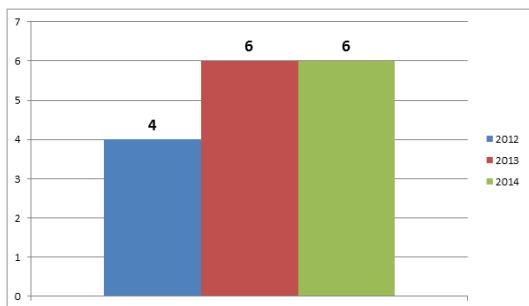


Рисунок 1 – Количество выполняемых проектов по годам

Общее финансирование на 2014 год составило 42 млн тенге. Все задачи, поставленные в календарных планах проектов успешно выполнены, и отчеты по ним сданы в МОН РК.

В рамках выполняемых проектов в 2014 году опубликованы 4 статьи в международном научном издании Scopus, 1 статья в зарубежном журнале, 10 статей в журналах РК, 19 статей в международных конференциях, 12 статей в конференциях РК, опубликованы 12 учебников, учебных пособий (рисунок 2).

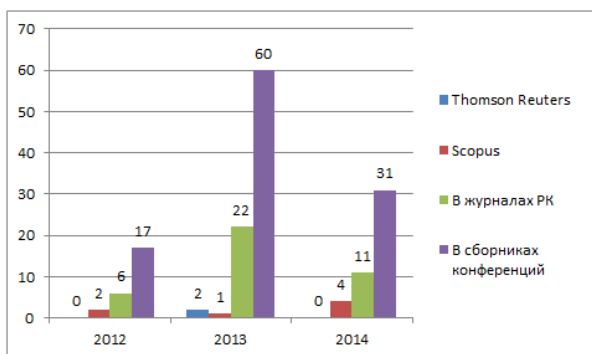


Рисунок 2 – Количество публикаций по годам

Количество зарегистрированных заявок на охранные документы – 6 (рисунок 3).

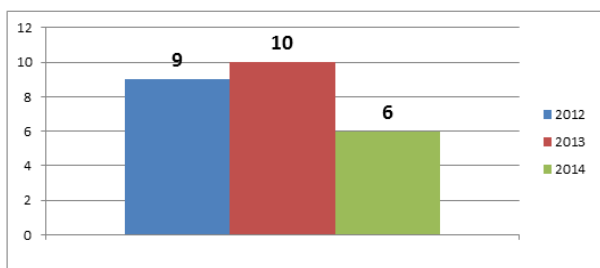


Рисунок 3 – Количество зарегистрированных охранных документов по годам

Ежегодно НИИ «Искусственный интеллект» проводит международные научные конференции (рисунок 4), так в 2012 году проведена III международная научно-практическая конференция «Информатизация общества», в 2013 году проведена I международная конференция «Компьютерная обработка тюркских языков», в 2014 году проведена IV международная научно-практическая конференция «Информатизация общества».

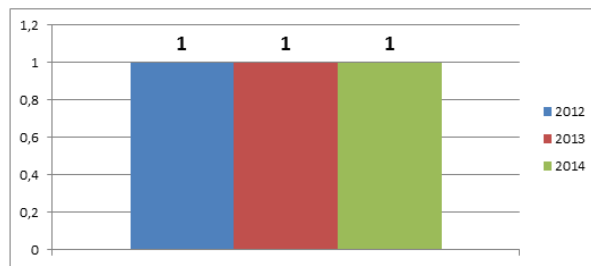


Рисунок 4 – Количество организованных мероприятий по годам

В научную деятельность института привлечены молодые ученые: докторанты, магистранты и студенты факультета информационных технологий. Докторанты участвуют в международных и республиканских конференциях. В 2014 году по результатам научной деятельности в НИИ «Искусственный интеллект» защищаются диссертации магистрантов и докторантов

НИИ «Искусственный интеллект» активно сотрудничает с соответствующими иностранными научными центрами из России (Институт рикладной семиотики АН РТ), Белоруссии (Кафедра интеллектуальных информационных технологий БГУИР, Украины (Институт информатики и искусственного интеллекта ДНТУ), Турции (Стамбульский технический университет), Японии (Университет Осака), и др. Иностранные профессора участвуют в качестве консультанта в подготовке докторантов. Так в 2014 году два докторанта защитили диссертацию PhD: Барлыбаев А. на тему «Модели и методы интеллектуального обучения» (иностранный консультант Голенков В.В., д.т.н., проф., БГУИР) и Есенбаев Ж. на тему «Распознавание казахской речи по определенной словарной базе в условиях шумов» (иностранный консультант Шелепов В.Ю., д.ф.м.-н., проф., ДНТУ).

НИИ «Искусственный интеллект» наметил следующие перспективы развития научной деятельности:

- Привлечение больше молодых ученых.
- Активное участие в конкурсах и грантах.
- Разработка и реализация совместных научно-исследовательских проектов с ведущими зарубежными научными центрами по искусственному интеллекту на базе обмена научными сотрудниками.
- Организации и проведения НИР по хозяйственным договорам.
- Участие в конкурсе по разработке стандартов и других нормативных документов технического регулирования в области информационных технологий и информационной безопасности.
- Проведение экспертизы разрабатываемых программных средств учебного назначения и оценки необходимых финансовых и материальных затрат на реализацию соответствующих проектов.

- Обеспечение информационного, рекламного и маркетингового сопровождения научных исследований.

Заключение

В итоге ожидаются следующие результаты:

- модели, методы, алгоритмы, стандарты и аппаратно-программные средства решения теоретических и практических задач искусственного интеллекта, которые позволят повысить эффективность научного, методического, нормативного и технологического обеспечения образования, науки, техники и экономики страны;
- научные и инновационные работы студентов, магистрантов и докторантов в области искусственного интеллекта, которые будут подтверждать создания необходимых условий молодым ученым университета для участия в научно-исследовательской и инновационной деятельности;
- дополнительные рабочие места за счет собственных средств, которые позволят аккумулировать в университете талантливых молодых специалистов в области искусственного интеллекта;

Библиографический список

- [Люгер, 2004] Люгер Дж. Ф. «Искусственный интеллект: стратегии и методы решения», 4-е издание: Пер. с англ., Москва, «Вильямс», 2004. - 864 стр.
- [Рассел, Норвиг, 2005] Рассел С., Норвиг П. «Искусственный интеллект: современный подход(AIMA)», 2-е издание : Пер. с англ., Москва, «Вильямс», 2005.-1424 с.
- [Поспелов, 1990] Поспелов Д.А. Искусственный интеллект. Справочник. Книга 2. Модели и методы. Москва, «Радио и связь», 1990, -306 с.
- [Поспелов, 1996] Поспелов Д.А. Десять "горячих точек" в исследованиях по искусственному интеллекту. Москва, «Интеллектуальные системы», 1996. - Т.1, вып.1-4. - С.47-56.
- [Айдарханов, 1994] Айдарханов М.Б. Метрический и структурный подходы к построению групповых классификаций . Алматы, «Гылым», 1994, - 55 с.
- [Дюсембаев, 2001] Дюсембаев А.Е. Математические модели сегментации программ. Москва, "Физматлит", Серия "Библиотечка программиста", 2001. - 208с.
- [Шарипбаев, 1996] Шарипбаев А.А. Доказательства правильности программных и аппаратных средств компьютеров. Алматы, «Гылым», 1996, - 240 с.

THE WAYS OF ARTIFICIAL INTELLIGENCE EVOLUTION IN KAZAKHSTAN

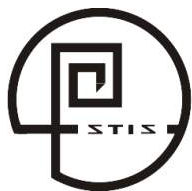
Sharipbay A.A.

*Eurasian national university of L.N. Gumilov, Astana,
Republic of Kazakhstan*

sharalt@mail.ru

The paper presents the basic directions research in artificial intelligence in the world. Information about research carried out in Kazakhstan gives in work. The problems and perspectives of their development are discussed.

Keywords: artificial intelligence; path of development; speech technologies



OSTIS-2015

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822:514

СОЗДАНИЕ ОНТОЛОГИЧЕСКОЙ МОДЕЛИ ЛОКАЛЬНОЙ КОМПЬЮТЕРНОЙ СЕТИ

Шарипбай А.А., Аскарова С.А., Муканова А.С.

Евразийский национальный университет им. Л.Н. Гумилева, г. Астана, Республика Казахстан

sharalt@mail.ru

sandugash.kz@gmail.com

asel_ms@bk.ru

Применение онтологии для передачи знаний в различных областях информационных технологий сделало их более «умными» в последние несколько десятилетий. Многие онтологии были построены для различных областей, таких как биология, медицина, физика, химия и математика. В рамках данной работы будет создана онтологическая модель локальной компьютерной сети с точки зрения ее физических составляющих, то есть аппаратных и программных компонентов. Онтология разработана в формате OWL, что обеспечивает легкую интеграцию с другими приложениями, основанными на семантике. Онтология локальной компьютерной сети может быть использована в семантических веб-приложениях для поиска понятий в области компьютерных сетей.

Ключевые слова: онтологическая модель; локальная вычислительная сеть.

Введение

Согласно определению консорциума W3C, Semantic Web представляет собой расширение существующей сети Internet, в котором информация представляется в четком и определенном смысловом значении, дающем возможность людям и компьютерам работать с более высокой степенью взаимопонимания и согласованности [Berners-Lee et al, 2001]. Обилие знаний, имеющихся в сети организована с помощью семантической сети. Онтология называется ядром Semantic Web, так как она необходим для разработки приложений семантической паутины. Разработка онтологии в различных областях доказала свою эффективность в различных направлениях.

Онтология является всеобъемлющей и детальной формализацией некоторой области знаний с помощью таксономии. Таксономия, в свою очередь, является базовой составляющей онтологии, которая определяет классы объектов и взаимодействие между ними. Таксономия иллюстрируется с помощью иерархической структуры данных, которая содержит классы объектов относящихся к области знаний, их связи, правила и ограничения, принятые в этой области [Gruber,1993]. Онтология является наиболее мощным и широко распространенным инструментом моделирования

отношений между объектами различных предметных областей.

Появление локальных и глобальных сетей предоставило пользователям компьютеров новые возможности быстрого обмена информацией, более продуктивной работы организаций в различных сферах деятельности. По мере развития средств, форм и методов процессов обмена и обработки информации, усложняется вопрос организации и мониторинга работоспособности, надежности и безопасности в сети. Способы коммуникаций компьютеров эволюционировали за последние несколько десятилетий. Эта эволюция приводит к частым введениям новых понятий и технологий для повышения скорости, эффективности, безопасности и различных аспектов в области компьютерных сетей.

1. Обзор исследований в области создания онтологии компьютерных сетей

Онтологии, разработанные для обеспечения знаний в более широкой предметной области как компьютерные сети, ограничены. Существует онтология компьютерных сетей, разработанная в образовательных целях, которая исследует понятия, такие подклассы как коммуникации, приложения, стандарты и безопасность сети, для использования в

качестве учебного пособия [Ling et al., 2008]. Основным недостатком существующей системы является то, что отношения между понятиями не проанализированы надлежащим образом. Связи типа «является» и «является частью» используется для всех отношений, что делает онтологию слабой.

Существует также работа, посвященная использованию гиперграфов для представления онтологий сетевого оборудования [Починский, 2011]. Данная работа предлагает онтологию, которая могла бы использоваться в телекоммуникационной компании в качестве базы знаний об оборудовании и клиентах, и включает такие классы как устройство, плата, порт, клиент. Однако, для создания полной онтологии локальной сети данных классов недостаточно.

Среди зарубежных публикаций можно отметить работу, посвященную онтологии домашней компьютерной сети [Docherty, 2010]. Данная работа рассматривает компоненты такой сети, требования и условия коммуникации между различными устройствами в сети.

Также существует множество работ, посвященных оценке сложности компьютерных сетей. Они рассматривают компьютерные сети с различных точек зрения и с различными целями. Исследование, посвященное оценке сложности сетей [Becheru and Vadica, 2014], авторы представляют новую онтологию, которая позволяет выполнять анализы, которые основаны на знаниях о сложных сетях с различными сетевыми атрибутами и метриками.

Создание онтологии локальной компьютерной сети обеспечит возможность рассмотреть компоненты сети, что в свою очередь, обеспечит знания, относящиеся к данной предметной области.

2. Процесс разработки онтологии

Общий процесс разработки онтологии включает следующие этапы:

- составление глоссария терминов (понятий)
- точные определения терминов на естественном языке
- построение деревьев классификации понятий (иерархии классов)
- определение атрибутов классов и их значений
- добавление экземпляров классов
- создание систем логических выводов.

Для создания онтологии локально компьютерной сети, в данной работе используется редактор онтологий Protégé 5.0.

В данной работе производится попытка формализации знаний для описания объекта «Локальная компьютерная сеть». Формализация знаний с помощью Protégé, означает представление знаний в виде онтологий с помощью языка OWL.

Локальные сети, как правило, являются частными сетями, которые обычно размещаются в одном или близко расположенных зданиях какой-либо организации. Предназначение таких сетей в объединении компьютеров и рабочих станций организации для предоставления совместного доступа к ресурсам и обмена информацией [Таненбаум и Уэзероло, 2012].

Рассматривать локальную сеть можно с различных точек зрения, однако в рамках данной работы, она будет рассматриваться с точки зрения ее функциональных компонентов, из которых она состоит. Таким образом, класс «Network» должен содержать следующие компоненты:

- NetworkHardware - все устройства и среды передачи данных, т.е. это физические и аппаратные компоненты сети
- NetworkSoftware – программные компоненты сети.

Далее, класс NetworkHardware классифицируется на следующие подклассы:

- EndDevices – конечные устройства (иногда называемы хостами или узлами), которые обмениваются информацией между собой, они инициализируют процесс передачи данных, начинают передавать или запрашивать данные у других конечных устройств (рисунок 1); в данный подкласс входят серверы, смартфоны, персональные компьютеры, рабочие станции, ноутбуки, планшеты и т.д.

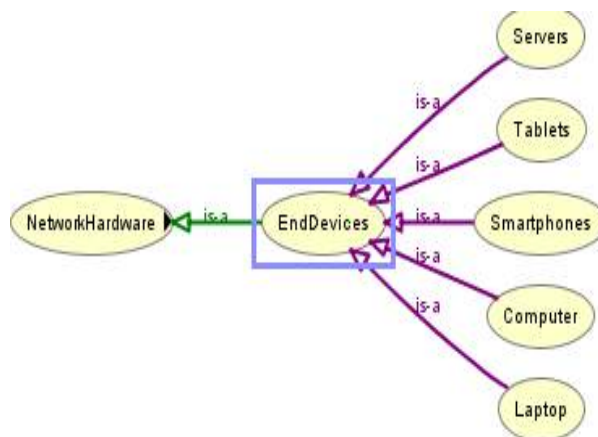


Рисунок 1 - Классификация класса «EndDevices»

- NetworkMedia – среды передачи данных (рисунок 2); в качестве среды передачи данных в локальных сетях используются металлы (в основном медь), сверхпрозрачное стекло (кварц) или пластик и эфир; физическая среда передачи данных может представлять собой кабель «витая пара», коаксиальный кабель, волоконно-оптический кабель и окружающее пространство; в рассматриваемой нами модели в данный подкласс входят такие средства передачи данных как медная витая пара, телефонный кабель, стекловолокно, Bluetooth, Wi-Fi.

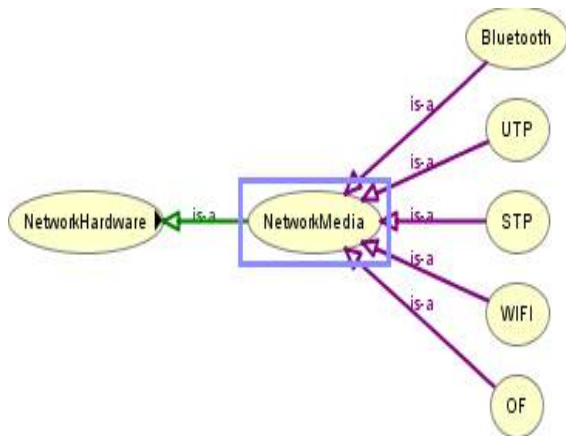


Рисунок 2 - Классификация подкласса «NetworkMedia»

• **IntermediaryDevices** - устройства, объединяющие конечные устройства в локальные (или глобальные) сети передачи данных (рисунок 3); отличие промежуточных устройств от конечных в том, что промежуточные устройства не инициализируют процесс передачи данных, они не начинают передавать или запрашивать данные у других устройств; в данный подкласс входят хабы (hubs), свитчи (switches, коммутаторы), роутеры (routers, маршрутизаторы), модемы (modems), беспроводные точки доступа (Wireless Access Point) и файрволлы (firewalls, брандмауеры).

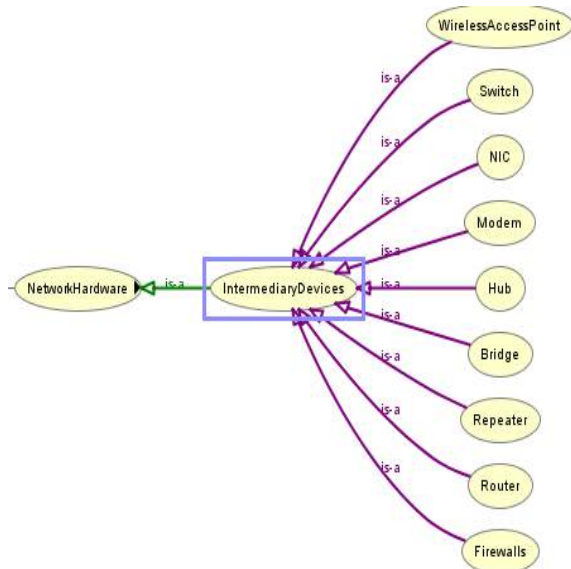


Рисунок 3 - Классификация подкласса «IntermediaryDevices»

Таким же образом классифицируются класс «NetworkSoftware», отвечающая за программные составляющие сети (рисунок 4).

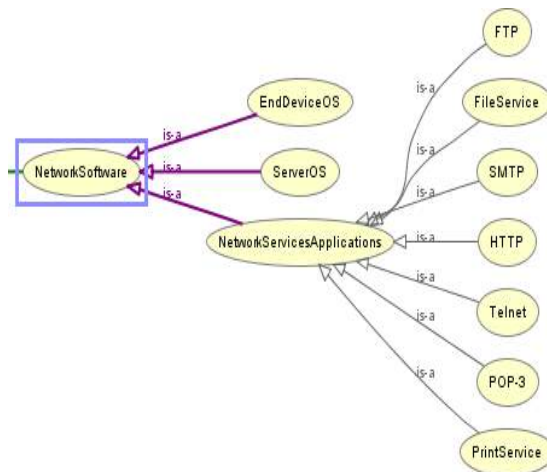


Рисунок 4 - Классификация класса «NetworkSoftware»

Далее, рассмотрев наиболее широко используемые аппаратные и программные средства сетей, определим свойства классов. Например, класс «Network» имеет компоненты «NetworkSoftware» и «NetworkHardware». Для установления такой связи между данными классами, создаем свойство объекта «hasComponent» (рисунок 5).

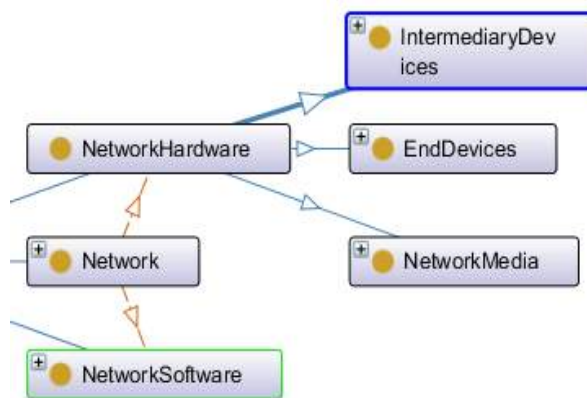


Рисунок 5 – Установление связей между классами

Общая онтология, полученная в результате проделанной работы, представлена на рисунке 6.

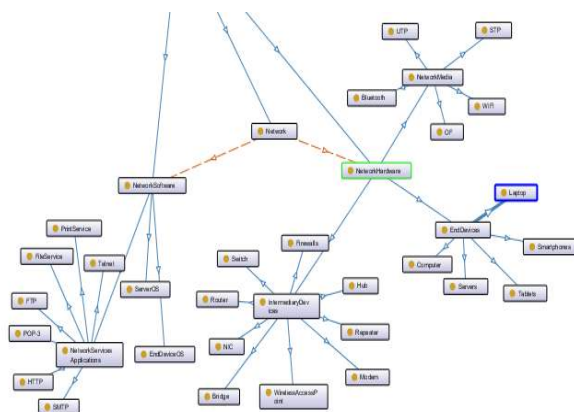


Рисунок 6 – Онтология локальной компьютерной сети

Заклучение

Данная работа рассматривает создание онтологии для такой выбранной предметной области как локальная компьютерная сеть. В ней приводится краткая информация о том, что такое Semantic Web, онтология и этапы ее создания. Также рассматривается онтология компьютерной сети, созданная для образовательных целей и ее недостатки. В статье предложена онтологическая модель локальной компьютерной сети, созданная с точки зрения ее функциональных компонентов. Предложена модель классификации понятий, относящихся к данной предметной области и отношения между этими понятиями.

Библиографический список

[Berners-Lee et al, 2001] Berners-Lee, J. Hendler, and O. Lassila, "The Semantic Web", Scientific American, vol. 284, no. 5, pp. 34-43, May 2001

[Gruber, 1993] Gruber T.R., A Translation Approach to Portable Ontology Specifications / Gruber T.R./Knowledge Acquisition, 1993, P.199-220

[Ling et al., 2008] Ling Jiang, Chengling Zhao and Haimei Wei, "The Development of Ontology-Based Course for Computer Networks", International Conference on Computer Science and Software Engineering, 2008

[Починский, 2011] Починский И.А. Использование гиперграфов для представления онтологии сетевого оборудования. // Проблемы информатики в образовании, управлении, экономике и технике: Сб. статей XI Междунар. научно-техн. конф. – Пенза: ПДЗ, 2011. – С. 74-78.

[Docherty, 2010] Liam S. Docherty, An Ontology Based Approach Towards A Universal Description Framework for Home Networks, Technical Report CSM-182, Department of Computing Science and Mathematics, University of Stirling, 2010

[Becheru and Badica, 2014] Alex Becheru and Costin Badica, Complex Networks' Analysis Using an Ontology-Based Approach: Initial Steps, Lecture Notes in Artificial Intelligence 8793, pp. 326–337, Springer International Publishing Switzerland, 2014

[Таненбаум и Уэзероло, 2012] Э. Таненбаум и Д. Уэзеролл, Компьютерные сети, Пятое издание, — СПб.: Питер, 2012. — 960 с.

CREATING AN ONTOLOGICAL MODEL FOR THE LOCAL COMPUTER NETWORK

Sharipbay A.A., Askarova S.A., Mukanova A.S.

*L.N. Gumilyov Eurasian National University,
Astana, Republic of Kazakhstan*

sharalt@mail.ru

sandugash.kz@gmail.com

asel_ms@bk.ru

The use of ontologies for knowledge transfer in various areas of information technology has made them more "intelligent" in the last few decades. There are many ontologies built for different areas, such as physics, chemistry, biology, medicine and mathematics. Within this work, ontological model of a local computer network is created in terms of its physical components, such as hardware and software components. Ontology is developed in the format of OWL, which provides easy integration with other applications based on semantics. Ontology for local network can be used in semantic web applications to search for concepts in the field of computer networks.

Introduction

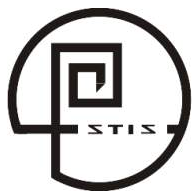
Ontology designed to provide knowledge to the wider subject area as computer networks, are limited. There is an ontology of computer networks designed for educational purposes [Ling et al., 2008]. Its main drawback is that the relationship between the concepts are not analyzed properly. Connection type "is" and "is part of" is used for all the relationships, and it makes ontology weak. Creating ontology local computer network will provide an opportunity to consider the components of the network, which in turn, will provide knowledge relating to this subject area.

Main Part

This part gives a short review on the general stages of the process of creating an ontology. In addition, it provides a definition the selected domain of local computer network, which is followed by consideration of concepts related to this area. Classification of these concepts and relationships between them are illustrated using the ontology editor Protégé 5.0.

Conclusion

This paper discusses the creation of ontology for a selected domain as the local computer network. It provides a short description of what is Semantic Web, ontology, and the stages of its creation. In addition, it considers ontology network, created for educational purposes and its weaknesses. The paper proposes an ontological model for the local computer network, developed in terms of its functional components. It proposes a model of the classification of concepts related to this subject area and the relationship between these concepts.



OSTIS-2015

(Open Semantic Technologies for Intelligent Systems)

УДК 004.432.4

ОНТОЛОГИЧЕСКАЯ МОДЕЛЬ ПРОЦЕССА ОБЕСПЕЧЕНИЯ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

Ниязова Р.С., Буданова Н.

*Евразийский национальный университет им. Л.Н.Гумилева,
г. Астана, Республика Казахстан*

Rozamgul@list.ru
Emma9876@mail.ru

В работе описаны основные понятия информационной безопасности, онтологические задачи обеспечения информационной безопасности, классификации угроз информационной безопасности, модели реализации угроз информационной безопасности.

Ключевые слова: информационная безопасность; онтологические задачи; концептуальная модель; модель информационной безопасности.

Введение

Информация – это ценность. Утечка конфиденциальной информации приносит моральный или материальный ущерб. Условия, способствующие неправомерному овладению конфиденциальной информацией, сводятся к ее разглашению и несанкционированному доступу к ее источникам.

В наше время безопасность информационных ресурсов может быть обеспечена только за счет комплекса системной защиты информации. Комплексная система защиты информации должна быть: непрерывной, плановой, целенаправленной, конкретной, активной, надежной и др.

Мир находится на пороге глобальных изменений: новое информационное общество приходит на смену обществу индустриальному, в связи с чем новые информационные технологии все более проникают в области деятельности человека, особенно в промышленность и общественную жизнь, ускоряя процессы глобализации и интеграции мировой экономики и мирового сообщества [8].

Таким образом, информационные системы являются одним из системообразующих факторов жизни современного общества, и влияние информационной безопасности и на все стороны жизни общества с течением времени будет только возрастать [9].

1. Основные составляющие информационной безопасности

Информационная безопасность – это защищенность информации и поддержка инфраструктуры от случайных или преднамеренных воздействий как естественного, так и искусственного характера, которые могут нанести неприемлемый ущерб субъектам информационных отношений, в том числе владельцам и пользователям информации и поддерживающей инфраструктуры. Под информацией в области защиты понимаются сведения, раскрываемые через демаскирующие признаки объектов защиты или путем несанкционированного доступа к техническим средствам обработки информации. Информационная технология – система технических средств и способов обработки информации. Информационный ресурс – совокупность данных и программ, задействованных при обработке информации техническими средствами.

Субъект информационных отношений – физическое или юридическое лицо, обладающее определенным правом по отношению к информационному ресурсу. В зависимости от уровня полномочий субъект информационных отношений может быть источником, собственником, владельцем или пользователем информации. Источник информации – материальный объект или субъект, способный накапливать, хранить, преобразовывать и выдавать информацию в виде сообщений или сигналов различной физической природы. Защита информации – комплекс

мероприятий, направленных на обеспечение информационной безопасности. Верный подход к проблемам информационной безопасности начинается с поиска субъектов информационных отношений и исследования интересов этих субъектов, связанных с использованием информационных систем (ИС). Угрозы информационной безопасности – это обратная сторона использования информационных технологий.

Доступность – это возможность за приемлемое время получить требуемую информационную услугу. Под целостностью подразумевается актуальность и непротиворечивость информации, ее защищенность от разрушения и несанкционированного изменения.

Конфиденциальность – защита от несанкционированного доступа к информации.

При построении системы информационной безопасности предприятия предлагается модель (рисунок 1), которая описывает совокупность объективных внешних и внутренних факторов и демонстрирует их влияние на состояние информационной безопасности на объекте. Данная модель включает следующие объективные факторы: угрозы информационной безопасности, характеризующиеся вероятностью возникновения и вероятностью реализации; уязвимости информационной системы или системы контрмер, влияющие на вероятность реализации угрозы; риск – фактор, отражающий возможный ущерб организации в результате реализации угрозы информационной безопасности – утечки информации и ее неправомерного использования.



Рисунок 1 – Концептуальная модель информационной безопасности

Благодаря многоступенчатой структуре концептуальной модели информационной безопасности предприятия, введения системы контроля и закрепления ответственности, существенно снижается риск утечек конфиденциальной информации по вине человеческого фактора.

2. Онтология задачи обеспечения информационной безопасности

Опираясь на введенные выше понятия, можно построить следующую онтологическую схему

задачи обеспечения информационной безопасности (рисунок 2)

Субъекты информационных отношений (источник, собственник, владелец или пользователь информации) определяют множество информационных ресурсов, которые должны быть защищены от различного рода атак. К активам

ИС обычно относят:

- материальные ресурсы;
- информационные ресурсы (аналитическая, служебная, управляющая информация на всех этапах своего жизненного цикла: создание, обработка, хранение, передача, уничтожение);
- информационные технологические процессы жизненного цикла автоматизированных систем;
- предоставляемые информационные услуги и т.п.[9].

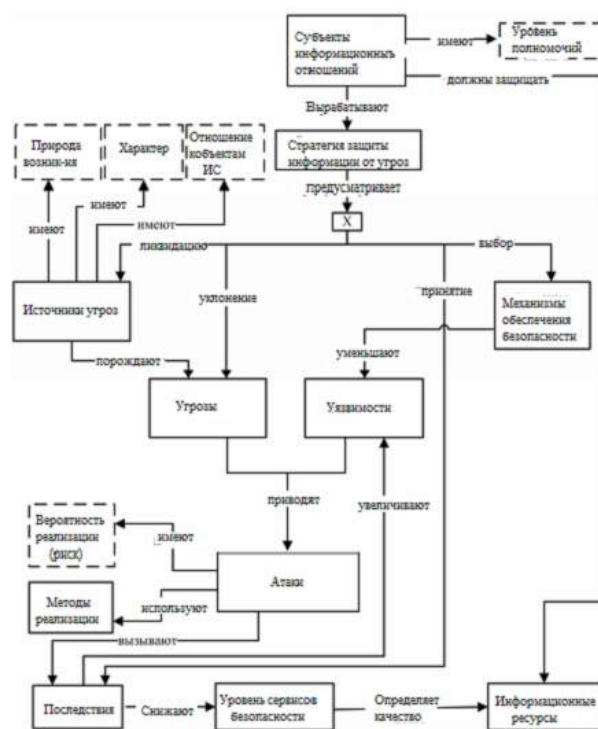


Рисунок 2 – Онтология задачи обеспечения безопасности информационных систем

Исходя из данного принципа, моделирование и классификацию источников угроз и их проявлений, целесообразно проводить на основе анализа взаимодействия логической цепочки:

Источник угрозы – Угроза – Уязвимость – Реализация угрозы (атака) – Последствия (ущерб).

При этом под термином угроза понимается возможная опасность (потенциальная или реально существующая) совершения какого-либо деяния (действия или бездействия), направленного против объекта защиты (информационных ресурсов), наносящего ущерб собственнику, владельцу или пользователю, проявляющегося в опасности искажения и/или потери информации.

3. Классификация угроз информационной безопасности

Обеспечение защиты информации невозможно без проведения системного анализа соответствующих угроз безопасности. Основу такого анализа должна составлять классификация угроз по определенным базовым признакам, дающая исследователю целостное представление о различных вариантах деструктивных воздействий и их последствиях.

Данные элементы при разработке классификации могут быть выбраны в качестве базовых классификационных признаков для последующей их декомпозиции.

Согласно ГОСТ Р 51275-2006 факторы, влияющие на информационную безопасность, можно подразделить по признаку отношения к природе их возникновения на объективные и субъективные, по отношению к объектам информационной системы - на внутренние и внешние.

Кроме этого, и внешние и внутренние источники могут носить как преднамеренный, так и непреднамеренный характер.

Непреднамеренные угрозы возникают независимо от воли и желания людей. Данный тип угроз связан чаще всего с прямым природным или техногенным физическим воздействием на элементы информационной системы и ведет к нарушению работы этой системы и/или физическому повреждению (уничтожению) носителей информации, средств обработки и передачи данных, телекоммуникационных каналов.

Причиной возникновения угроз непреднамеренного характера могут быть как сбои вследствие конкретных ошибок персонала и прямых действий иных лиц, так и случайные нарушения в работе системы.

Преднамеренные угрозы, в отличие от непреднамеренных, могут быть созданы только людьми, действующими целенаправленно с целью дезорганизовать работу информационной системы. Преднамеренные угрозы, в свою очередь, подразделяются на пассивные и активные.

Пассивные угрозы связаны с несанкционированным доступом к информации без каких-либо ее изменений. Активные угрозы связаны с попытками изменения информации или попытками лишения доступа к информационным ресурсам легитимных пользователей.

Общая схема классификации угроз информационной безопасности показана на рис.3

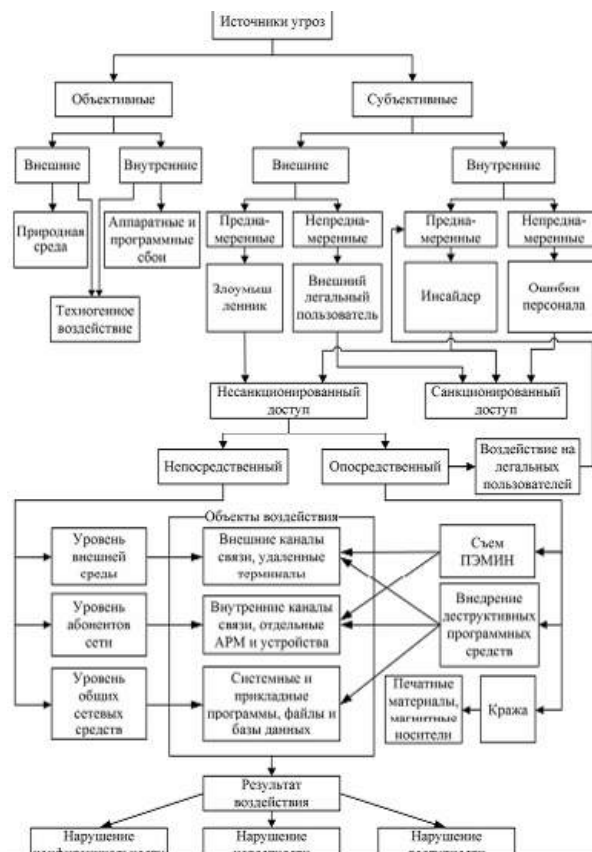


Рисунок 3 – Классификация угроз информационным ресурсам

Заключение

В современном мире уже не осталось сфер деятельности без применения компьютера и информационных систем. Человечество совершило большой рывок в области новых технологий. Поэтому люди беспокоятся о безопасности информации и наличии рисков, связанных с автоматизацией и предоставлением гораздо большего доступа к конфиденциальным, персональным или другим критическим данным. С каждым днем увеличивается число компьютерных преступлений, что может привести в конечном счете к подрыву экономики. И поэтому должно быть ясно, что информация - это ресурс, который надо защищать.

Способы обеспечения информационной безопасности должны быть ориентированы на упреждающий характер действий, направляемых на заблаговременные меры предупреждения возможных угроз коммерческим секретам. Обеспечение информационной безопасности достигается организационными, организационно-техническими и техническими мероприятиями, каждое из которых обеспечивается специфическими силами, средствами и мерами, обладающими соответствующими характеристиками.

Библиографический список

[Шарипбаев А. А., Баймуратова Г. Г., 2005] Шарипбаев А.А., Баймуратова Г.Г. О требованиях к стандартизации программных средств, 2005г.

[Попов В. Б., 2005] Попов В.Б., Основы информационных и телекоммуникационных технологий. Основы информационной безопасности, 2005г.

[Грушо А. А., Тимонина Е. Е., 1996] Грушо А.А., Тимонина Е. Е., 1996 Теоретические основы защиты информации, 1996г.

[Малюк А. А., 2004] Информационная безопасность - концептуальные и методологические основы защиты информации, 2004г.

[Бармен С., 2002] Бармен С. Разработка правил информационной безопасности, 2002г.

[Степанов Е. А., Корнеев И. К., 2001] Степанов Е.А., Корнеев И.К. Информационная безопасность и защита информации, 2001г.

[Барсуков В. С., 2001] Барсуков В. С. Безопасность - технологии, средства, услуги., 2001г.

[Кастельс М., 2000] Информационная эпоха: экономика, общество и культура / Пер.с англ. под науч. ред. О. И. Шкаратана. М., 2000.

[Ажмухамедов И. М., 2011] Принципы обеспечения комплексной безопасности информационных систем // Вестник АГТУ. Серия: "Управление, вычислительная техника и информатика" №1/2011, С.7-11.

[Ажмухамедов И. М., 2012] Решение задач обеспечения информационной безопасности на основе системного анализа и нечеткого когнитивного моделирования, 2011

ONTOLOGIC MODEL OF PROCESS OF ENSURING INFORMATION SECURITY

Niyazova R.S., Budanova N.

*L.N. Gumilyov Eurasian University,
Astana, Republic of Kazakhstan*

Rozamgul@list.ru

Emma9876@mail.ru

Introduction

Information – this value. Leaks of confidential information bring moral or material damage. Conditions conducive to the mastery of inappropriate confidential information, reduced to its disclosure or unauthorized access to its source.

Main Part

In our time, the security of information resources can only be achieved through a complex system of information protection. Integrated security system should be: continuous, planned, focused, specific, active, reliable, and others. Methods of information security should be directed to the precise nature of actions directed to early action to prevent possible threats to commercial secrets.

Conclusion

Providing information security is achieved organizational, logistical and technical measures, each of which provides specific powers, means and measures with relevant characteristics.



OSTIS-2015

(Open Semantic Technologies for Intelligent Systems)

УДК УДК 004.9

КОНЦЕПЦИЯ ИНСТРУМЕНТАЛЬНОЙ ПЛАТФОРМЫ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ В СРЕДЕ ОБЛАЧНЫХ ВЫЧИСЛЕНИЙ С ИСПОЛЬЗОВАНИЕМ ИНТЕЛЛЕКТУАЛЬНЫХ ТЕХНОЛОГИЙ

Вишняков В.А., Гондаз Саз М.М., Моздуоани Шираз М.Г.

*Белорусский государственный университет информатики и радиоэлектроники,
г. Минск, Республика Беларусь*

vish2002@list.ru

Проанализированы основные проблемы информационной безопасности при использовании облачных вычислений (ОВ) и направления их решений. Показано использование интеллектуальных технологий в информационной безопасности. Представлены направления совершенствования интеллектуального управления с использованием ОВ и получение корпоративного интеллекта. В качестве концепции предложено создание инструментальной платформы, объединяющей интеллектуальные системы защиты для облачной среды на основе семантических технологий.

Ключевые слова: интеллектуальные технологии, информационная безопасность, облачные вычисления, инструментальная платформа

Введение

Развитие технологий и сред облачных вычислений (СОВ) вносит новые источники угроз, которые необходимо учитывать при обеспечении безопасности компьютерных систем и сервисов. При этом динамический характер процессов информационного взаимодействия затрудняет возможности оперативной оценки рисков нарушения конфиденциальности, целостности и доступности программных и инфраструктурных ресурсов, предоставляемых в режиме удаленного доступа [Моляков А.С., 2014].

Традиционные средства обеспечения информационной безопасности (ИБ) такие как средства разграничения доступа, межсетевые экраны, системы обнаружения вторжений, контролируют только те информационные потоки, которые проходят по каналам, предназначенным для их передачи. Поэтому угрозы, реализуемые посредством скрытых каналов передачи информации не могут быть блокированы. В связи с этим необходимо использовать интеллектуальные технологии защиты от угроз, которые формируются посредством скрытых каналов информационного воздействия или внутри периметра безопасности корпоративной компьютерной сети [Электр. ресурс, 2013].

1. Угрозы информационной безопасности в среде ОВ

Исследование области обеспечения безопасности СОВ проводилось как российскими, так и зарубежными учеными, среди которых следует отметить: Danish Jamil провёл типизацию угроз для сред облачных вычислений и предложил ряд решений, позволяющих противодействовать рассмотренным угрозам; Michael Miller выполнил анализ механизмов безопасности сред облачных вычислений и выделил общие неустраняемые недостатки [Туманов Ю.М., 2012].

Анализ состояния ИБ в СОВ выявил применение технологий адаптивных систем защиты, которые не всегда позволяют осуществлять контроль за информационными потоками, поскольку они функционируют на верхних уровнях иерархии. Классические методы поиска вредоносного программного кода не позволяют обнаруживать новые образцы вредоносного ПО (ВПО), реализующего технологии ДКОМ и VICE, так как они встраиваются в ОС на более «низком» уровне, чем модули адаптивных систем защиты. Традиционные методы перехвата системных функций гостевых ОС не позволяют обнаруживать программные «закладки», которые внедряются в ОС на этапе загрузки [Моляков А.С., 2014].

Применение сред ОВ ведет к появлению новых проблем ИБ, таких как: распространение ВПО посредством сред ОВ; доверие поставщику услуг

среды ОВ; выявление ВПО, ориентированного на среды ОВ; выявление ПО, не являющегося вредоносным, но содержащим в себе ошибки разработчика.

Построение перспективных механизмов обеспечения безопасности в среде облачных вычислений связывается не только с защитой информации (ЗИ) от выявленных уязвимостей, а с возможностью предотвращения новых неизвестных методов проведения атак, а также в разработке новых моделей угроз и методов предотвращения или отражения компьютерных атак на информационные ресурсы, которые используют возможности предикативной идентификации скрытых каналов и потенциально опасных процессов информационного взаимодействия. Для этого необходимо разработать: модели скрытых угроз информационной безопасности в среде ОВ; модели операций, происходящих с данными при их обработке в СОВ; метод обнаружения скрытых угроз; алгоритм предикативной идентификации скрытых угроз в гостевой ОС и гипервизоре.

Механизмы аутентификации в среде ОВ классифицируются в зависимости от факторов: «знание» используется при вводе пароля или ответа на секретный вопрос, «электроника» означает применение электронных идентификаторов (USB-ключи, смарт-карты, другие e-токены), «био» применяется в системах распознавания отпечатков пальцев, геометрии руки, оболочки глаза, голоса, почерка и т.д., «социальный» использует разговор с оператором. Механизмы аутентификации можно рассмотреть по приоритету их использования: основные – при штатном входе в систему, резервные (почтовый ящик) – при потере пароля либо взломе учетной записи, последние (last resort) – при вмешательстве администрации информационной системы [Малков А.А., 2013].

2. Направления интеллектуализации в информационной безопасности

Основные задачи, которые должны решать интеллектуальные системы ЗИ СОВ (ИСЗИ): обнаружение неизвестных вторжений; поддержки принятия решения о перераспределении ресурсов систем ЗИ; возможности автоматического изменения свойств и параметров в зависимости от изменения условий среды функционирования; дезинформации нападающей стороны о свойствах и параметрах защиты. ИСЗИ, обеспечивающие обнаружения атак, в качестве интеллектуального инструмента используют нейронные сети (НС), системы нечеткой логики и экспертные системы (ЭС).

В ИСЗИ на основе ЭС в базе знаний содержится описание правил, соответствующим профилям легальных пользователей и сценариям атак. Недостатки ИСЗИ на базе ЭС: система не является

адаптивной и не обнаруживаются неизвестные атаки. Если НС представлена в виде системы обнаружения атак, при обработке трафика происходит анализ информации на наличие злоупотреблений. Случаи с указанием на атаку перенаправляются к администратору безопасности. Подход быстродействующий, поскольку используется один уровень анализа. Основной недостаток НС является «непрозрачность» формирования результатов анализа. В системах обнаружения атак с применением НС и ЭС, если идентифицируются новые атаки, то базу следует обновить. Использование гибридных нейро-экспертных или нейро-нечетких систем позволяет хранить нечеткие предикатные правила, которые автоматически корректируются в процессе обучения НС. Свойство адаптивности нечетких НС позволяет решать задачи идентификации угроз, сопоставления поведения пользователей с имеющимися в системе шаблонами, автоматически формировать новые правила при изменении типа угроз [Калач А.В., 2011].

Предложена концепция построения ИСЗИ предприятия, основанная на сочетании принципов функциональной интеграции, иерархической организации, комплексирования моделей, методов и алгоритмов, стандартизации систем ЗИ, что позволяет построить архитектуру автоматизированной системы защиты информации, основанной на многоагентном подходе [Погорелов Д. Н., 2008].

3. Направления защиты информационных ресурсов в среде ОВ

Одним из направлений является разработка, включающая: математическую модель представления ПО, позволяющая получать формальный вывод о наличии или отсутствии его деструктивных свойств; описание классифицирующего признака ПО, обладающего деструктивным свойством; алгоритм классификации ПО, использующий подход к оценке подобия различных экземпляров ПО; методику верификации ПО на наличие деструктивных свойств для сред ОВ [Туманов 2012];

Для решения задачи обнаружения вирусных атак в сети Интернет предложена архитектура на основе продукционной системы с многоуровневой вертикальной моделью агентов. Архитектура включает базу знаний в виде правил продукций, механизм логического вывода, рецепторы и эффекторы агента, модуль коммуникации с другими агентами. Применительно к задаче обнаружения вирусных атак, рецепторы передают факты о внешних воздействиях в базу знаний. В результате логического вывода вырабатывается решение, которое передается эффектору об изменениях внешней среды. Для распределенного решения задач используются разные типы агентов: агент-субординатор, множество агентов исполнителей, агент-интегратор. Агенты связаны между собой в

виде многоуровневой архитектуры, которая может быть горизонтальной или вертикальной. Для решения задачи обнаружения вирусных атак подходит вертикальная многоуровневая архитектура агентов. С учетом специфики решаемой задачи многоагентная система должна включать несколько агентов, которые выполняют в системе различные функции. В результате анализа информационного процесса обнаружения вирусных атак в сетях ОВ можно рассматривать агентов: разграничивающих права доступа пользователей сети, обнаружения вторжений, обнаружения типа атаки, строящих сценарий поведения для отражения вирусной атаки, агент, являющийся посредником-координатором всей многоагентной системы [Берестов А.А., 2011].

Проанализированы системы обнаружения атак (СОА): Snort, Bro, Prelude, OSSEC, Suricata, тенденции их развития. Определен перечень критериев, которым должна удовлетворять СОА: многоуровневость наблюдения за системой; адаптивность (способность обнаруживать модифицированные реализации известных атак и новые виды атак); проактивность, (обладание встроенными механизмами реакции на атаку); открытость (возможность добавления новых анализируемых ресурсов); совмещение централизованного и распределенного управления; защищенность (иметь средства защиты своих компонентов [Никишева А.В. 2013].

Модель, включающая формализацию и контроль информационного взаимодействия в форме виртуальных соединений с помощью межсетевых экранов учитывает динамический характер выделяемых ресурсов и структуру протоколов сетевого взаимодействия. Входом модели является поток сетевых пакетов, которые поступают в межсетевые экраны системы ЗИ в среде ОВ, а выходом является разделение пакетов на виртуальные соединения. Классификация пакетов дана на принадлежность соединению и определению подмножества правил фильтрации для них [Лукашин А.А., 2013].

Модель противодействия угрозам ИБ, в которой решение о варианте реагирования принимается в зависимости от вероятности атаки, оцениваемой с использованием механизма нечеткого логического вывода [Машкина И.В., 2009].

4. Интеллектуализация управления в средах ОВ

На базе Web 3.0 развивается управление в СОВ. Онтология формирует семантику, создавая новые возможности для интеллектуальных агентов (ИА), выполняющих запросы пользователей. Открытое извлечение информации (Information Extraction – IE) обеспечит работу новых форм поиска, освобождая пользователей от задачи по исследованию документов, выданных поисковой машиной. Широко применяются серверы исполнения деловых регламентов (BRE – Business

Rules Engine). Чтобы справиться со сложностью бизнес-процессов, связывающих несколько предприятий или цепочку создания инноваций в Web 3.0, компании требуется создания новых процессов, превосходящих современные серверы исполнения регламентов [Вишняков В.А. 2014, Фингар П. 2011].

Распределенный искусственный интеллект – DAI (Distributed Artificial Intelligence) основывается на агентных технологиях. Стандартный программный агент имеет три свойства: автономность, способность реагировать и выйти на связь. Простые программные агенты могут общаться с другими «сущностями»: пользователями, программными агентами или объектами. Добавив к этому способность планировать и ставить цели, поддерживать модели представлений, рассуждать о действиях и повышать уровень знаний и качество работы через обучение, получим главные компоненты ИА.

ИА могут быть интегрированы в структуры ОВ, содержащие конкретные функции по решению задач, обработки данных и управления. Они поддерживают естественное соединение информации и технологий, основанных на знаниях и процесс логических рассуждений, образуя интеллектуальные КИС (ИКИС). ИА позволяют включить функцию обучения и самосовершенствования как на уровне инфраструктуры (адаптивная маршрутизация), так и на уровне приложения (адаптивные пользовательские интерфейсы) [Фингар П. 2011].

ИА используются для сбора бизнес-аналитики – BI (Business Intelligence) и процессов обработки сложных событий – CEP (Complex Event Processing). Показатель посещений страниц устарел, важно количество связей в социальных сетях, отправленных сообщений и время, проведенное на конкретном сайте. Получение информации и непрерывный анализ в реальном времени в ОВ – это следующая задача для ИКИС, для этого надо переходить от «поиска в данных» к «поиску в блогах». В ИКИС необходимо выйти за пределы поисковиков, обработать Интернет-данные, чтобы понять, что же происходит в отрасли, оценить ситуацию о товарах и услугах компании, т.е. нужна аналитика Web 3.0. Когда компании выводят управление бизнес-процессами в сложную деловую экосистему, ценность обработки сложных событий становится опорой для корпоративного интеллекта необходимых для того, чтобы создать и улучшить постоянно меняющиеся бизнес-процессы [Фингар П. 2011].

Данные принципы интеллектуального управления будут использованы в ИС ЗИ для сред ОВ при построении инструментальной платформы [Вишняков В.А., 2014].

5. Концепция инструментальной платформы ЗИ в среде ОВ

Предложены тенденции развития ИТЗИ в СОВ [Машкина И.В., 2009, Вишняков В.А., 2014]:

- совершенствование архитектуры систем ЗИ, обеспечивающих эффективное управление в условиях неопределенности состояния информационной среды;
- разработка новых моделей противодействия угрозам нарушения ИБ на основе выбора оптимального варианта реагирования на события безопасности;
- совершенствование инструментальных программных комплексов с интеллектуальной поддержкой принятия решений и исследованием эффективности методов, моделей и алгоритмов;
- развитие технологий многоагентных систем для обнаружения атак, противодействия угрозам нарушения ИБ, оценки уровня защищенности информации;
- разработка основ, моделей и средств защиты облачной инструментальной платформы проектирования интеллектуальных систем на основе семантических технологий.

Для реализации этих тенденций необходимо построение инструментальной платформы ЗИ в среде ОВ. Платформа будет использоваться для создания пользовательских прикладных ИС. Предложены следующие решения по инструментальной платформе на базе многоагентной системы:

- разработка структура и состав многоагентной системы обнаружения атак, включающая в себя агентов рабочих станций, серверов, маршрутизаторов и сетей и позволяющая делать вывод о атаках, состоянии и перспективах ее защиты;
- получение метода принятия агентами совместного решения, позволяющего сформировать общение агентов и на основании результатов анализа сведений, полученных из различных источников, оценить состояние ОВ в целом;
- выработка методики обнаружения атак с использованием многоагентных технологий, позволяющая обучить многоагентную систему и использовать ее для дальнейшего обнаружения неизвестных воздействий ОВ.
- расчет эффективности предложенных методов, используя разработанные решения.

Заключение

Одним из направлений в СЗИ СОВ является разработка моделей, методов, архитектур и аппаратно-программных средств управления ИБ для решения проблемы защиты на базе облачной инструментальной платформы, созданной на основе семантических технологий.

Библиографический список

[Моляков А.С. 2014] Моляков, А.С. Модели и метод противодействия скрытым угрозам информационной

безопасности в среде облачных вычислений / А.С. Моляков / Автореферат канд. дисс. по спец. 05.13.19. СПб, 2014. – С.17.

[Электронный ресурс, 2013]. Intelligence Community Information Technology Enterprise (ICITE) [Электронный ресурс], режим доступа: http://www.insonline.org/i/d/a/Resources/ICITE_Doing.aspx (дата доступа 22.10.2013).

[Туманов Ю.М., 2012] Туманов Ю.М. Защита сред облачных вычислений путём верификации программного обеспечения на наличие деструктивных свойств. /Ю.М. Туманов / Автореферат канд. дисс. по спец. 05.13.19, М.: МИФИ, 2012. – 19 с.

[Малков А.А., 2013] Малков, А.А. Технология аутентификации с помощью доверенных лиц / А.А. Малков / Автореферат канд. дисс. по спец. 05.13.19. Уфа: УГАТУ, 2013. – С.16.

[Калач А.В., 2011] Калач, А.В., Немтина Е.С. Интеллектуальные средства и моделирование систем защиты информации @Технологии техносферной безопасности" (<http://ipb.mos.ru/ttb>) Выпуск № 3 (37) – 2011. – С.3-11.

[Погорелов Д.Н., 2006] Погорелов, Д. Н. Защита информационных ресурсов предприятия на основе многоагентной технологии / Д. Н. Погорелов // Автореферат канд. дисс. по спец. 05.13.19. Уфа, 2007. – С.16.

[Берестов А.А., 2011] Берестов, А.А. Архитектура интеллектуальных агентов на основе производной системы для защиты от вирусных атак в сети Интернет / А.А. Берестов // Материалы . XV Всероссийской научной конференции «Проблемы информационной безопасности в системе высшей школы» М.: МИФИ, 2011. – С.24-25.

[Никишева А.В., 2013] Никишева, А.В. Многоагентная система обнаружения атак на информационную систему предприятия /А.В. Никишева / Автореферат канд. дисс. по спец. 05.13.19. Волгоград, 2013. – С.19.

[Лукашин, А.А., 2012] Лукашин, А.А. Система защиты информационного взаимодействия в среде облачных вычислений /А.А. Лукашин / Автореферат канд. дисс. по спец. 05.13.19, СПб, 2012.

[Машкина И.В., 2009] Машкина, И.В. Модели и метод принятия решений по оперативному управлению защитой информации / И.В. Машкина // Системы управления и информационные технологии. Москва - Воронеж, 2008. №2 (32). С. 98 – 104.

[Фингар П., 2011] Фингар П. Облачные вычисления – бизнес-платформа XXI века. Пер. с англ. Захаров А.В. / П. Фингар / – М.: Акваринная Книга, 2011. – 256 с.

[Вишняков, В.А., 2014] Вишняков, В.А. Информационное управление и безопасность: методы, модели, программно-аппаратные решения. Монография. / В.А. Вишняков. – Минск: МИУ, 2014. – 287с.

CONCEPTION INSTRUMENTAL PLATFORM INFORMATION SECURITY IN CLOUD COMPUTING WITH INTELLIGENCE TECHNOLOGIES

Vishniakou U.A., Gongas Sas M.M., Mosdurani Shiras M.G.

*Belorussian State University of Informatic and Radioelectronic,
Minsk, Republic of Belarus*

vish2002@list.ru

The main problems of information defense in cloud computing and its decisions are presented. The using of intelligence technologies in information defense are shown. The directions of intellectual management development with cloud computing use (corporate intelligence) are presented. The concept of instrumental platform joined intelligence subsystems defended for cloud computing area on the base of semantic technologies is proposed.



OSTIS-2015

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822:514

ОНТОЛОГИЧЕСКИЙ ПОДХОД К СОЗДАНИЮ НАУЧНЫХ ИНТЕРНЕТ-РЕСУРСОВ

Загорулько Ю.А., Загорулько Г.Б.

*Институт систем информатики им. А.П. Еришова Сибирского отделения
Российской академии наук, г. Новосибирск, Россия*

zagor@iis.nsk.su

gal@iis.nsk.su

В докладе представлен подход к построению научных интернет-ресурсов, предназначенных для информационной и аналитической поддержки научной и производственной деятельности в определенной области знаний. Особенностью подхода является использование онтологии для систематизации и интеграции знаний, информационных ресурсов, релевантных моделируемой области знаний, и средств интеллектуальной обработки содержащихся в них данных. Кроме того, на основе онтологии организуется удобная навигация по научным знаниям и информационным ресурсам, содержательный поиск данных и средств интеллектуальной обработки, а также строятся шаблоны для автоматического извлечения информации из Интернет.

Ключевые слова: научный интернет-ресурс, область знаний, информационный ресурс, онтология, специализированная программная оболочка.

Введение

Несмотря на то, что значительная часть информации практически по всем областям знаний представлена в Интернет, проблема эффективного обеспечения научного сообщества информацией по интересующим его тематикам еще далека до своего решения. Нерешенной остается и проблема доступа к методам и средствам обработки знаний и данных, собранных по этим тематикам. Большей частью, даже уже представленные в сети Интернет реализации методов обработки остаются недоступными из-за отсутствия содержательной информации о них, по которой их могли бы найти как программные агенты, так и члены научного сообщества.

Для представления знаний о некоторой области исследований и содержательного описания релевантных ей информационных ресурсов и методов обработки содержащихся в них данных и знаний удобным средством являются онтологии [Guarino, 19981] и построенные на их базисе семантические сети [Осипов, 1990; Загорулько, 2013]. Именно эти формализмы были положены в основу интеллектуальных систем, предназначенных для информационной и аналитической поддержки научной и производственной деятельности в определенных областях знаний. Такие системы

получили название тематические интеллектуальные научные интернет-ресурсы (ИНИР).

Создание тематических ИНИР позволит решить следующие задачи:

(1) сведение информационных ресурсов, относящихся к определенной области знаний, а также средств их интеллектуальной обработки в единое информационное пространство;

(2) поддержка логической целостности системы семантических описаний информационных ресурсов и средств их интеллектуальной обработки, сведенных в единое информационное пространство;

(3) обеспечение возможности открытого содержательного доступа не только к интегрированным информационным ресурсам, но и к средствам их интеллектуальной обработки.

Актуальность первой из этих задач обусловлена тем, что российская наука, образование и производство испытывают в наши дни потребность в концентрации и обобщении накопленной информации по различным отраслям знаний и эффективном ее использовании, но удовлетворение этой потребности затрудняется тем, что в силу своей многоплановости и многоаспектности научные информационные ресурсы рассредоточены на удаленных страницах множества сайтов, а также в распределенных электронных библиотеках и

архивах. Сведение в единое информационное пространство такого рода ресурсов, относящихся к требуемой области знаний, будет способствовать удовлетворению указанной потребности.

Поддержка логической целостности системы семантических описаний ресурсов и средств их интеллектуальной обработки необходима для обеспечения пользователей актуальной, достоверной и достаточно полной информацией по интересующей его теме. Решение этой задачи будет обеспечено тем, что в основе создаваемых ИНИР будут лежать онтологии, дающие целостное и непротиворечивое представление о моделируемой области знаний, а также семантические сети, построенные на концептуальном базисе онтологий.

Сведение информационных ресурсов и средств их интеллектуальной обработки в единое информационное пространство и обеспечение логической целостности системы их семантических описаний создаст предпосылки для решения третьей задачи – содержательного доступа к этим ресурсам и механизмам их интеллектуальной обработки широкому кругу пользователей.

Ввиду высокой потребности в системах такого класса нами разрабатывается технология создания и сопровождения ИНИР, ориентированная не на программистов и инженеров знаний, а непосредственно на специалистов в областях знаний, для которых такие ресурсы разрабатываются. Эта технология является развитием технологии [Загоруйко, 2011], успешно применявшейся при построении порталов научных знаний для многих предметных областей.

1. Концепция и архитектура тематического ИНИР

Тематический ИНИР представляет собой доступную через Интернет информационную систему, обеспечивающую систематизацию и интеграцию научных знаний и информационных ресурсов определенной области знаний, содержательный эффективный доступ к ним и средствам их интеллектуальной обработки. ИНИР позволяет исследователям значительно сократить время, требуемое для обеспечения доступа к необходимой информации и ее анализа, за счет аккумуляции описаний сущностей моделируемой области знаний и релевантных ей информационных ресурсов (в том числе, web-сервисов) непосредственно в контенте ИНИР.

Ядром системы знаний ИНИР является онтология, которая наряду с описанием моделируемой области знаний содержит соотношенное с ним описание структуры и типологии интегрируемых информационных ресурсов и методов интеллектуальной обработки содержащихся в них данных.

Семантическая сеть, структура которой определяется онтологией ИНИР, играет роль

интеллектуального хранилища данных, в котором накапливается информация о релевантных научных информационных ресурсах и web-сервисах, реализующих методы обработки содержащихся в них знаний и данных.

На основе онтологии организуется удобная навигация по научным знаниям и информационным ресурсам, интегрированным в ИНИР, а также содержательный поиск данных и средств их интеллектуальной обработки.

Кроме онтологии и семантической сети в систему знаний ИНИР включен тезаурус, который рассматривается в качестве лексического дополнения онтологии. Он содержит описание терминов моделируемой области знаний, выраженных словами или словосочетаниями, с помощью которых понятия онтологии представляются в текстах и пользовательских запросах. Тезаурус задает смысл понятий посредством соотнесения одних терминов с другими, главным образом, с помощью семантических отношений. Благодаря этому он может применяться как при обработке пользовательских запросов, так и при поиске, обработке и аннотировании информационных ресурсов, интегрируемых в ИНИР.

ИНИР имеет традиционную для информационных систем трехуровневую архитектуру (см. Рис. 1), включающую уровень представления информации, уровень обработки информации и уровень хранения и доступа к информации (базовый уровень).

Первый уровень обеспечивается пользовательским интерфейсом, главными функциями которого являются представление пользовательских запросов и результатов поиска и решений задач, а также обеспечение управляемой онтологией навигации в информационном пространстве ИНИР. При этом пользовательский интерфейс обеспечивает не только содержательный доступ к контенту ИНИР, но и к средствам аналитической обработки информации.



Рисунок 1 – Архитектура тематического ИНИР

На уровне обработки информации обеспечиваются различные виды поиска и обработки информации, а также ее передача между уровнями. Для этих целей данный уровень включает средства поиска информации в контенте ИНИР, а также средства ее аналитической обработки, реализованные, в том числе, и в виде web-сервисов [McIlraith, 2001].

Средства поиска позволяют осуществлять как поиск информации по ключевым словам, так и расширенный семантический поиск с представлением запроса в терминах понятий и отношений онтологии и ограничений на них. Эти же средства поддерживают навигацию по контенту ИНИР, поставляя в пользовательский интерфейс семантическую окрестность просматриваемых понятий и объектов.

В качестве аналитических инструментов используются средства фильтрации и визуализации понятий онтологии и объектов семантической сети. Фильтрация позволяет из большого списка выбрать объекты, значения атрибутов которых удовлетворяют указанным ограничениям. Сервис визуализации [Апанович, 2011] обеспечивает отображение полной системы понятий онтологии в виде графа, предоставляет пользователю возможность оценить объем каждого понятия с помощью круговой диаграммы (т.е. количество объектов, относящихся к данному понятию), позволяет графически отобразить либо полную сеть объектов, либо только граф объектов, связанных выбранным отношением.

Базовый уровень обеспечивает выполнение функций хранения и управления знаниями (онтологией и тезаурусом) и данными (контентом ИНИР) с использованием средств реляционных СУБД (MySQL), технологий Semantic Web (OWL, RDF) [OWL, 2004] и семантических web-сервисов (WSDL, OWL-S) [Chinnici, 2007; Martin et al., 2004].

2. Методологическая и техническая поддержка построения ИНИР

В настоящее время разрабатывается технология построения ИНИР. Как было сказано выше, ее особенностью является ориентация на широкий круг пользователей – экспертов, т.е. специалистов в определенных областях знаний. Такая технология позволяет экспертам собирать и систематизировать в рамках единого информационного пространства обширные знания и данные, релевантные требуемой области знаний, а также средства их интеллектуальной обработки.

Так как основу ИНИР составляют онтологии, центральным элементом технологии их создания является методология построения онтологий, к которой примыкает методология построения тезаурусов. Технология базируется на программных средствах, поддерживающих построение онтологий, тезаурусов и управление контентом ИНИР.

2.1. Методология построения онтологии ИНИР

Онтология конкретного ИНИР строится в соответствии с методологией, главными принципами которой являются:

- структурирование онтологии ИНИР путем разделения ее на относительно независимые онтологии, каждая из которых представляет самостоятельный компонент знаний;
- использование базовых онтологий, которые включают только самые базовые сущности, не зависящие от области знаний ИНИР;
- построение всех онтологий ИНИР на основе базовых онтологий путем их доработки и развития.

Использование такой методологии значительно упрощает создание онтологии ИНИР и ее дальнейшее сопровождение.

Как было сказано выше, онтология ИНИР кроме описания понятий и отношений моделируемой области знаний включает соотнесенное с ним описание структуры и типологии интегрируемых информационных ресурсов и методов интеллектуальной обработки данных и знаний. В связи с этим онтология ИНИР состоит из трех онтологий, отвечающих за представление указанных выше компонентов знаний, а именно: онтологии области знаний ИНИР, онтологии научных информационных ресурсов и онтологии задач и методов.

В качестве базовых онтологий предложены онтология научной деятельности и онтология научного знания, на основе которых строится онтология области знаний ИНИР, а также базовые онтологии научных информационных ресурсов и задач и методов.

Онтология научного знания содержит классы, задающие структуры для описания понятий конкретных областей знаний, такие как Раздел науки, Метод исследования, Объект исследования, Научный результат и др. Эта онтология также включает отношения, связывающие между собой объекты указанных выше классов.

Онтология научной деятельности включает классы понятий, относящиеся к организации научной и исследовательской деятельности, такие как Персона, Организация, Событие, Научная деятельность, Проект, Публикация и др. Она включает также отношения, связывающие понятия данной онтологии как между собой, так и с понятиями онтологии научного знания.

Базовая онтология научных информационных ресурсов включает класс Информационный ресурс в качестве основного класса. Этот класс служит для описания, релевантных области знаний информационных ресурсов. Набор атрибутов и связей этого класса основан на стандарте Dublin core [Hillmann, 2005]. Его атрибутами являются: название ресурса, язык ресурса, тематика ресурса,

тип доступа к ресурсу и т.п. Объекты этого класса могут быть связаны семантическими отношениями с другими информационными объектами, представляющими в контенте ИНИР организации, персоны, публикации, события, разделы науки и т.д.

Базовая онтология задач и методов включает такие классы как Задача, Метод и Web-сервис, а также отношения, связывающие эти понятия между собой и понятиями других базовых онтологий. С помощью понятий и отношений данной онтологии могут быть описаны задачи, для решения которых предназначен ИНИР, методы их решения, а также реализующие их web-сервисы.

Описания web-сервисов онтологии задач и методов базируются на онтологии OWL-S [Martin et al., 2004], предназначенной для описания семантических web-сервисов. Благодаря этому с web-сервисом связываются не только описание его интерфейса в терминах типов входных и выходных данных, но и описание его семантики, т.е. того, что сервис делает, его предметной области, ограничений на область применения и качество сервиса и т.п. Наличие семантического описания у web-сервисов обеспечивает не только реализацию их поиска и корректного использования (исполнения), но и создает предпосылки и для их успешной интеграции в ИНИР. При этом будет обеспечиваться содержательный доступ к ним не только для программных агентов, но и для исследователя, желающего найти необходимые для решения его задач средства интеллектуальной обработки информации.

2.2. Построение тезауруса

Методология построения тезауруса области знаний ИНИР базируется на тех же принципах, что и методология построения онтологии.

Так, тезаурус области знаний ИНИР строится путем дополнения ядра тезауруса, содержащего описания терминов, представляющих понятия базовых онтологий, в том числе в интернет-ресурсах.

Ядро тезауруса, как и весь тезаурус, строится на основе онтологии представления тезауруса [Загоруйко, 2012], которая включает набор базовых понятий и отношений, присутствующих в любом тезаурусе. В частности, она содержит классы, описывающие такие сущности тезауруса, как термины, которые подразделяются на дескрипторы (предпочтительные термины) и аскрипторы (текстовые входы, которые при поиске и индексировании документов могут быть заменены на соответствующие дескрипторы), источники терминов (web-ресурсы, текстовые документы или коллекции текстов, в которых встречаются или определяются термины) и области/подобласти знаний, с которыми могут быть соотнесены термины. В онтологии также представлены отношения, связывающие объекты перечисленных выше классов между собой.

2.3. Управление системой знаний ИНИР

Поддержка процесса управления системой знаний ИНИР обеспечивается редакторами онтологий и данных, а также подсистемой сбора онтологической информации из Интернет.

Указанные редакторы реализованы как web-приложения, поэтому поддерживают удаленную коллективную разработку и поддержку системы знаний ИНИР авторизованными экспертами через Интернет.

Редактор онтологий предназначен для построения онтологий и управления ими в процессе жизненного цикла ИНИР. Этот редактор проектировался таким образом, чтобы им могли пользоваться эксперты, не являющиеся специалистами в области информатики и программирования.

Управление контентом ИНИР осуществляется с помощью редактора данных. Этот редактор работает под управлением онтологии, что позволяет не только значительно облегчить ввод данных (путем генерации удобных форм для ввода объектов и связей между ними), но и обеспечить их логическую целостность.

Построение тезауруса и последующее редактирование его содержания осуществляется с помощью того же редактора данных, но работающего уже под управлением онтологии представления тезауруса, что позволяет обеспечить логическую целостность его терминологической системы.

Для того чтобы ИНИР был полезным ресурсом его система знаний должна содержать достаточно полную информацию о моделируемой области знаний и выполняемой в рамках ее научной и/или производственной деятельности. Создание такого ИНИР – довольно трудоемкая задача, требующая значительных усилий разработчиков. Для ее автоматизации предназначена подсистема сбора информации из Интернет, реализующая методы поиска и извлечения информации, базирующиеся на онтологии и механизмах метапоиска.

Сбор информации для ИНИР включает следующие задачи: поиск релевантных его области знаний интернет-ресурсов и документов, извлечение информации из них и занесение полученной информации в контент ИНИР. Для выполнения этих задач в подсистему сбора информации входят модуль поиска релевантных интернет-ресурсов, модуль извлечения информации из интернет-ресурсов, модуль занесения найденной информации в контент ИНИР, а также база данных ссылок на интернет-ресурсы (БД СИР).

При настройке ИНИР на область знаний эксперт заполняет БД СИР ссылками на релевантные по его мнению интернет-ресурсы. При этом для каждой ссылки указывается класс онтологии, объекты которого описывает соответствующий ей ресурс.

Список ссылок может пополняться не только вручную – экспертами, но и автоматически – модулем поиска интернет-ресурсов, который выполняет сбор ссылок на релевантные интернет-ресурсы по поисковым запросам, сформированным на основе названий классов онтологии и терминов тезауруса, представляющих понятия моделируемой области знаний. Этот модуль запускается с заданной при настройке ИНИР периодичностью. При этом модуль поиска обращается к поисковым системам Google, Яндекс и Bing через их программные интерфейсы, т.е. использует механизм метапоиска с последующей фильтрацией дубликатов и нерелевантных ссылок [Ахмадеева, 2013].

Контент ИНИР заполняется информацией из таких источников, как порталы знаний, сайты организаций, ассоциаций, проектов и конференций, новостные ленты, социальные научные сети, вики-ресурсы, реестры (каталоги) веб-сервисов и др. Из этих источников извлекается информация о проектах, организациях, персонах, конференциях и публикациях, т.е. обо всех объектах классов онтологии научной деятельности, а также о самих источниках, как объектах класса Информационный ресурс онтологии научных информационных ресурсов.

Для каждого из этих классов разрабатывается свой метод извлечения информации, включающий набор шаблонов и связанных с ними обработчиков. В шаблонах для каждого типа извлекаемой информации указываются обработчики, реализующие алгоритмы обхода и/или анализа соответствующих фрагментов интернет-страниц или документов. Указанные шаблоны генерируются на основе онтологии, а затем уже при необходимости дополняются экспертами. Для повышения полноты извлечения информации увеличивается вариативность этих шаблонов за счет использования в них альтернативных терминов из тезауруса (синонимов и гипонимов).

Например, шаблон для извлечения информации о проектах, построенный на основе описания класса Проект, имеет вид, представленный на рисунке 2.

```
<Class Name="Проект" engine="FragmentSearch">
  <Marker Term="О проекте" PType="Menu/Head" FragType="Page/Block"/>
  <Marker Term="Проект" PType="Head" FragType="Block"/>
  <Attr Name="Название" type="string" engine="NameEntity">
    <Marker Term="Проект" PType="link" FragType="LinkText"/>
    <Marker Term="Проект" PType="sentence" FragType="QuoteText"/>
    <Marker Term="Проект" PType="Head" FragType="Head"/>
  </Attr>
  <Attr Name="Аннотация" type="text">
    <Marker Term="Аннотация/Содержание проекта/Описание
    проекта/ О проекте" PType="Head" FragType="Block/Page"/>
  </Attr>
  <Relation Name="Публикация_о_Проекте">
    <Marker Term="Публикации" PType="Menu/Head" FragType="Page/Block"/>
    <Marker Term="Список публикаций" PType="Menu" FragType="Page"/>
    <Marker Term="Литература" PType="Menu/Head" FragType="Page/Block"/>
    <Marker Term="Библиография" PType="Menu/Head" FragType="Page/Block"/>
    <Object Name="Публикация" engine="PublicationsList"/>
  </Relation>
  <Relation Name="Участник проекта">
    <Marker Term="Об участниках" PType="Menu" FragType="Page"/>
    <Marker Term="Список участников" PType="Head" FragType="Block"/>
    <Marker Term="Исполнители" PType="Head" FragType="Block"/>
    <Marker Term="Участники" PType="Head" FragType="Block"/>
    <Object Name="Персона" engine="PersonList"/>
  </Relation>
</Class>
```

Рисунок 2 – Шаблон для извлечения информации о проекте

Шаблон содержит указание на то, для извлечения объектов какого класса он предназначен (блок Class). Он также включает блоки атрибутов (Attr), отношений (Relation) и аргументов отношений (Object). Каждый из этих блоков может описываться одним или группой альтернативных маркеров (Marker), задающих свойства фрагмента текста, содержащего извлекаемую информацию.

Заключение

В докладе представлен подход к построению тематических ИНИР, обеспечивающих систематизацию и интеграцию информационных ресурсов определенной области знаний и средств интеллектуальной обработки содержащейся в них информации, а также содержательный эффективный доступ к ним и их использование при решении различных задач.

Важным преимуществом ИНИР является то, что он позволяет исследователям значительно сократить время, требуемое для доступа к необходимой информации и ее анализа, за счет аккумуляции описаний сущностей моделируемой области знаний и релевантных ей информационных ресурсов непосредственно в своем контенте.

Использование онтологии в качестве основы ИНИР, создает предпосылки для того, чтобы технология построения ИНИР стала действительно массовой. С одной стороны, онтология является удобным средством формирования и фиксации общего разделяемого экспертами-разработчиками знания о данной предметной области, обеспечивая при этом возможность переиспользования знаний, что упрощает и ускоряет разработку новых приложений. С другой стороны, базирование средств описания области знаний ИНИР, как и создания и сопровождения его контента, на онтологии делает их доступными для использования непосредственно экспертами, так как представление знаний и данных в виде объектов и отношений между ними, принятое в онтологии, является наиболее естественным для человека.

Чтобы данной технологией мог воспользоваться широкий круг экспертов, в ее рамках должны быть разработаны программные оболочки разных типов ИНИР, отличающиеся набором базовых онтологий и, возможно, программных компонентов. Такие специализированные оболочки будут представлять собой «пустые» ИНИР, т.е. в них будут представлены все необходимые структурные компоненты будущего ИНИР, но не достроены нижние уровни онтологии области знаний и не заполнен контент.

Примером такой специализированной оболочки является оболочка портала научных знаний [Загорулько, 2008], включающая рассмотренные выше базовые онтологии – онтологию научной деятельности и онтологию научного знания, с помощью которых эксперт может, не прибегая к помощи инженеров знаний и программистов,

построить онтологию требуемой области знаний. С помощью этой оболочки были построены научные интернет-порталы по археологии [Андреева, 2006] и компьютерной лингвистике [Боровикова, 2006].

Работа выполнена при финансовой поддержке РФФИ проект № 13-07-00422 «Методы и технологии создания и управления интеллектуальными научными Интернет-ресурсами на основе онтологий и семантических сетей».

Библиографический список

[Андреева, 2006] Андреева О.А., Боровикова О.И., Булгаков С.В. и др. Археологический портал знаний: содержательный доступ к знаниям и информационным ресурсам по археологии // Тр. 10-й национальной конференции по искусственному интеллекту с международным участием КИИ-2006 (25–28 сентября 2006 г., Обнинск). М.: Физматлит, 2006. Т. 3. С. 832–840.

[Апанович, 2011] Апанович З.В., Винокуров П.С., Кислицина Т.А. Методы и средства визуализации информационного наполнения больших научных порталов // Вестник НГУ Серия: Информационные технологии. 2011. — Т.9, выпуск 3. — С. 5-14.

[Ахмадеева, 2013] Ахмадеева И.Р., Загорулько Ю.А., Саломатина Н.В., Серый А.С., Сидорова Е.А., Шестаков В.К. Подход к формированию тематических коллекций текстов на основе интернет-ресурсов // Вестник НГУ. Серия: Информационные технологии. 2013. Т.11, выпуск 4.

[Боровикова, 2006] Боровикова О.И., Загорулько Ю.А., Загорулько Г.Б. и др. Разработка портала знаний по компьютерной лингвистике // Тр. 11-й национальной конференции по искусственному интеллекту с международным участием КИИ-2008 (Дубна, 2008 г.). М.: ЛЕНАНД, 2008. Т. 3. С. 380–388.

[Загорулько, 2008] Загорулько Ю.А., Боровикова О.И. Подход к построению порталов научных знаний // Автометрия. № 1, 2008, т. 44. –с. 100–110.

[Загорулько, 2011] Загорулько Ю.А. Технология построения порталов научных знаний: опыт применения, проблемы и перспективы // Труды 21-й Международной Крымской конференции «СВЧ-техника и телекоммуникационные технологии» – КрыМиКо-2011 –Севастополь, Крым, Украина, изд. Севастополь: Вебер, 2011. –Т.1. –С.51–54.

[Загорулько, 2012] Загорулько Ю.А., Боровикова О.И. Программная оболочка для построения многоязычных тезаурусов предметных областей, ориентированная на экспертов // Труды 13-й национальной конференции по искусственному интеллекту с международным участием КИИ-2012. – Белгород: Изд-во БГТУ, 2012. -Т.4. -С. 76-83.

[Загорулько, 2013] Загорулько Ю.А. Технологии разработки интеллектуальных систем, основанные на интегрированной модели представления знаний // Открытые семантические технологии проектирования интеллектуальных систем = Open Semantic Technologies for Intelligent Systems (OSTIS-2013): материалы III Междунар. научн.-техн. конф. (Минск, 21-23 февраля 2013 г.) / редкол. : В. В. Голенков (отв. ред.). – Минск: БГУИР, 2013. –С. 31-42.

[Осипов, 1990] Осипов Г.С. Построение моделей предметных областей. Неоднородные семантические сети // Известия АН СССР. Техническая кибернетика. –1990. – №5. – с. 32–45.

[Chinnici, 2007] Chinnici R., Moreau J.-J., Ryman A., Weerawarana S. Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language. W3C Recommendation 26 June 2007. Available at: <http://www.w3.org/TR/wsd120/> (accessed 1 December 2014).

[Guarino, 1998] Guarino N. Formal Ontology in Information Systems // Proceedings of FOIS'98 (Trento, Italy, 1998). Amsterdam: IOS Press, 1998. pp. 3-15.

[Hillmann, 2005] Hillmann D. Using Dublin Core. 2005. Available at: <http://dublincore.org/documents/usageguide/> (accessed 8 May 2014).

[Martin et al., 2004] Martin D. et al. OWL-S: Semantic Markup for Web Services. W3C Member Submission 22 November 2004.

Available at: <http://www.w3.org/Submission/OWL-S/> (accessed 1 December 2014).

[McIlraith, 2001] McIlraith S.A., Son T.C., Zeng H. Semantic Web Services. Intelligent Systems, IEEE, 2001, 16(2). pp. 46-53.

[OWL, 2004] OWL Web Ontology Language Guide. W3C Recommendation 10 February 2004. Available at: <http://www.w3.org/TR/owl-guide/> (accessed 1 December 2014).

ONTOLOGICAL APPROACH TO DEVELOPMENT OF SCIENTIFIC INTERNET RESOURCE

Zagorulko Yu.A. *, Zagorulko G.B. *

* *A.P. Ershov Institute of Informatics Systems
Siberian Branch of the Russian Academy of
Sciences, Novosibirsk, Russia*

zagor@iis.nsk.su

gal@iis.nsk.su

The paper presents an approach to development of intelligent scientific Internet resources (ISIR) intended to providing information and analytical support of research and production activity in certain area of knowledge. An important advantage of ISIR is its ability to appreciably reduce time of access to information required by researchers and processing it due to accumulation of descriptions of entities of the modelled knowledge area and Internet resources relevant it directly in the ISIR content.

A main feature of the approach is using ontology for systematization and integration of knowledge, information resources and intelligent processing facilities relating to the modeled area of knowledge. Besides, a convenient navigation through scientific knowledge and information resource. content-based access to data and intelligent processing facilities as well as information extraction from Internet are also organized on the basis of ontology.



OSTIS-2015

(Open Semantic Technologies for Intelligent Systems)

УДК 004.92

РАЗРАБОТКА И ГЕНЕРАЦИЯ ВИРТУАЛЬНЫХ СРЕД НА ОСНОВЕ СЕМАНТИЧЕСКИХ ПРЕДСТАВЛЕНИЙ

Грибова В.В., Федорищев Л.А.

*Институт автоматизации и процессов управления Дальневосточного отделения
Российской академии наук, г. Владивосток, Россия*

`gribova@iacp.dvo.ru`

`fleo1987@mail.ru`

В работе приводится описание подхода к разработке интеллектуальных систем с виртуальной реальностью (виртуальных сред) на основе семантических представлений. Данный подход позволяет заменить кодирование в разработке и сопровождении виртуальных сред их естественным декларативным описанием.

Ключевые слова: семантические сети, виртуальные среды, онтологии, облачные технологии.

Введение

В настоящее время во всем мире активно используются интеллектуальные системы для решения различных профессиональных задач. Особое место среди них занимают системы виртуальной реальности (виртуальные среды), предназначенные для создания симуляторов, компьютерных обучающих тренажеров, которые позволяют в виртуальной реальности проводить отработку моторных навыков, знаний, ставить безопасные виртуальные эксперименты и т.п.

Однако разработка виртуальных сред (ВС) до сих пор связана с различными трудностями [Трухин, 2008]. Существующие средства ориентированы на программистов, а не на экспертов предметных областей. При этом основу виртуальных сред составляют предметные знания, которые используются и реализуются в рассмотренных средствах неэффективно (встроены в код программного средства).

В лаборатории интеллектуальных систем ИАПУ ДВО РАН реализован комплекс Интернет Разработки Виртуальных Интерактивных Сред – ИРВИС [Gribova et al., 2014]. Данный комплекс работает на облачной платформе IASaaS [Клещев и др., 2011], которая предоставляет широкий спектр современных технологий, включая: прикладные облачные сервисы и средства их разработки.

Целью работы является описание метода создания виртуальных сред на основе семантических представлений и его реализация.

1. Подход к генерации виртуальных сред на основе семантических представлений

1.1. Основные принципы

Основные принципы подхода к генерации виртуальных сред на основе семантических представлений заключается в следующем [Грибова и др., 2012]:

1) Виртуальная среда создается как интерпретируемая декларативная модель, представленная в форме семантической сети, вместо прямого проектирования и реализации ее на языках программирования.

2) Модель виртуальной среды формируется на основе онтологии, которая также представляется семантической сетью.

3) Семантическая сеть декларативной модели включает в свой состав подсети для различных разработчиков: экспертов, дизайнеров, программистов. Это позволяет достичь разделения разработки между специалистами.

4) Все семантические сети находятся на облачной платформе. Это позволяет обеспечить жизнеспособность ВС, поскольку в процессе всего жизненного цикла модель ВС доступна для сопровождения.

5) Семантическая сеть каждой ВС делится на две независимые подсети: среда и ее интерфейс. Данный подход позволяет добиться произвольной гибкости в настройке программы ВС в форме

различных конечных сервисов, построенных на одной и той же семантической сети.

1.2. Семантические представления компонентов виртуальных сред

Архитектура программного комплекса состоит из программных и информационных компонентов. Программными компонентами являются: структурный и графический редакторы, интерпретатор модели. Структурный и графический редакторы предназначены для создания и модифицирования информационных компонентов. Все информационные компоненты представляются семантическими сетями. Программный комплекс состоит из следующих информационных компонентов:

- онтология ВС – семантическая сеть, описывающая структуру декларативных моделей ВС;
- онтология интерфейса ВС – семантическая сеть, описывающая структуру WIMP-интерфейса ВС;
- декларативная модель ВС – семантическая сеть, конкретизирующая онтологию ВС;
- декларативная модель WIMP-интерфейса ВС – семантическая сеть, конкретизирующая онтологию WIMP-интерфейса ВС;
- онтология библиотеки мультимедиа-данных – семантическая сеть структуры библиотеки для хранения и использования мультимедиа-данных в различных ВС; независима от других семантических сетей ВС;
- библиотека мультимедиа-данных – единственная семантическая сеть, конкретизирующая онтологию библиотеки мультимедиа-данных;
- онтология записей – семантическая сеть структуры записей, сделанных на основе использования ВС; независима от других семантических сетей ВС;
- записи – единственная семантическая сеть, конкретизирующая онтологию записей.

Ключевыми информационными ресурсами комплекса ИРВИС являются онтология ВС и онтология WIMP-интерфейса ВС. Онтология ВС состоит из трех основных подсетей: *Объекты*, *Действия* и *Сценарий*.

Объекты = {Простой объект, Изменяемый объект, Составной объект, Таблица, Источник света, Камера}; $0 \leq i \leq$ количество объектов

Простой объект = <Имя объекта, Описание, Логические атрибуты, Презентационные атрибуты >

Имя объекта \in Строка, имя объекта, его идентификатор, по которому к нему можно получить доступ

Логические атрибуты = {Логический атрибут}; $0 \leq i \leq$ количество логических атрибутов

Логический атрибут = <Имя \in Строка, Значение \in Типы данных >

<...>

Действия = {Действие_i | Действие_i \in Действие}, $0 \leq i \leq$ количество действий

Действие = <Имя действия, Описание, Способы выполнения, Входные параметры, Изменение состояния объектов, Получение оценки, Параметры обработки, Сообщение >

Входные параметры = {Входной параметр_i | Входной параметр_i \in Параметр}, $1 \leq i \leq$ количество входных параметров

Параметр = <Значение, Ожидаемое значение, Операция сравнения >

Операция сравнения \in {=, !=, >, <, >=, <=}

<...>

На рис. 1 представлена семантическая сеть "Действия" в виде схемы.



Рисунок 1 - Семантическая сеть "Действия"

Сценарий = <Имя сценария, Этапы, Последовательность этапов >

Этапы = {Этап_i}, $0 \leq i \leq$ количество этапов

Этап = <Входные параметры, Вершины, Дуги, Начальная вершина, Конечная вершина >

Вершины = {Вершина_i \in Вершина}, $0 < i \leq$ количество вершин

Дуги = {Дуга_i \in Дуга}, $0 < i \leq$ количество дуг

Вершина = <Действие, Параметр >

Дуга = <Вершина, из которой выходит дуга, Вершина, в которую входит дуга, Переход>

<...>

Онтология WIMP-интерфейса ВС описывает функциональный состав каждого из типов виртуальной среды (неинтерактивные и интерактивные, со сценарием и без, с роликами и без роликов) и сопоставляет каждому типу набор интерфейсных элементов, реализующих указанную функциональность. Функциональный состав каждого типа виртуальной среды может быть расширен по усмотрению разработчика, при этом он может выбрать произвольный набор интерфейсных элементов и их атрибутов.

WIMP-Интерфейс ВС строится по принципу независимых блоков: Задание, Исследование, Запись и Сценарий с обязательными и необязательными параметрами.

Интерфейс = <Задание, Исследование, Запись, Сценарий>

Задание = <Обучение, Тест, Сообщение, Вопросы, Результат>

Обучение ∈ <Надпись, Расположение>

<...>

2. Реализация семантического подхода

Реализация рассмотренного подхода выполнена на платформе IASPaas [Клещев и др., 2011]. Платформа IASPaas изначально спроектирована и построена с учетом представления различных знаний и программных сервисов в форме семантических сетей, размещенных на облачном сервере. Семантические сети на платформе представлены в виде информационных ресурсов, связанных между собой в одну общую сеть.

Все приведенные семантические сети виртуальных сред для комплекса ИРВИС также представлены отдельными информационными ресурсами. Доступ к информационным ресурсам осуществляется через административные сервисы, главным образом, через редактор информационных ресурсов (IWE).

Редактор IWE - составной компонент платформы IASPaas. IWE является двухуровневым редактором. На первом уровне формируется семантическая сеть метаинформации, по которой автоматически создается редактор для создания семантической сети информации. В инструментальном комплексе ИРВИС метаинформацией являются онтологии среды, интерфейса и т.д., информацией – соответствующие онтологиям модели.

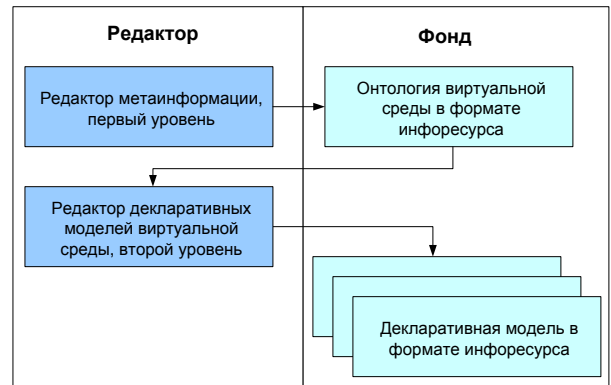


Рисунок 2 - Схема работы редактора семантических сетей

На рис. 2 представлена схема работы редактора семантических сетей IWE. Редактор обеспечивает создание, редактирование, удаление элементов семантической сети виртуальной среды, осуществляет связь между ними, обеспечивает целостность. На втором уровне на вход редактора поступает онтология виртуальной среды, эксперт предметной области, конкретизируя ее элементы, получает логическое представление конкретной модели виртуальной среды. На рис. 3 приведен скриншот интерфейса редактора для формирования модели виртуальной среды в семантическом представлении.

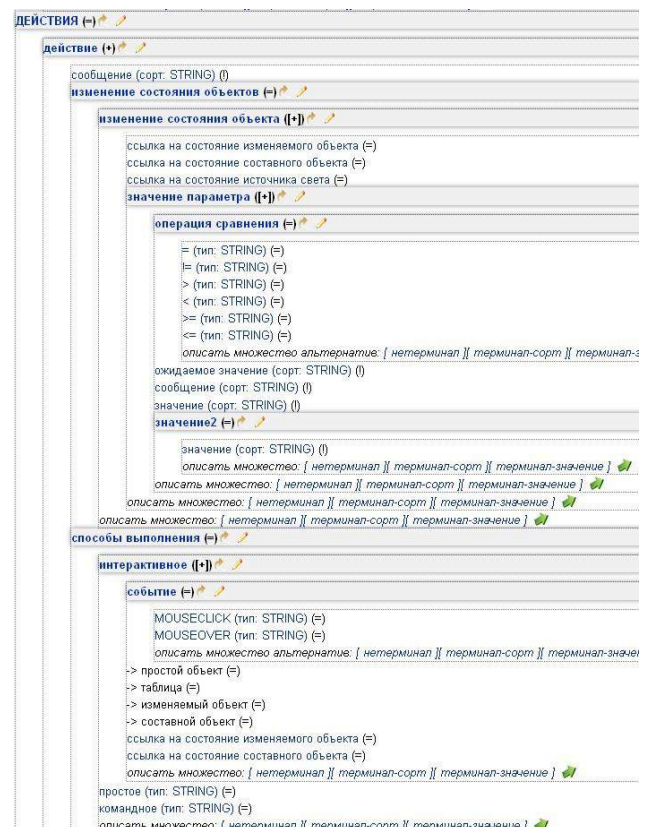


Рисунок 3 - Интерфейс редактора

3. Пример виртуальной среды

На базе комплекса ИРВИС разработаны различные виртуальные среды [Грибова и др., 2014]. В качестве примера рассмотрим обучающий тренажер по классическим методам исследования

зрения в офтальмологии [Черняховская и др., 2011]. Тренажер включает обучающие задания по классическим методам исследования в офтальмологии: определение клинической рефракции (скиаскопии), исследование поля зрения методом кампиметрии, определение остроты зрения по таблицам Сивцева, Головина, определению остроты зрения по оптотипам Б.Л. Поляка, и другие.

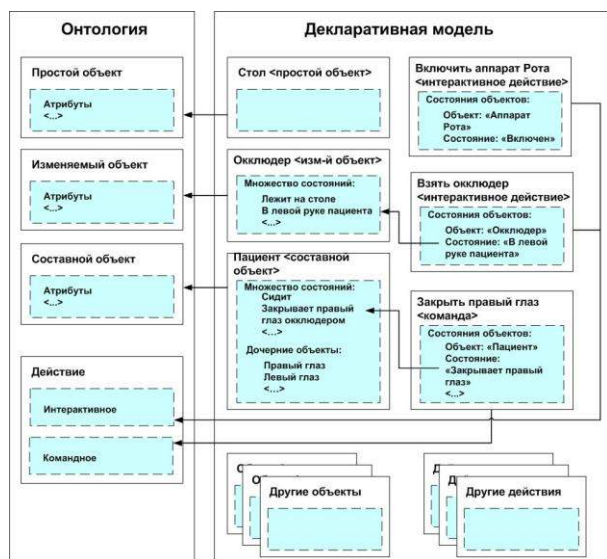


Рисунок 4 - Фрагмент семантической сети тренажера

На рис. 4 приведен фрагмент семантической сети декларативной модели тренажера.

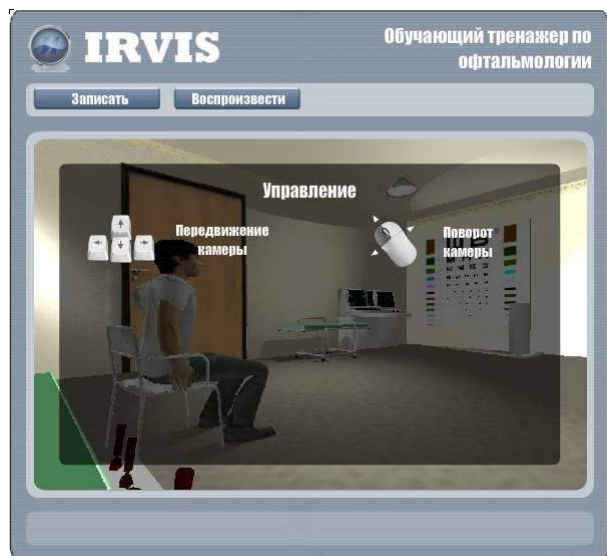


Рисунок 5 - Виртуальный тренажер

На рис. 5 представлен скриншот тренажера.

Заключение

В данной статье описаны принципы применения семантических сетей для разработки такого класса интеллектуальных систем как виртуальные среды. Описывается метод создания виртуальных сред на основе семантических представлений. Приводятся формализованные семантические сети различных компонентов виртуальных сред.

Работа выполнена при финансовой поддержке РФФИ, проект № 13-07-00024-а, 15-07-03193.

Библиографический список

[Gribova et al., 2014] Gribova V, Fedorischev L. Cloud service for development of virtual interactive environments // Proc. of 2014 2nd International Conference on Advanced ICT for Education (ICAICTE 2014), August 16-17, 2014 in Dalian, China. Atlantis Press, 2014. Vol. 1. Pp. 20-23

[Грибова и др., 2012] Грибова В.В., Федорищев Л.А. Обучающие виртуальные системы и средства их создания // Вестник информационных и компьютерных технологий. – 2012. – №3. – С. 48-51

[Грибова и др., 2014] Грибова В.В., Федорищев Л.А. Создание интеллектуальных сервисов с виртуальной реальностью // Материалы 14-й национальной конференции по искусственному интеллекту с международным участием (КИИ 2014). – 24-27 сентября 2014 Казань, Россия.

[Клещев и др., 2011] Клещев, А.С. и другие Облачная платформа для разработки и управления интеллектуальными системами // Труды конференции OSTIS-2011. С. 5-14.

[Трухин, 2008] Трухин, А.В. Анализ существующих в РФ тренажерно-обучающих систем // Открытое и дистанционное образование. - 2008. - №1. - С. 32-39

[Черняховская и др., 2011] Черняховская, М.Ю. и другие. Модель виртуального мира мультимедиа тренажера для медицинского образования // International Book Series "Information Science and Computing", Sofia, Bulgaria, 2011. – №22. – Pp. 140-148

DEVELOPMENT AND GENERATION OF VIRTUAL ENVIRONMENTS ON BASE OF SEMANTIC FORMS

Gribova V.V. *, Fedorischev L.A. *

* Institute of Automation and Control Processes, the Far Eastern Branch of Russian Academy of Science Vladivostok, Russia

gribova@iacp.dvo.ru

fleo1987@mail.ru

The article presents an approach to development of virtual environments by semantic forms. The approach is to change coding of a program by declarative description of the program.

Introduction

At present virtual environments is perspective direction of development. However, development and maintenance of virtual interactive environments are still a complicated process. At this paper we want to show a new approach on the new program complex IRVIS on the cloud platform IACPaaS.

Conclusion

Principles of applying of semantic nets to virtual environments are described at the article. There are presented the method of development, formalized semantic nets and an example of a virtual environment.



УДК 004.89

КЛАССИФИКАЦИЯ ДЛЯ ЗАДАЧ, ВЫЯВЛЯЕМЫХ ПРИ СИСТЕМНОМ АНАЛИЗЕ ИНТЕЛЛЕКТУАЛЬНОЙ ДЕЯТЕЛЬНОСТИ

Шалфеева Е.А.*

* *Институт автоматизации и процессов управления, г. Владивосток, Россия*

shalf@iacp.dvo.ru

В данной статье показаны некоторые результаты литературного обзора классификаций экспертных задач и постановок этих задач, пригодных для практического использования на этапе системного анализа и интеллектуальной деятельности. Обоснована потребность аналитиков и разработчиков систем, основанных на знаниях, в новой открытой классификации задач с их математическими постановками, с которыми могут быть связаны известные методы решения.

Ключевые слова: интеллектуальная деятельность; экспертная задача; классификация задач; постановка задачи.

Введение

Автоматизация профессиональной интеллектуальной деятельности и управления ее качеством требует детальной разработки всех тех этапов, от которых зависит решение проблемы поддержки такой деятельности и правильности применения знаний при этом. Первый из них - этап системного анализа, в рамках которого моделируется предметная область, выявляются задачи, которые решаются специалистами и требуют программной поддержки [Клещев и др., 2014].

Множество исследований было посвящено обсуждению разных интеллектуальных задач и их постановок и предметно-независимых подходов к автоматизации их решения. *Интеллектуальные* или *экспертные* задачи (т.е. задачи, решаемые в областях деятельности человека, которые требуют мышления, определенного мастерства и накопленного опыта [Джексон, 2001]) анализируются в работах Уотермана, Хейеса-Рота, Кленси и в более поздних исследованиях.

Цель настоящего исследования – проанализировать наличие в литературе полной классификации всех экспертных задач и постановок этих задач, пригодных для их практического использования на этапе системного анализа предметной области и интеллектуальной деятельности. Это является частью исследовательской работы по созданию инструментальных средств поддержки разработки систем, основанных на знаниях, в которых форма представления знаний предусматривает возможность непрерывного совершенствования знаний на

протяжении периода эксплуатации системы [Клещев и др., 2014].

1. Цели классификации задач и постановок задач

Работы, представляющие и анализирующие постановки и классификации задач, созданы (если судить по их предисловиям) для тех, кто собирается профессионально заниматься конструированием экспертных систем [Джексон, 2001], иногда только для инженеров по знаниям, системных аналитиков, постановщиков задач [Гаврилова и др., 2000], для формирования знаний и построения обучаемых алгоритмов в прикладных задачах [Воронцов, 2012].

Цель многих «работ по онтологиям», авторы которых: Vargas, L. N. (1996), Guarino (1995), Mizoguchi (1995), Calero C. (2010), - определить способы решения разных задач, чтобы обеспечить использование множества известных методов. Такова же цель в [Chandrasekaran et al., 1998].

Подходы и методы к автоматизации решения задач различают в зависимости от «сложности области приложения», «особенностей решаемых задач», «характера доступного знания» (используемых баз знаний) [Хейес-Рот и др., 1987]. Каждый из методов "эвристическая классификация", "сопоставление", "предложение и применение", "предложение и проверка" и т.п. оказывается эффективным в определенных условиях [Джексон, 2001].

Разработка и анализ методик, аналогичных *эвристической классификации*, - шаг на пути к созданию специальных инструментальных средств

инженерии знаний для этих целей [Джексон, 2001; Clancey, 1985]. И некоторые оболочки (формирующие прототип системы искусственного интеллекта) воплощают методы, применимые для решения экспертных задач [Rychener, 1985].

2. Термины в разных классификациях, определениях и постановках задач

2.1. Названия типичных задач

Термин *задача* (problem) нередко употребляется как стоящая в предметной области профессиональная задача (проблема); которая может быть декомпозирована на непустое множество под-задач (операций [Clancey, 1985]), в том числе – экспертных, связанных с синтезом или анализом некоторых систем в предметной области.

Самые популярные в доступных источниках названия решаемых экспертами задач – *диагностика, интерпретация, планирование и проектирование*. Названиями «типичных», «характерных» экспертных задач являются следующие шесть – *интерпретация, диагностика, предсказание, проектирование, планирование, мониторинг* [Хейес-Рот и др., 1987].

Уотермен к этим шести добавляет еще задачи *отладка, управление поведением*. Кроме того, добавляет задачу *ремонт* и задачу *постановка диагноза и предписание определенных действий* [Уотермен, 1989]. В [Гаврилова и др., 2000] к шести «типичным» добавляются три задачи – *обучение, управление, поддержка принятия решений*.

В более современной публикации к типовым задачам, решаемым экспертными системами, не отнесены задачи *мониторинга* и *предсказания* [Джексон, 2001]. Спектр типовых задач включает: *диагностику, проектирование, планирование* и еще две задачи: *извлечение информации из первичных данных* (что близко к Интерпретации), а также *структурный анализ сложных объектов* (который тоже чаще всего называют интерпретацией, хотя применительно к структурному анализу химических соединений в DENDRAL можно встретить формулировки «выдает диагноз в виде соответствующей химической структуры», и «решает задачу построения возможных химических структур»).

Названия вышеупомянутых шести «типичных» экспертных задач использованы и в [Clancey, 1985], но кроме них есть другие, а *интерпретация* – синоним термина *анализ*. Тот факт, что разные авторы дают разные классификации с разным числом задач, связан не только в их желании использовать классификацию, более адекватную их целям и накопленному опыту, но и с разницей в базовой терминологии. Из-за этого предлагаемые авторами подходы к решению задач трудно применимы для произвольной ситуации: различаются определения задач, их постановки,

определение сущности, по отношению к которой решается задача.

2.2. Термины для сущностей предметной области

Термин *система* часто означает проблемную область со сложными реальными объектами или явлениями [Гаврилова и др., 2000; Попов и др., 1996]. Некоторые авторы вместо термина *система* употребляют *объект* [Попов и др., 1996; Тельнов, 2000] или *ситуация*. Некоторые пользуются несколькими терминами: *system, structure, object* [Rychener, 1985].

Объект чаще встречается в задачах построения\синтеза [Rychener, 1985]. При этом часто имеют в виду *спецификацию на создание объектов* (набор необходимых документов – чертеж, пояснительная записка и т.д.) [Гаврилова и др., 2000]. Нередко термин *объект* употребляется как часть *системы* [Clancey, 1985].

Некоторые авторы задачу построения\проектирования рассматривают применительно к *системам*, технология построения которых сходна с инженерией программного обеспечения. [Clancey, 1985; Sriram et al., 1985; Rychener, 1985].

Но, рассматривая задачи анализа, Кленси имеет в виду т. наз. «системы управления», тип и основные характеристики которых зависят от пар «стимул/реакция»).

Используя термины *динамическая* предметная область (или динамическая проблемная среда), авторы говорят о задачах, учитывающих «фактор времени», например, входные данные изменяются за время сеанса работы приложения (Попов и др., 1996; Тельнов, 2000), и о задачах с событиями, упорядоченными во времени [Heyes-Roth et al., 1983]. Чаще *динамичность* относят к объектам, связывая ее с функционированием объектов [Rychener], изменением объектов с ходом времени [Хейес-Рот и др., 1987] или необходимостью их обслуживания [Rychener].

2.3. Толкования используемых терминов

Чаще всего в доступных публикациях отсутствуют формальные постановки задач, дается только их описание (характерные особенности, отличающие от других задач). При этом даже наиболее «распространенные» задачи описываются по-разному.

2.3.1. Распространенные задачи

Чаще всего *диагностика* описывается как процесс обнаружения в системе неисправностей [Гаврилова и др., 2000] некоторого класса или отклонения параметров системы от нормативных [Павлов, 2011]), но нередко отмечают другую цель задачи, а именно – выявление причин, приведших к возникновению ситуации [Clancey, 1985; Тельнов, 2000; Rychener, 1985]. Есть пример того, что в задаче *диагностики* системы учитывают свойства ее

составных элементов (если во фразе «необходимость понимания функциональной структуры ...диагностирующей системы» [Гаврилова и др., 2000] допущена опечатка и имеется в виду диагностируемая система.)

Мониторинг - непрерывная интерпретация данных в реальном масштабе времени и сигнализация о выходе тех или иных параметров за допустимые пределы (обычно на основе показаний датчиков) [Гаврилова и др., 2000]. Кленси связывает мониторинг с обнаружением «отклонения в поведении» (получаемом в ответ на поданный сигнал) [Clancey, 1985]. Но иногда *мониторинг* определяется, скорее, как измерительная (не экспертная) задача (слежение за текущей ситуацией [Тельнов, 2000]).

Описания задачи *проектирование* от разных авторов похожи друг на друга: *проектирование* - определение конфигурации объектов [Тельнов, 2000]) или синтез потенциальной структурной конфигурации [Srigam et al., 1985] или построение структурной организации компонентов [Clancey, 1985], которая удовлетворяет заданным ограничениям, критериям. Вариант описания «подготовка спецификаций на создание объектов с заранее определенными свойствами» [Гаврилова Т.А., Хорошевский, 2000] можно считать аналогичным, если в спецификации-чертеже «видеть» структурную организацию компонентов объекта. Но вариант от [Ruchener, 1985] «построение объекта\ системы по данной спецификации» выглядит как задача, «следующая за» проектированием (как *сборка* у Кленси).

Интерпретация (данных) - процесс определения смысла данных, результаты которого должны быть согласованными и корректными [Гаврилова Т.А., Хорошевский, 2000], формирования описания ситуаций по результатам наблюдений [Хейес-Рот и др., 1987] или определение сущности рассматриваемой ситуации [Тельнов, 2000]. Под это описание попадают два разных случая. Один – когда данные невербальны: «непосредственно реальные данные» или «естественная информация», противопоставляемые четким символическим представлениям проблемной ситуации. Второй - когда данные представляют собой символические представления - набор признаков и их значений (например, упоминаемые 16 независимых факторов, описывающих Личность в [Гаврилова и др., 2000] или «факты, на основе которых ... анализируют финансовое состояние предприятия [Тельнов, 2000]).

Не являются одинаковыми описания задачи *прогнозирование*: предсказывание хода событий в будущем (на основании модели прошлого и настоящего) [Хейес-Рот и др., 1987]; предсказание последствия некоторых событий или явлений (на основании анализа имеющихся данных) [Гаврилова и др., 2000] или предсказание последствий развития текущих ситуаций (на основе математического и эвристического моделирования) [Тельнов, 2000].

Описания задачи *планирование* от разных авторов похожи друг на друга: выбор последовательности действий по достижению поставленной цели [Тельнов, 2000; Джексон, 2001] или «нахождение планов действий» для объектов-исполнителей [Гаврилова и др., 2000]. В подробных описаниях этой задачи можно заметить, что шаги плана рассматривают как компоненты конфигурируемого решения [Джексон, 2001]. (Стоит отметить, что в [Тельнов, 2000] составление расписаний не отнесено к планированию, а выделено в отдельную задачу «Диспетчирование».)

2.3.2. Менее распространенные задачи

Управление - управление поведением (функция, поддерживающая определенный режим деятельности) сложных систем [Гаврилова и др., 2000], управление поведением системы [Уотермен, 1989]. Но у [Кленси] Управление - это «простая задача» (операция) определения того, какие стимулы (входные сигналы) следует подать на вход системе, чтобы получить желаемую реакцию [Clancey, 1985; Джексон, 2001]. Однако на практике реализация функции поддержки определенного режима функционирования системы связана с определением *управления* как составной задачи: функция ... управления поведением сложной системы в соответствии с заданными спецификациями – это «интерпретация, прогноз, планирование, моделирование, оптимизация выработанных решений, мониторинг» [Павлов, 2011]. Встречаются описания\определения *управления* как расширение другой задачи: мониторинг, дополненный реализацией действий [Тельнов, 2000].

Задачу *Управление* сводят и к «принятию решений» (например, на биржевых торгах [Тельнов, 2000]). Но *поддержка принятия решений* иногда выделяется в самостоятельную задачу (или совокупность процедур) обеспечения необходимой информацией и рекомендациями, помогающими выбрать и/или сформировать нужную альтернативу среди множества выборов при принятии ответственных решений (напр., «выбор стратегии выхода из кризисной ситуации; выбор страховой компании или инвестора» [Гаврилова и др., 2000]).

Распространенная на практике задача *коррекции* (называемая также *отладкой* или *лечением* в соответствующих предметных областях) - выработка рекомендаций по устранению неисправностей [Павлов, 2011] или рекомендации действий по исправлению отклонений от нормального состояния рассматриваемых ситуаций [Тельнов, 2000]. *Формирование курса лечения* - определение комбинации назначенных больному препаратов и их дозировок [Джексон, 2001]; *отладка* - определение способов устранения неисправностей [Уотермен, 1989]. Эту задачу часто вместе с диагностикой рассматривают как единую задачу [Тельнов, 2000]), а Уотермен дает название задаче «постановка диагноза и предписание определенных действий».

Ряд авторов упоминают и определяют задачу *определения принадлежности ситуации к некоторому классу*, при этом называют ее по-разному: задача *распознавания различных ситуаций* [Тельнов, 2000], диагностика - процесс соотнесения объекта с некоторым классом объектов [Гаврилова и др., 2000]. Часто взаимозаменяемо используют названия *классификация* и диагностика, поясняя, что решение задачи классификации – главный шаг диагностики или употребляя как синонимы «классифицировать вид заболевания» и «дифференциальная диагностика» [Воронцов, 2012].

2.3.3. Редкие задачи

Идентификацией в [Clancey, 1985] названа задача анализа, использующая в качестве анализируемой информации пар сигналов «стимул/реакция».

Структурный анализ сложных объектов (например, химических соединений) [Джексон, 2001].

Диспетчирование - распределение работ во времени, составление расписаний, например, планирование графика освоения капиталовложений. [Тельнов, 2000].

В [Clancey, 1985] даны задачи построения - *спецификация* (ее результат - ограничения, которым должна удовлетворять синтезируемая система) и *сборка* («реализует спроектированную систему, собирая воедино отдельные ее компоненты»).

Довольно часто при обсуждении спектра решаемых задач упоминается необходимость *определения того, какой информации не хватает*, чтобы получить множество решений. Обычно такую задачу совмещают с той задачей, для решения которой не хватает информации: на абстрактном уровне (задачи с неопределенными знаниями - *задачи доопределения недостающей информации* и получения множества возможных решений [Тельнов, 2000]) или на более конкретном уровне (классифицировать вид заболевания и *находить совокупности симптомов*, наиболее характерные для данного заболевания) [Воронцов, 2012].

Стоит отметить, что, характеризуя *задачу анализа* и *задачу синтеза*, авторы делают акценты на разных особенностях получения их решения. По [Clancey, 1985] и [Попов, 1996] в задачах анализа требуется определить неизвестные характеристики \ свойства системы \ модели объекта. Результатом задач синтеза считают построение модели объекта \ системы (по заданным условиям) [Попов, 1996] или изменение ее конструкции [Clancey, 1985]. В [Гаврилова и др., 2000] задачи синтеза отличаются по построению \ конструированию решений из частей (компонентов или под-проблем), при этом множество решений потенциально не ограничено в отличие от задач анализа.

Параллельно с задачами анализа и синтеза, решаемыми с использованием заранее построенных баз знаний (БЗ), другие авторы определяют *задачи*

построения самих БЗ (рассматривая их даже как «прикладные задачи» - задачи классификации, регрессии и прогнозирования в самых разных областях человеческой деятельности) [Воронцов, 2012]. Накопив достаточное количество прецедентов (признаковое описание объектов: бинарных порядковых и количественных признаков, а также применявшиеся методы воздействия на объект), можно формировать знания для решения разных задач. Знания моделируют зависимость совокупных свойств от его атрибутов и воздействий на него (множества влияющих факторов) с помощью обучения по прецедентам [Воронцов, 2012]. Речь обычно идет о решении на основе таких знаний задач классификации, лечения; предсказания.

Из описания «систем экспертного типа» ясно, что отдельно могут быть рассмотрены *задачи критики принятых решений* [Кобринский 2010] (постановки таких задач не приводятся, но указывается на полезность, как минимум, в учебном процессе "критикующих систем экспертного типа», приводятся примеры системы поиска ошибок в предлагаемом решении задачи планирования предоперационной подготовки и системы оценки решения задачи *задачи доопределения недостающей информации* – «назначения дополнительного обследования больному»).

Таким образом, понимание спектра задач и сути задач у разных авторов значительно отличаются, что не позволяет выбрать какой-то один из источников для применения накопленного опыта в своих целях.

3. Собственно классификации задач

Поскольку почти всегда предлагаемые исследователями задачи нельзя рассматривать как взаимно исключающие, другие авторы занимались дальнейшим их совершенствованием.

Говоря о разных экспертных задачах, авторы чаще всего базировались на дихотомии «задача анализа - задача синтеза» [Clancey, 1985; Попов и др., 1996; Гаврилова и др., 2000; Тельнов, 2000]. (Кленси задачу анализа называет *интерпретацией*, и эта его «родовая операция» «накрывает» любые другие задачи, в том или ином виде выполняющие «описание обслуживаемой системы», например, задачу предсказания).

Но следующий уровень рассмотрения задач у разных авторов отличается. В [Гаврилова и др., 2000] задачами анализа являются: *интерпретация данных, диагностика, поддержка принятия решения*. У [Clancey, 1985] подклассы другие: *идентификация, предсказание, управление*. В этом случае появляется еще один уровень, на котором подклассы *задачи идентификации* различаются получаемым результатом: *мониторинг* и *диагностирование*.

В [Тельнов, 2000] различаются (для динамической предметной области) другие задачи

анализа: *распознавание, интерпретация, диагностика и коррекция*. При этом вводится дополнительный уровень разделения: задачи с детерминированными знаниями (задачи распознавания принадлежности ситуации к некоторому классу) и задачи с неопределенными знаниями (интерпретация, диагностика и коррекция) [Тельнов, 2000].

К задачам *синтеза* все авторы относят *проектирование* и *планирование*. Проектирование и планирование принято рассматривать на одном уровне [Rychener, 1985]. К этим двум добавляют управление в [Гаврилова и др., 2000]. А в иерархии задач синтеза [Clancey, 1985] проектирование дополнено спецификацией и сборкой (однако, судя по их описанию, это подзадачи, решаемые обычно вместе: по спецификации решается задача проектирования, после чего требуется сборка).

Кленси считает частными случаями (или частью) *проектирования* задачи *конфигурирование* и *планирование*; здесь это две взаимосвязанные подзадачи, и можно сказать, что наличие плана построения заданной конфигурации позволяет рассматривать результат проектирования как проект, реализуемый на практике. Планирование поставили на следующий уровень классификации и Reichgelt и van Harmelen (1986) [Джексон, 2001], трактуя планирование как «проектирование последовательности операций».

Другой пример следующего уровня задач «под» проектированием - *conceptual design, analysis и detailed design* [Sriram et al., 1985]. Это согласуется с мнением «решение задачи синтеза включает в себя анализ» [Попов и др., 1996], потому что процесс решения содержит, в частности, шаги: создание исследовательской модели объекта и анализ этой модели.

Формирование курса лечения - пример задачи, в которой решение конструируется [Джексон, 2001], но в классификации Кленси среди задач синтеза нет ни лечения, ни отладки.

Тот факт, что *управление* оказывается у одних задачей анализа, а у других – синтезом означает, что «управление» - наиболее неоднозначно используемый термин в качестве экспертных задач.

При рассмотрении задач в динамических областях [Попов, 1996; Тельнов, 2000] или в связи с «реальным временем» [Гаврилова и др., 2000]: предлагаются названия трех групп задач – *статические, квазидинамические, динамические* [Попов, 1996], [Гаврилова и др., 2000]. Но классификации задач здесь не появляется, а приводимые примеры показывают, что задачи анализа попадают в каждую группу.

Таким образом, нет стройной проблемно-ориентированной классификации, позволяющей сопоставить вновь выявленную или возникшую задачу некоторой ранее поставленной и/или решенной.

4. О постановках задач

Постановки задач встречаются в [Rychener, 1985] (для трех задач: *Diagnosis, Design и Planning*), в [Воронцов, 2012]. Пример постановки (для задачи диагностики) таков.

Дано: ситуация работы со сбоями, необычные признаки; стандартный набор диагностических тестов;

цели: сопоставить известной категории болезней; найти вероятные объяснения (причины) признаков; рекомендовать методы лечения;

ограничения (например): о важности выбора некоторых тестов из больших наборов, возможно, очень дорогих тестов;

шаги решения: вывести возможные причины признаков; собрать данные о ситуации через опросы и выполнение тестов; сгруппировать возможные причины в гипотезу о болезни (ошибке); отличать конкурирующие гипотезы; принимать во внимание взаимодействия нескольких причин; принимать во внимание историю системы; вести рассуждение на основе «общего причинного» знания системы, или на основе теории [Rychener, 1985].

Наличие таких постановок позволяет видеть различия и общие черты разных задач. Например, *задача проектирования* имеет те же *ограничения*, что *задача диагноза*, а *планирование* - ту же *цель*, что касается *проектирования* [Rychener, 1985].

Постановки задач не одинаковы для разных уровней подробности их рассмотрения. Так, «ограничения» для более абстрактной задачи *планирование* содержат: «непредвиденные обстоятельства в фактической процедуре выполнения; ограничения на то, как процедурные шаги взаимодействуют» [Rychener, 1985]. А для более «узкой» задачи («планирование действий робота») ограничения специфичны: «физические ограничения, предписывающие, как можно захватывать и размещать; временные ограничения на то, как связывать воедино операции захвата и размещения» [Джексон, 2001].

Очевидна нехватка полного набора постановок всех задач, которые позволили бы устанавливать соответствие своей задачи уже ранее определенной (или увидеть необходимость в уточнении постановки некоторой общей задачи).

5. Отличительные черты новой классификации

Для того чтобы при проведении системного анализа для автоматизации профессиональной интеллектуальной деятельности рационально выделять отдельные подзадачи, методы решения которых уже известны, и необходимая для решения информация специфицирована (на некотором уровне абстракции), требуется единая классификация постановок задач. Задачи новой классификации должны быть представлены так, чтобы их отличия были явны и однозначны.

Предполагается, что весь спектр задач, относящихся к области разработки систем, основанных на знаниях, должен быть охвачен этой классификацией. Однако не менее важна открытость классификации как возможность естественного добавления к ней возможных новых задач как подклассов задач, уже представленных на одном из уровней абстракции.

Основными понятиями при описании задач новой классификации должны стать достаточно абстрактные *предметные символы*, *функциональные символы*, и *предикаты* а в постановках задач должны присутствовать *база знаний* (непустое множество предложений на логическом языке, представляющее свойства предметной области) и *алгебраическая система*, представляющая систему или ситуацию в предметной области.

Предполагается провести различие не только между *задачами анализа* (или более конкретно - *задачами анализа результатов наблюдений*), и *задачами синтеза* (или более общей группы задач *анализа условий на решения*), но различать задачи *индукции* и *дедукции* (в зависимости от того, задана ли база знаний или требуется ее сформировать).

Ожидается, что новая классификация должна будет представить задачи, возникающие в предметных областях, где существует конечное множество «классов» ситуаций («пространство альтернатив»).

Аналогичное «внимание» ожидается по отношению к:

- задачам, решаемым для систем, состоящих из компонентов, с учетом пространственных отношений между ними;
- задачам, в которых определяющую роль играет время, а понятия представлены функциональными соответствиями и отношениями, зависящими от времени;
- задачам, в которых от времени зависят действия, приводящие к некоторой заданной цели (или действия упорядочены хотя бы частично);
- задачам, решаемым для ситуаций, описываемых с помощью причинно-следственных отношений.

Тогда каждая выделенная при проведении системного анализа профессиональной деятельности отдельная подзадача либо сможет быть «спроецирована» на такую классификацию либо «расширит» классификацию: будет уточнением существующей (абстрактной) задачи или будет добавлена как новая подзадача на некотором уровне.

Заключение

В данной статье обоснована потребность аналитиков и разработчиков систем, основанных на знаниях, в новой открытой классификации задач с их математическими постановками.

Работа выполнена при финансовой поддержке грантов РФФИ № 15-07-03193: «Облачные технологии обеспечения жизнеспособности и качества интеллектуальных систем» и № 13-07-00024 «Облачная платформа для создания и использования интеллектуальных сервисов».

Библиографический список

- [Воронцов, 2012] Воронцов К. В. Математические методы обучения по прецедентам. 2012. [http://www.machinelearning.ru/wiki/index.php?title=Машинное_обучение_\(курс_лекций_К.В.Воронцов\)](http://www.machinelearning.ru/wiki/index.php?title=Машинное_обучение_(курс_лекций_К.В.Воронцов)).
- [Гаврилова и др., 2000] Гаврилова, Т. А., Хорошевский В.Ф. Базы знаний интеллектуальных систем. СПб: Питер, 2000, 384 с.
- [Джексон, 2001] Джексон П. Введение в экспертные системы: Уч. пос. – М.: Издательский дом "Вильямс", - 2001. – 624 с.
- [Клещев и др., 2014] Клещев А.С., Шалфеева Е.А. Содержание системного анализа при автоматизации интеллектуальной деятельности на уровне отрасли // Открытые семантические технологии проектирования интеллектуальных систем (OSTIS-2014): материалы IV междунар. науч.-техн. конф. (Минск, 20–22 февраля 2014 года)/ редкол.: В.В. Голенков (отв. ред.) [и др.]. – Минск: БГУИР, 2014. С. 285–290.
- [Кобринский, 2010] Кобринский Б. А. Системы поддержки принятия решений в здравоохранении и обучении // Врач и информационные технологии : Научно-практический журнал. - 2010. - N2. - С. 39-45.
- [Павлов, 2011] Павлов С. Н. Системы искусственного интеллекта : учеб. пособие. В 2-х частях. / Томск: Эль Контент, 2011. — Ч. 1. — 176 с.
- [Попов и др., 1996] Попов Э.В., И.Б. Фоминых, Е.Б. Кисель, М.Д. Шапот. Статические и динамические экспертные системы, учебное пособие. - М: «Финансы и статистика», 1996, с. 318.
- [Тельнов, 2000] Тельнов Ю.Ф. Интеллектуальные информационные системы: учебное пособие. М.: МЭСИ, - 2000. - 81 с.
- [Уотермен, 1989] Уотермен Д. Руководство по экспертным системам: Пер. с англ. М: Мир, - 1989 г. 388 с.
- [Хейес-Рот и др., 1987] Хейес-Рот Ф., Уотермен Д., ЛенатД. Построение экспертных систем. М: Мир, - 1987. - 441 с.
- [Chandrasekaran et al., 1998] Chandrasekaran B., Josephson J. R., Richard B.V. Ontology of Tasks and Methods // The Eleventh Workshop on Knowledge Acquisition, Modeling and Management, 18-23 April, Banff, Alberta, Canada. - 1998.
- [Clancey, 1985] Clancey W. J. Heuristic Classification // Artificial Intelligence, 1985, #27, 289-350.
- [Rychener, 1985] Rychener M.D. Expert systems for engineering design // Expert Systems. - 1985. vol. 2. № 1, - P. 30-44.
- [Sriram et al., 1985] Sriram D., Maher. M. L. and Fenves, S. J. Knowledge-Based Expert Systems for Structural Design // Computers and Structures, January - 1985. - P. 1-9.

A CLASSIFICATION FOR THE TASKS REVEALED DURING INTELLIGENT ACTIVITY SYSTEM ANALYSIS

Shalfееva E. *

* The Institute of Automation and Control Processes, Vladivostok, Russia

shalf@iacp.dvo.ru

Some results of review of expert tasks classifications and task statements suitable for practical use during system analysis are presented in this article. The need of new open classification of tasks for analysts and developers of the systems founded on knowledge is proved. Such classification has to contain mathematical statements of tasks with which known methods of task decision can be connected.



OSTIS-2015

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822

ТЕХНОЛОГИЯ РАЗРАБОТКИ РЕШАТЕЛЕЙ ЗАДАЧ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ С ИСПОЛЬЗОВАНИЕМ ИНСТРУМЕНТАЛЬНЫХ СЕРВИСОВ ОБЛАЧНОЙ ПЛАТФОРМЫ IASRAAS

Крылов Д.А., Москаленко Ф.М., Тимченко В.А.

Федеральное государственное бюджетное учреждение науки Институт автоматизации и процессов управления Дальневосточного отделения Российской академии наук, г. Владивосток, Россия

dmalkr@gmail.com

philipmm@iacp.dvo.ru

vadim@dvo.ru

В работе описана технология разработки агентов, входящих в состав мульти-агентных решателей задач прикладных интеллектуальных облачных сервисов, с использованием инструментальных (системных) сервисов платформы IASraaS. Технология направлена на снижение за счет ее применения трудоемкости разработки и, прежде всего, сопровождения интеллектуальных систем как облачных мульти-агентных сервисов.

Ключевые слова: интеллектуальные системы, мульти-агентные системы, агентно-ориентированное программирование, облачные сервисы.

Введение

Разработка и сопровождение интеллектуальных систем (ИС) или систем, основанных на знаниях (состоящих, в общем случае, из решателя задач, пользовательского интерфейса и базы знаний), является чрезвычайно сложным и трудоемким процессом. Данный процесс имеет свои особенности, поэтому стандартная технология разработки программных систем не может быть напрямую спроецирована на разработку ИС. В первую очередь это связано с тем, что база знаний ИС должна разрабатываться и сопровождаться экспертами предметной области и быть им понятной. Описанные в литературе технологии разработки ИС предполагают, в частности, что база знаний формируется в результате взаимодействия эксперта предметной области с инженером знаний. Однако в работах [Грибова и др., 2010], [Kleshchev, 2011], [Грибова и др., 2013a], [Грибова и др., 2013b] указываются недостатки такого подхода, особенно проявляющиеся при сопровождении базы знаний. В результате на сегодняшний день можно говорить об отсутствии принятой в качестве стандарта технологии, позволяющей разрабатывать жизнеспособные адаптивные ИС. Наряду с этим по-прежнему ощущается острая потребность в средствах разработки ИС, а также в повторном использовании компонентов ИС. Важной задачей при этом также является обеспечение доступа

пользователей к ИС и средствам их разработки на протяжении всего жизненного цикла ИС. Существующие же технологии основаны на традиционном подходе к сопровождению программных средств, при котором версия системы передается конечному пользователю и который не предусматривает оперативное устранение возможных ошибок, централизованного хранения и обновления онтологий и баз знаний, модификацию решателей задач, сводя эти процессы лишь к смене/обновлению версий системы.

Современный подход к разработке программных систем основан на использовании технологии облачных вычислений, которые, с одной стороны, обеспечивают доступность программных систем широкому кругу пользователей, с другой – позволяют на протяжении всего жизненного цикла программной системы осуществлять ее сопровождение, поскольку все компоненты системы остаются доступными ее разработчику. К настоящему времени при использовании данного подхода для разработки ИС созданы редакторы информационного наполнения ИС, компоненты которых можно использовать при создании новых ИС [Gennari et al., 2003], [Орлов и др., 2006a], [Protege, 2014]; разработаны и широко используются платформы [Орлов и др., 2006b], [Sanderson, 2009], [TopQuadrant, 2014], которые поддерживают отдельные этапы цикла разработки облачных программных систем.

Однако на сегодняшний день отсутствуют облачные платформы, полностью поддерживающие разработку, функционирование и сопровождение всех трех компонентов ИС (базы знаний, решателя задач, интеллектуального интерфейса), а также концепцию единообразного представления, хранения и повторного использования информационного наполнения и программных компонентов ИС.

Для решения вышеперечисленных проблем в области разработки и сопровождения жизнеспособных ИС в работах [Грибова и др., 2011], [Gribova et al., 2013] предложена концепция облачной платформы IACPaaS, поддерживающей следующие технологические принципы разработки, сопровождения и использования облачных ИС:

- все информационные ресурсы (онтологии, знания, данные) и декларативные компоненты решателей задач имеют единое унифицированное представление (в виде иерархической семантической сети) [Lehmann, 1992];
- формирование и сопровождение знаний осуществляется экспертами предметной области на основе моделей онтологий;
- пользовательский интерфейс редакторов для эксперта генерируется по модели онтологии;
- метод решения задачи разбивается на под-

задачи, где каждой подзадаче соответствует агент;

- для доступа агентов к информационным ресурсам, имеющим унифицированное представление, предусмотрены программные интерфейсы;

- ИС предоставляется пользователю как облачный мульти-агентный сервис.

В данной работе описывается технология разработки решателей задач ИС на платформе IACPaaS, каждый из которых представляет собой множество взаимодействующих между собой посредством обмена сообщениями агентов.

1. Облачная платформа IACPaaS

Облачная платформа IACPaaS представляет собой программно-информационный интернет-комплекс, предоставляющий контролируемый доступ и единую систему администрирования для создания и использования интеллектуальных сервисов и их компонентов, поддержку функционирования агентов (через передачу и обработку сообщений, запуск блоков продукций – обработчиков сообщений). Комплекс основан на технологии облачных вычислений и обеспечивает удаленный доступ конечным пользователям к интеллектуальным системам, а разработчикам и управляющим – к средствам создания интеллектуальных систем и управления ими. Концептуальная четырехуровневая архитектура платформы IACPaaS представлена на рис. 1.

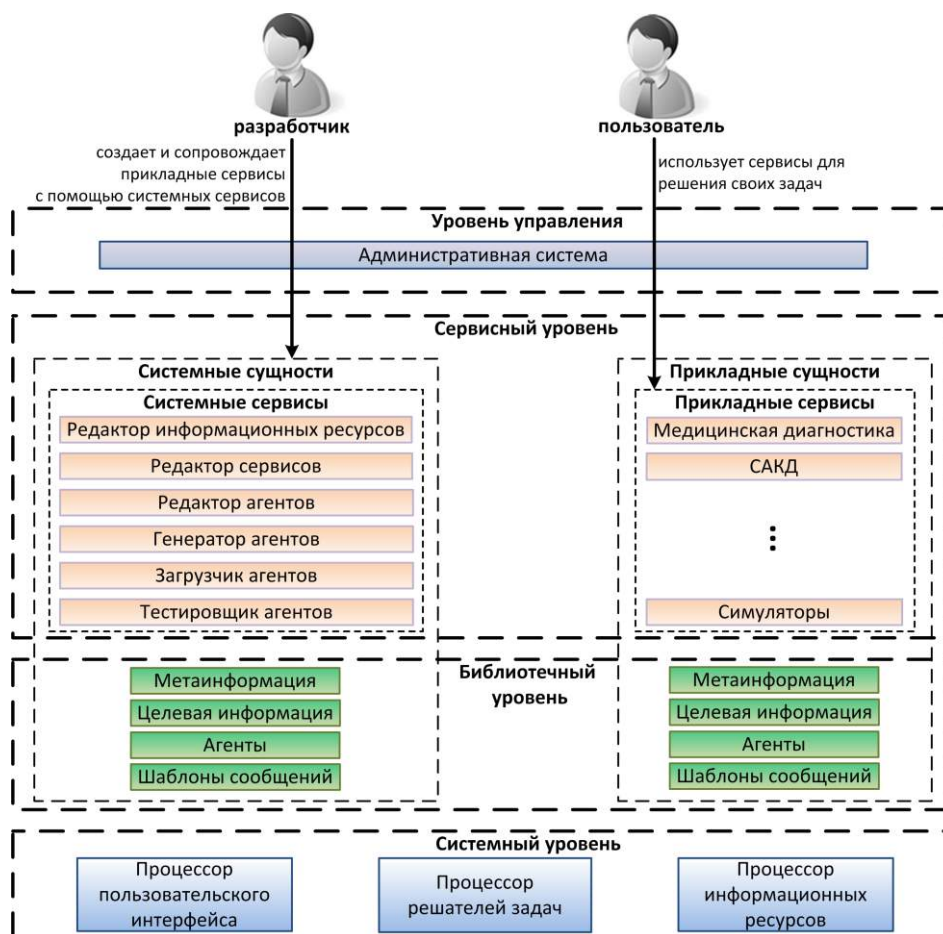


Рисунок 1 – Концептуальная четырехуровневая архитектура платформы IACPaaS

1. *Системный уровень* (уровень *виртуальной машины*). Виртуальная машина платформы IASaaS состоит из процессора информационных ресурсов (ПИР), процессора решателей задач (ПРЗ) и процессора пользовательского интерфейса (ППИ), каждый из которых предназначен для поддержки соответствующих компонентов интеллектуальных систем. На данном уровне обеспечиваются доступ к Фонду информационных ресурсов платформы, запуск и работа сервисов, а также взаимодействие сервисов с пользователями.

2. *Библиотечный уровень*. Это уровень отдельных (повторно используемых при разработке и сопровождении) компонентов сервисов, к которым относятся агенты, шаблоны сообщений, используемые для взаимодействия между агентами, а также информационные ресурсы двух типов: представляющих метаинформацию и целевую информацию.

3. *Сервисный уровень*. Данный уровень представляет собой совокупность сервисов, каждый из которых представлен множеством взаимодействующих посредством обмена сообщениями агентов, обрабатывающих информационные ресурсы Фонда. Выделяются прикладные сервисы, создаваемые прикладными разработчиками для решения задач пользователей, и системные сервисы, необходимые для функционирования платформы и развития ее Фонда информационных ресурсов.

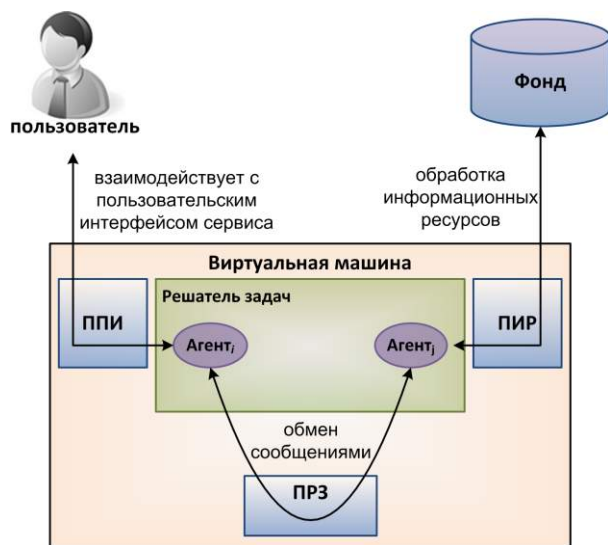


Рисунок 2 – Архитектурно-контекстная диаграмма сервисов

4. *Уровень управления*. Данный уровень представлен системным сервисом платформы *Административная система*, которая предназначена для обеспечения контролируемого доступа к функциональным возможностям платформы IASaaS и управления правами доступа на использование прикладных и сервисов, разработанных на базе данной платформы.

Системные сервисы обеспечивают инструментальную поддержку технологии разработки прикладных облачных мульти-агентных

сервисов и их компонентов (в т.ч. входящих в состав сервисов агентов). Технология разработки агентов и шаблонов сообщений, которые они могут принимать и/или отправлять, состоит из следующих традиционных для создания программных средств этапов [Фатрелл и др., 2004]: *разработка требований, проектирование, реализация, тестирование и отладка, ввод в эксплуатацию*.

2. Разработка агентов и шаблонов сообщений

2.1. Разработка требований к агенту

На этом этапе формулируются требования к функциональности агента; структуре его входных и выходных данных; структуре сообщений, посредством которых он взаимодействует с другими агентами; структуре информации, доступной в любом блоке продукции агента (при каждом обращении к агенту во время работы сервиса) и используемой для хранения собственных данных, настроек агента, управляющих логикой его работы и т.п. – локальной структуре данных (может отсутствовать, если все нужные данные доступны через сообщения и/или информационные ресурсы). Платформа IASaaS не предоставляет средств для разработки и управления требованиями к агентам.

2.2. Проектирование агента

На данном этапе, также выполняемом без использования средств платформы, проектируется устройство и поведение агента как совокупности блоков продукции, обрабатывающих принимаемые сообщения. Для каждого блока продукции определен свой шаблон входных сообщений. Ввиду использования платформы IASaaS на следующих этапах, при проектировании необходимо соблюсти следующие условия:

- проанализировать содержимое фонда платформы IASaaS на наличие шаблонов сообщений, которые можно использовать при проектировании работы некоторых блоков продукции агента, и в случае обнаружения таких сущностей, их необходимо задействовать на этапе реализации агента;
- *Корневой агент* должен содержать блок продукции, выполнение которого инициируется сообщением по шаблону *Инициализирующее сообщение*;
- если агент должен завершать работу сервиса, то у него должен быть блок продукции, в процессе выполнения которого посылается сообщение по шаблону *Финализирующее сообщение*;
- агент *Интерфейсный контроллер* должен иметь блок продукции, выполнение которого инициируется сообщением по шаблону *Запрос от агента Вид*;
- для ответа агенту платформы *Вид* (с целью отображения интерфейса) агент сервиса должен

содержать блок продукций, в процессе выполнения которого посылается сообщение по одному из шаблонов: *Отобразить окно, Вернуть инфоресурс в окно, Вернуть строку в окно.*

2.3. Реализация агента

Реализация агента состоит в формировании в фонде платформы IACPaaS информационных ресурсов, представляющих декларативные спецификации разрабатываемого агента и используемых в нем новых шаблонов входных и выходных сообщений; генерации заготовок исходного кода агента и шаблонов сообщений по их декларативному описанию; написании кода агента и шаблонов сообщений (в частности, кода блоков продукций агента); получении байт-кода агента и шаблонов сообщений и загрузки его в фонд.

2.3.1. Формирование информационных ресурсов

Формирование в фонде информационного ресурса, представляющего декларативную спецификацию шаблона сообщений, выполняется по следующей схеме.

1. Создание с использованием Административной системы в фонде платформы IACPaaS нового информационного ресурса (с названием *Шаблон <имя>*, где *<имя>* – название шаблона, присвоенное ему на этапе проектирования) по метаинформации *Структура шаблона сообщений*, описывающей онтологию декларативных представлений шаблонов сообщений платформы IACPaaS.

2. Формирование содержимого созданного информационного ресурса с использованием системного сервиса *Редактор информационных ресурсов*, в котором процесс редактирования управляется метаинформацией *Структура шаблона сообщений*. Специфицирование шаблона сообщений состоит в задании:

- описания шаблона (на естественном языке);
- внутреннего имени шаблона (используемое при формировании имени класса шаблона сообщений на этапе генерации заготовки его исходного кода);
- требуемой структуры содержимого сообщений (если необходимо).

Формирование в фонде платформы информационного ресурса, представляющего декларативную спецификацию агента, выполняется по следующей схеме.

1. Создание с использованием Административной системы в фонде платформы IACPaaS нового информационного ресурса, представляющего декларативную спецификацию разрабатываемого агента, по метаинформации *Структура агента*, описывающей онтологию декларативных представлений агентов платформы.

2. Формирование содержимого созданного информационного ресурса с использованием системного сервиса *Редактор агентов*, в котором

процесс редактирования управляется метаинформацией *Структура агента*. Специфицирование агента состоит в задании:

- описания агента (на естественном языке);
- внутреннего имени агента (используемое при формировании имени класса агента на этапе генерации заготовки его исходного кода);
- локальной структуры данных требуемого вида (если необходимо);
- для каждого блока продукций агента:
 - описания;
 - шаблона входных сообщений – сообщений, инициирующих выполнение данного блока продукций агента (путем создания ссылки на соответствующий информационный ресурс в фонде платформы IACPaaS);
 - шаблонов выходных сообщений – сообщений, создаваемых в процессе выполнения данного блока продукций агента и рассылаемых адресатам после завершения выполнения данного блока (если они есть).

2.3.2. Генерация заготовок исходного кода

На данном шаге с помощью системного сервиса *Генератор агентов* выполняется генерация заготовок исходного кода разрабатываемого агента и используемых в нем новых шаблонов входных и выходных сообщений по их декларативному описанию, а также получение байт-кода повторно используемых (пользовательских и встроенных в платформу IACPaaS) шаблонов сообщений (если таковые имеются) в виде jar-архивов сгруппированных в пакеты файлов, содержащих набор необходимых классов на языке программирования (в текущей реализации это Java), а также файлов, содержащих байт-код. Анализируя декларативную спецификацию агента, сервис формирует абстрактный класс на языке программирования с именем, совпадающим с внутренним именем агента, указанном в спецификации, который является подклассом системного класса облачной платформы *Agent* и содержит:

- конструктор класса, используемый платформой для создания и инициализации данного агента в виртуальной машине платформы IACPaaS;
- статический инициализирующий блок, содержащий описание блоков продукций агента (по количеству блоков продукций агента, указанных в его декларативном описании), используемый диспетчером сообщений платформы;
- набор внутренних статических классов (“создателей результатов”, количество которых совпадает с количеством блоков продукций агента, указанных в его декларативном описании), описывающих шаблоны тех (и только тех) сообщений, которые могут быть созданы и отправлены после окончания работы соответствующего блока продукций (т.е. выходных

для блока продукции сообщений) агента. Имя каждого такого класса есть внутреннее имя шаблона входного для блока продукции сообщения, к которому добавлен суффикс *MessageResultCreator*;

- набор методов-заглушек (количество которых совпадает с количеством блоков продукции агента, указанных в его декларативной спецификации) *void runProduction(...)*, соответствующих блокам продукции агента и имеющих два параметра (эти методы необходимо переопределить в классе-реализации):

- типом первого параметра является класс, соответствующий шаблону входного для блока продукции сообщения;
- типом второго параметра является класс “создатель результатов”, описывающий шаблоны выходных для блока продукции сообщений.

На основе данного класса-заготовки агента разработчику необходимо создать класс-реализацию агента (наследующий от класса-заготовки) с именем *<внутреннее имя агента>Impl*. В теле каждого переопределяемого метода *void runProduction(...)* необходимо реализовать логику (алгоритм) работы соответствующего блока продукции. В классе-реализации можно, также, описать и реализовать множество вспомогательных методов, внутренних классов и т.п. для использования последних внутри методов *void runProduction(...)*.

Помимо абстрактного класса-заготовки агента сервис генерации заготовок исходного кода, анализируя декларативные спецификации всех шаблонов сообщений (входных и выходных), на которые присутствуют ссылки в спецификации агента, на основе каждого из них формирует класс на языке программирования с именем, *<внутреннее имя шаблона сообщений>Message*. Каждый такой класс является подклассом системного класса облачной платформы IASPaas *Message* и содержит:

- конструктор класса, используемый платформой для создания и инициализации данного шаблона сообщений в виртуальной машине платформы IASPaas;
- внутренний статический класс, содержащий два метода создания сообщений по данному шаблону для агентов и агентов-экземпляров, которым эти сообщения должны быть отправлены:
 - параметром первого метода является строковое имя агента, которому должно быть послано созданное сообщение;
 - параметром первого метода является указатель на экземпляр агента, которому должно быть послано созданное сообщение.

Сгенерированный класс кода шаблона сообщений можно оставить без изменений, либо дополнить вспомогательными методами (обычно, это методы чтения/модификации информационного ресурса сообщения), внутренними классами и т.п.

2.3.3. Подготовка и загрузка байт-кода в фонд

На данном шаге выполняется компиляция кода классов агента и шаблонов сообщений, объединение полученного в результате компиляции байт-кода в jar-архив и загрузка (или обновление) *class*-файлов с байт-кодом агента и шаблонов сообщений в фонд платформы IASPaas – в информационные ресурсы, представляющие сформированные на первом шаге этапа реализации декларативные описания агента и шаблонов сообщений соответственно. Последнее выполняется с использованием системного сервиса *Загрузчик агентов* и необходимо для запуска и функционирования разработанных агентов на виртуальной машине платформы IASPaas.

Необходимо отметить, что *Генератор агентов* и *Загрузчик агентов* проверяют полноту декларативного описания агента и используемых в нем шаблонов сообщений. Если декларативное описание не полно, то разработчику отображается сообщение, локализирующее соответствующее место в описании. *Загрузчик агентов* проверяет, также, корректность байт-кода загружаемого агента и шаблонов сообщений, а также байт-кода используемых в них внутренних классов (если байт-код не корректен, то его загрузка или обновление не выполняется, а разработчику отображается сообщение о найденной ошибке).

2.4. Тестирование и отладка агента

На данном этапе для функционального тестирования разработанного агента используется системный сервис *Тестирующий агентов*, который обеспечивает многократный запуск и выполнение агентов на заданном множестве тестов, а также формирование и сохранение отчетов о результатах испытаний. Множество тестов для агента формируется с помощью *Редактора информационных ресурсов* по метайнформации *Структура тестов агента*. Тест для агента, в общем случае, есть четверка *<входное сообщение, множество ожидаемых выходных сообщений, множество изменяемых агентом информационных ресурсов, ожидаемые состояния изменяемых информационных ресурсов>*. Обязательным является только первый компонент. Тест считается успешно пройденным, если содержимое ожидаемых сообщений совпало с содержимым соответствующих сообщений, которые агент послал в процессе работы блока продукции, а содержимое изменяемых в процессе работы блока продукции информационных ресурсов совпало с содержимым соответствующих информационных ресурсов, перечисленных в последнем компоненте теста.

Для просмотра отчетов о результатах испытаний и журналов используется *Редактор информационных ресурсов* (в режиме просмотра). Журнал работы агента на конкретном тесте присоединяется к отчету и также доступен для просмотра. Журналирование используется для получения информации о том, какие события и в какой последовательности происходят во время

работы блоков продукции агента, а также для того, чтобы локализовать место возникновения ошибки и дать достаточно информации для ее воспроизведения. Сформированные наборы тестов могут использоваться для регрессионного тестирования в процессе сопровождения агента.

2.5. Ввод агента в эксплуатацию

Для включения агента в состав сервисов он с разрешения администратора предметной области (для которой разработан агент) переводится в режим публичного использования.

Заключение

Использование облачной платформы IACPaas для разработки компонентов ИС соответствует современным требованиям к проектированию и реализации ИС и обеспечивает их жизнеспособность за счет того, что все компоненты доступны разработчикам и экспертам для поддержания их в актуальном состоянии. Разработка каждого агента, входящего в состав решателей задач ИС, состоит в описании компонентов его декларативной спецификации и написанию кода множества блоков продукции. На основе декларативных спецификаций выполняется генерация заготовок исходного кода агентов (и шаблонов сообщений), в них же помещается и хранится процедурная часть (байт-код) соответствующих агентов и шаблонов сообщений. Сопровождение агента состоит в изменении исходного кода агента и, возможно, шаблонов сообщений, используемых им, с последующей загрузкой обновленного байт-кода в фонд платформы. Также перед этим может быть изменена декларативная спецификация агента и используемых им шаблонов сообщений и выполнена повторная генерация заготовок исходного кода.

С использованием предложенной технологии и системных сервисов платформы созданы следующие облачные прикладные сервисы: сервис разработки профессиональных виртуальных облачных сред, компьютерные обучающие тренажеры по классическим методам исследования в офтальмологии, виртуальная химическая лаборатория, виртуальная модель городского района. В процессе разработки находятся сервисы по автоматизированному конструированию доказательств математических теорем и медицинской диагностике.

Работа выполнена при финансовой поддержке РФФИ (проект 13-07-00024а, проект 14-07-00299а).

Библиографический список

- [Грибова и др., 2010] Грибова В.В., Клещев А.С., Шалфеева Е.А. Управление интеллектуальными системами // Известия РАН. Теория и системы управления. – 2010. – №6. – С.122-137.
- [Kleshchev, 2011] Kleshchev, A.S. How can ontologies contribute to software development? // Lecture Notes in Artificial

Intelligence. Springer-Verlag Berlin Heidelberg. – 2011. – Vol. 6581/2011. – Pp. 121-135.

[Грибова и др., 2013а] Грибова В.В., Клещев А.С. Технология разработки интеллектуальных сервисов, ориентированных на декларативные предметные базы знаний. Часть 1. Информационные ресурсы // Информационные технологии. 2013. №9. С. 7-11.

[Грибова и др., 2013б] Грибова В.В., Клещев А.С. Технология разработки интеллектуальных сервисов, ориентированных на декларативные предметные базы знаний. Часть 2. Решатель задач. Пользовательский интерфейс // Информационные технологии. 2013. №10. С. 10-14.

[Gennari et al., 2003] Gennari, J.H., Musen, M.A., Fergerson, R.W., etc. The evolution of Protege: An environment for knowledge-based systems development/International Journal of Human-Computer Studies. 2003. 58(1):89-123.

[Орлов и др., 2006а] Орлов В.А., Клещев А.С. Компьютерные банки знаний. Универсальный подход к решению проблемы редактирования информации. – Информационные технологии. – 2006. – №5 – С. 25-31.

[Protege, 2014] The Protégé Ontology Editor and Knowledge Acquisition System. [Electronic resource]. URL: <http://protege.stanford.edu/> (дата обращения 25.11.2014).

[Орлов и др., 2006б] Орлов В.А., Клещев А.С. Компьютерные банки знаний. Многоцелевой банк знаний // Информационные технологии. – 2006. – №2. – С. 2-8.

[Sanderson, 2009] Sanderson D. Programming Google App Engine: Build and Run Scalable Web Apps on Google's Infrastructure. – Sebastopol, California : O'Reilly Media, 2009. – 394 p.

[TopQuadrant, 2014] TopQuadrant. [Electronic resource]. URL: <http://www.topquadrant.com/> (дата обращения : 25.11.2014).

[Грибова и др., 2011] Грибова В.В., Клещев А.С., Крылов Д.А., Москаленко Ф.М., Смагин С.В., Тимченко В.А., Тютюнник М.Б., Шалфеева Е.А. Проект IACPaas. Комплекс для интеллектуальных систем на основе облачных вычислений // Искусственный интеллект и принятие решений. – 2011. – №1. – С.27-35.

[Gribova et al., 2013] Gribova V.V., Kleschev A.S., Krylov D.A., Moskalenko Ph.M., Timchenko V.A., Shalfeyeva E.A., Goldstein M.L. A software platform for the development of intelligent multi-agent internet-services // Proceedings of the Distributed Intelligent Systems and Technologies Workshop (DIST'2013). – 1-4 July 2013. – St. Petersburg, Russia. – Pp. 29-36.

[Lehmann, 1992] Lehmann F.W. Semantic Networks // Computers & Mathematics with Applications. – 1992. – Vol.23. – №2-5.

[Фатрелл и др., 2004] Фатрелл Р.Т., Шафер Д.Ф., Шафер Л.И. Управление программными проектами. Достижение оптимального качества при минимуме затрат: Пер. с англ. – М.: Вильямс, 2004. – 1136 с.

A TECHNOLOGY FOR DEVELOPMENT OF PROBLEM SOLVERS OF INTELLIGENT SYSTEMS WITH THE USE OF IACPAAS CLOUD PLATFORM TOOLS

Krylov D.A., Moskalenko Ph.M.,
Timchenko V.A.

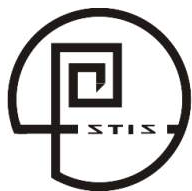
*Federal State Budget Institution of Science
Institute for Automation and Control Processes
Far Eastern Branch of the Russian Academy of
Sciences, Vladivostok, Russia*

dmalkr@gmail.com

philipmm@iacp.dvo.ru

vadim@dvo.ru

The paper presents a technology for development of multi-agent problem solvers of applied intelligent cloud services with the use of system tools of IACPaas platform. The technology is put to reduce the labour-intensiveness of development and primarily of support for intelligent cloud services.



OSTIS-2015

(Open Semantic Technologies for Intelligent Systems)

УДК 004.8+620

МЕТОДЫ И СРЕДСТВА СИТУАЦИОННОГО УПРАВЛЕНИЯ В ЭНЕРГЕТИКЕ НА ОСНОВЕ СЕМАНТИЧЕСКОГО МОДЕЛИРОВАНИЯ

Массель Л.В., Массель А.Г.

*Институт систем энергетики им. Л.А. Мелентьева СО РАН,
г. Иркутск, Россия*

massel@isem.sei.irk.ru

amassel@gmail.com

В статье описан предлагаемый авторами подход к ситуационному управлению в энергетике на основе семантического моделирования. Современная трактовка ситуационного управления рассматривается как отображение идей Д.А. Поспелова и его учеников на современные информационные технологии. В качестве основных методов ситуационного управления предлагаются методы ситуационного анализа и ситуационного моделирования, реализуемые с помощью технологий и инструментальных средств семантического моделирования в энергетике, к которому отнесены онтологическое, когнитивное, событийное и вероятностное (на основе Байесовских сетей доверия) моделирование. Для обоснования предлагаемого подхода используется авторская фрактальная стратифицированная модель (ФС-модель), с помощью которой рассмотрены структура и основные составляющие предлагаемого языка ситуационного управления.

Ключевые слова: ситуационное управление, семантическое моделирование, язык ситуационного управления, онтологическое, когнитивное, событийное и вероятностное моделирование.

Введение

Авторы развивают высказанную ими ранее идею применения концепции ситуационного управления для управления в энергетике в условиях экстремальных ситуаций. Предлагается трактовка идей Д.А. Поспелова и его учеников, опирающаяся на достижения современных информационных технологий, в частности, на результаты в области семантического моделирования в энергетике, к которому отнесены онтологическое, когнитивное, событийное и вероятностное (на основе Байесовских сетей доверия) БСД-моделирование. Концепцию ситуационного управления предлагается развивать для управления в энергетике в условиях экстремальных ситуаций, к которым отнесены как чрезвычайные, так и критические ситуации, причем основное внимание предлагается уделять последним, для предотвращения перерастания критических ситуаций в чрезвычайные. Обращается внимание на актуальность моделирования и анализа критических ситуаций, позволяющих сформировать перечень превентивных и оперативных мер, предотвращающих возможную чрезвычайную ситуацию, для чего предлагается интегрировать технологии семантического и математического

моделирования. Для методологического обоснования данного подхода предлагается использовать авторскую фрактальную стратифицированную модель (ФС-модель), которая была предложена ранее для структурирования пространства знаний.

1. Современное состояние и тенденции развития ситуационного управления.

Основоположниками направления «Ситуационное управление» по праву считаются Д.А. Поспелов и его ученики: Железов Ж.И., Клыков Ю.И. и др., результаты исследований которых были обобщены в [Поспелов. 1986]. Существенный вклад в развитие этого направления внес белорусский ученый В.В. Мартынов, предложивший универсальный семантический код (УСК). Спад интереса к ситуационному управлению в России, наступивший в 90-х гг., можно объяснить как изменениями экономико-политических условий в стране и наступившей «зимой искусственного интеллекта», так и трудностями, с которыми столкнулись разработчики, пытаясь построить модели сложных объектов управления с помощью предлагаемых подходов.

Развитие информационных технологий позволяет дать современную трактовку этого

направления, основанную на появлении как более совершенной техники, так и новых методов и подходов, в том числе семантического моделирования. Кроме того, претерпела изменения сама парадигма искусственного интеллекта. Если на ранних этапах его развития предполагалось, что системы ИИ могут заменять, в ряде случаев, естественный интеллект, сейчас получают распространение интеллектуальные вычисления (Intelligent Computing), под которыми понимаются методы и системы искусственного интеллекта, направленные на усиление и поддержку естественного интеллекта (поддержку принятия решений экспертами).

В настоящее время получают распространение более прагматические трактовки ситуационного управления. В словаре терминов МЧС (2010) ситуационное управление определяется как деятельность органов управления, при которой решения и управляющие воздействия субъекта управления основываются на анализе вариантов принятия решения с учетом: текущего состояния объекта управления, располагаемых вариантов действий и прогноза последствий принимаемых управленческих воздействий. В ряде случаев ситуационное управление определяют как оперативное управление, осуществляемое в дополнение к стратегическому, перспективному и заключающееся в принятии управленческих решений по мере возникновения проблем в соответствии со складывающейся экономической ситуацией (авторы не разделяют эту точку зрения, о чем будет сказано ниже).

В [Васильев, 2012] используется идея ситуационного управления, суть которой заключается в выборе управленческих решений с учетом сложившейся ситуации из некоторого набора допустимых (типовых, стандартных) управляющих воздействий. Под текущей ситуацией S при этом понимается совокупность текущего состояния объекта (вектор состояния X) и его внешней среды (вектор возмущений F). Тогда $S = \langle X, F \rangle$. Вводится также понятие полной ситуации $S = \langle C, G \rangle$, где C – текущая ситуация, G – цель управления. В свою очередь, цель управления G может быть представлена в виде целевой ситуации G_g , к которой должна быть приведена имеющаяся текущая ситуация. Тогда $S = \langle C, G_g \rangle$. Полагая, что текущая ситуация C принадлежит некоторому классу Q' , а целевая (заданная) ситуация G_g – классу Q'' , ищется такое управление (вектор управляющих воздействий U), которое принадлежит множеству допустимых управлений Ω_u и обеспечивает требуемое преобразование одного класса ситуаций в другой:

$$C \in Q' \xrightarrow{U \in \Omega_u} G_g \in Q''.$$

Таким образом, ситуационное управление выступает как отображение:

$$(Q', Q'') \rightarrow U \in \Omega_u.$$

сопоставляющее паре «текущая ситуация - целевая ситуация» требуемый результат – управление U .

Другими словами, при ситуационном управлении проблема выбора управляющих воздействий сводится к адекватной оценке состояния объекта и среды (что усложняется при наличии факторов неопределенности), отнесению соответствующей текущей ситуации к одному из типовых классов и выбору такого управления (из определенного набора альтернатив), которое приводит к достижению поставленной цели управления (целевой ситуации) [Васильев, 2012]. Исходя из этого, авторы предлагают в качестве основных методов ситуационного управления методы ситуационного анализа и ситуационного моделирования, основанные, в свою очередь, на семантическом моделировании.

2. Ситуационный анализ и семантическое моделирование в энергетике.

Рассмотренный выше подход был интерпретирован авторами в [Массель и др., 2014] на примере исследований проблем энергетической безопасности (ЭБ), или, иначе, оценки состояния ТЭК в условиях возможных сценариев угроз ЭБ с учетом мероприятий, направленных на повышение уровня ЭБ. Авторами предложено использовать в этих исследованиях ситуационный анализ и ситуационное моделирование, как основные методы ситуационного управления. При этом предлагается в первую очередь использовать идеи ситуационного управления для разработки стратегий развития энергетики России с учетом требований энергетической безопасности. Учитывая, что наличие факторов неопределенности усложняет адекватную оценку состояния объекта и среды, предложено использовать семантические технологии ситуационного анализа, к которым отнесены онтологическое, когнитивное, событийное и вероятностное (на основе Байесовских сетей доверия) моделирование. В коллективе, возглавляемом авторами, разработаны средства поддержки этих видов моделирования, интегрированные в рамках интеллектуальной ИТ-среды. Подробнее технологии семантического моделирования и инструментальные средства их поддержки (интеллектуальная ИТ-среда) рассматривались авторами в [Массель, 2010], [Массель А., 2010], [Массель и др., 2012], а также на конференциях OSTIS- 2013 [Массель и др., 2013a] и OSTIS -2014 [Массель и др., 2014].

Авторы предлагают также использовать концепцию ситуационного управления в энергетике для управления в условиях экстремальных ситуаций (ЭкС). Под экстремальными ситуациями в энергетике понимаются как чрезвычайные, так и критические ситуации, определение которых

базируется на оценке состояний систем или объектов по шкале: «норма», «предкризис» - критическая ситуация, «кризис» - чрезвычайная ситуация. Исходя из этого, под *критическими ситуациями* понимаются ситуации, когда возникают угрозы бесперебойному функционированию технических объектов и объектов обеспечения жизнедеятельности и /или угрозы жизни или здоровью, как отдельных людей, так и социальных (профессиональных) групп [Массель и др., 2013б].

Эти угрозы могут быть устранены принятием соответствующих превентивных и оперативных мер, которые не позволят критической ситуации перерасти в чрезвычайную. В приведенной выше трактовке ситуационного управления эти мероприятия соответствуют управляющим воздействиям, обеспечивающим переход из текущей (критической) ситуации в целевую (безопасную). В настоящее время внимание уделяется преимущественно поддержке принятия решений и управлению в чрезвычайных ситуациях, когда требуется реализация оперативных и ликвидационных мероприятий, на это направлена, в том числе, деятельность подразделений МЧС. Авторы обращают внимание на актуальность моделирования и анализа критических ситуаций, позволяющих сформировать перечень превентивных и оперативных мер, предотвращающих возможную чрезвычайную ситуацию, для чего предлагается интегрировать технологии семантического и математического моделирования. Для методологического обоснования данного подхода предлагается использовать авторскую фрактальную стратифицированную модель (ФС-модель), которая была предложена ранее для структурирования пространства знаний.

3. Фрактальный подход к построению инструментальных средств ситуационного управления в энергетике.

ФС-модель была предложена как концептуальная модель структурирования знаний, основанная на представлении разных форм (видов) знаний как объектов расслоенного (стратифицированного) пространства [Массель, 1994]. При построении ФС-модели вводится понятие *информационных миров* - подпространств, объединяющих однотипные информационные объекты, т.е. объекты, определяемые совокупностью свойств, являющихся существенными в этом рассмотрении (координат данного подпространства).

ФС-модель определяется как совокупность непересекающихся слоев (информационных миров), и их отображений в информационном пространстве. Каждому уровню соответствует свой слой (страта) этого пространства, и, следовательно, свой информационный мир; последовательность отображений отражает процесс познания.

Графически ФС-модель удобно изображать в виде совокупности вложенных сферических оболочек. Информационный объект, обозначаемый условно точкой на одной из сфер, в свою очередь, может быть расслоен при необходимости более детального его рассмотрения (в одном случае может быть удобно рассматривать сложный объект как точку, а в другом - перейти к более детальному рассмотрению выбранного объекта при сохранении «точечного» представления других).

Поскольку знания любого исследователя не всеобъемлющи, каждый практически работает со своим «фракталом» знаний - «вырезкой» из информационного пространства, которую можно представить в виде конуса или пирамиды, что соответствует, например, выделению дисциплин при изучении реального мира (рис.1). Тогда наша сфера оказывается состоящей из множества «конусов», что согласуется с «пирамидами знаний» в когнитологии, предложенными Т.А. Гавриловой [Гаврилова и др., 1992]. Математическое описание ФС-модели приведено в [Массель, 1995].

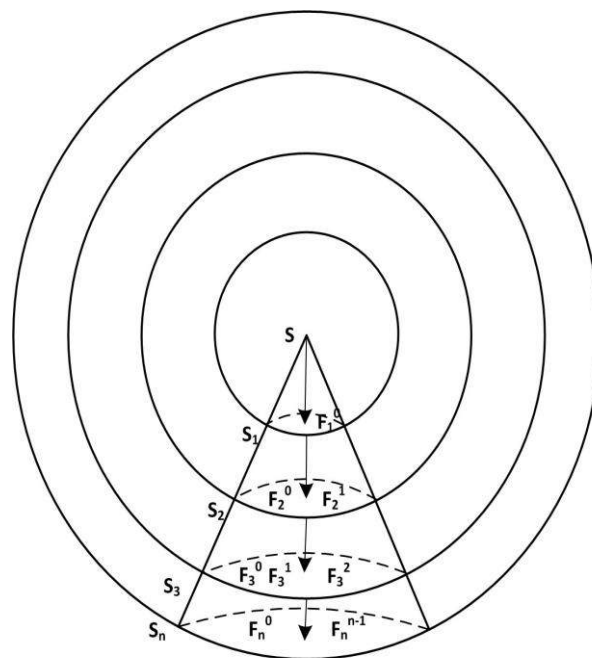


Рисунок 1 – Фрактальная стратифицированная модель.
 S_i - слои (страты) ФС-модели,
 F_j^i - отображения из i -го слоя в j -ый.

Проиллюстрируем этот подход на примере энергетике. Упомянутые выше исследования имеют многовариантный характер (рис. 2).

На основе базового варианта могут быть сгенерированы множество вариантов как по территориальному и/или временному признаку (периоды прогнозирования), так и в зависимости от преобладающего вида топлива в перспективе. В свою очередь, на каждый из вариантов могут быть наложены сценарии критических и/или чрезвычайных ситуаций S_i и перечень мероприятий (управляющих воздействий) M_j , предотвращающих ЭЧС или ликвидирующих ее последствия.

ФС-модель может быть использована двояко. В первом случае с ее помощью можно построить «дерево вариантов», т.е. использовать ее для планирования вычислительного эксперимента. Традиционно для выполнения этих исследований используются математические модели большой размерности и решается задача линейного программирования. Принятый в последнее время комбинаторный подход позволял рассчитывать до пяти миллионов вариантов, которые затем анализировались экспертами. Авторами была предложена двухуровневая технология исследований, поддерживаемая интеллектуальной ИТ-средой, которая предусматривает на первом, верхнем уровне, проведение качественного «экспресс-анализа» с помощью семантического моделирования, а отобранные несколько вариантов рассчитываются на втором уровне с помощью традиционных программных комплексов (например, с помощью многоагентного программного комплекса ИНТЭК-М, разработанном в коллективе, возглавляемом авторами).

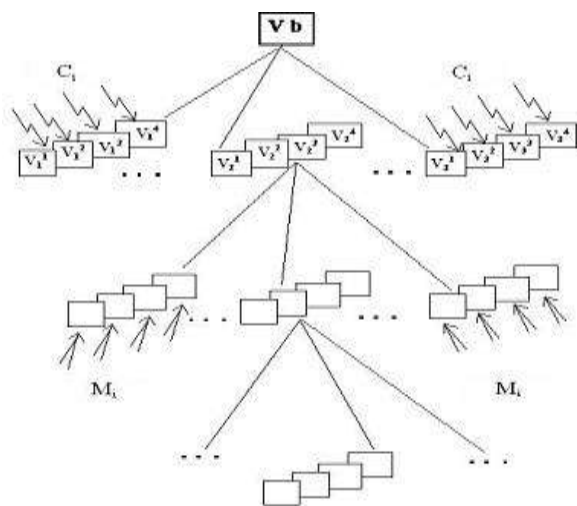


Рисунок 2 – Многовариантный характер исследований направлений развития ТЭК с учетом требований энергетической безопасности

Второе применение ФС-модели может быть использовано для методологического обоснования разработки средств ситуационного управления, что будет рассмотрено ниже.

4. Языки и инструментальные средства ситуационного управления.

Для описания объектов управления Д.А. Поспеловым и его учениками предлагалось использовать семиотические, или знаковые модели, включающие как необходимые формализмы, так и правила их изменения, а также дискретные ситуационные сети (ДСС). В качестве языков ситуационного управления рассматривались: язык исчисления предикатов первого порядка, язык RX-кодов (прообраз современных онтологий), универсальный семантический код; предлагалось создать универсальный ЯСУ на основе ДСС

(подробнее это рассматривалось авторами в [Массель и др., 2014]). К сожалению, учитывая как усовершенствование технической платформы, так и появление новых языков программирования, в настоящее время невозможно использовать программные средства, разработанные в 80-х годах, которые, если и сохранились, перешли в категорию унаследованного программного обеспечения.

Современные языки и инструментальные средства ситуационного управления в энергетике авторы предлагают разрабатывать на основе принципа сетцентричности, в основе которого - наличие интегрированной информационной среды, позволяющей обеспечивать информационный обмен между участниками информационного процесса с использованием ситуационной осведомленности, обеспечиваемой средствами ГИС-сервисов (3D-визуализации) (рис. 3) [Массель и др., 2013с].

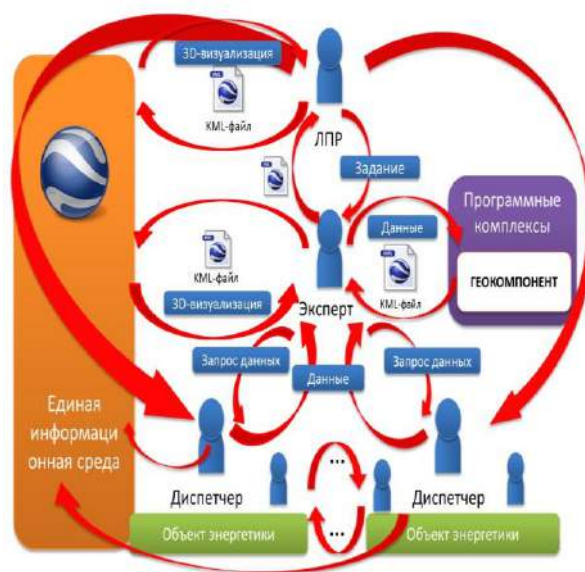


Рисунок 3 – Обобщенная схема поддержки принятия решений в энергетике в рамках сетцентрического подхода

С помощью ФС-модели можно представить язык ситуационного управления как пятерку $L = (C, G, E, U, F)$, где C и G - описание исходной и целевой ситуаций, E - сценарий ЭкС, U - описание управляющих воздействий (событий), F - описание отображений.

В соответствующей ФС-модели могут быть выделены слои как исходных и целевых ситуаций C и G , так и E - сценариев ЭкС. Управляющие воздействия U могут быть описаны как события, образующие отдельный слой. Отображения F есть не что иное, как наложение сценариев E на исходные ситуации C , порождающие «переходные ситуации» P , а также воздействие управляющих воздействий U на «переходные ситуации» P , с тем чтобы перевести их в целевые (устойчивые) ситуации G .

Соответственно язык ситуационного управления должен включать две составляющих (D, M): средства описания знаний D (для описания

ситуаций, сценариев, управляющих воздействий), и средства манипулирования знаниями M (для отображений информационных объектов из любого слоя в каждый). Последние включают как средства поддержки отображений F , так и, в идеале, средства логического вывода на цепочках (C, E, P, U, G) (исходная ситуация, сценарий ЭкС, переходная ситуация, управляющее воздействие, целевая ситуация).

На данном этапе в качестве средств описания знаний (D) предлагается использовать на первом шаге онтологии ситуаций, сценариев и событий и метаонтологии, интегрирующие детальные онтологии. На втором и последующих шагах онтологии отображаются в когнитивные модели, отражающие причинно-следственные связи концептов (понятий онтологий), событийные модели, описывающие развитие ситуаций, и БСД-модели, моделирующие риски возникновения ЭкС.

В качестве средств манипулирования знаниями (M) разрабатываются средства поддержки отображений F : переход от онтологий к когнитивным, событийным и БСД-моделям; переход от когнитивных моделей к событийным; переход от когнитивных и событийных моделей к БСД-моделям. В предыдущих работах авторов они назывались средствами интеграции когнитивных, событийных и БСД-моделей.

Для реализации средств логического вывода на первом шаге может быть использовано исчисление предикатов (например, конструкции языка Пролог). В последующем эти средства могут быть объединены в ситуационное исчисление [Рассел и др.]. В настоящее время средства ситуационного управления, включающие библиотеки онтологического (OntoMap), когнитивного (CogMap), событийного (EventMap) и вероятностного (BayNet) моделирования реализуются в составе Ситуационного полигона, разрабатываемого под руководством и при участии авторов [Массель А. и др., 2014].

Заключение

В статье рассмотрен подход к разработке методов и средств ситуационного управления в энергетике, основанный на семантическом моделировании. Реализация данного подхода рассматривается на основе принципа сетцентричности, подразумевающего наличие интегрированной информационной среды, позволяющей обеспечивать информационный обмен между участниками информационного процесса с использованием ситуационной осведомленности, обеспечиваемой средствами ГИС-сервисов (3D-визуализации). В качестве основных методов ситуационного управления рассматриваются методы ситуационного анализа и ситуационного моделирования, которые реализуются с помощью технологии и инструментальных средств онтологического, когнитивного, событийного и

вероятностного (на основе Байесовских сетей доверия) моделирования. Для методологического обоснования предложенного подхода используется авторская ФС-модель. Выделены две основные составляющие языка ситуационного управления: средства описания знаний (для описания слоев ФС-модели) и манипулирования знаниями (для поддержки отображений слоев). Показано, что на первом этапе для реализации этих составляющих могут быть использованы разработанные в авторском коллективе средства семантического моделирования: онтологического, когнитивного, событийного и вероятностного моделирования, которые интегрируются в рамках разрабатываемого под руководством и при участии авторов Ситуационного полигона.

Результаты, представленные в статье, получены при частичной финансовой поддержке грантов РФФИ № 13-07-00140, № 15-07-01284, №15-37-20710, №15-57-04074, гранта Программы Президиума РАН №229.

Библиографический список

- [Поспелов, 1986] Ситуационное управление. Теория и практика / Поспелов Д.А. // М.: Наука, 1986, 284 с.
- [Васильев, 2012] Интеллектуальные системы защиты информации / Васильев В.И. // М.: Машиностроение, 2012, 171 с.
- [Массель и др., 2014] Ситуационное управление и семантическое моделирование в энергетике / Массель Л.В., Массель А.Г. // Труды IV Международной конференции OSTIS, Беларусь, Минск: БГУИР, 2014, С. 111-116.
- [Массель, 2010] Применение онтологического, когнитивного и событийного моделирования для анализа развития и последствий чрезвычайных ситуаций в энергетике / Массель Л.В. // Проблемы безопасности и чрезвычайных ситуаций. - №2. - 2010. - С. 34-43.
- [Массель А., 2010] Методологический подход к организации интеллектуальной поддержки исследований проблемы энергетической безопасности / Массель А.Г. // «Информационные технологии». - №9. - 2010. - С. 32-36.
- [Массель и др., 2012] Интеллектуальные вычисления в исследованиях направлений развития энергетики / Массель Л.В., Массель А.Г. // Известия Томского политехнического университета. - 2012. - Т. 321. - № 5. Управление, вычислительная техника и информатика. - С. 135-141.
- [Массель и др., 2013а] Семантические технологии на основе интеграции онтологического, когнитивного и событийного моделирования / Массель Л.В., Массель А.Г. // Материалы III международной научно-технической конференции «OSTIS-2013» – Беларусь, Минск: БГУИР, 2013. – С. 247-250.
- [Массель и др., 2013б] Технологии и инструментальные средства интеллектуальной поддержки принятия решений в экстремальных ситуациях в энергетике / Массель Л.В., Массель А.Г. // Вычислительные технологии. - 2013. - Т.18. - Специальный выпуск. - С. 37-44
- [Массель, 1994] Фрактальная модель структурирования знаний / Массель Л.В. // Сб. науч. трудов Национальной конференции с международным участием “Искусственный интеллект-94”. - Рыбинск, 1994. - т.1. - С. 46-49.
- [Гаврилова и др., 1992] Извлечение и структурирование знаний для экспертных систем / Гаврилова Т.А., Червинская К.Р. // М.: Радио и связь, 1992. - 200 с.
- [Массель, 1995] Фрактальный подход к построению информационных технологий / Массель Л.В. // В кн: Кривоуцкий Л.Д., Массель Л.В. // Информационная технология исследований развития энергетики // Новосибирск: “Наука”, Изд. фирма РАН, 1995. - С. 40-67.
- [Массель и др., 2013с] Моделирование этапов принятия решений на основе сетцентрического подхода / Массель Л.В., Иванов Р.А., Массель А.Г. // Вестник ИРГТУ. - №10 (81). - 2013. - С. 16-22.

[Рассел и др.] Действия, ситуации и события.(онтология ситуационного исчисления)/ В кн.: Рассел С. , Норвиг П. Искусственный интеллект: современный подход, 2-е изд.: Пер. с англ.- М.: Изд. Дом «Вильямс», 2006.- С. 451-466.

[Массель А. и др., 2014] Ситуационный полигон как инструмент ситуационного управления в энергетике / Массель А.Г. Иванов Р.А. // Труды IV Международной конференции OSTIS, Беларусь, Минск: БГУИР. - 2014. – С. 277-280.

METHODS AND TOOLS OF CONTINGENCY MANAGEMENT IN ENERGY SECTOR BASED ON SEMANTIC MODELING

Massel L.V., Massel A.G.

*Melentiev Energy Systems Institute of Siberian Branch of the Russian Academy of Sciences
Irkutsk, Russia*

massel@isem.sei.irk.ru

amassel@gmail.com

This article describes the proposed authors approach to contingency management in the energy sector based on semantic modeling. The modern interpretation of situational management is seen as a reflection of D.A. Pospelov and his colleague's ideas on the modern information technologies. The main methods of contingency management include situational analysis and situational modeling implemented using technologies and tools of semantic modeling in the energy sector, including the ontological, cognitive, event and probabilistic (based on Bayesian belief networks) modeling. The authors use for justification of the proposed approach own fractal stratified model (FS-model), by means of which the structure and the main components of the proposed contingency management language are described.

Key words: contingency management, semantic modeling, contingency management language, ontological, cognitive, event and probabilistic modeling.

Introduction

The authors propose to develop the concept of situational management for contingency management in the energy sector in extreme situations to which assigned as emergency so critical situations, predominantly to pay attention on the last to prevent escalation of critical situations in the emergency. The attention is paid to the relevance of the modeling and analysis of critical situations, what allows you to create a list of preventive and operational measures to prevent a possible emergency situation by means of technologies integrating of semantic and mathematical modeling. It is proposed to use the network-centric principle and develop special language of contingency management, including components of the knowledge description and knowledge manipulation. For justification of the proposed approach the authors use own fractal stratified model (FS-model), by means of which the structure and the main components of the proposed contingency management language are described.

Main Part

The modern interpretation of situational management in the energy sector is considered as a reflection of D.A. Pospelov and his colleague's ideas on the modern information technologies.

The authors propose to use in these studies, situational analysis and situational modeling as the main methods of contingency management. It is proposed to primarily use the idea of contingency management to elaborate strategies for the development of Russian energy sector to meet the requirements of energy security.

Modern languages and tools of contingency management in the energy sector the authors propose to develop on the basis of network-centric principle, based on the presence of integrated information environment, allowing data exchange between the participants of the information process using situational awareness provided by means of GIS-services (3D-visualization).

The fractal stratified model is described and shown its interpretation by the example of energy security problems research. The contingency management language is described as $L = (C, G, E, U, F)$, where C and G - description of the source and target situations, E - emergency scenario, U - description of management actions (events), F – descriptions of the mappings.

There are selected components of knowledge description and knowledge manipulation. The first includes the ontology's of situations, scenarios and events, as well as cognitive, event and probabilistic models.

The second includes a means to support mappings from any layer in each and means of inference on the chains (C, E, P, U, G) (the initial situation, the emergency scenario, a transitional situation, the control action, the target situation).

To implement the means of inference in the first step can be used predicate calculus (eg, constructions of Prolog). Subsequently, these tools can be combined into a situational calculus.

Currently, situational management tools, including libraries ontological (OntoMap), cognitive (CogMap), event (EventMap) and probabilistic (BayNet) simulations are implemented as part of the Situation polygon, which developed under the supervision and with the participation of authors.

Conclusion

The article describes the approach to the development of methods and means of contingency management in the energy sector based on semantic modeling.

It is proposed to use in the first stage a means of semantic modeling, such as ontological, cognitive, event and probabilistic modeling, which are integrated into the framework of the Situation polygon which is developed under the supervision and with the participation of the authors.

The results presented in this paper were obtained with the partial financial support by RFBR grants № 13-07-00140, № 15-07-01284, №15-37-20710, №15-57-04074, by grant of RAS Presidium Program № 229.



OSTIS-2015

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822:514

ОТ ИНТЕЛЛЕКТУАЛЬНЫХ ПОРТАЛОВ К ПОРТАЛАМ ЗНАНИЙ: О РОЛИ ИНТЕЛЛЕКТУАЛЬНЫХ СЕРВИСОВ

Страхович Э.В., Власов С.А., Гаврилова Т.А.

*Институт «Высшая школа менеджмента» Санкт-Петербургского государственного
университета,
г. Санкт-Петербург, Российская Федерация*

strakhovich@gsom.pu.ru

vlasov@gsom.pu.ru

gavrilova@gsom.pu.ru

В статье рассматриваются интеллектуальные сервисы, выполняющие функцию поддержки корпоративных порталов знаний. Ставятся задачи определения и построения современной классификации интеллектуальных сервисов для инженерии знаний, а также обсуждаются основные тенденции в их исследовании, разработке и применении.

Ключевые слова: порталы знаний; интеллектуальные сервисы; онтологии.

Введение

Возможности корпоративных информационных порталов во многом определяются функциональностью реализованных в их составе сервисов. При развитии системы управления знаниями в составе портала предполагает «интеллектуализацию» сервисов. При этом единого общепринятого определения интеллектуального сервиса пока не существует. Задача данного исследования состоит в определении и построении современной классификации интеллектуальных сервисов для инженерии знаний, а также в выявлении основных тенденций в их исследовании и разработке.

1. Информационные порталы и порталы знаний

Интеллектуальные сервисы как предмет исследования находятся на пересечении научных и технологических парадигм в области информационных систем, систем управления знаниями (СУЗ), корпоративных информационных порталов, сервисных систем и «умных» сервисов.

К СУЗ относятся технологии, включающие комплекс формализованных методов для поиска, извлечения знаний из носителей знания, структурирования и систематизацию знаний для удобного хранения и поиска, анализа, обновления и распространения знаний. Изначально в области

искусственного интеллекта использовалось понятие «системы, основанной на знаниях» (knowledge-based system, KBS), тогда как понятие «системы управления знаниями» (knowledge management system) появилось значительно позже в литературе по менеджменту и оно шире, чем KBS.

Термин «корпоративный информационный портал» (КИП) был введен впервые в 1998 г. и обозначает приложения, позволяющие компаниям раскрыть информацию, хранящуюся как внутри, так и вне организации, и обеспечить пользователей единой точкой доступа и персонализированной информацией необходимой для принятия обоснованных бизнес-решений. Следует отметить, что некоторые авторы разделяют корпоративные информационные порталы на два типа: корпоративные информационные порталы и корпоративные порталы управления знаниями [Benbya et al, 2004]. К порталам первого типа относятся порталы с сервисами поиска, обмена и совместного использования информации, порталы второго типа включают сервисы, разработанные с использованием методов искусственного интеллекта, и выполняют задачи управления знаниями. Порталами знаний мы будем называть системы управления знаниями, доступ к которым реализован посредством корпоративных порталов.

Корпоративным информационным порталам отводится значительная роль как на уровне государственных и муниципальных структур

[Стратегия..., 2013], так и на уровне отдельных компаний, профессиональных и любительских сообществ [Шатовская и др., 2013]. Особое внимание уделяется развитию корпоративных информационных порталов в области образования [Балова и др., 2012] и здравоохранения. Так, через разработку веб-порталов реализуются проекты создания открытых образовательных ресурсов (ООР). Подобные проекты разрабатываются как отдельными странами (например, Российский общеобразовательный портал), так и группой стран (например, проект Open Education Europa). При создании ООР ставится целью решение задач повышения качества образования, доступа к информации, обмена знаниями, привлечения широкого круга пользователей — преподавателей, учащихся, исследователей. Разработки по созданию ООР предполагают организацию сетевого взаимодействия на основе порталов учебных заведений, что приведет к расширению требований к функциональности коммуникационных сервисов КИПов.

Структуры КИП разрабатываются на основе сервис-ориентированной архитектуры, предполагающей выделение сервиса в независимый модуль. Среди сервисов КИП в первую очередь следует упомянуть специализированные или те, которые отражают бизнес-процессы предприятия или учреждения и во многом определяют специфику конкретного портала. Так, на портале высшего учебного заведения может быть представлен сервис регистрации на учебный курс, на портале медицинских услуг — сервис записи на прием к специалисту, а на городском портале — сервис записи ребенка в детский сад и т.д.

2. Роль сервисов информационных порталов

Как уже отмечалось, КИП создаются прежде всего с целью поиска и обмена информацией, общения пользователей, совместного использования информации. Для реализации перечисленных функций представлены базовые сервисы, составляющие сущность портала [Львович и др., 2010]:

- коммуникационные сервисы, обеспечивающие функции обмена информацией и совместной работы пользователей и технического персонала портала, реализующие современные информационные технологии проведения опросов и голосований;
- информационные сервисы, предназначенные для оповещения пользователей об изменениях и событиях в предметной области или в области интересов пользователя;
- навигационные сервисы, предназначенные для поиска или повышения эффективности поиска информации;
- тематические сервисы поиска, анализа и визуализации пространственных данных (ГИС-порталы);

- персонафикационные (идентификационные) сервисы, обеспечивающие функции, необходимые для идентификации, авторизации и аутентификации пользователей портала, а также адаптации визуального оформления и представления информации в соответствии с потребностями пользователя портала; сюда же можно отнести и наличие «личного кабинета», запоминаящего профиль пользователя и настройки системы, сделанные пользователем;

- образовательные сервисы часто включаются в корпоративные порталы, так как обучение сотрудников входит в задачи информационного обмена.

Для поддержания работы портала реализованы так называемые технические сервисы, к которым можно отнести:

- статистические сервисы, реализующие функции сбора и анализа статистической информации, накапливаемой в процессе эксплуатации портала;
- сервис аудита, обеспечивающий протоколирование (логирование) всех действий, осуществляющихся в рамках системы безопасности;
- сервис мониторинга.

Вопросы реализации эффективного поиска, где под эффективностью понимается как собственно сокращение времени поиска, так и оценка результата с точки зрения удовлетворения информационных и консультационных запросов пользователя [Шатовская и др., 2013], занимают сейчас одно из центральных мест в задачах развития КИП.

Особое внимание при разработке порталов уделяется интерактивным сервисам.

3. Интеллектуальные сервисы

Понятие «сервисные системы» (service systems) возникло недавно и сейчас достаточно размыто. Под сервисными системами подразумевается такие социальные, технические, информационные и организационные системы, которые создают дополнительные ценности для их элементов, как в экономическом, так и в социальном плане, а также с точки зрения защиты окружающей среды. В основе сервисных систем лежат синергетические эффекты и процесс со-создания ценности за счет интенсивных взаимодействий между элементами системы [Maglio, 2014; Barile et al, 2010].

Если рассматривать только технологическую компоненту сервисных систем, которая представляется совокупностью взаимодействующих информационных систем, то можно говорить о наличии или отсутствии свойства интеллектуальности (intelligent service). Свойство интеллектуальности системы может проявляться только при наличии базы знаний и при взаимодействии с некой внешней средой, в которой система применяет различные поисковые стратегии

достижения определенной цели. Таким образом, необходимым условием интеллектуальности является непрерывное взаимодействие с внешней средой, которое обеспечивается, например, посредством сенсоров различного рода, динамической масштабируемостью и др. свойствами.

Категорию «умных» или «разумных» сервисов (smart service systems) можно рассматривать как подкласс интеллектуальных сервисов. К «умным» сервисным системам относятся системы, обладающие рядом характеристик, среди которых часто отмечаются само-конфигурация (или по крайней мере легко запускаемая реконфигурация) [Barile et al, 2010], способность к прогнозированию и принятию предупредительных мер (в отличие от простого реактивного поведения), а также взаимосвязанность и непрерывное взаимодействие с внутренними и внешними элементами системы. Тем не менее, общепринятые определения как интеллектуальных, так и умных сервисов еще не установлены. В самом общем понимании к умным сервисам иногда относят не только информационные системы с определенными возможностями, но и другие типы сервисов [Allmendinger et al, 2005].

Можно предложить новую обобщенную таксономию [Гаврилова др., 2014] на основе ряда критериев классификации для интеллектуальных сервисов, обслуживающих порталы знаний (рис.1):

- типы элементов, формирующих тот или иной сервис,
- структура взаимосвязей сервисной системы между различными типами элементов [Maglio, 2014];
- уровень «интеллектуальности» или «разумности» сервиса [Barile et al, 2010],
- динамические аспекты работы сервиса,
- типы доступной сервису информации и др.

Интеллектуальные сервисы часто создаются на основе онтологий предметной области. Применение онтологических моделей представления знаний повышает релевантность информационного поиска и способствует развитию сервисов портала. Онтологический подход применяется также в описании и постановке задач, реализуемых при создании портала, при описании знаний и архитектуры портала [Балова и др., 2012; Gavrilova et al, 2009], при описании предметной области и наполнении контента портала [Загорулько и др., 2007].

В качестве тенденций следует отметить подход к проектированию и разработке порталов, заключающийся в ориентации на удовлетворение потребностей пользователя, предъявляемых им при обращении к конкретному portalу, так называемый user-centered подход [Jámbor et al, 2012].



Рисунок 1 – Классификация интеллектуальных сервисов

Реализация такого подхода использует методы искусственного интеллекта при описании пользователя и предметной области. С user-centred подходом связано решение задачи обеспечения комфортности пользователя, что в свою очередь повышает уровень пользования порталом. В частности, такие задачи ставятся при развитии портала государственных услуг [Стратегия..., 2013].

Заключение

Порталы знаний на основе онтологий успешно развиваются и используются уже более 10 лет, начиная от пионерских работ Staab and Maedche (2001), фундаментального обзора Loebke & Crowston (2010), и последних работ (Milosz, Borys & Grzegorski, 2012). В России хорошо известны работы московской (Лукашевич Н.В., Хорошевский В.Ф., Осипов Г.С.), петербургской (Смирнов А.В., Рубашкин В.Ш.), дальневосточной (Клещев А.С., Грибова В.В., Артемьева И.Л.) и новосибирской школ (Загоруйко Н. Г., Загоруйко Ю.А.).

Разрабатываемый группой сотрудников СПбГУ проект ИнС-ПОРТ (Интеллектуальные Сервисы поддержки ПОРТалов знаний на основе онтологий) ориентирован на решение научной проблемы по созданию системы интеллектуальных сервисов поддержки жизненного цикла портала знаний - от проектирования до эксплуатации. Целью проекта является создание методологии и технологии, позволяющих создавать и тиражировать порталы знаний предприятий, научных проектов, учебно-методических центров и др. Методология ПРОТЕСИС (ПРОект-ТЕхнология-СИСтема), разрабатываемая в рамках проекта, позволит создавать, объединять, редактировать и ликвидировать онтологии предметных областей на основе структурно-визуального подхода. Также она позволит объединить современные представления об интеллектуальных сервисах с теорией создания и поддержки систем управления знаниями [Гаврилова и др., 2009].

На текущий момент проведен анализ существующих подходов к формированию корпоративных порталов знаний, разработана новая таксономия интеллектуальных сервисов, проведен анализ платформ SaaS (Software as a Service), выбраны инструменты для работы с корпоративными (совместными) онтологиями. Следующей задачей стоит разработка методики создания прототипов порталов знаний на основе онтологий и их развития до полнофункциональных систем управления знаниями. Планируется, что эта методика будет основываться на идеях платформ SaaS.

Разработка проекта ведется при финансовой поддержке РФФИ (проект 14-07-00294 А) и Санкт-Петербургского государственного университета.

Библиографический список

[Балова и др., 2012] Балова, Т.Г. Проектирование сервисов портала ВУЗа для интеграции информационных ресурсов

образовательных программ / Т. Г. Балова, В. О. Мокеров, Н. М. Темирбеков // Проблемы информатики 2012, № 3, с.86-90.

[Гаврилова и др., 2014] Гаврилова, Т.А. Интеллектуальные сервисы поддержки порталов знаний / Гаврилова Т.А. Власов С.А. // Труды 2-ого Международного Поспеловского симпозиума «Гибридные и синергетические интеллектуальные системы ГИСИС-2014», Светлогорск, 2014. – с. 94-100.

[Гаврилова и др., 2009] Гаврилова, Т.А. Информационные технологии управления знаниями / Гаврилова Т.А., Кудрявцев Д.И. // В книге «Инновационное развитие: экономика, интеллектуальные ресурсы, управление знаниями» (глава 26) под ред. Мильнера Б.З., М.: ИНФРА-М, 2009. – с.500-516.

[Загоруйко и др., 2007] Загоруйко, Ю.А. Технология построения онтологий для порталов научных знаний / Загоруйко Ю.А., Боровикова О.И. // Вестник НГУ. Серия: Информационные технологии. 2007. Том 5, выпуск 2, с.42-52.

[Львович и др., 2010] Львович, Я.Е. Проблемы организации и управления в единой информационной образовательной среде/ Львович Я.Е., Львович И.Я., Волкова Н.В. // Вестник Воронежского государственного технического университета. 2010. Т. 6. № 9. С. 4-8.

[Стратегия..., 2013] Стратегия развития отрасли информационных технологий в Российской Федерации на 2014 - 2020 годы и на перспективу до 2025 года. Утверждена распоряжением Правительства Российской Федерации от 1 ноября 2013 г. № 2036-р

[Шатовская и др., 2013] Шатовская, Т.Б. Исследование адаптации интерактивных сервисов портала автовладельцев на основе адаптивных алгоритмов / Шатовская Т.Б., Негурица Д.С. // Восточно-Европейский журнал передовых технологий. 2013. Т. 3. № 10 (63). С. 25-28.

[Allmendinger et al, 2005] Allmendinger, G. Four strategies for the age of smart services / Allmendinger G., Lombreglia R. // Harvard Business Review. 2005. Vol. 83, 10 10. P. 131.

[Barile et al, 2010] Barile, S. Smart service systems and viable service systems: Applying systems theory to service science / Barile S., Polese F. // Service Science. 2010. Vol. 2, 1-2 1-2. P. 21-40.

[Gavrilova et al, 2009] Gavrilova, T. Ontology-Based Conceptual Domain Modeling for Educational Portal. / Gavrilova, T., Gorovoy, V., & Petrashen, E. // IEEE International Conference on Advanced Learning Technologies, JUL 15-17, 2009, Riga, LATVIA, 495-496.

[Benbya et al, 2004] Benbya, H. Corporate portal: a tool for knowledge management synchronization / Hind Benbya, Giuseppina Passiante, Nassim Aissa Belbaly. // International Journal of Information Management 24 (2004) 201–220.

[Maglio, 2014] Maglio, P. P. Editorial Column—Smart Service Systems/ Maglio P. P. // Service Science. 2014. Vol. 6, 1 1. P. i-ii.

[Jámbor et al, 2012] Jámbor, T. A user-centered design approach to the development of web-portal: the NOBIT experience / Timea Jámbor, Peter Seprenyi, Luca Morganti, Andrea Gaggioli, Silvio Bonfiglio. // Tomorrow in sight: from design to delivery. Proceedings of the 4th AAL Forum. Eindhoven, the Netherlands, 24-27 September 2012, p.429-434.

FROM ENTERPRISE INFORMATION PORTALS TO KNOWLEDGE MANAGEMENT PORTALS: THE ROLE OF INTELLIGENT SERVICES

Strakhovich E.V., Vlasov S.A., Gavrilova T.A.

Graduate School of Management St.-Petersburg State University, St.-Petersburg, Russia

strakhovich@gson.pu.ru

vlasov@gson.pu.ru

gavrilova@gson.pu.ru

The intelligent (smart) services, supporting corporate knowledge portals, are considered. The problem of definition and construction of their modern classification is raised. The key trends in the intelligent (smart) services research and development are discussed.



УДК 004.8

ОБЗОР СИСТЕМ КОЛЛАБОРАТИВНОЙ ФИЛЬТРАЦИИ

Шереметова Е.И.

*Санкт-Петербургский государственный политехнический университет,
г. Санкт-Петербург, Россия*

sheremetova.ei@gmail.com

В обзоре представлены различные виды систем коллаборативной фильтрации и их алгоритмы. Основным параметром для объединения алгоритмов в группы является сходство в их математической реализации. Предложенная классификация позволяет облегчить выбор алгоритма для конкретной задачи, имеющей ряд особенностей (количество пользователей и объектов в системе, время вычислений и др.).

Ключевые слова: рекомендующие системы, коллаборативная фильтрация, интеллект-карта.

Введение

Рекомендующие системы широко используются в области информационных технологий. Такие системы работают с информацией, полученной после обработки истории предпочтений пользователей.

В последнее время появилось большое количество коммерческих приложений, основанных на построении предположений о том, какие позиции могут заинтересовать пользователя системы. Именно это сделало актуальным вопрос выбора наиболее подходящего алгоритма для реализации системы.

Ниже рассматриваются получившие распространение на сегодняшний день алгоритмы коллаборативной фильтрации, а также их классификация в зависимости от типа используемых при реализации математических методов.

1. История рекомендующих систем

Первой рекомендующей системой, ставшей прародителем современных систем, является система оценки текста – одного из наиболее сложных материалов для анализа. Рекомендующая система Information Tapestry project от компании Xerox Palo Alto Research Center была разработана в 1992 году и позволяла фильтровать текстовые сообщения.

Впервые термин «collaborative filtering» был введен ее разработчиками в статье «Using

collaborative filtering to weave an information tapestry» [Goldberg et al., 1992]. Кроме того, в этой статье была сформулирована идея о том, что в процесс создания рекомендаций должны быть вовлечены люди, дающие оценки изученным ими документам.

Рекомендующие системы получили известность относительно недавно – в середине 1990-х годов. В это же время появилось четкое разделение в понятиях «системы коллаборативной фильтрации» и «рекомендующей системы». Это вызвано тем, что рекомендующие системы могут основываться на разных подходах к решению главной задачи – вычислению предположительной оценки объекта конкретным пользователем. Рассмотрим существующие на сегодняшний день классификации рекомендующих систем.

2. Современные рекомендующие системы

2.1. Классификация рекомендующих систем

Рекомендующие системы в работе А.В. Пономарева были разделены на три вида – контентные, коллаборативной фильтрации и гибридные системы [Пономарев, 2013]. На рисунке 1 представлена интеллект-карта, описывающая виды рекомендующих систем. Рассмотрим каждую из этих систем подробнее.

1. Content-based – контентные системы, базирующиеся на оценке схожести собственных характеристик объектов.

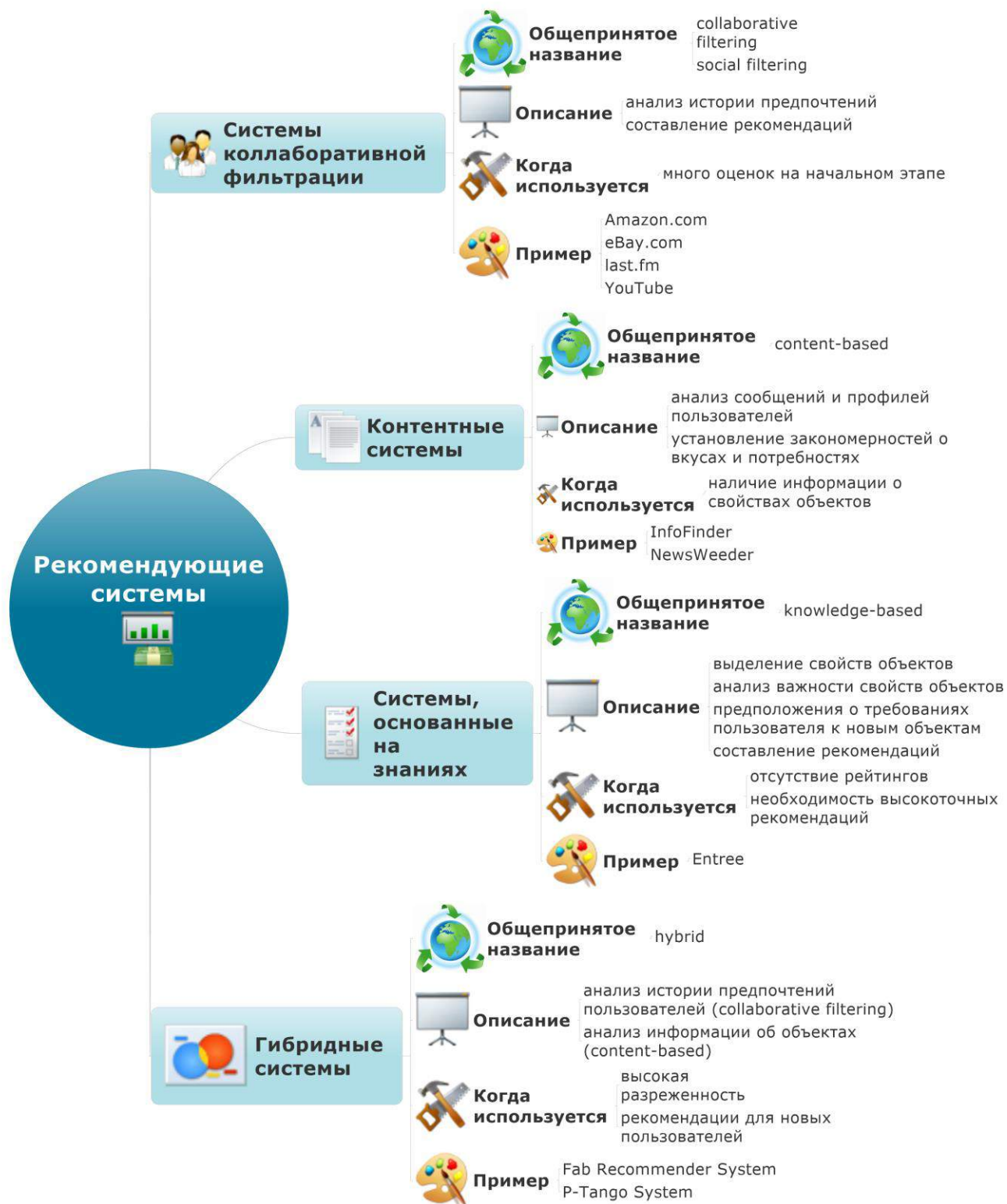


Рисунок 1 – Типы рекомендующих систем

Принцип работы заключается в анализе текстовой информации (документы, URL, сообщения пользователей, профили пользователей и др.) и установлении на ее основе закономерностей о вкусах, предпочтениях и потребностях пользователя [Su&Khoshgoftaar, 2009].

Примером content-based является система NewsDude, рекомендующая новости на основе

анализа их содержания.

2. Collaborative filtering – системы коллаборативной фильтрации. Такие системы используются в сервисах Amazon.com, eBay.com, YouTube и др.

Основу их работы составляет анализ истории предпочтений пользователей (совершенные

покупки, просмотренные фильмы и т.д.). Этот подход исключает проблему зависимости работы системы от предметной области (content-based). Однако у таких систем возникают трудности при отсутствии достаточной «истории» предпочтений [Burke, 2002]. Эта проблема решена в гибридных рекомендующих системах, сочетающих в себе несколько различных подходов к построению рекомендующих систем.

3. Hybrid – гибридные системы, которые комбинируют принципы работы систем коллаборативной фильтрации и контентных систем. Преимущество этого типа систем заключается в возможности работы с разреженными матрицами оценок и с новыми пользователями системы.

В работе Robin Burke описывается новый вид гибридной системы – EntreeC, которая включает в себя методы системы коллаборативной фильтрации и методы систем, основанных на знании о предпочтении пользователя не в области конечного объекта, а в области совокупности предъявляемых к нему требований [Burke, 2000].

Таким образом, автор выделяет четвертый вид рекомендующих систем – knowledge-based recommender systems. Такой подход более сложен, однако позволяет значительно увеличить точность генерируемых рекомендаций и уменьшить количество ошибок.

Стоит принять во внимание, что точность предположений не является единственным критерием выбора алгоритма рекомендующей системы. Помимо точности к таким критериям относятся скорость вычислений, количество пользователей и объектов, разреженность матрицы и др.

2.2. Системы коллаборативной фильтрации

При реализации алгоритмов коллаборативной фильтрации в рекомендующих системах используются исходные данные в виде разреженной матрицы оценок (Пользователи-Объекты) Основной задачей алгоритма является заполнение этой

матрицы и, таким образом, предоставление данных о том, какие объекты получили наивысшие предположительные оценки для каждого пользователя.

Появление большого многообразия способов вычисления неизвестных оценок привело к необходимости объединения их в группы по определенному признаку. Большинство современных авторов статей, например [Su&Khoshgoftaar, 2009], [Das et al., 2007], [Burke, 2002], выделяют 2 основных типа алгоритмов: model-based и memory-based.

1. Алгоритмы memory-based получили такое название из-за способа вычислений предположительных оценок – предположения строятся на базе рейтингов других пользователей и весов конкретного пользователя/объекта (в зависимости от метода) [Das et al., 2007]. Таким образом, оценивается показатель сходства, от которого зависит результат. Это обеспечивает значительную простоту реализации.

К методам memory-based авторы статьи Xiaoyuan Su и Taghi M. Khoshgoftaar относят методы Neighbor-based, основанные на «соседстве» (сходстве оценок) пользователей, и Item-/User-based top-N, которые из числа соседей выбирают N наиболее часто оцениваемых соседями элементов [Su&Khoshgoftaar, 2009].

2. Алгоритмы model-based основаны на построении «модели пользователя» в соответствии с историей его предпочтений. Среди них автор статьи «Google News Personalization: Scalable Online Collaborative Filtering» Abhinandan Das выделяет следующие алгоритмы: latent semantic indexing (LSI), singular value decomposition (SVD), Bayesian clustering, probabilistic latent semantic indexing (PLSI), multiple multiplicative Factor Model (MMF), Markov Decision process и Latent Dirichlet Allocation (LDA) [Das et al., 2007].

Описанные типы алгоритмов можно представить в виде интеллект-карты (рисунок 2). Типичные ошибки оформления статей.

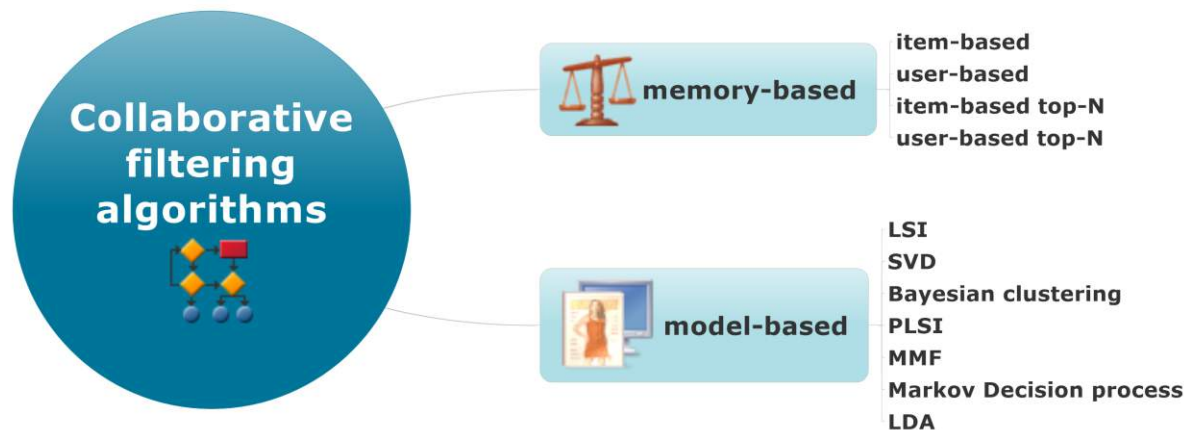


Рисунок 2 – Типы алгоритмов коллаборативной фильтрации. Классический подход

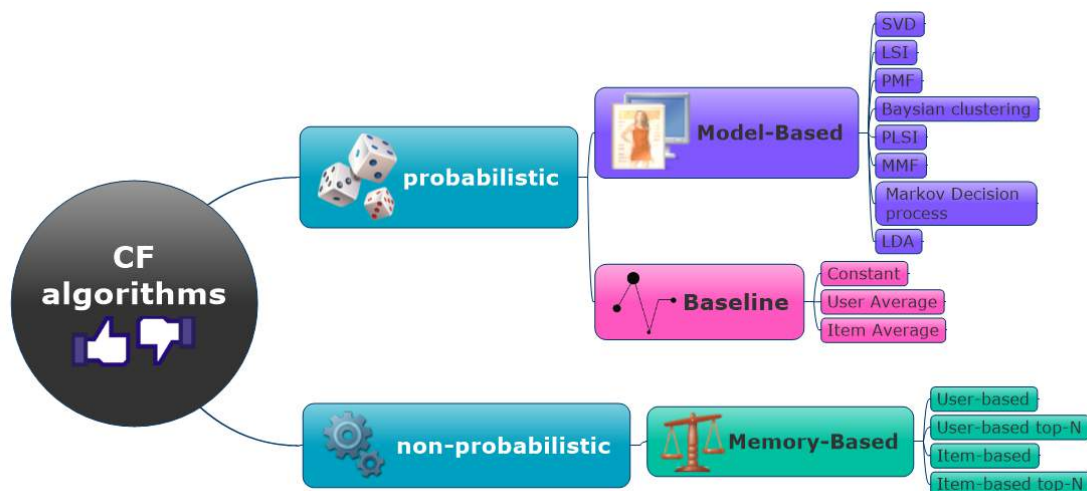


Рисунок 3 – Типы алгоритмов коллаборативной фильтрации. Расширенная классификация

3. Обзор современных классификаций алгоритмов коллаборативной фильтрации

Некоторые исследователи наряду с общепринятой классификацией, приведенной в предыдущей главе, выделяют также еще несколько типов алгоритмов. В статье [Ekstrand et al., 2011] описывается иной подход к классификации. Авторы выделяют отдельную группу вероятностных методов которые основывается на построении вероятностной модели поведения пользователя и использовании ее для построения предположений о его будущем поведении.

К вероятностным методам коллаборативной фильтрации авторы относят LDI, PLSI, SVD, Bayesian networks, Markov Decision process и Bayesian clustering (которые в предыдущей главе рассматривались как методы model-based).

Кроме того, авторы указанной статьи также выделяют еще одну группу методов Baseline Predictors и описывают ее на примере метода User Average.

Однако некоторые исследователи относят к этой группе методов также Constant и Item Average [Su&Khoshgoftaar, 2009]. Их также можно отнести к вероятностным методам, так как они основываются на вычислении вероятностных оценок.

Методы memory-based нельзя классифицировать как вероятностные, поскольку в этих алгоритмах отсутствует этап построения профиля (вероятностной модели) пользователя. Они основываются исключительно на информации о прошлых оценках пользователей и соседстве (пользователей или объектов). Таким образом, эти алгоритмы следует отнести к детерминированному подходу.

Объединив рассмотренные выше современные подходы к классификации, можно построить

интеллект-карту методов коллаборативной фильтрации (рисунок 3).

4. Тенденции к развитию систем коллаборативной фильтрации

Дальнейшее развитие систем коллаборативной фильтрации неразрывно связано с решением тех проблем, которые существенно влияют на качество их работы. Ряд исследователей [Sindhwani&Melville, 2008], [Su&Khoshgoftaar, 2009] выделяют три основных вида проблем. Они представлены на интеллект-карте (рис. 4).

4.1. Проблема «холодного старта»

Согласно статье [Schein et al., 2002], такая ситуация возникает при появлении в системе нового пользователя. Многие алгоритмы не предусматривают возможность составления рекомендаций для пользователей, которые еще не оценили ни одной позиции.

Попытки решения этой проблемы были предприняты в алгоритмах model-based, в которых оцениваются скрытые характеристики пользователей, что позволяет обоснованно составлять рекомендации на этапе первого обращения пользователя к системе. Однако такие рекомендации все же имеют низкую точность.

4.2. Проблема индивидуального подхода

Среди пользователей системы, как правило, находится человек, который по своим предпочтениям значительно отличается от большинства людей. Решение этой проблемы в большинстве современных систем ограничивается использованием алгоритмов, которые не позволяют оценкам таких пользователей вносить значительные изменения в какие-либо общие результаты (таким образом, проблема просто игнорируется).



Рисунок 4 – Проблемы в коллаборативной фильтрации

Однако в статье [Ghazanfar&Prugel-Bennett, 2011] было предложено решение, которое предполагает использование SVM-регрессии для составления рекомендаций пользователям-индивидуалистам и методов кластерного анализа для обычных пользователей.

Следует обратить внимание на то, что такая система использует также и подходы, относящиеся к content-based рекомендующим системам, что обязывает отнести рассмотренный пример к гибридным рекомендующим системам (рисунок 1). Таким образом, вопрос поиска подхода на основе алгоритмов коллаборативной фильтрации все еще остается нерешенным.

4.3. Псевдоклиентские атаки

Данная проблема возникает, когда пользователи намеренно ставят высокие оценки одним товарам и низкие – товарам-конкурентам, пытаясь, таким образом, в коммерческих целях искусственно повлиять на рейтинг товаров. Такие атаки оказывают значительное воздействие на точность рекомендаций, поэтому поиск решения является наиболее актуальной темой из всех рассмотренных в этой главе.

Различные варианты решения описываются в многих современных статьях [Mehta &Nejdl, 2008], [Zou&Fekri, 2013], [Zhan&Kulkarni, 2014]: авторы предлагают использовать комбинации нескольких алгоритмов, среди которых SVD и PSA, спектральная кластеризация и user-based, Belief Propagation (model-based) и PCA – все они относятся к разным группам алгоритмов и дают разные результаты в борьбе с атаками. В данный момент продолжаются поиски решения проблемы с целью создания более точных и универсальных подходов.

Заключение

В обзоре рассмотрены основные алгоритмы коллаборативной фильтрации. В соответствии с

принятой классификацией, отраженной на рисунке 3, эти методы были разделены на группы:

- вероятностные методы, которые основываются на построении вероятностных моделей пользователей;
- детерминированные методы, работающие непосредственно с матрицей оценок без создания каких-либо дополнительных профилей пользователей или объектов.

Детерминированные методы исторически были разработаны раньше и являются простыми в реализации. Они по-прежнему применяются в рекомендующих системах, в особенности в сочетании с другими алгоритмами.

Преимущество вероятностного подхода заключается в получении более точных результатов. Эти методы широко используются для решения различных проблем в рекомендующих системах, описанных в главе 4, что указывает на наличие значительных перспектив развития вероятностных алгоритмов коллаборативной фильтрации.

Библиографический список

- [Пономарев, 2013] Пономарев А. В. Обзор методов учета контекста в системах коллаборативной фильтрации / А.В. Пономарев //Труды СПИИРАН. – 2013. – Т. 7. – №. 30. – С. 169-188.
- [Burke, 2002] Burke R. Hybrid recommender systems: Survey and experiments / R. Burke //User modeling and user-adapted interaction. – 2002. – Т. 12. – №. 4. – С. 331-370.
- [Burke, 2000] Burke R. Knowledge-based recommender systems / R. Burke //Encyclopedia of library and information systems. – 2000. – Т. 69. – №. Supplement 32. – С. 175-186.
- [Das et al., 2007] Das A. S. et al. Google news personalization: scalable online collaborative filtering / A. S. Das et al. //Proceedings of the 16th international conference on World Wide Web. – ACM, 2007. – С. 271-280.
- [Ekstrand et al., 2011] Ekstrand M. D., Riedl J. T., Konstan J. A. Collaborative filtering recommender systems / M. D. Ekstrand, J. T. Riedl, J. A. Konstan //Foundations and Trends in Human-Computer Interaction. – 2011. – Т. 4. – №. 2. – С. 81-173.
- [Ghazanfar&Prugel-Bennett, 2011] Ghazanfar M., Prugel-Bennett A. Fulfilling the needs of gray-sheep users in recommender systems, a clustering solution / M. Ghazanfar, A. Prugel-Bennett; – 2011.

[Goldberg et al., 1992] Goldberg D. et al. Using collaborative filtering to weave an information tapestry / D. Goldberg et al. // Communications of the ACM. – 1992. – T. 35. – №. 12. – C. 61-70.

[Mehta & Nejdil, 2008] Mehta B., Nejdil W. Attack resistant collaborative filtering / B. Mehta, W. Nejdil // Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval. – ACM, 2008. – C. 75-82.

[Sindhvani&Melville, 2008] Sindhvani V., Melville P. Document-word co-regularization for semi-supervised sentiment analysis / V. Sindhvani, P. Melville //Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on. – IEEE, 2008. – C. 1025-1030.

[Schein et al., 2002] Schein A. I. et al. Methods and metrics for cold-start recommendations / A. I. Schein et al. //Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval. – ACM, 2002. – C. 253-260.

[Su&Khoshgoftaar, 2009] Su X., Khoshgoftaar T. M. A survey of collaborative filtering techniques / X. Su, T. M. Khoshgoftaar //Advances in artificial intelligence. – 2009. – T. 2009. – C. 4.

[Zhan&Kulkarni, 2014] Zhang Z., Kulkarni S. R. Detection of shilling attacks in recommender systems via spectral clustering / Z. Zhang, S. R. Kulkarni //Information Fusion (FUSION), 2014 17th International Conference on. – IEEE, 2014. – C. 1-8.

[Zou&Fekri, 2013] Zou J., Fekri F. A belief propagation approach for detecting shilling attacks in collaborative filtering / J. Zou, F. Fekri //Proceedings of the 22nd ACM international conference on Conference on information & knowledge management. – ACM, 2013. – C. 1837-1840.

COLLABORATIVE RECOMMENDER SYSTEMS

Sheremetova E.I.

*St. Petersburg State Polytechnic University, St.
Petersburg, Russia*

sheremetova.ei@gmail.com

This paper presents different types of recommender systems, especially collaborative filtering systems and their methods. To classify collaborative filtering techniques I researched similarities in their mathematical implementation. The proposed classification can facilitate further assumptions of using algorithms for specific tasks depending on several criteria (number of users, number of items, computation time etc.).

Keywords: recommender systems; collaborative filtering; mind map.

Introduction

Recommender systems are widely used in information technologies.

Nowadays such systems became an important tool in commercial applications based on large information and product spaces. It puts the problem of choosing the most suitable algorithm for implementing in the system.

This paper discusses common collaborative filtering algorithms and their classifications depending on used mathematical methods.

Main Part

The first recommender system Information Tapestry project was developed in 1992 by Xerox Palo Alto Research Center and allowed filtering text messages.

For today there are 4 main types of recommender systems:

1. content-based system use similarity between items a given user has liked in the past.

2. collaborative filtering (CF) identifies users that are similar in their tastes to given user and recommends items they have liked.

3. hybrid systems combine CF algorithms and content-based approach.

4. knowledge-based systems use a knowledge of how an item meet a user's needs.

Each of these approaches has its strengths and weaknesses. Now the most widely implemented technology is collaborative filtering.

CF algorithms use a matrix of preferences for items by users to predict a product a new user might wish to purchase or examine.

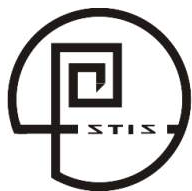
Most researchers are describing 2 groups of CF algorithms: memory-based and model-based. But there are several approaches allocating a number of different types of CF techniques which allows developing a new classification proposed in the third section of the paper.

Conclusion

The overview shows an approach to classifying CF techniques which includes 2 main types – determinate algorithms and probabilistic algorithms.

Determinate algorithms were developed earlier. There are a variety of determinate CF algorithm implementations because it is easy to implement them but low accuracy is often the main weakness.

Probabilistic methods, on the contrary, are more complicated but on the other hand they can make more accurate predictions. It indicates to the presence of great prospects for further development.



OSTIS-2015

(Open Semantic Technologies for Intelligent Systems)

УДК 004.89

МЕТОДЫ И ИНСТРУМЕНТЫ РАБОТЫ С БИЗНЕС-ПРАВИЛАМИ

Савин А.М.

*Санкт-Петербургский государственный политехнический университет,
г. Санкт-Петербург, Российская Федерация*

Savin.Alexey.M@gmail.com

В настоящее время многие компании в целях достижения более высокой гибкости и наилучшей производительности в процессе своей работы внедряют подход, основанный на применении бизнес-правил. Это позволяет им максимально быстро реагировать на постоянно меняющиеся потребности рынка, однако выбор наиболее подходящей системы управления может быть весьма непростой задачей в связи с большим количеством различных вариантов. Для того, чтобы облегчить выбор в подобной ситуации, в рамках данной работы предлагается обзор различных систем, применяемых в процессе использования подобного подхода.

Ключевые слова: бизнес-правила; системы управления бизнес-правилами; подход использования бизнес-правил

Введение

На протяжении последних десятилетий большинство компаний в процессе своей деятельности так или иначе сталкиваются с необходимостью автоматизации и управления бизнес-процессами [Hildreth, 2005], которые обычно включают в себе некоторые наборы правил, определяющие логику поведения в различных ситуациях. Долгое время подобная логика непосредственно закладывалась в создаваемое программное обеспечение [Hildreth, 2005], а любые изменения приводили к необходимости переписывания различных участков кода и последующего развёртывания новых версий систем автоматизации.

Со временем стало понятно, что подобный подход является слишком дорогостоящим в силу необходимости привлечения технических специалистов, а также высокой длительности внесения изменений, что особенно актуально для компаний, работающих в банковской и финансовой сферах [BSI, 2010b], для которых чрезвычайно важно максимально быстро реагировать на изменения рынка. Это привело к возникновению нового подхода, основанного на использовании так называемых бизнес-правил (англ. business rules approach).

Основная идея данного подхода заключается в отказе от жёсткого программирования логики в пользу создания эквивалентного ей набора правил, преимущественно выраженного в форме

утверждений вида “если <условие> - то <действие>”, которые, в свою очередь, должны быть записаны в максимально простой и понятной нетехническим специалистам форме, например, в графическом виде, в виде таблицы решений или на некотором языке близком к естественному. На основе созданных правил формируется некоторое внешнее хранилище, в котором они могут быть легко изменены, и на которое в процессе своей работы должно опираться программное обеспечение для автоматизации.

Таким образом, использование бизнес-правил позволяет максимально быстро вносить изменения в реализованную логику и как следствие добиться более высокой гибкости работы используемого программного обеспечения. Для полного понимания решений используемых в процессе применения описываемого подхода рассмотрим историю его возникновения.

1. История возникновения концепции бизнес-правил

Подход создания и использования бизнес-правил зародился под влиянием трёх различных областей [BSI, 2010b]:

- Искусственный интеллект, в частности, экспертные системы
- Моделирование данных
- Реинжиниринг бизнес-процессов

В 1980-х годах многие компании были заняты

разработкой так называемых экспертных систем, основная задача которых заключалась в предоставлении ответов на интересующие вопросы или предложении решений различного рода проблем в областях, где в обычной ситуации необходимо привлечение соответствующих экспертов [BSI, 2010b]. Одним из классических примеров подобного рода задач можно считать постановку медицинского диагноза.

Однако постепенно рынок становился всё более насыщенным и, кроме того, крупные компании, в частности работавшие в банковской и финансовой сферах, стали сталкиваться с проблемой быстрого реагирования на изменения рынка, влекущей за собой потребность внесения изменений в бизнес-логику, реализованную в рамках программного обеспечения. В результате в конце 80-х – начале 90-х годов зародилась концепция бизнес-правил [Hildreth, 2005] и часть компаний, занимавшихся разработкой экспертных систем, начали исследование данной проблемы, при этом подобный подход потребовал значительных усилий по выполнению формализации различных бизнес-понятий.

Следующим шагом в развитии бизнес правил стало создание в 1989 году организации под названием Business Rules Group, которая обратила внимание на моделирование данных (англ. data modeling), в частности на то, что многие реляционные базы данных, используемые организациями, включали в себя таблицы, которые можно трактовать как таблицы принятия решений [BSI, 2010b]. Основной характерной чертой данного этапа являлся поиск вариантов упрощения ввода правил на основе разбора некоторых заданных в текстовом виде критериев, описывающих бизнес-логику, с использованием заранее определённых терминов, что в дальнейшем позволило бы представить их в форме, допускающей непосредственное выполнение [BRC, 2008].

Деятельность данной организации оказала непосредственное влияние на будущее создание консорциумом Object Management Group (OMG) стандартов Semantics of Business Vocabulary and Business Rules (SVBR) и Business Motivation Model (BMM) [BRC, 2008].

Последний на сегодняшний день этап становления бизнес-правил связывают с развитием концепции управления бизнес-процессами (англ. business process management, BPM), в частности с публикацией в 2007 книги Тома Дэбейра “Business Process Management with a Business Rules Approach” [BSI, 2010b]. Основная идея заключается в том, что любой бизнес-процесс, как правило, представляет собой сложную структуру с некоторым количеством дочерних процессов и принятием решений на различных его этапах, которые в свою очередь должны контролироваться бизнес-правилами. Таким образом, в настоящее время происходит постепенная интеграция концепций бизнес-правил и управления бизнес-

процессами, что особенно хорошо видно на примере компании IBM и её системы управления бизнес-правилами JRules, которая постепенно объединяется с другими продуктами компании.

2. Современное состояние концепции бизнес-правил

В общем случае бизнес-правило – это утверждение, которое определяет или ограничивает тот или иной аспект деятельности организации, с целью управления этим аспектом или влияния на него посредством наложения структуры [Бирн и др., 2012a].

Значительная часть бизнес-правил формулируются в форме выражений вида “если – то”, где левая часть содержит некоторые условия, а правая – соответствующие действия, которые необходимо выполнить в случае их истинности, например: “Если сумма договора превышает 5000\$, то отправить сообщение финансовому директору”. Именно для таких правил осуществляется непосредственное выполнение различными методами, в то время их формальное описание происходит на основе сформированных моделей с различными атрибутами и связями, которые в свою очередь создаются с использованием правил, не попадающих под эту структуру.

Бизнес-правила существуют в любой организации, вне зависимости от того, записаны ли они в явном виде. Как уже было сказано ранее, основная идея рассматриваемого подхода заключается в реализации бизнес-логики посредством правил, сформулированных в максимально простой и понятной форме и находящихся в некотором внешнем, по отношению к использующему их программному обеспечению, хранилище. Однако, уже исходя из этого, можно предположить, что их использование отрицательно скажется на производительности в сравнении с явной программной реализацией бизнес-логики, в результате чего применение подобного подхода имеет смысл лишь для наиболее критичных правил. Их выбор осуществляется из соображений изменчивости, сложности, а также потенциальной переиспользуемости [Бирн и др., 2012b].

Процесс создания, управления и выполнения бизнес-правил осуществляется с помощью так называемых систем управления бизнес-правилами (англ. business rules management system, BRMS), которые в свою очередь состоят из нескольких компонентов [Ritchie, 2011]:

- Хранилище бизнес-правил.
- Инструменты работы с бизнес-правилами – позволяют изменять существующие правила, а также создавать новые и добавлять их в хранилище.
- Машина исполнения бизнес-правил (англ. business rules engine, BRE) – осуществляет

выполнение правил с использованием определённых методов и выдачу соответствующего результата.

Архитектура типичной системы управления бизнес-правилами основывается на веб-технологиях, в частности, среда для работы с правилами, и машина для их исполнения могут быть развёрнуты в качестве веб-сервисов [Бирн и др., 2012b]. В общем случае можно выделить следующие достоинства использования подобного подхода [HSG, 2012]:

- Повышение контролируемости бизнес-логики
- Уменьшение зависимости от технических специалистов
- Возможность переиспользования бизнес-правил

К основным недостаткам следует отнести такие факторы, как:

- Высокая сложность формализации правил
- Увеличение времени разработки систем автоматизации бизнес-процессов
- Зависимость от конкретной системы управления бизнес-правилами

На сегодняшний день использование бизнес-правил является устоявшейся практикой в крупных организациях, работающих в сферах страхования и здравоохранения, а также банковской и финансовой сферах. Перейдём к рассмотрению различных систем управления бизнес-правилами.

2.1. Системы управления бизнес-правилами

2.1.1. *WebSphere ILOG JRules*

Первоначально данная система была разработана компанией ILOG, которая в свою очередь в 2009 году была приобретена IBM, в результате чего JRules вошла в семейство продуктов WebSphere, а на сегодняшний день также является одним из компонентов более широкой продукта под названием IBM Operational Decision Manager. В рамках данной системы управления бизнес-правила могут быть представлены в одной из следующих форм [Бирн и др., 2012b; ILOG, 2008]:

- Business Action Language (BAL) – близкий к естественному языку описания правил в виде утверждений, включающих в себя набор условий, а также действий, которые должны быть выполнены в случае их истинности.
- Таблица принятия решений, где каждая строка характеризует некоторое правило.
- Дерево решений, где ветви представляют собой условия, а листья – действия, которые должны быть выполнены в случае их истинности.

Сами правила составляются с использованием словаря терминов, выводимого автоматически из, так называемых, моделей бизнес-объектов (англ. business object model, BOM), которые являются некоторым внутренним форматом представления и, в свою очередь, выводятся из моделей исполняемых объектов (англ. executable object model, XOM). Последние непосредственно являются классами, используемыми при разработке программного обеспечения. Таким образом, схема классов разрабатываемой системы может быть автоматически использована для написания правил [Бирн и др., 2012b; ILOG, 2008].

Каждое созданное правило транслируется в язык ILOG Rule Language (IRL) [ILOG, 2008], который используется непосредственно при их исполнении. Кроме того, данный язык может быть использован для написания сложных правил, работающих непосредственно с BOM представлением, которые в общем случае сможет редактировать только технический специалист.

Подобная система выведения представлений позволяет создать двухстороннюю интеграцию с языками программирования [ILOG, 2008], при которой результат исполнения правил передаётся программе непосредственно в виде объектов. При этом JRules существует в версиях, осуществляющих интеграцию с Java через Eclipse, и с .NET через Visual Studio. Кроме того, JRules также позволяет опубликовать правила в качестве веб-сервиса.

При исполнении правил JRules может использовать несколько различных подходов [ILOG, 2005]:

- Логический вывод (Inferencing Execution Mode) – применяет модификацию алгоритма Rete под названием RetePlus для выполнения сложных наборов правил, когда выполнение одного правила может приводить к срабатыванию нескольких других.
- Последовательное выполнение (Sequential Execution Mode) – последовательно выполняет все удовлетворяющие правила, позволяет достичь максимальной скорости для простого набора.

Машина исполнения бизнес-правил JRules поддерживает стандарт JSR 94, что позволяет взаимодействовать с ним сторонним приложениям на языке Java. Кроме того, сама система поддерживает интеграцию с пакетом Microsoft Office, позволяя редактировать правила с использованием Word и Excel [Ritchie, 2011]. К недостаткам JRules можно отнести высокую стоимость, а также довольно высокую сложность освоения, ввиду того, что данная система представляет собой комплексное решение, рассчитанное на интеграцию с другими продуктами IBM [Вейнберг, 2013], однако последнее, в частности, позволяет при написании правил

использовать в качестве моделей UML-диаграммы, созданные в IBM Industry Model.

2.1.2. Drools

Drools представляет собой свободно распространяемую систему управления бизнес-правилами, разрабатываемую компанией Red Hat. Кроме того, существует её коммерческая версия под названием JBoss Enterprise BRMS, которая также включает в себя техническую поддержку для корпоративных клиентов. В рамках данной системы управления бизнес-правила могут быть представлены в одной из следующих форм [Оливьери, 2009; RedHat, 2013]:

- Drools Rule Language (DRL) - собственный язык описания правил, каждое из которых включает в себя некоторое условие и действие, которое необходимо совершить при его истинности, например, выполнить некоторый Java-код.
- Таблица принятия решений в форматах Excel или CSV.

Создание правил осуществляется путём интеграции соответствующей функциональности в среду разработки Eclipse, а для их описания в качестве моделей используются непосредственно классы разрабатываемого программного обеспечения, при этом данная система управления бизнес-правилами рассчитана только на интеграцию с системами, написанными на Java.

Для исполнения правил Drools позволяет выбрать один из следующих методов [RedHat, 2013]:

- Модификацию алгоритма Rete под названием ReteOO, оптимизированную для объектно-ориентированных систем.
- Собственный алгоритм PHREAK, использующий концепцию ленивых вычислений.

Подробную информацию о каждом из них можно найти в официальной документации [RedHat, 2013]. Кроме того, для каждого правила может быть задан некоторый приоритет, влияющий на порядок исполнения правил [Оливьери, 2009]. Как и JRules, Drools также поддерживает стандарт JSR 94. В качестве основного недостатка системы можно выделить отсутствие возможности интеграции с другими языками программирования, в частности, с платформой .NET, однако существует ряд свободно распространяемых проектов, осуществляющих соответствующее портирование.

2.1.3. Visual Rules

Visual Rules – система управления бизнес-правилами на базе Eclipse, разработанная компанией Bosch, основная идея которой заключается в создании средств, позволяющих полностью осуществлять создание и управления бизнес-правилами в визуальной форме без необходимости

привлечения технических специалистов, что является основным отличием от рассмотренных ранее систем: в то время как последние делают упор на автоматическое выведение необходимых для написания правил моделей из классов разрабатываемого программного обеспечения, Visual Rules предоставляет возможность описания этих моделей на языке UML с соответствующими ограничениями на языке OCL [BSI, 2010a]. В рамках данной системы бизнес-правила могут быть представлены в одной из следующих форм:

- Flow rule – по сути представляют собой дерево решений, аналогичное JRules с упором на максимальное удобство редактирования, посредством тщательно проработанного пользовательского интерфейса.
- Decision table - таблица принятия решений, где каждая строка характеризует некоторое правило.
- State flow – диаграмма переходов между состояниями системы, которые срабатывают при выполнении заданных условий, что также можно интерпретировать как некоторые правила.

Visual Rules, как и предыдущие системы, позволяет провести интеграцию с Java приложением, а также осуществить развёртывание правил в виде web-сервиса, что позволяет использовать их на любой платформе. Другим немаловажным преимуществом является возможность осуществления визуальной отладки правил.

Важным отличием Visual Rules является отказ от использования алгоритмов логического вывода при исполнении правил в пользу генерации Java кода, имплементирующего соответствующую логику [BSI, 2010c]. К недостаткам системы можно отнести высокую цену, а также, в некоторых случаях, отказ от текстового написания правил, однако несмотря на это она обладает более низким порогом вхождения по сравнению с JRules (Вейнберг, 2013).

2.1.4. Сравнительный анализ

Помимо рассмотренных ранее, также существует и ряд других систем управления бизнес-правилами, таких как Corticon, InRule, OpenRules, Blaze Advisor и Oracle Business Rules, которые на сегодняшний день получили менее широкое распространение [Вейнберг, 2013], и кроме того не используют принципиально других подходов в сравнении с рассмотренными системами. На основании описанных систем управления бизнес-правилами на сегодняшний день можно выделить следующие их ключевые характеристики, которые также могут являться критериями для проведения классификации (Рисунок 1):

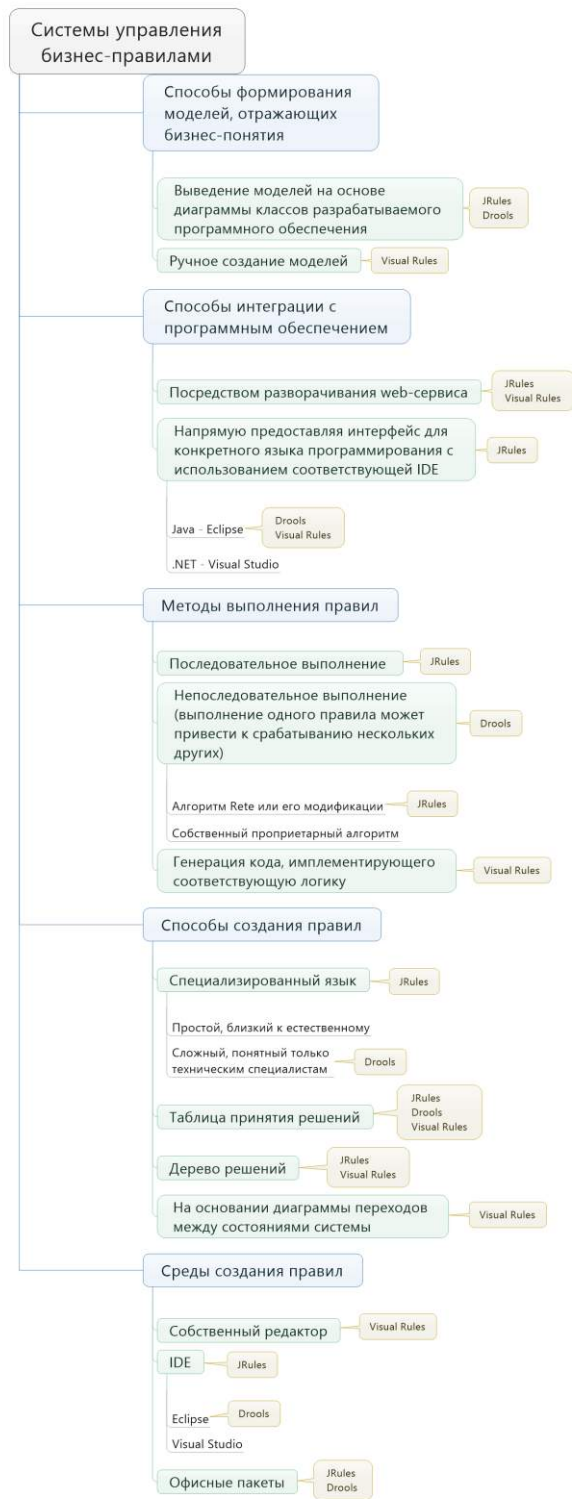


Рисунок 1 – Ключевые характеристики систем управления бизнес-правилами

Приведённая интеллектуальная карта является аннотированной, что позволяет достаточно просто оценить характеристики конкретных систем, в частности хорошо видно, что каждая из них предлагает несколько различных вариантов выбора в рамках каждой из категорий категорий.

3. Перспективы развития концепции бизнес-правил

Основываясь на текущих тенденциях применения подхода использования бизнес-правил, а также на прогнозах производителей систем управления [BSI, 2010b] и независимых исследователей [Rymer et al., 2011], можно выделить несколько наиболее вероятных путей дальнейшего развития данной концепции:

3.1. Объединение с обработкой бизнес-событий

Одним из подходов, развивавшихся параллельно с бизнес-правилами является так называемая обработка бизнес-событий, которая заключается в построении приложений, способных путём выявления закономерностей в потоках входной информации в реальном времени ответить на вопрос, какие именно события происходят в рамках бизнес-процесса, на которые следует обратить внимание.

Используемые информационные потоки практически всегда отражают информацию о реальном мире, например, действия посетителей казино или текущее положения транспортных средств. Для построение таких приложений некоторые компании используются так называемые системы обработки сложных событий (англ. complex event processing, CEP), в то время как другие используют для этого системы управления бизнес-правилами.

Подобный дуализм позволяет сделать предположение о возможном будущем объединении данных подходов [Rymer et al., 2011]. CEP-системы теоритически могут осуществлять автоматическое уведомление систем управления бизнес-правилами, и, кроме того, многие из них уже используют собственные “правила” в качестве обработчиков для поступающих событий. Подобного направления развития придерживается Bosch, вместе с их рассмотренной ранее системой Visual Rules [Rymer et al., 2011; BSI, 2010b].

3.2. Интеграция в управление решениями

Другим возможным путём развития для бизнес-правил может служить некоторое расширение контекста в рамках подхода управление решениями (англ. decision management) [Rymer et al., 2011]. Примером применения данного подхода являются моделирование розничными сетями склонности посетителей к покупке тех или иных продуктов и использованием результатов для принятия различные решения, закладывая соответствующую логику в системы управления бизнес-правилами.

Таким образом, в рамках данного подхода часть бизнес-правил формируется динамически на основе прогнозирующих моделей, построенных с использованием накопленной статистической информации, что, в целом, позволяет сделать их более адаптивными.

Заключение

В рамках данной работы были рассмотрены различные аспекты подхода использования бизнес-правил. Как видно из истории, наибольшее влияние на его становление оказали экспертные системы. В настоящее время существует значительное количество поставщиков систем управления бизнес-правилами, и хотя на первый взгляд они существенно отличаются друг от друга по целому ряду параметров, можно выделить группу наиболее значимых с точки зрения внедрения характеристик, что позволит существенно сократить выбор в контексте потребностей конкретно взятой организации.

На сегодняшний день также наблюдаются тенденции постепенного объединения данного подхода с концепциями обработки бизнес-событий и управления решениями. В целом, подход использования бизнес-правила прочно вошёл в современные методы организации бизнес-процессов, и в дальнейшем подобная ситуация будет лишь усиливаться.

Библиографический список

- [Бирн и др., 2012a] Бирн, Б. Системы управления бизнес-правилами IBM Industry Models и ILOG: Часть 1. Определение бизнес-правил при помощи моделей IBM Industry Models [Электронный ресурс] / Б. Бирн, Г. Аймирзян. – Режим доступа: <http://goo.gl/gUEwbH>, свободный.
- [Бирн и др., 2012b] Бирн, Б. Системы управления бизнес-правилами Industry Models и ILOG: Часть 2. Добавление системы управления бизнес-правилами [Электронный ресурс] / Б. Бирн, Г. Аймирзян, Д. Уолл. – Режим доступа: <http://goo.gl/9lAvpl>, свободный.
- [Вейнберг, 2013] Вейнберг, Р. Р. Мировой рынок и критерии оценки выбора и использования систем управления бизнес-правилами // Наука и практика, 2013, № 2(10), С. 61–66
- [Оливьери, 2009] Оливьери, Р. Реализация бизнес-логики при помощи процессора правил Drools [Электронный ресурс]. – Режим доступа: <http://goo.gl/QWzE5d>, свободный.
- [Hildreth, 2005] Hildreth, S. Rounding Up Business Rules [Электронный ресурс]. – Режим доступа: <http://goo.gl/8xhNC0>, свободный.
- [Ritchie, 2011] Ritchie, A. Patterns integrating WebSphere ILOG JRules with IBM software. - Poughkeepsie, N.Y.: IBM, International Technical Support Organization.
- [Rymer et al., 2011] Rymer, J. R. The future of business rules platforms: customers are moving to event and decision management [Электронный ресурс] / J. R. Rymer, M. Gualtieri – Режим доступа: <http://goo.gl/DbFAH7>, свободный.
- [BRC, 2008] Business Rules Community. History of the Business Rules Approach [Электронный ресурс]. – Режим доступа: <http://goo.gl/xujNMMH>, свободный.
- [BSI, 2010a] Bosch Software Innovations. The benefits of Visual Rules for business applications [Электронный ресурс]. – Режим доступа: <http://goo.gl/Eyh1CC>, свободный.
- [BSI, 2010b] Bosch Software Innovations. The past, present, and future of business rules [Электронный ресурс]. – Режим доступа: <http://goo.gl/GaUDaC>, свободный.
- [BSI, 2010c] Bosch Software Innovations Visual Rules Suite Modeler. Java integration guide. Version 6.1 [Электронный ресурс]. – Режим доступа: <http://goo.gl/WUlyH1>, свободный.
- [HGS, 2012] Hartmann Software Group. Business Rule Management System [Электронный ресурс]. – Режим доступа: <http://goo.gl/R7H0OE>, свободный.
- [ILOG, 2005] ILOG. ILOG JRules: Leading the way in Business Rule Management Systems [Электронный ресурс]. – Режим доступа: <http://goo.gl/65KXem>, свободный.

[ILOG, 2008] ILOG. ILOG JRules 6.7: Getting Started [Электронный ресурс]. – Режим доступа: <http://goo.gl/IOljEJ>, свободный.

[RedHat, 2013] Drools Documentation [Электронный ресурс]. – Режим доступа: <http://goo.gl/6zdL1U>, свободный.

TOOLS AND METHODS IN BUSINESS RULES APPROACH

Savin A.M.

*Saint-Petersburg State Polytechnical University,
Saint-Petersburg, Russian Federation*

Savin.Alexey.M@gmail.com

Nowadays many companies follow the business rules approach to attain greater agility and excellent performance in their work, using Business Rule Management Systems. It allows them to react quickly to ever-changing market needs. However, selection of the most suitable one can be rather challenging due to the amount of distinct variants. All of them implement different languages to define rules and use different engines to execute them. In order to help with the choice, the paper suggests review of different systems, which are used in business rules approach.

Keywords: business rules, business rule management systems, business rules approach

Introduction

For a long time, business logic in many companies was directly implemented in business process management systems. However, disadvantages of such approach became obvious rather soon, especially for companies, which had needed to react quickly to ever-changing market needs. As a result, such situation led to birth of the new approach, based on the using of business rules.

Main Part

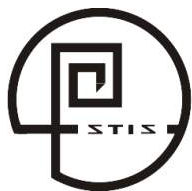
The history of business rules has three main phases: expert systems, activity of Business Rules Group and integration with business process management

Nowadays, there are many different business rules management systems. All of them implement different languages to define rules, use different engines to execute them and have some other differences. However, it is possible to seriously reduce amount of distinct variants by determining with the needs of the specific company.

In the future, the business rules approach may be integrated with the complex event processing and decision management.

Conclusion

This paper suggests a review of different business rules management systems, in order to help with the selection of the most suitable one. It is also review main historical phases of this approach and make some assumptions about its future.



УДК 005.336.4:004.9

СИСТЕМЫ УПРАВЛЕНИЯ ЗНАНИЯМИ В АТОМНОЙ ПРОМЫШЛЕННОСТИ

Костебелова В.К.

*Санкт-Петербургский государственный политехнический университет
г. Санкт-Петербург, Россия*

kostebelovavk@gmail.com

Данная статья посвящена изучению связи между процессом управления знаниями и производительностью организаций на атомных электростанциях. Исследуются механизмы управления знаниями, оказывающие существенное влияние на организационную эффективность, и примеры лучшей практики в создании, сохранении и обмена знаниями в атомной промышленности.

Ключевые слова: ядерные знания; системы управления знаниями; результаты интеллектуальной собственности; Госкорпорация «Росатом»

Введение

Современную экономику невозможно представить без особого подхода к знаниям, формирующимся и впоследствии применяющимся в различных отраслях народного хозяйства. В зависимости от размеров предприятия, управление знаниями может быть представлено эффективной многоуровневой системой, а может отсутствовать вовсе. Кроме того, немаловажными факторами, влияющими на использование управления знаниями, выступают специфика деятельности компании, заинтересованность и опытность руководства, финансовое состояние.

Так, в атомной промышленности отмечается повышенное внимание к вопросам создания, накопления, хранения, передачи и применения знаний. Это не случайно, поскольку эта сфера является одной из самых важных для жизнеобеспечения населения, и недостаточное внимание к циркулирующим в ней знаниям может оказаться опасным и даже вызвать катастрофу. Поэтому компаниям атомной отрасли необходимо не просто быть в курсе существования управления знаниями, но и внедрять доказавшие свою полезность соответствующие механизмы и методы.

1. Краткая история данной проблематики

Для начала разберемся с основными понятиями данной проблематики.

В работе [Лещенко В.О, 2014] дается следующее определение: «Управление ядерными знаниями (Nuclear knowledge management) – это

интегрированный, системный подход к процессу идентификации, приобретения, преобразования, развития, распространения, использования, разделения и сохранения знаний, создания условий для коллективной выработки новых знаний». Таким образом, процесс управления ядерными знаниями не отличается от процесса управления знаниями в целом. Но с другой стороны недостаток ядерных знаний увеличивает риск возникновения аварии на атомной станции, что может привести к необратимой катастрофе и гибели тысячи людей.

В своей работе Вильгельм Е.В. [Вильгельм Е.В, 2013] выделяет следующие основные тенденции в управлении знаниями на предприятиях ядерной энергетики:

- 1) Становление комплексного бизнес-процесса управления знаниями, включающего все стадии жизненного цикла знаний;
- 2) Более детальное изучение рисков для бизнеса, деятельности и безопасности, вызванных пробелами в знаниях;
- 3) Совершенствование стратегического планирования благодаря доступу к знаниям в большем объеме с повышенной надежностью.

В атомной промышленности активно используются 4 основные концепции управления знаниями: концепция обучающейся организации, модель динамической трансформации знания, процессная модель создания знания, концепция интеллектуального капитала.

На рисунке 1 изображены основные концепции управления знаниями в форме диаграммы Исикавы. У каждой концепции выделены представители, время зарождения и основная идея.



Рисунок 1 – Основные концепции управления знаниями

1.1. Концепция обучающейся организации

Питер Сенге [Першукова В. А. стр. 16] выделил пять универсальных взаимодополняющих элементов (дисциплин) обучающейся организации:

1) личное совершенствование - непрерывный процесс прояснения и углубления личного видения, ориентация на постоянное обучение. Разрыв между мечтой и реальностью представляет собой источник творческой энергии. При этом ошибки индивида - это события, из которых человек ещё не извлёк пользу;

2) интеллектуальные модели - тающиеся глубоко в сознании предположения, обобщения или даже картинки или образы, которые влияют на наше понимание мира и на выбираемый нами образ действия;

3) общее видение - общая для всех картина будущего. Достаточно часто общее видение является следствием харизмы лидера. Для организации необходимы лидеры, которые имеют собственное видение и способны воодушевлять;

4) групповое обучение, которое начинается с "диалога", предполагающего свободный обмен мнениями, с отбрасывания штампов и предрассудков;

5) системное мышление, которое позволяет развивать предыдущие четыре дисциплины в ансамбле, когда целое может стать больше, чем сумма его отдельных частей.

1.2. Модель динамической трансформации знания

Нонака и Такеучи [Першукова В. А., стр. 18] предложили:

1. Спираль знания
2. Условия создания организационного знания:
3. Дихотомии стратегического менеджмента в области управления знаниями:

1.3. Процессная модель создания знания

В процессной модели Холсэпла и Сайна [Першукова В. А., стр. 21-22] предложены

основные и вспомогательные виды деятельности цепочки создания знаний.

Основные:

- a) получение и отбор знаний из внешних и внутренних источников;
- b) генерирование новых знаний;
- c) изменение состояния организационного знания посредством распространения и хранения полученных, отобранных или сгенерированных знаний;
- d) материализацию знаний в конечный продукт компании для передачи этого знания во внешнюю среду.

Вспомогательные:

- a) лидерство, активизирующее все процессы создания знаний;
- b) координацию взаимозависимых процессов в управлении знаниями для обеспечения ресурсами в нужное время в нужном месте;
- c) контроль за доступностью знаний в необходимом качестве и количестве, а также за доступностью средств и ресурсов для управления знаниями с учётом ограничений и требований по обеспечению безопасности (юридическая, социальная и техническая защита);
- d) измерение ценности знаний ресурсов и процессов, включая количественные и качественные методы измерения.

2. Краткое изложение современного состояния вопроса с выделением различных подходов

В настоящее время в атомной промышленности всего мира прослеживается тенденция увеличения расходов на обучение, подготовку и повышение квалификации персонала.

В первую очередь такой рост популярности к процессу управления знаниями обусловлен резкому снижению экспертов в данной области в начале

2000-х. Как заметил в своей работе Yanko Yanev [Yanev Y., 2009] количество специалистов в области ядерной энергетики резко упало. Если в 1993 количество дипломированных специалистов в Германии приближалось к 40 тысячам, то в 1999 это число было близко к 1 тысяче, что соответственно очень сильно отразилось на квалификации рабочих.

А. Косилов. [Косилов А., 2010] в своем докладе выделил следующие проблемы ядерных знаний:

- Стареющие кадры и уход на пенсию
- Потеря знаний
- Деградация технологических навыков и потеря “know-how”
- Возможная деградация безопасности
- Снижение (исчезновение) инновационного потенциала

Активно вопросом процесса передачи и сохранения знаний в атомной промышленности занялось Международное агентство по атомной энергии (МАГАТЭ) (англ. IAEA, сокр. International Atomic Energy Agency)- международная организация для развития сотрудничества в области мирного использования атомной энергии.

В России Госкорпорация «Росатом» проводит Международную конференцию по управлению знаниями и инновациям в высокотехнологичных компаниях (Rosatom Knowledge Management). В своей статье Вильгельм Е.В. [Вильгельм Е.В., 2013] пишет, что «перед школой ставятся цели, отвечающие глобальным изменениям, а именно подготовка единой методической основы сохранения критически важных знаний, ее гармонизация, как внутриотраслевая, так и с существующими подходами МАГАТЭ; апробация подготовленных методик и процессов в организациях, их последующая доработка с учетом реалий организаций. Причем при реализации проекта учитываются лучшие практики уже осуществленных проектов в отрасли (таких как «Мост поколений», ОАО «Атомэнергомаш»), а также богатейший опыт МАГАТЭ в этой области».

Рассмотрим российскую и зарубежную практику управления знаниями. Можно выделить два подхода: традиционный подход (создание

«внутреннего» продукта) и «инновационный» подход (коммерциализация технологий).

На рис.2 представлены основные подходы к процессу управления знаниями. Традиционный подход - сохранение и передача знаний новым сотрудникам. Инновационный подход предполагает коллективное использование знаний организации и коммерциализацию знаний.

Как пишет Лещенко В.О [Лещенко В.О., 2014] Госкорпорация «Росатом» активно начинает использовать «инновационный» подход и внедряет коммерциализацию технологий.

Система управления знаниями Госкорпорации «Росатом» - это комплексный продукт: 16 объектов интеллектуальной собственности по 3 функциональным блокам.

3. Основные перспективы

В 2011 г. Госкорпорация «Росатом» начала активное внедрение системы управления знаниями в атомную промышленность. В своем докладе Лещенко В.О [Лещенко В.О., 2014] выделяет «что в силу высокой наукоемкости атомной отрасли знания и результаты интеллектуальной деятельности (далее - РИД) являются основой конкурентоспособности госкорпорации «Росатом». В связи с этим особую значимость для компании имеет совокупность стратегий и процессов по выявлению, приобретению, распространению и использованию знаний. Таким образом, системы управления знаниями в госкорпорации рассматривалась как один из инструментов реализации Программы инновационного развития».

Из доклада Лещенко В.О [Лещенко В.О., 2014] можно выделить основные функциональные блоки (подсистемы) системы управления знаниями в госкорпорация «Росатом»:

- управление научным сообществом;
- управление научно-технической информацией;
- управление результатами интеллектуальной деятельности.



Рисунок 2 – Основные подходы к управлению знаниями

Российские атомные компании только начинают создавать системы корпоративных знаний. Главная проблема российских компаний — точно определить, какие знания нужны и сегодня — крайне необходимы, где они имеются, как их можно приобрести и передавать, чтобы обеспечить фирме конкурентные преимущества.

Как пишет Лазарев С.В. [Лазарев С.В., 2009] на предприятиях энергетики перед внедрением системы управления знаниями (СУЗ) «необходимо разработать определенные подходы: идентифицировать деятельность компании на стратегическом и процессном уровне, рассмотреть бизнес - модель компании. Затем необходимо идентифицировать места использования этих знаний на этапах выполнения бизнес-процессов, провести анализ разрыва между существующими и необходимыми знаниями, на основе которого выбирается СУЗ».

Таким образом, для процесса внедрения системы управления знаниями в компании необходимо пройти множество шагов.

4. Заключение

Процесс управления ядерными знаниями не отличается от процесса управления знаниями в целом, основные концепции и подходы остаются такими же. Но потеря или искажение ядерных знаний может привести к риску возникновения аварии на атомной станции, поэтому безопасность играет важную роль.

С каждым годом все больше и больше компаний атомной промышленности занимаются вопросом управления знаниями и внедряют системы корпоративных знаний в свою деятельность. В России госкорпорация «Росатом» активно продвигает использование систем управления знаниями и занимает лидирующую позицию. Каждый год проводится международная конференция Rosatom Knowledge Management, что, безусловно, содействует развитию данного вопроса.

Библиографический список

[Вильгельм Е.В., 2013] Вильгельм Е.В. Современные тенденции в управлении знаниями на российских предприятиях энергетики – Журнал «Системное управление». – № 4 (22), 2013. – Электрон. дан. / Режим доступа: <http://sisupr.mrsu.ru/2013-4/PDF/vilgelm.pdf>, свободный

[Косилов А., 2010] Косилов А. Управление знаниями: основные понятия и подходы, Декабрь 2010. Электрон. дан. / Режим доступа: <http://www.bs.u.by/main.aspx?guid=97471>, свободный

[Лещенко В.О., 2014] Лещенко В.О. Система управления знаниями в атомной отрасли [Электронный ресурс] / ЗАО "Наука и инновации", Февраль 2014 – Режим доступа: <http://www.atomic-energy.ru/video/48100>, свободный.

[Лазарев С.В., 2009] Лазарев С.В. Внедрение систем управления знаниями в энергетике – Журнал «Энергетик». – №2, 2009. – Электрон. дан. / Режим доступа: <http://www.transform.ru/articles/html/10it/it000015.article>, свободный

[Першукова В. А., 2012] Першукова В. А. «Росатом» делится знаниями: Knowledge management в высокотехнологичных компаниях // Москва, 2012.— 152 с.

[Yanev Y., 2009] Yanev Y. Managing Nuclear Knowledge. Retrieved 2009, from <http://www.irpa.net/members/54302/%7B994C880E-7AE0-4A55A074BE9BEC722AA%7D/RC%208%20Managing%20Nuclear%20Knowledge%20PPT.pdf>

[Comparative Analysis, 2011] Comparative Analysis of Methods and Tools for Nuclear Knowledge Preservation - Nuclear Energy Series No.NG-T-6.7 STI/PUB/1494. Retrieved 2011, from http://www-pub.iaea.org/MTCD/publications/PDF/Pub1494_web.pdf

NUCLEAR KNOWLEDGE MANAGEMENT

Kostebelova V.K.

Saint Petersburg State Polytechnical University

St. Petersburg, Russia

kostebelovavk@gmail.com

The paper investigates the link between knowledge management practices and organizational performance in nuclear power plants. The research provides insights into the mechanisms with the help of which knowledge management practices have an impact on organizational effectiveness. Furthermore, it gives examples of current best practices concerning the knowledge creation, preservation and sharing in the nuclear industry.

Introduction

The nuclear sector is one of the most important parts of the energy industry because it can provide a considerable amount of electricity at low costs. But insufficient attention to circulating knowledge can be dangerous and lead to devastating accidents, such as Chernobyl and Fukushima. In this light, nuclear companies should not only know about existence of knowledge management, but also implement useful mechanisms and methods.

Main Part

Given relevant Russian and foreign practices, knowledge management can be divided into two approaches: the traditional approach and the so called 'innovative' approach. The traditional approach implies preservation and transfer of knowledge to new employees. Innovative approach involves the use of collective knowledge of organizations and commercialization of knowledge.

Conclusion

The process of nuclear knowledge management is no different from the process of knowledge management as such; the basic concepts and approaches are the same. But the loss or distortion of nuclear knowledge can increase the risk of having accidents.

Every year, more and more companies analyse nuclear knowledge management systems and implement corporate knowledge into their activities. In Russia, 'Rosatom' is actively promoting the use of knowledge management systems and holds a leading position in this respect.



OSTIS-2015

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822: 004.912

СЦЕНАРНЫЙ ПОДХОД ПРИ ИССЛЕДОВАНИИ ДИНАМИКИ ИНФОРМАЦИОННЫХ ПОТОКОВ В СЕТИ ИНТЕРНЕТ

Додонов А.Г., Ландэ Д.В., Бойченко А.В.

*Институт проблем регистрации информации НАН Украины,
г. Киев, Украина*

dodonov@ipri.kiev.ua, dwlande@gmail.com, boychenko.a@gmail.com

Рассмотрен сценарный подход к исследованию динамики информационных потоков при анализе и прогнозировании социальных явлений. Определены и подробно описаны этапы информационно-аналитического исследования на основе анализа контента сети Интернет.

Ключевые слова: сценарный подход; динамика информационных потоков; контент-мониторинг; аналитические исследования.

Введение

Сценарии или сценарные модели является одним из видов логико-лингвистических моделей, предназначенных для отображения развернутых во времени последовательностей взаимосвязанных событий, операций или процессов. Сценарии могут иметь структуры, в которых установлены условия перехода к той или иной частной стратегии, или просто отражены возможные альтернативы без указания условий. Требование взаимосвязанности в сценарных моделях не является строгим и носит достаточно условный характер, так как устанавливается на основе субъективных суждений экспертов, а также определяется спецификой формулировки целей деятельности.

Понятие сценарной модели более широкое, чем понятие алгоритма. При этом сценарная модель накладывает менее суровые ограничения на характер причинно-следственных отношений.

Типовой сценарий аналитической деятельности состоит из следующих элементов:

- краткое описание сути обрабатываемой в сценарии функциональной задачи;
- ресурсы (участники сценария, базы данных, знаний, программы);
- операции, выполняемые в ходе сценария;
- этапы выполнения функциональной задачи;
- шаги сценария (включая временные требования);
- содержание действий участников на каждом шагу сценария;
- описание условий выполнения сценария;
- описание ограничений на выполнение;

- ссылки на экранные формы, иллюстрирующие содержание действий;
- ссылки на фрагменты документации, поясняющие содержание шага сценария;
- комментарии.

Сценарий аналитической деятельности можно представить в виде кортежа (S, P, R, E, Z) , где:

- S – множество состояний системы,
- P – множество параметров,
- R – множество ресурсов,
- E – множество переходов между состояниями,
- $Z \subseteq S$ – множество результатов выполнения сценария.

Для построения сценария аналитик, используя средства реализации сценариев (обычно, АРМ некоторого моделирующего комплекса), выполняет обход графа, описывая действия, предписанные его дугам и, таким образом, решает задачу формирования последовательностей взаимодействия между участниками.

В настоящее время контент сети Интернет образует значимый динамический сегмент информационного пространства, информационные потоки, содержание и объемы которых необходимо учитывать при проведении аналитических исследований практически в любой области. Основным объектом анализа при этом являются событийные или тематические срезы этих потоков – массивы информационных сообщений, документов, соответствующих определенным событиям или тематикам. Динамика информационных потоков определяется комплексом как внутренних, так и

внешних нелинейных механизмов, которые отражаются, возможно, в неявном виде. Зачастую удовлетворительным оказывается упрощенное понимание информационного сюжета как некоторой зависимой от времени величины $n(t)$, поведение которой описывается нелинейными уравнениями [Додонов и др., 2009]. Таким информационным потокам можно ставить в соответствие временные ряды, для анализа которых все чаще обоснованно применяются формальные методы: статистического, фрактального, Фурье или вейвлет-анализа.

Постановка задачи

Для эффективного проведения информационно-аналитических исследований на основе анализа контента сети Интернет (а точнее ее веб-ресурсов) авторами предлагается последовательность шагов, этапов обработки информации, каждый из которых сам по себе обеспечивает получение продукта. Совокупность таких этапов, базирующихся на использовании необходимых и доступных инструментальных средств, специальных приемов, можно рассматривать как методику, процедуру проведения действий, нацеленных на получение аналитических материалов, которые могут использоваться для поддержки принятия решений.

Любая методика рассчитана на решение конкретных задач. При проведении информационно-аналитических исследований на базе интернет-контента к таким задачам можно отнести:

1. Нахождение релевантных публикаций по тематике.
2. Определение динамики тематических публикаций.
3. Определение критических точек в динамике тематических публикаций.
4. Выявление объектов мониторинга.
5. Выявление и визуализация взаимосвязей событий и объектов мониторинга, а также объектов мониторинга между собой.
6. Прогнозирование развития событий.

Этапы информационно-аналитического исследования

В соответствии с приведенными выше задачами предлагается выделить следующие этапы информационно-аналитического исследования:

1. Выбор системы интеграции интернет-документов.
2. Формирование запроса в среде выбранной системы. Нахождение тематических публикаций по запросу с помощью систем контент-мониторинга.
3. Определение динамики тематических публикаций по запросу.
4. Определение критических точек в динамике тематических публикаций.

5. Определение основных событий в критических точках.
6. Выявление объектов мониторинга.
7. Выявление и визуализация взаимосвязей.
8. Прогноз развития событий.

Рассмотрим эти этапы подробно.

Выбор системы интеграции интернет-документов

Для получения репрезентативной информации об объекте исследования необходимо воспользоваться поисковой системой с аналитическими функциями, охватывающей достаточный объем информации, относящейся к исследуемому объекту/событию. Для анализа динамики информационных потоков необходимо каким-то образом получить соответствующую статистику, представленную в виде временных рядов. Многие современные информационно-аналитические системы содержат в своем составе средства отображения статистики вхождения в базы данных понятий, соответствующих пользовательским запросам. В настоящее время существует несколько открытых информационных сервисов, в рамках которых можно наблюдать временную динамику объемов публикаций по тематикам, определяемым запросами. Так Google books Ngram Viewer (<http://ngrams.googlelabs.com/>), предоставляет визуализацию динамики количества книг, в которых упоминаются слова. Сервис «Яндекс пульс блогосферы» (<http://blogs.yandex.ru/pulse/>) также позволял отображать динамику публикаций в блогах, содержащих заданные пользователем ключевые слова, однако был закрыт ввиду малой посещаемости. Сегодня этот сервис доступен лишь по специальному разрешению компании «Яндекс». На сайте Национального корпуса русского языка (НКРЯ) в бета-режиме запущен сервис N-грамм (<http://www.ruscorpora.ru/ngram.html>), близкий по функциональности сервису Google books Ngram Viewer.

Безусловно, самыми эффективными для решения задач анализа динамического контента являются специализированные системы интеграции сетевого контента. В частности, в рамках исследований авторами использовалась система контент-мониторинга веб-ресурсов InfoStream (www.infostream.ua), реализующая необходимую функциональность и охватывающая около 100 тыс. документов в сутки с 7000 веб-сайтов.

Формирование запроса в среде выбранной системы

Массив тематических документов (тема – события, связанные с Евромайданом в Киеве 2013-2014 гг.) выбирается с помощью системы InfoStream путем ввода запроса на языке данной системы:

(майдан|евромайдан)&(избиен|разгон|штурм|беркут|молотов|титишк|погиб)&lang.RUS,

по которому в период с ноября 2013 года по март 2014 года было опубликовано свыше 200 тысяч *тематических публикаций* на веб-сайтах (рис. 1).

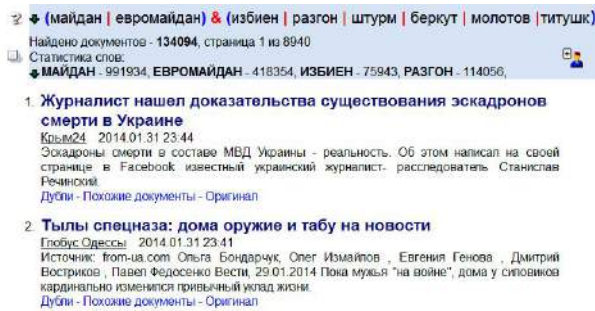


Рисунок 1 – Фрагмент поискового интерфейса системы InfoStream

Система InfoStream обеспечивает поиск, а также просмотр списка и полных текстов релевантных документов.

Определение динамики тематических публикаций по запросу

Режим динамики событий системы интеграции интернет-ресурсов позволяет получить данные о количестве публикаций по заданному запросу за указанный промежуток времени. Эти данные могут быть загружены в настольную систему обработки данных и отображаются в виде графика (рис. 2). В интерфейсе пользователя обеспечивается переход к просмотру релевантных документов по выбранной дате. После этого данные временной динамики за каждые сутки нормируются, т.е. в среде системы Excel формируется временной ряд, содержащий относительные значения, равные отношению количества тематических сообщений к общему потоку сообщений за сутки.

Это, в частности, позволяет избавиться от недельной периодичности в количестве тематических публикаций. Затем происходит переход к процедуре определения критических точек в данном временном ряду.

Определение критических точек в динамике сообщений

Критические точки как локальные максимумы временного ряда динамики публикаций можно определить, например, визуально по графику, представленному на рис. 2. Вместе с тем, существуют несколько научно-обоснованных методик, базирующихся на методах цифровой обработки сигналов.

Кривая цикла информационных операций

В результате анализа многочисленных диаграмм поведения ТИП, были выявлены наиболее типичные, базовые профили их поведения [Ландэ, 2013], [Додонов и др., 2013]. Предложенные модели

полностью соответствуют реальным данным, которые экстрагируются системами контент-мониторинга. Поэтому приведенные зависимости могут быть использованы как шаблоны, например, для выявления информационных операций – как путем анализа ретроспективного фонда сетевых публикаций, так и для оперативного мониторинга появления некоторых их признаков в реальном времени.

Понятия в динамике :

• (майдан | евромайдан) & (избиен | разгон | штурм | беркут | молотов | титишк)



Рисунок 2 – Режим «Динамика событий» системы интеграции

В частности, для выявления информационных операций [Горбулин и др., 2009] следует внимательно следить за динамикой публикаций по целевой теме и, если есть возможность, пользоваться доступными аналитическими средствами, средствами цифровой обработки данных и распознавания образов, например, вейвлет-анализом.

На рисунке 3 приведен обобщенная диаграмма, соответствующая всем этапам жизненного цикла информационных операций, обоснованная авторами в [Ландэ, 2008].

Для выявления степени «близости» фрагментов исследуемого временного ряда приведенной диаграмме в различных масштабах предлагается использовать так называемый «вейвлет-анализ» [Астафьева, 1996], который в настоящее время нашел широкое применение как в естественных науках, так и в социологии [Давыдов, 2008].

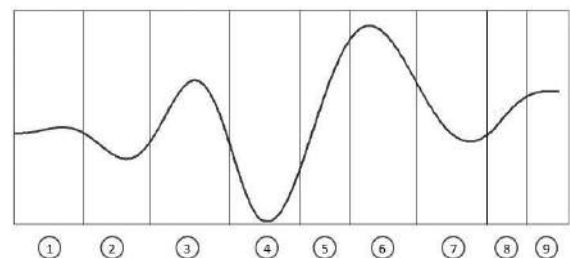


Рисунок 3 – Жизненный цикл информационных операций: 1 – фон; 2 – затишье; 3 – «артподготовка»; 4 – затишье; 5 – атака/триггер роста; 6 – пик завышенных ожиданий; 7 – утрата иллюзий организаторов; 8 – общественное осознание; 9 – продуктивность/фон

Главная идея вейвлет-преобразования заключается в том, что нестационарный временной ряд разделяется на отдельные промежутки (так называемые «окна наблюдения»), и на каждом из них вычисляется величина, показывающая степень

близости закономерностей исследуемых данных с разными сдвигами некоторого вейвлета (специальной функции) в разных масштабах. Вейвлет-преобразование генерирует набор коэффициентов, которые являются функциями двух переменных: времени и частоты, и потому образуют поверхность в трехмерном пространстве.

Непрерывное вейвлет-преобразование для функции $f(t)$ строится с помощью непрерывных масштабных преобразований и переносов выбранного вейвлета $\psi(t)$ с произвольными значениями масштабного коэффициента a и параметра сдвига b :

$$W(a,b) = (f(t), \psi(t)) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} f(t) \psi^* \left(\frac{t-b}{a} \right) dt.$$

Полученные вейвлет-коэффициенты можно представить в графическом виде, если по одной оси отложить сдвиг вейвлета (ось времени), а по другой – масштабы (ось масштабов), и раскрасить точки полученной схемы в зависимости от величины соответствующих коэффициентов (чем больше коэффициент, тем ярче цвета).

Эти коэффициенты, которые показывают, насколько поведение процесса в данной точке аналогично вейвлету в данном масштабе. Чем ближе от анализируемой зависимости в окрестности данной точки к виду вейвлета, тем большую абсолютную величину имеет соответствующий коэффициент. Применение этих операций, с учетом свойства локальности вейвлета в частотно-временной области, позволяет анализировать данные на разных масштабах и точно определять положение их характерных особенностей во времени.

На скейлограмме видны все характерные особенности исходного ряда: масштаб и интенсивность периодических изменений, направление и значение трендов, наличие, расположение и продолжительность локальных особенностей.

В работе [Ландэ, 2013] показано, что вейвлеты «мексиканская шляпа» и Морле (рис. 4) наиболее точно отражает динамику информационных операций, результаты применения этого вейвлета приведены на рис. 5, благодаря чему были выбраны три даты (2013.11.30, 2014.01.22, 2014.02.19), соответствующие критическим точкам исследуемого процесса.

Следует отметить, что инструменты построения вейвлет-спектограмм доступны как в ряде пакетов математических программ, например, в Matlab, так и через Интернет в режиме онлайн (<http://ion.researchsystems.com/cgi-bin/ion-p?page=wavelet.ion>).

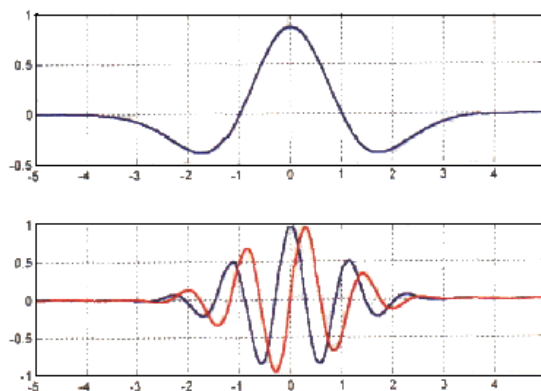


Рисунок 4 - Вейвлеты mexh, Морле

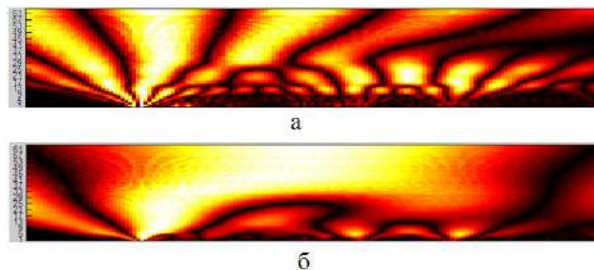


Рисунок 5 – Вейвлет-спектрограммы исследуемого временного ряда (а – «мексиканская шляпа», б – вейвлет Морле)

Определение основных событий в критических точках

После определения критических точек с помощью системы контент-мониторинга выполняется построение основных сюжетных цепочек из сообщений, соответствующих запросу за выбранные даты. Таким образом определяются основные события за указанные даты (рисунок 6).

Для последующего анализа отбирается три массива сообщений, соответствующие трем выбранным датам, объекты из которых (в простейшем случае – персоны и веб-источники) могут рассматриваться как объекты мониторинга.

Выявление объектов мониторинга

С помощью методов экстрагирования фактических данных, применяющихся в системах интеграции интернет-ресурсов, в интерфейсе пользователя формируются так называемые «информационные портреты», охватывающие списки персон, топонимов, языков, компаний и т.п., содержащиеся в документах, релевантных некоторому заданному запросу.

В нашем случае из «информационного портрета», соответствующего тематическому запросу выбираются наиболее упоминаемые персоны и/или веб-ресурсы за выбранные даты (рис. 7 и 8). Эти списки могут агрегироваться, в результате чего возможно определение взаимосвязей событий и объектов (рис. 9).

В качестве системы визуализации авторами выбрана система анализа и отображения сетей Gephi (www.gephi.org).

2013.11.30 Разгон демонстрантов на Майдане

Азаров считает разгон демонстрантов на Майдане в Киеве провокацией

Премьер-министр Украины Николай Азаров считает разгон демонстрантов на Майдане Незападности в Киеве провокацией и обещает, что ситуация будет тщательно образом расследована. Об этом УНИАН сообщил пресс-секретарь премьер-министра Виталий Лыщенко. "Планируется проведение расследования в том, что необходимо провести в краткие сроки тщательное и объективное расследование, и для этого создана оперативно-

2013.11.30 14:52 Пятеро участников Евромайдана госпитализированы из Шевченковского райотдела милиции Zhas.info

238

2013.11.30 23:53 Янукович приказал Генпрокуратуре наказать виновных в разгоне Евромайдана korabelnaja.info

2014.01.22 Штурм на ул. Грушевского

В центре Киева стягивают бронетехнику

В центре Киева сосредотачиваются крупные силы боевых внутренних войск МВД. Известно, что к стадиону "Динамо", где собралась протестующая бригада БТР. Значительное количество силовиков стоят рядом, прикрываясь щитами, перегородив улицу Грушевского. Передают "Интерфакс-Украина". 22 января в Киеве произошла очередная столкновения радикальной оппозиции с милицией.

2014.01.22 13:11 "Беркут" разогнал протестующих на Грушевского в центре Киева драки Газард

479

2014.01.22 23:58 В Киеве объявлена эвакуация Гляй-Полс

2014.02.19 Штурм правительственного портала

Кровавая ночь в Киеве: сможет ли Янукович удержать власть?

Ситуацию на Украине в интервью ИА "Медиафакс" оценивают ведущие украинские эксперты. ПОЧЕМУ УКРАИНА НЕ ИЗРАЙЛЕТ? Минувшей ночью в столице Киева вспыхнувшая драма перешла в трагедию: в бою между силовиками и сторонниками Майдана погибли по меньшей мере 36 человек, из которых 26 - активисты оппозиции, а 11 - милиционеры.

2014.02.19 14:51 PR и оппозиция готовы провести экстренное заседание парламента НОВОСТИ Виртулет

543

2014.02.19 23:59 Украина на краю пропасти и в трауре Ежаквекис

Рисунок 6 – Основные сюжетные цепочки за выбранные даты

Информационный портрет	
Уточнить запрос	
Рубрики	(26)
Языки	(1)
Страны источников	(20)
Источники	(50)
Размер	(3)
Цифровая насыщенность	(3)
Тональность	(2)
География	(50)
Персоны	(50)
AND	NOT
<input type="checkbox"/> Янукович	<input type="checkbox"/>
<input type="checkbox"/> Кличко	<input type="checkbox"/>
<input type="checkbox"/> Нигояна	<input type="checkbox"/>
<input type="checkbox"/> Луценко	<input type="checkbox"/>
<input type="checkbox"/> Яценюк	<input type="checkbox"/>
<input type="checkbox"/> Тягнибок	<input type="checkbox"/>
<input type="checkbox"/> Азаров	<input type="checkbox"/>
<input type="checkbox"/> Вербицкий	<input type="checkbox"/>
<input type="checkbox"/> Жизневский	<input type="checkbox"/>
<input type="checkbox"/> Мусий	<input type="checkbox"/>
<input type="checkbox"/> Ключев	<input type="checkbox"/>
<input type="checkbox"/> Лукаш	<input type="checkbox"/>

Рисунок 7 - Списки «доминирующих» персон

Эти же данные позволяют выявлять взаимосвязи между объектами, например, между указанными аналитиками веб-ресурсами и персонами (рис. 10).

На рис. 9 можно видеть, что каждому массиву (узлы, идентифицированные датами) соответствуют объекты. При этом в центральной части сети располагаются объекты, общие для нескольких событий (О-зона), а «гребешки» на периферии соответствуют специальным объектам, отражающим специфику конкретных событий (С-зоны).

Также можно предложить критерий релевантности события, связанного с конкретной датой, общей тематике: чем большая часть объектов из него попадает в О-зону, тем он более релевантен тематике. Формально значение этого критерия $k_{i,N}$

для сюжета i тематики s может быть записано следующим образом:

$$k_{i,N} = \frac{T_{i,N} \cap T_{s,N}}{N},$$

где N – количество объектов, $T_{i,N}$ – множество значимых объектов события i , $T_{s,N}$ – множество значимых объектов для всей тематики.

Информационный портрет	
Уточнить запрос	
Рубрики	(26)
Языки	(1)
Страны источников	(20)
Источники	(50)
AND	NOT
<input type="checkbox"/> ГигаМир.net	<input type="checkbox"/>
<input type="checkbox"/> Сегодня.ua	<input type="checkbox"/>
<input type="checkbox"/> Цензор.Нет	<input type="checkbox"/>
<input type="checkbox"/> ICTV Факти	<input type="checkbox"/>
<input type="checkbox"/> Независимое бюро новостей	<input type="checkbox"/>
<input type="checkbox"/> Полемика	<input type="checkbox"/>
<input type="checkbox"/> Телеканал новин "24"	<input type="checkbox"/>
<input type="checkbox"/> Gazeta.ua	<input type="checkbox"/>
<input type="checkbox"/> Новый Регион	<input type="checkbox"/>
<input type="checkbox"/> Главком	<input type="checkbox"/>
<input type="checkbox"/> Обзоратель	<input type="checkbox"/>
<input type="checkbox"/> 112.ua	<input type="checkbox"/>
<input type="checkbox"/> Вести.ua	<input type="checkbox"/>
<input type="checkbox"/> SiteUA	<input type="checkbox"/>
<input type="checkbox"/> Главред	<input type="checkbox"/>
<input type="checkbox"/> ЛІГАБізнесінформ	<input type="checkbox"/>
<input type="checkbox"/> УРА-Інформ	<input type="checkbox"/>
<input type="checkbox"/> Известия в Украине	<input type="checkbox"/>

Рисунок 8 - Списки «доминирующих» веб-ресурсов



Рисунок 9 - Пример визуализации связей событий и объектов



Рисунок 10 - Визуализация взаимосвязей веб-ресурсов и персон

Подход к прогнозу: R/S-анализ

Для решения задач прогнозирования перспективным представляется применение теории фракталов при анализе информационного пространства. Фрактальный анализ самоподобия информационных массивов может рассматриваться

как технология, предназначенная для осуществления аналитических исследований с элементами прогнозирования, пригодная к экстраполяции полученных зависимостей.

Важнейшей характеристикой рядов, которые имеют хаотичное поведение, является, как известно, показатель Херста (H), определяемый в результате так называемого R/S -анализа [Федер, 1991]. Этот показатель базируется на анализе нормированного разброса – отношения разброса значений исследуемого ряда R к стандартному отклонению S .

Достаточно часто, когда соотношение R/S

$$R/S = (N/2)^H,$$

имеет постоянный тренд, можно говорить о соотношении: где H – показатель Херста, который для достаточно широкого класса рядов связан с хаусдорфовой (фрактальной) размерностью D постоянной формулой: $D+H=2$.

На рис. 11 представлено соотношения R/S для рассматриваемого в этой работе временного ряда. Как можно видеть, кривая нормированного размаха удовлетворительно аппроксимируется прямой в двойном логарифмическом масштабе. Численные значения H характеризуют разные типы коррелированной динамики (персистентности). При $H=0,5$ наблюдается некоррелированное поведение значений ряда, а значение $0,5 < H < 1$ соответствует степени автокорреляции ряда.

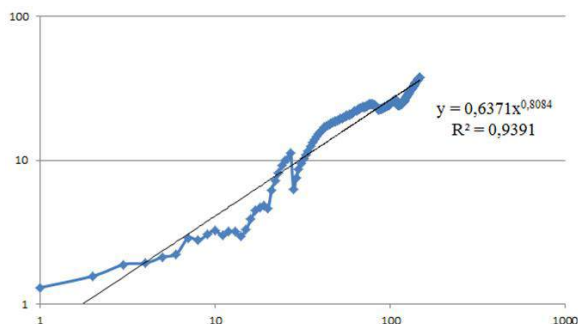


Рисунок 11 - Кривая R/S в двойной логарифмической шкале

Как можно видеть, значение показателя Херста для исследуемого информационного потока соответствует величине 0,81, что подтверждает предположение о самоподобии и итеративности рассматриваемых процессов в информационном пространстве. Это означает, что общая информационная напряженность остается на большом уровне, как только исчезает «шлейф» одного сюжета по выбранной тематике информационных, ему на смену возникает новый сюжет, т. е. его поведение в дальнейшем будет близко к предшествующему поведению.

Заключение

В докладе представлена методика аналитического исследования, которая базируется

на сценарном подходе и использовании современных инструментальных средств анализа и визуализации информационных потоков и временных рядов. Как показывает опыт, применение сценарного подхода при исследовании контента и структуры информационных ресурсов сети Интернет позволяет существенно повысить эффективность аналитической деятельности.

Предложенная методика, по мнению авторов, позволяет решить сформулированную задачу, ее можно использовать в качестве основы для проведения аналитической и прогнозной деятельности на основе исследования контента современных компьютерных сетей. В дальнейшем на основе полученных результатов планируется разработка моделирующего комплекса.

Библиографический список

- [Астафьева, 1996] Астафьева Н.М. Вейвлет-анализ: основы теории и примеры применения // Успехи физических наук, 1996. – 166. – № 11. – Р. 1145-1170.
- [Горбулин и др., 2009] Горбулин В.П., Додонов О.Г., Ланде Д.В. Інформаційні операції та безпека суспільства: загрози, протидія, моделювання: монографія. – К.: Інтертехнологія, 2009. – 164 с.
- [Давыдов, 2008] Давыдов А.А. Системная социология. – М.: Издательство ЛКИ, 2008. – 192 с.
- [Додонов и др., 2009] Додонов О.Г., Ланде Д.В., Путятин В.Г. Інформаційні потоки в глобальних комп'ютерних мережах – К: Наукова думка, 2009, – 295 с.
- [Додонов и др., 2013] Додонов А.Г., Ланде Д.В., Прищеп В.В., Путятин В.Г. Конкурентная разведка в компьютерных сетях. – К.: ИПРИ НАН Украины, 2013. – 248 с.
- [Ланде, 2008] Ланде Д.В. Новітні підходи й технології інформаційно-аналітичної підтримки прийняття рішень // Національна безпека: український вимір: шокв. наук. зб. / Рада нац. безпеки і оборони України, Ін-т пробл. нац. Безпеки, 2008. – Вип. 1-2 (20-21). – С. 87-105.
- [Ланде, 2013] Ланде Д.В. Тренди відображення інформаційних операцій в інформаційному просторі // Інформація і право, 2013. – N 1 (7). – С. 82-88.
- [Федер, 1991] Федер Е. Фракталы. – М., Мир, 1991. – 261 с.

A SCENARIO APPROACH IN THE RESEARCH OF DYNAMICS OF INFORMATION STREAMS IN THE INTERNET

Dodonov A.G., Lande D.V., Boychenko A.V.

Institute for Information Recording NAS of Ukraine, Kiev, Ukraine

dodonov@ipri.kiev.ua, dwlande@gmail.com

boychenko.a@gmail.com

A scenario approach to the study of the dynamics of information flows in the analysis and prediction of social phenomena are considered. Identified and described in detail the stages of information-analytical study based on the analysis of content on the Internet.

Keywords: scenario approach; dynamics of information streams; content-monitoring; analytical research.



OSTIS-2015

(Open Semantic Technologies for Intelligent Systems)

УДК 004.96

ПОДХОД К ФОРМИРОВАНИЮ СЛОЖНОГО ИНЖЕНЕРНОГО РАСЧЕТА НА ПОРТАЛАХ

Новогрудская Р.Л., Глоба Л.С.

*Национальный технический университет Украины «Киевский политехнический институт»,
г. Киев, Украина*

lgloba@its.kpi.ua

rinan@ukr.net

В работе представлен подход к проектированию расчетных задач на инженерных порталах знаний, основанный на алгебре расчетов, которая позволяет получить требуемый общий расчет как результат выполнения определенной системы алгебраических формализмов. Алгебра расчетов описывает последовательность выполнения частных расчетов, при этом формализмы алгебры расчетов портала знаний представляются множеством операций заданного вида на хранимом наборе информационных и вычислительных ресурсов портала.

Ключевые слова: портал знаний, алгебра расчетов, расчетная задача, функциональный элемент.

ВСТУПЛЕНИЕ

Интеграция данных и вычислительных ресурсов осуществляется с помощью порталных решений, которые содержат ссылки на территориально-распределенные ресурсы, позволяя не только получить доступ к информационным ресурсам, хранящимся в их базах, но и, используя данные, знания и сервисы, реализовать определённые вычислительные задачи предметной области. Когда количество ссылок на различные информационно-вычислительные ресурсы становится достаточно большим, то организация эффективного управления информационно-вычислительными процессами становится практически невозможным. Кроме того, ряд информационных и вычислительных ресурсов могут динамически менять свое месторасположение и не всегда является возможным централизованно управлять этим процессом, в связи с этим возникает проблема эффективной организации информационно-вычислительных ресурсов на порталах, что позволило бы реализовать решение наукоемких и интеллектуальных задач в различных сферах деятельности, а в инженерной деятельности это становится особенно важным.

1. Введение в проблему

В последнее время при построении порталов научных знаний все чаще используются

онтологические модели. Такой подход к построению порталов знаний освещен в частности в работах Загорюлько Ю. А., Боровиковой А. И. Гавриловой Т. А., Хорошевского В. Ф. и др [John, 2010], [Bobbilo, 2013], [Guarino, 2006], [Movshovitz-Attias, 2012], [Загорюлько, 2008], [Гаврилова, 2000], [Колб, 2012]. В качестве примера можно предложить порталы знаний в области археологии и компьютерной лингвистики [Загорюлько, 2011] при построении которых в качестве модели представления знаний использовалась онтологическая модель [Глоба, 2012]. Онтология позволяет объекты порталов и их информационное окружение, структурировать и систематизировать информацию, представленную на порталах. Но, когда речь идет об инженерных порталах знаний с большим количеством различных расчетных задач нерешенными остаются

– задача качественной интеграции расчетных задач в информационную структуру портала,

– задача установления взаимосвязи между предметной областью (элементами онтологии знаний и онтологии предметной области) и непосредственно сервисами, с помощью которых выполняются вычислительные инженерные задачи,

Важное место в построении портала имеет интеграция расчетных задач, которых существует огромное множество и которые занимают важное место при проведении исследований в той предметной области, которая рассматривается.

Существует необходимость задать корректную реализацию этих расчетных задач в информационной среде портала.

Сложность структуры портала обеспечивается вложенностью и соподчиненностью расчетных задач, из чего следует наличие на порталах сложных функциональных элементов.

В результате выполнения полного расчета можно полностью охарактеризовать объект исследования и определить подходит он или нет для использования при создании определенной детали необходимой проектировщику конструкции. Таким образом необходимо не только описать доступны пользователю на портале расчетные задачи, но и установить связи между конкретными сервисами (набором сервисов) для реализации конкретных рабочих процессов расчетных задач в рамках совместных рабочих процессов портала, а также установить связь сервисов с информационными ресурсами что необходимо для решения производственных задач пользователя. Важным является решение задачи оптимизации этих связей.

Также, как было определено, при построении реальных научных и инженерных задач используется их декомпозиция на подзадачи, что в зависимости от направленности общей задачи и параметров, используемых при расчете, могут компоноваться в общую задачу динамично и одна подзадача может использоваться в нескольких общих задачах. Исходя из этого необходимо учесть и описать логику связности расчетных задач.

Опишем общие проблемы, возникающие при проектировании порталов знаний:

- описание элементов инженерного портала знаний,
- определение механизма связывания информационных ресурсов с вычислительными,
- описание характеристик вычислительных и информационных ресурсов в зависимости от предметной области, которая рассматривается,
- описание механизма определения результата выполнения вычислительного ресурса,
- описание последовательности взаимодействия workflow, формирующие инженерные расчетные задачи.

Решение таких проблем влечет за собой ряд задач:

- построение модели инженерного портала знаний,
- разработка формального аппарата, позволяющего оперировать элементами портала в процессе проектирования и решение инженерных задач портала,

- разработка подхода к формированию последовательности элементов частных расчетов для их объединения в общий расчет.

Построение модели портала знаний позволит систематизировать и структурировать как информационные, так и вычислительные ресурсы портала. Разработка формального аппарата – формальной алгебраической модели портала – позволит задать формальные контекстно-независимые алгебраические структуры для описания элементов портала и операций над ними. Подход к динамическому формированию общей расчетной задачи, позволит, используя формальную алгебраическую модель портала реализовать динамическое объединение частичных расчетов для формирования общего комплексного инженерного расчета.

2. Формальная алгебраическая модель инженерного портала

Рассмотрим алгебру расчетов, предназначенную для динамического получения частичных расчетов при построении общих расчетов за счет выполнения определенной системы алгебраических формализмов, которая формирует последовательность их обработки. Формальная алгебра расчетов на портале знаний представлена множеством операций заданного вида на заданном наборе его информационных и вычислительных ресурсов.

Расчетом C_k является система $\langle R_k A_k \rangle$, где R_k - отношение со схемой $R_k = (R_1, R_2, \dots, R_n)$, A_k – логическая формула правила.

Правило расчета можно представить любой логической формулой, в которой:

Переменные – имена атрибутов $N_1 N_2, \dots, N_n$,

Константы – элементы соответственных множеств $M_1 M, \dots, M_n$,

Предикатные символы – символы отношений $\langle \rangle \leq \geq = \neq$,

Символы логический связок – символы логический операций $\wedge, \vee, -$

Пусть **схему отношений** R_k можно записать как $R_k = (R_1, R_2, \dots, R_n)$, где множества можно разделить на три подмножества согласно выделенным типам метаописаний (стандартные метаописания, специализированные метаописания, метаописания, устанавливающие связь) [Теленик, 1999].

Выделим следующие базовые элементы алгебры расчетов:

Объекты. Объектами являются основные элементы алгебры, над которыми проводятся все операции алгебры. Для алгебры расчетов - это фактические расчеты. Определим множество объектов как множество расчетов.

Операции. В алгебре расчетов выделяют два типа операций простые и сложные:

Элементарные операции, которые являются представлением обычных алгебраических операций по определению, это: Добавление +, которое обозначает процесс суммирования значений элементов схем отношений, или метаописаний; Умножение *, которое обозначает поиск произведения значений элементов схем отношений или метаописаний.

Множественные операции [Kenneth, 2007]: \subseteq - нестрогое включение, \subset - строгое включение, \in - принадлежность, \notin - не принадлежность;

Логические операции – это аналоги теоретико-множественных операций алгебры отношений [Глоба, 2014]. Среди *логических операций* выделяют унарные и *n*-арные [Павлов, 2002]:

Определение 1. Пусть дано расчеты $C_1 = \langle R_1 A_1 \rangle$ и $C_2 = \langle R_2 A_2 \rangle$, где $R_1 = R_2$. Тогда:

Расчет $C_3 = \langle R_3 A_3 \rangle$ – объединение расчетов C_1 и C_2 ($C_3 = C_1 \cup C_2$), если C_3 содержит все кортежи из C_1 и все кортежи из C_2 , которые не совпадают ни с одним из кортежей из C_1 , причем C_3 не содержит никаких других кортежей.

Расчет $C_3 = \langle R_3 A_3 \rangle$ – пересечение расчетов C_1 и C_2 ($C_3 = C_1 \cap C_2$), если C_3 содержит те, и только те кортежи, которые содержатся и в C_1 и C_2 , при чем C_3 не содержит никаких других кортежей.

Расчет $C_3 = \langle R_3 A_3 \rangle$ – разность расчетов C_1 и C_2 ($C_3 = C_1 \setminus C_2$), если C_3 содержит те, и только те кортежи, которые являются кортежами C_1 и не содержатся в кортеже C_2 , при чем C_3 не содержит никаких других кортежей.

Расчет $C_3 = \langle R_3 A_3 \rangle$ называется композицией расчетов C_1 и C_2 ($C_3 = C_1 \circ C_2$), если существует расчет C_4 , такой что содержит некоторые кортежи из C_1 и C_2 , тогда расчет C_3 содержит те, и только те кортежи из C_1 , которые есть в C_4 и те кортежи из C_2 , которые содержатся в C_4 , при чем C_3 не содержит никаких других кортежей.

Расчет $C_3 = \langle R_3 A_3 \rangle$ называется эквивалентным расчетом C_1 ($C_3 \sim C_1$), если C_3 содержит те, и только те кортежи, которые являются кортежами C_1 , и количество кортежей в C_1 и C_3 совпадает, причем C_3 не содержит никаких других кортежей.

Расчет $C_3 = \langle R_3 A_3 \rangle$ называется дополнением расчета C_1 , ($C_3 = C_1^d$) к расчету C_1^y , который содержит все кортежи схемы R_1 системы, если C_3 содержит те, и только те кортежи, которые принадлежат C_1^y и не принадлежат C_1 . Дополнение является унарной операцией над расчетами.

Сложные операции.

Сложные операции состоят из множества объектов и простых операций: операцию последовательного соединения, операцию

параллельного соединения, операцию логического объединения, операцию инверсии [Глоба, 2014].

Определение 2. Расчет $C_3 = \langle R_3 A_3 \rangle$ – есть результатом операции параллельного соединения расчетов C_1 и C_2 ($C_3 = C_1 \exists_{\text{пр}} C_2$), если $x_i \in C_1$ и $x_j \in C_2$, то C_3 исчисляется как $(x_i \vee x_j)$.

Определение 3. Расчет $C_3 = \langle R_3 A_3 \rangle$ – является результатом операции последовательного соединения расчетов C_1 и C_2 ($C_3 = C_1 \exists_{\text{пс}} C_2$), если $x_i \in C_1$ и $x_j \in C_2$, то C_3 исчисляется как $(x_i \wedge x_j)$.

Операция логического объединения – $O_{\text{лн}}$ позволяет сочетать расчеты, связь между которыми задана через третий элемент. Операция логического сочетания позволяет сочетать два расчета при известном третьем расчете на базе операции композиции.

Определение 4 Расчет $C_3 = \langle R_3 A_3 \rangle$ называется инверсией расчета C_1 , ($C_3 = C_1^{-1}$), если C_3 содержит те, и только те кортежи, которые содержатся в C_1 в обратном порядке.

Данные. Элементы, которые подаются на вход и на выход системы. В качестве качества данных в алгебре расчетов рассматривают разнообразные константы, переменные, а также результат операций над ними, на физическом уровне данными могут быть значения разнообразных параметров или характеристик, формулы, границы значений, а также частичные расчеты. Все виды данных определены как носители данных [Шаховская, 2008].

3. Метод формирования сложного функционального элемента

Основной задачей при проектировании последовательности сложного функционального элемента на инженерном портале знаний является объединение соответствующих частичных функциональных элементов для включения их в сложный функциональный элемент.

Множество частичных функциональных элементов – Φ_e , хранится в независимом хранилище, которое не имеет связей с хранилищем сложных функциональных элементов – $s\Phi_e$, (Рис. 1). Структура сложных и частичных Φ_e включает название и параметры Φ_e .

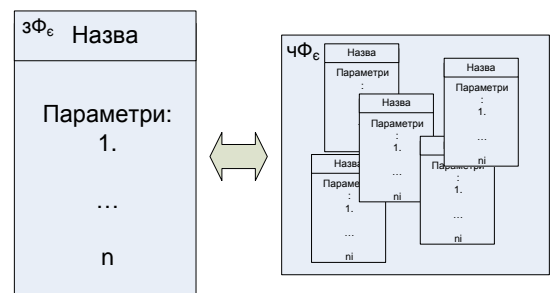


Рисунок 1 - Множество сложных и частичных Φ_e

Поиск частичных Φ_3 , которые необходимо включить в сложный может базироваться на сравнении параметров Φ_3 , поскольку в сложный Φ_e включаются только те частичные, у которых есть общие параметры, или области значений соответствующих параметров пересекаются.

Пусть множество сложных Φ_3 :

$$R^3 \ni r_k^3, r_k^3 = \langle T_k^3, p_{kj}^3 \rangle,$$

где

R^3 – множество сложных Φ_3 ,

r_k^3 – k -й Φ_3 из множества сложных Φ_3 ,

T_k^3 – название k -ого Φ_3 из множества сложных Φ_3 ,

p_{kj}^3 – j -й параметр k -ого Φ_3 из множества сложных Φ_3 ,

Множество частичных Φ_3 :

$$R^4 \ni r_l^4, r_l^4 = \langle T_l^4, p_{lq}^4 \rangle$$

R^4 – множество частичных Φ_3 ,

r_l^4 – l -й Φ_3 из множества частичных Φ_3 ,

T_l^4 – название l -ого Φ_3 из множества частичных Φ_3 ,

p_{lq}^4 – q -й параметр l -ого Φ_3 из множества частичных Φ_3 ,

Множества P_{kj}^3 и P_{lq}^4 могут пересекаться или не пересекаться, что отображает наличие у обоих Φ_3 одинаковых параметров или их отсутствие соответственно. Используя введенные обозначения передоложим метод, который обеспечит формирование сложного Φ_3 из частичных Φ_3 на базе сравнения значений параметров, которые используются в Φ_3 , который предоставит возможность динамически устанавливать порядок проведения расчетов представленных соответствующими Φ_3 . Предлагается процесс формирования сложного Φ_e представить с помощью дерева, что упростит процедуру модификации Φ_3 . Исходя из того, что полученное дерево является упорядоченным, то есть – деревом с корневым узлом и заданным порядком прохождения дочерних узлов, следует, что использования упорядоченных деревьев для представления сложного Φ_3 позволит определить последовательность выполнения частичных Φ_3 динамически.

Рассмотрим метод формирования дерева сложного Φ_3 (Рис. 2).

Опишем этапы метода:

Этап 1. На первом этапе происходит отбрасывание из множества R^4 расчеты, в которых ни один p_{kj}^3 не равняется никакому p_{lq}^4 .

Этап 2. На этом этапе происходит сравнение значений p_{lq}^4 расчетов отобранных на этапе 1, для отсека подмножества расчетов, которые имеют общие параметры, но множества их значений не пересекаются.

Этап 3. На этапе 3 проводится упрощение формулы частичного Φ_3 .

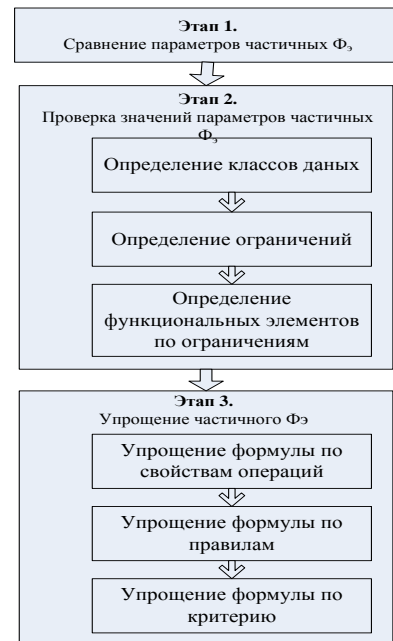


Рисунок 2 – Метод формирования дерева сложного Φ_3 ,

Предлагается выделить конечные частичные Φ_3 и промежуточные частичные Φ_3 . Конечные частичные Φ_3 – $чк\Phi_3$ есть неделимыми частичными Φ_3 , они являются листьями дерева, промежуточные частичные Φ_3 – $чп\Phi_3$ это частичные Φ_3 , которые в свою очередь состоят еще из некоторого множества частичных Φ_3 и отражены в дереве всеми вершинами кроме корневой и листовых.

4. Пример использования метода формирования сложного функционального элемента

Предположим, задано общий функциональный элемент «Расчет на прочность силовых элементов магнитных систем ИТЕР» - $з\Phi_1$ с структурой описания:

$$r_1^3 = \langle T_1^3, p_{1j}^3 \rangle, j = 1, n$$

где

T_1^3 – «Расчет на прочность силовых элементов магнитных систем ИТЕР»

$p_{11}^3 - a_1 = 10$ – наружное давление

$p_{12}^3 - E_2 = 52$ – усредненная напряженность

$p_{13}^3 - E_1 = 48$ – базовая напряженность

$p_{14}^3 - R = 5$ – радиус обмотки

$p_{15}^3 - c = 10$ – длина связи корпуса с обмоткой

$p_{16}^3 - l = 15$ – критическая длина

$p_{17}^3 - D_m = \langle 10 \dots 12 \rangle$ – максимальный наружный диаметр

$p_{18}^3 - Pi = 18$ – размерность корпуса

$p_{19}^3 - \eta = 20$ – критическое давление

$p_{110}^3 - \phi = 5$ – допустимое давление напряженности

$p_{111}^3 - v = 28$ – длина обмотки

$p_{112}^3 - d = 44$ – прочность обмотки
 $p_{113}^3 - e = 6$ – константа соответствующая уровню статичности
 $p_{114}^3 - M = 12$ – приведенное общее мембранное напряжение

На первом этапе анализируем множество частичных ФЭ, выбираем те, параметры которых совпадают с параметрами з ФЭ1 за правилом:

$$P_1^3 = \bigcup_i^m P_i^3$$

При чем, возможно $P_i^3 \cap P_j^3$, где $i, j = \overline{1, m}$

Сначала рассмотрим множество промежуточных частичных расчетов. Из них условию удовлетворяют «Расчет основных параметров» и «Проверочный расчет»:

Структура «Расчета основных параметров»:

$$r_1^{пч} = \langle T_1^{пч}, p_{1j}^{пч} \rangle, j = 1, n$$

где

$T_1^{пч}$ – «Расчет основных параметров»

$$\begin{array}{ll}
 p_{11}^{пч} - d & p_{13}^{пч} - M \\
 p_{12}^{пч} - e &
 \end{array}$$

Структура «Проверочного расчета»:

$$r_2^{пч} = \langle T_2^{пч}, p_{2j}^{пч} \rangle, j = 1, n$$

где

$T_2^{пч}$ – «Проверочный расчет»

$$\begin{array}{ll}
 p_{21}^{пч} - a1 & p_{27}^{пч} - D_m \\
 p_{22}^{пч} - E_2 & p_{28}^{пч} - P_i \\
 p_{23}^{пч} - E_1 & p_{29}^{пч} - \eta \\
 p_{24}^{пч} - R & p_{210}^{пч} - \phi \\
 p_{25}^{пч} - c & p_{211}^{пч} - v \\
 p_{26}^{пч} - l &
 \end{array}$$

Далее необходимо проанализировать частичные расчеты, найдено три расчета, удовлетворяющих условию:

«Расчет на статическую прочность» и его структура:

$$r_1^4 = \langle T_1^4, p_1^4 \rangle$$

T_1^4 – «Расчет на статическую прочность»

$$\begin{array}{ll}
 p_{11}^4 - a1 & p_{15}^4 - c \\
 p_{12}^4 - E_2 & p_{16}^4 - l \\
 p_{13}^4 - E_1 & p_{17}^4 - D_m \\
 p_{14}^4 - R &
 \end{array}$$

«Расчет на критическую постоянную» и его структура:

$$r_2^4 = \langle T_2^4, p_2^4 \rangle$$

T_2^4 – «Расчет на критическую постоянную»

$$\begin{array}{ll}
 p_{21}^4 - a1 & p_{25}^4 - c \\
 p_{22}^4 - E_2 & p_{26}^4 - l \\
 p_{23}^4 - E_1 & p_{27}^4 - D_m \\
 p_{24}^4 - R &
 \end{array}$$

«Расчет на стойкость» и его структура:

$$r_3^4 = \langle T_3^4, p_3^4 \rangle$$

T_3^4 – «Расчет на стойкость»

$$\begin{array}{ll}
 p_{31}^3 - P_i & p_{33}^3 - \phi \\
 p_{32}^3 - \eta & p_{34}^3 - v
 \end{array}$$

В результате получаем подмножество множества чФЭ, которое удовлетворяет условию и сформированное дерево зФЭ, вершинами второго уровня которого являются эти расчеты:

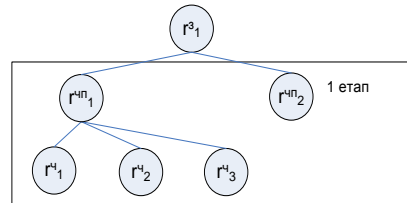


Рисунок 4 – Дерево зФЭ этап 1

На втором этапе необходимо сравнить значения тех параметров частичных ФЭ, которые совпадают:

Значение $(p_{ik}^3) \diamond$ Значение $(p_{jt}^4) \diamond$ Значение (p_{lp}^4) ,

при условии, что $p_{ik}^3 = p_{jt}^4 = p_{lp}^4$.

Рассмотрим параметр D_m отобранных на этапе 1 расчетов: 1. для расчета на прочность силовых элементов магнитных систем ИТЕР $D_m = \langle 10, \dots, 12 \rangle$, для расчета на статическую прочность $D_m = \langle 0, \dots, 15 \rangle$, для расчета на критическую постоянную $D_m = \langle 15, \dots, 52 \rangle$.

Сравниваем:

Значение $(p_{17}^3) \diamond$ Значение $(p_{17}^4) \diamond$ Значение (p_{27}^4)

Область значения параметра 7 расчета на прочность силовых элементов магнитных систем ИТЕР включает область значения параметра 7 расчета на статическую прочность и не включает область значения параметра 7 расчета на критическую постоянную:

$$Z(p_{17}^3) \subset Z(p_{17}^4), Z(p_{17}^3) \neq Z(p_{27}^4)$$

Это свидетельствует про сужение подмножества частичных ФЭ, полученного на 1-ом этапе с исключением из него ФЭ на критическую постоянную. Построенное дерево зФЭ после этапа 2 представлено на Рис. 5

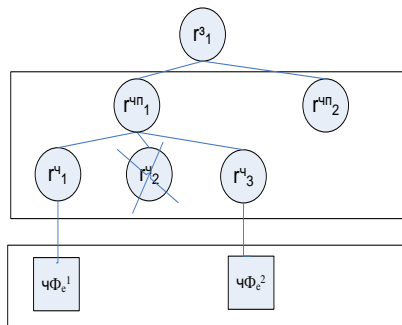


Рисунок 5 – Дерево 3ФЭ этап 2

Рассмотренный пример доказал возможность использования предложенного в разделе 3 метода для формирования законченных комплексных инженерных расчетов на порталах знаний.

Выводы

В статье представлен подход к структуризации и систематизации информационных и вычислительных ресурсов на инженерных порталах знаний. Предложена формальная алгебраическая модель портала знаний, которая представлена множеством операций заданного вида на заданном наборе информационных и вычислительных ресурсов портала, позволяющая задать формальные контекстно-независимые структуры для описания элементов портала знаний.

Предложен метод динамического формирования комплексных инженерных расчетов, который позволяет динамически устанавливать структуру сложных функциональных элементов, повышать эффективность процесса их выполнения за счет параллельной обработки независимых друг от друга ветвей дерева сложного функционального элемента, а также сочетать частичные расчетные задачи инженерного портала в общий расчет, который выполняется по запросу конечного пользователя.

Представлено применение предложенного подхода на примере реального инженерного расчета, что доказывает его эффективность.

СПИСОК ЛИТЕРАТУРЫ

- [John, 2010] John F. Sowa The role of logic and ontology in language and reasoning, Chapter 11 of Theory and Applications of Ontology: Philosophical Perspectives, edited by R. Poli & J. Seibt, Berlin: Springer, 2010, pp. 231-263.
- [Bobillo, 2013] F. Bobillo, U. Straccia. Aggregation Operators for Fuzzy Ontologies. Applied Soft Computing 13(9):3816-3830, 2013
- [Guarino, 2006] Guarino N., Guizzardi G. In Defense of Ontological Foundations for Conceptual Modeling. In: Scandinavian Journal of Information Systems, vol. 18(1) pp. 115 - 126. 2006.
- [Movshovitz-Attias, 2012] D. Movshovitz-Attias and W.W. Cohen. 2012. Boot-strapping biomedical ontologies for scientific text using nell. Technical report, Carnegie Mellon University, CMU-ML-12-101.
- [Загоруйко, 2008] Загоруйко Ю. А. и др. Подход к построению порталов научных знаний // Автотметрия. 2008. Т. 44. № 1. С. 100—110.

[Гаврилова, 2000] Гаврилова Т. А., Хорошевский В. Ф. Базы знаний интеллектуальных систем. Учебник. — СПб.: Питер, 2000

[Колб, 2012] А. Колб, Д.Г. Web-ориентированная реализация семантических моделей интеллектуальных систем / Д.Г. Колб // Открытые семантические технологии проектирования интеллектуальных систем (OSTIS-2012): материалы Междунар. науч.-техн. конф., Минск, 16–18 февр. 2012 г. / Белорус. гос. ун-т информатики и радиоэлектроники; редкол.: В.В. Голенков (отв. ред.) [и др.]. – Минск, 2012. – С. 111–122.

[Загоруйко, 2011] Загоруйко Ю. А. Технология построения порталов научных знаний: опыт применения, проблемы и перспективы. материалы 21-й Международной Крымской конференции «СВЧ-техника и телекоммуникационные технологии» (Крымико' 2011) / Загоруйко Ю. А. - Севастополь: Вебер, 2011. – С.51-54.

[Глоба, 2012] Глоба Л.С., Новогрудская Р.Л. Модели и методы интеграции информационных и вычислительных ресурсов // Труды второй международной научно-технической конференции «Открытые семантические технологии проектирования интеллектуальных систем» OSTIS-2012. Минск, 2012

[Теленик, 1999] Теленик С.Ф., Логика представления вычислительных процессов в интеллектуальной системе SmartBase // Системные технологии. Системное моделирование технологических процессов: Сб. науч. Тр. – 1999. – Вып. 6. – С.131-139.

[Kenneth, 2007] Kenneth H. Rosen. Discrete Mathematics: And Its Applications. McGraw-Hill College. ISBN 978-0-07-288008-3. (2007)

[Глоба, 2014] Глоба Л.С., Новогрудская Р.Л., Проектирование инженерных расчетов для порталов знаний, Материалы III Международной научно-технической конференции «Open Semantic Technologies for Intelligent Systems (OSTIS - 2014)» 137-143сс.

[Павлов, 2002] Павлов А.А., Теленик С.Ф. Информационные технологии и алгоритмизация в управлении. – К.:Техніка, 2002.–344с.

[Глоба, 2014] Л.С. Глоба, Р.Л. Новогрудская, Подход к построению формальной алгебраической системы порталов знаний, Онтология проектирования. - 2014. - №2(11). – ISSN 2223-9537- С. 40-59

[Шаховська, 2008] Шаховська Н. Б. Простір даних області наукових досліджень // Моделювання та інформаційні технології. – 2008. – № 45. – С. 132–140

APPROACHES OF THE FORMATION OF COMPLEX ENGINEERING CALCULATIONS ON THE PORTAL

Novogradskaya R.L., Globa L.S.

National thechnical university of ukraine «Kyiv Politecnick Institute», Kyiv, Ukraine

lgloba@its.kpi.ua

rinan@ukr.net

The paper presents an approach to the design of computational tasks in engineering knowledge portal based on the algebra of calculation tasks, which allows you to obtain the required total calculation tasks as a result of the algebraic formalism system. Algebra of calculation tasks includes the sequence of partial tasks using formalisms, the set of the operations, the sets of data and objects.



OSTIS-2015

(Open Semantic Technologies for Intelligent Systems)

УДК 004.82

МЕТАГРАФЫ КАК ОСНОВА ДЛЯ ПРЕДСТАВЛЕНИЯ И ИСПОЛЬЗОВАНИЯ БАЗ НЕЧЕТКИХ ЗНАНИЙ

Глоба Л. С., Терновой М. Ю., Штогрин Е. С.

*Национальный технический университет Украины «Киевский политехнический институт»,
Институт телекоммуникационных систем,*

г. Киев, Украина

lgloba@its.kpi.ua

ternovoy@its.kpi.ua

L_Shtogrina@mail.ru

В работе рассмотрены проблемы представления и обработки нечетких знаний в базах нечетких знаний. Определен метаграф для представления базы нечетких знаний Мамдани и рассмотрены его свойства. Описан подход к представлению баз нечетких знаний в виде метаграфа. Приведена постановка задачи формирования метаграфа из неупорядоченной базы нечетких знаний Мамдани, а также постановка задачи перенумерации вершин метаграфа и предложено их решение.

Ключевые слова: база нечетких знаний, лингвистические переменные, метаграф, метавершина.

Введение

Качество и эффективность работы экспертных систем, которые основываются на базах нечетких знаний (БНЗ) зависит от форм представления БНЗ и методов проведения нечеткого логического вывода. Большинство существующих систем для работы с экспертными знаниями предполагают хранения знаний в виде неупорядоченного набора правил вида «Если - То», но это дополнительно усложняет процесс вывода, поскольку требует полного перебора правил [Субботин, 2008; Деменков, 2005]. В работах [Штовба, 2007; Глоба и др., 2008] предложены методы, позволяющие повысить эффективность нечеткого логического вывода, а также описываются условия, при выполнении которых можно ускорить процесс вывода. Однако предложенные методы рассчитаны на работу только с древовидными БНЗ. Как следствие, повышение эффективности работы с БНЗ, которые не являются древовидными и определение их особенностей является актуальной задачей.

В работе [Basu et al., 2010] предложен новый вид графического представления – метаграфы и описано применение метаграфов для логического вывода. В работе [Zheng-Hua, 2006] предложено использовать метаграфы для представления нечетких знаний. В работе [Штогрин и др., 2014] предложен метод графического представления метаграфов, который

может быть использован для изображения метаграфов, представляющих БНЗ. Графическое представление БНЗ позволяет наглядно проследить связи между лингвистическими переменными, находить и исправлять противоречия, а также определять другие свойства БНЗ за меньшее время, по сравнению с анализом правил, записанных в текстовом виде. В данной работе предложены модификация и расширение данного подхода для представления и обработки знаний, которые представлены в базах нечетких знаний Мамдани.

1. Базы нечетких знаний и метаграфы. Основные определения

Для того чтобы рассмотреть свойства БНЗ, а также предложить дальнейшее их определение с помощью метаграфа выпишем определение БНЗ и метаграфа. Отметим, что в дальнейшем изложении в данной работе под БНЗ подразумеваются именно базы нечетких знаний Мамдани. Также считается, что БНЗ сформирована для определения одной лингвистической переменной (ЛП), которую будем называть результирующей. Лингвистические переменные, значение которых определяется с помощью фаззификации или сразу задается в качественном виде будем называть входными. Для входных переменных не существует правил их определяющих.

Определение 1. База нечетких знаний Мамдани – это модель представления знаний, которая состоит из правил вида:

$$P_g = (P_l^z)_j = \text{ЕСЛИ} (X_{j_1} = t_{j_1}^{k_1}) \text{И} (X_{j_2} = t_{j_2}^{k_2}) \text{И} \dots \text{И} (X_{j_{n_j}} = t_{j_{n_j}}^{k_{n_j}}) \text{ТО} (X_l = t_l^z) \quad (1)$$

где $(P_l^z)_j$ – j -е правило для определения z -го термина ЛП с идентификатором l , в котором X_{j_s} – лингвистическая переменная, которая оценивается качественным термом $t_{j_s}^{k_s}$; $s = \overline{1, n_j}$ номер ЛП в левой части j -го правила, n_j – количество ЛП, которые находятся в левой части j -го правила.

Для сокращения записи введем обозначения для левой и правой части правила P_g :

$$P_g = (P_g^A, P_g^C),$$

где $P_g^A = \{t_{j_s}^{k_s} \mid s = \overline{1, n_j}\}$ – множество термов, которые входят в левую часть правила; $P_g^C = t_l^z$ – терм, который является результатом правила.

Определение 2. Частичная база нечетких знаний – это БНЗ для определения одной лингвистической переменной, включающая в себя правила, результатами которых являются термы данной ЛП. При этом такую ЛП будем называть ЛП верхнего уровня, а лингвистические переменные, входящие в левые части правил, будем называть ЛП нижнего уровня.

Определение 3. Метаграф – это тройка множеств $S = \langle V, M, E \rangle$, где $V = \{v_r \mid r = \overline{1, N_v}\}$ – порождающее множество (множество вершин метаграфа), $M = \{m_q \mid q = \overline{1, N_m}\}$ – множество метавершин, $E = \{e_h \mid h = \overline{1, N_e}\}$ – множество ребер, где N_v – количество вершин метаграфа, N_m – количество метавершин, N_e – общее количество ребер в метаграфе.

Метавершина метаграфа – определяется как множество вершин $m_q = \{v_r \mid v_r \in V, r = \overline{1, N_{m_q}}\}$, где N_{m_q} – количество вершин, входящих в метавершину m_q . Если две или больше метавершин соответствуют одному и тому же множеству вершин, то такие вершины считаются одинаковыми и рассматривается только одна из таких метавершин.

Для упрощения дальнейшей подачи материала введем понятие узла метаграфа $mv \in (V \cup M)$, таким образом, это вершина или метавершина.

Тогда ребро метаграфа определяется как $e_h = (mv_{out}, mv_{in})$, где mv_{out} – узел, из которого выходит ребро, mv_{in} – узел, в который входит ребро.

Определение 4. Будем называть метаграфом, представляющим БНЗ Мамдани, такой метаграф $S' = \langle V, M, E \rangle$, который удовлетворяет следующим условиям:

1. Каждая вершина $v(t_i^k)$ метаграфа S' соответствует терму ЛП t_i^k .

2. Каждая метавершина $m_g = \{mv_s \mid s = \overline{1, n_g}\}$ метаграфа S' соответствует P_g^A – левой части правила P_g . Метавершина m_g содержит в себе вершины, которые взаимно-однозначно соответствуют термам, которые входят в P_g^A .

3. Каждое ребро e_g метаграфа S' , соответствует правилу P_g БНЗ. Ребро $e_g = (m_g, v(t_l^z))$ выходит из метавершины m_g , и входит в вершину $v(t_l^z)$, которая соответствует терму t_l^z в P_g^C .

Исходя из данного определения, метаграф S' , который представляет БНЗ Мамдани не может содержать:

1. Ребер, которые выходят из вершины $\forall v_r : \text{deg}_{out}(v_r) = 0$.

2. Ребер, которые входят в метавершину $\forall m_g : \text{deg}_{in}(m_g) = 0$.

Также у него можно выделить следующие свойства:

1. Для любой метавершины существует хотя бы одно ребро, которое соединяет ее с какой-то вершиной $\forall m_g : \exists e_h = (m_g, v_r)$.

2. Вершины, которые не включены ни в одну метавершину $\exists v_{rez} = \{v_r \mid \forall q : v_r \notin m_q\}$.

3. Вершины, соответствующие термам одной лингвистической переменной или все не имеют входящих ребер $\forall v_r : \text{deg}_{in}(v_r) = 0$, или все имеют хотя бы одно входящее ребро $\forall v_r : \text{deg}_{in}(v_r) > 0$ одновременно.

2. Подход к представлению и использованию баз нечетких знаний в виде метаграфа

На основе представленных выше определений базы нечетких знаний Мамдани и метаграфа ниже

предложен поход к представлению и использованию БНЗ в виде метаграфа. Данный подход включает метод формирования метаграфа, метод нумерации узлов метаграфа и метод нечеткого логического вывода на основе метаграфа, представляющего БНЗ Мамдани.

Ниже сформулирована постановка задачи формирования метаграфа.

Дано:

$$\text{БНЗ} = \bigcup_{l,z,j} (P_l^z)_j \quad - \quad \text{неупорядоченная БНЗ}$$

Мамдани.

Найти:

Метаграф $S' = \langle V, M, E \rangle$, который представляет БНЗ.

Решение:

Метод формирования метаграфа, представляющего БНЗ состоит в определении отображения F , которое позволяет представить термы в виде вершин, а правила в виде метавершин и ребер метаграфа (рис. 1).



Рисунок 1 – Метод формирования метаграфа из БНЗ

В результате такого представления БНЗ все правила, для определения одного терма ЛП будут сгруппированы и связаны с этим термом. Это позволит уйти от полного перебора правил при проведении нечеткого логического вывода. Также появится возможность определения ряда противоречий еще до начала процесса вывода.

Особенность предложенного метода состоит в том, что полученный в результате такого

отображения метаграф будет изоморфным к БНЗ. Изоморфность гарантирует возможность однозначного восстановления БНЗ из метаграфа. Для этого предлагается связать с вершиной дополнительную информацию про терм лингвистической переменной, которому данная вершина соответствует и название ЛП.

Для последующего использования метаграфа необходимо провести нумерацию его вершин. Постановка задача нумерации приведена ниже.

Дано:

Метаграф S' , который соответствует неупорядоченной БНЗ - $\text{БНЗ} = \bigcup_{z,l,j} (P_l^z)_j$.

Найти:

Метаграф S'' , который соответствует БНЗ, с введенной на множестве узлов нумерацией, соответствующей порядку использования правил при проведении нечеткого логического вывода.

Решение:

Метод нумерации узлов состоит в проведении топологической сортировки узлов метаграфа. Последовательность действий представлена на рисунок 2.

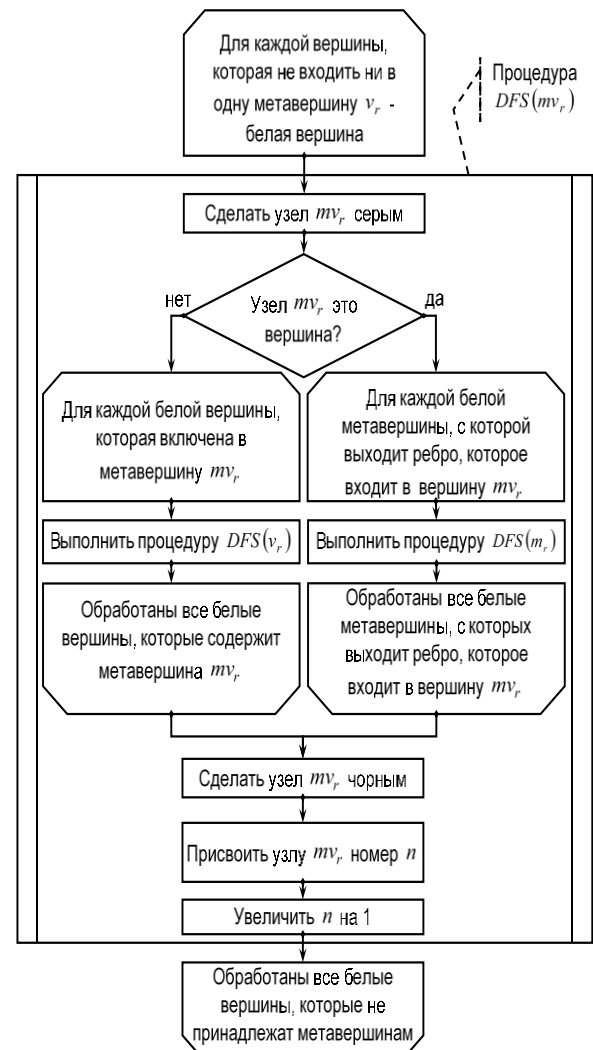


Рисунок 2 – Метод нумерации вершин метаграфа

Метод логического вывода на основе метаграфа с введенной нумерацией узлов базируется на обходе вершин и метавершин в порядке увеличения их номеров. Это позволяет сократить вычислительную сложность вывода.

Заключение

В работе предложен подход к представлению баз нечетких знаний в виде метаграфа. Рассмотрены особенности метаграфа, представляющего базу нечетких знаний Мамдани. Предложен подход к построению метаграфа на основе базы нечетких знаний, а также механизмы для повышения эффективности нечеткого логического вывода.

В дальнейшей работе планируется исследовать взаимосвязь свойств метаграфа и свойств баз нечетких знаний таких как полнота, непротиворечивость и избыточность. Это позволит проводить предварительную обработку баз нечетких знаний еще на этапе их построения.

Библиографический список

- [Субботин, 2008] Субботин С. О. Подання й обробка знань у системах штучного інтелекту та підтримки прийняття рішень / С. О. Субботин // Навчальний посібник. — Запоріжжя: ЗНТУ, 2008. — 341 с.
- [Деменков, 2005] Деменков Н. П. Нечеткое управление в технических системах / Н. П. Деменков // Учебное пособие. — М.: Изд. МГТУ им. Н.Э. Баумана, 2005. — 200 с.: ил.
- [Штовба, 2007] Штовба С. Д. Проектирование нечетких систем средствами Matlab / С. Д. Штовба // М.: Горячая линия – Телеком, 2007. — 288 с., ил.
- [Глоба и др., 2008] Глоба, Л. С. Створення баз нечітких знань для інтелектуальних систем управління / Л. С. Глоба, М. Ю. Терновой, О. С. Штогрин // Комп'ютинг. - Міжнародний науково-технічний журнал. - том 7, випуск 1. -Тернопіль, «Економічна думка» – 2008. – С.70-79.
- [Basu et al., 2010] Basu Amit Blanning Metagraphs and their applications / Amit Basu, W. Robert // Springer, 2010. 160 p.
- [Zheng-Hua, 2006] Zheng-Hua Tan Fuzzy Metagraph and Its Combination with the Indexing Approach in Rule-Based Systems / Tan Zheng-Hua // IEEE Transactions on Knowledge and Data Engineering. 2006. Vol. 18, No. 6. pp.829-841.
- [Штогрин и др, 2014] Штогрин Е.С. Метод визуализации метаграфа / Е.С. Штогрин, А.С. Кривенко / Научно-технический вестник информационных технологий, механики и оптики Санкт-Петербургского национального исследовательского Университета ИТМО, 2014, №3 (91) С. 124-130.

METAGRAPH BASED REPRESENTATION AND PROCESSING OF FUZZY KNOWLEDGBASES

Globa L.S., Ternovoy M.Y., Shtogrina O.S.

National Technical University of Ukraine “Kyiv Polytechnic Institute”, Kiev, Ukraine

lgloba@its.kpi.ua

ternovoy@its.kpi.ua

L_Shtogrina@mail.ru

The paper describes problems of fuzzy knowledge representation and processing in fuzzy knowledgebases. The metagraph for Mamdani fuzzy knowledgebase is defined. The method for metagraph based representation of fuzzy knowledgebase is described. The metagraph construction from Mamdani fuzzy knowledgebase and metagraph vertices numbering problems statement and their solutions are proposed.

Introduction

The efficiency of based on fuzzy knowledgebase expert systems depend on knowledgebase representation and inference mechanisms.

This paper describes approach for fuzzy knowledgebase representation and processing based on metagraph.

Main Part

The based on metagraph approach for fuzzy knowledgebase representation and processing consists of the method for metagraph construction, the method for numbering metagraph vertices and based on metagraph fuzzy inference.

The main idea of the method for metagraph construction form fuzzy knowledgebase is to define special mapping. This mapping allows one to represent linguistic variable terms as metagraph vertices and rules as metagraph metaverices and edges.

The method for metagraph vertices numbering is based on topological sorting.

The main idea of based on metagraph inference is to use vertices and metaverices in the order of increasing their numbers.

Conclusion

The approach for metagraph based representation of fuzzy knowledgebases is proposed in the paper. The properties of metagraph that representing Mamdani fuzzy knowledgebase are described. The methods for metagraph construction from fuzzy knowledgebases and for metagraph based fuzzy inference are proposed.



OSTIS-2015

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822:514

ДЕТЕКТИРОВАНИЕ ИЗМЕНЕНИЙ В XML-ПОДОБНЫХ СТРУКТУРАХ ДОКУМЕНТОВ

Глоба Л.С., Молчанов Ю.Н.

*Национальный технический университет Украины «Киевский политехнический институт»,
г. Киев, Украина*

lgloba@its.kpi.ua

molchanov@ukr.net

В работе представлен метод детектирования изменений в XML-подобных структурах документов, в основу которого положен метод генетического локального поиска для детектирования изменений, что обеспечивает удовлетворительное время детектирования изменений при сохранении достоверности полученных результатов. Предложенный метод обеспечивает удовлетворительную временную сложность решения задачи БЛП: при детектировании изменений в деревьях среднего размера (10-100 элементов) удалось значительно сократить минимальное время детектирования изменений по сравнению с точными методами.

Ключевые слова: детектирования изменений в XML-подобных структурах; древовидные структуры; метод генетического локального поиска; задача поиска хорошего соответствия.

Введение

В среде Internet в настоящее время распространена такая схема взаимодействия между публикующими информацию и потребителями информации, когда каждый человек, имеющий доступ к сети Internet, в любое время может выступать в роли как публикующего, так и потребителя. Эта ситуация приводит к постоянному неконтролируемому обновлению информации и проблемы отслеживания изменений на определенном ресурсе или по определенной тематике во всей сети Internet. В таких условиях от пользователя требуется постоянно обрабатывать большой объем данных только для поиска полезной информации. Вместе с тем, общий объем информации в сети Internet будет постоянно увеличиваться, поэтому пользователь вынужден постоянно увеличивать время для поиска необходимой информации.

Данная проблема решается в рамках разработки событийно-ориентированных систем публикации/подписки, обеспечивающих полное разделение во времени, пространстве и синхронизацию между тем, кто публикует, и подписчиком. В таких системах потребитель сообщает о своем интересе в определенной информации или изменениях на определенном ресурсе, и проблема отслеживания, детектирования и оповещения об изменениях возлагается на

соответствующие функции системы. Однако в настоящее время разработанные системы публикации/подписки характеризуются следующими недостатками [Chakravarthy et al., 2004], [Coruscio et al., 2002], [Chand et al., 2004]:

- недостаточной гибкостью в определении информационного объекта – невозможность определения отдельного объекта или группы объектов на веб-странице, на которой необходимо отслеживать изменения, недостаточное удобство представления информации пользователю, ограниченный набор значений параметров, который можно передать в запросе, возможность отслеживания только определенного типа изменений (например, невозможность определения вставки или обновления узлов в документе);
- некоторые системы не ориентированы на среду Internet и требуют заранее заданной информации о топологии системы;
- отсутствие поддержки структурированных источников информации (HTML, XML);
- необходимость в установке дополнительного программного обеспечения на стороне публикующего и потребителя;
- некорректное или неудобное отображение найденных изменений в документе и т.д.

Основным узлом веб-ориентированной системы публикации/подписки считается модуль отслеживания изменений в иерархически структурированной информации, входные данные

которого задают в виде XML-документа любой структуры, а алгоритм определения изменений работает в условиях неопределенности структуры (DTD, XML-schema) документов и должен поддерживать корректное определение добавления, удаления, обновления узла, а также изменение структуры, включая и изменение порядка следования узлов одного уровня, что не всегда реализовано в существующих системах публикации/подписки.

1. Анализ подходов к определению изменений в XML-документах

Система публикации/подписки требует сравнительно быстрого алгоритма детектирования изменений, обеспечивающего минимальную задержку определения и доставки уведомления об изменении определенной информации.

В настоящее время не существует эффективного механизма, который мог бы корректно определять и удобно представлять изменения в веб-страницах. Существующие решения, представленные в [changedetection.com, followthatpage.com, changedetect.com] основаны на детектировании изменений в текстовых файлах и некорректно отражают изменения в структуре документов.

Анализ существующих алгоритмов детектирования изменений в XML-документах позволил определить основные типы данных алгоритмов: алгоритмы, вычисляющие минимальную последовательность редактирования всех возможных редакционный скриптов [Chawathe, et al., 1997], [Chawathe, et al., 1998], [Алексеев и др., 2009], алгоритмы, основанные на хэш-подписях [Khan et al., 2003], [Jyoti et al., 2005], [Khandagale et al., 2010] и семантические алгоритмы [Flesca et al., 2007].

Основной задачей метода детектирования изменений является поиск соответствия между узлами двух древовидных структур - «хорошего соответствия», которое согласно работам [Chawathe, et al., 1997], [Chawathe, et al., 1998] предусматривает наличие такого соответствия между узлами древовидных структур, когда превращение старого дерева в новое требует минимального количества элементарных операций редактирования. Учитывая, что перед системой детектирования изменений ставится задача отображения произведенных изменений, а не задача преобразования старой древовидной структуры в новую структуру, то понятие «хорошего соответствия» было пересмотрено и сформулировано следующим образом. Хорошим соответствием между двумя древовидными структурами называется такое соответствие, когда суммарный критерий соответствия между всеми парами соответствующих узлов в двух деревьях является максимальным. Здесь под критерием соответствия понимается численная величина, показывающая степень соответствия двух узлов по критериям

сходства содержания, атрибутов и положения в иерархии дерева.

Существующие методы сравнения древовидных структур основываются на сравнении всех пар узлов двух деревьев на соответствующих местах в иерархии дерева. Если положение узлов совпадает, то проводится сравнение содержания узлов. Остальные узлы, которые не совпали, считаются добавленными или удаленными соответственно. К данным методам относится метод сравнения узлов на основе хэш-подписей, а также семантические методы. К недостаткам этих методов следует отнести рассмотрение только одного возможного соответствия между деревьями, когда структура дерева не меняется. Тогда как другие варианты соответствия, учитывающие все возможные операции редактирования над узлами деревьев, могли бы привести к более корректному результату. Данные методы имеют значительные ограничения и не подходят для решения задачи детектирования изменений в XML-документах.

При выборе математического аппарата для решения поставленной задачи был также проанализирован математический аппарат теории графов, так как древовидная структура в общем виде представляет собой связанный ациклический граф.

Можно выделить два основных подхода при сравнении графов - поиск максимального изоморфного пересечения и поиск минимальной последовательности редактирования.

Существующие методы поиска максимального изоморфного пересечения проводят поиск изоморфного пересечения, которое имеет максимальное количество вершин, определяющих меру сходства между графами. Основной особенностью методов поиска минимальной последовательности редактирования является предположение о наличии «хорошего соответствия» между древовидными структурами.

В существующих системах хранения структурированных документов как структура документа может изменяться, так и содержание в узлах дерева, причем как в случае изменившейся структуры, так и при той же самой структуре. Это требует решения о «хорошем соответствии» на основании критерия, который бы учитывал все возможные изменения в документах, а не предполагал изменение только содержания или только структуры.

Кроме того, два графа называются изоморфными, если существует взаимно-однозначное соответствие между их вершинами и ребрами, которое сохраняет смежность и инцидентность. Понятие изоморфности предусматривает наличие большего количества ограничений при поиске соответствия графов, чем поставленная задача поиска изменений между древовидными структурами, которая

предусматривает возможность изменения структуры дерева, удаление и добавление узла и т.п., исключает возможность рассмотрения соответствующих деревьев как изоморфных графов. Таким образом, существующие методы поиска максимального изоморфного пересечения не могут быть использованы для решения поставленной задачи.

Проведенный анализ показал, что задача поиска хорошего соответствия должна рассматриваться как задача поиска соответствия между связанными ациклическими графами, и на данный момент не существует метода сравнения таких структур. Это обусловлено тем, что при преобразовании древовидных структур не будут сохранены такие свойства, как смежность, инцидентность узлов и ребер, а также изоморфность графов, так как в рассматриваемой предметной области при преобразовании заданных деревьев могут быть изменены как отдельные узлы (содержание, атрибуты), так и вся структура дерева.

Рассмотренные алгоритмы имеют ряд недостатков:

- сосредоточены на производительности, временной сложности операций детектирования изменений, а также оптимизации полученных «дельт», но инструментарию и моделям для эффективного внедрения данных алгоритмов в системах публикации/подписки на практике уделяется мало внимания;
- не поддерживают отслеживание всех возможных типов изменений в структурированных документах, поэтому некорректно отражают результирующие изменения;
- не поддерживают отслеживание изменений в условиях неопределенности структуры (DTD, XML-schema) документов;
- обладают слишком большой временной сложностью для анализа XML документов большого объема.

Таким образом, можно сделать вывод, что задача поиска «хорошего соответствия» между древовидными структурами на данный момент требует разработки алгоритма детектирования изменений в древовидных структурах в условиях неопределенности структуры дерева. Такой алгоритм даст возможность разработки системы публикации/подписки, которая позволит уменьшить количество отсылаемой информации об изменениях в документах конечному пользователю, предоставляя ему только данные о проведенных изменениях и только после того, как они реально сделаны.

2. Метод определения изменений в древовидных структурах

Метод детектирования изменений между древовидными структурами состоит из следующих шагов:

1. Введение исходных данных в виде начальной и обновленной древовидных структур.

В задаче детектирования изменений в древовидных структурах на основе веб-страниц формируют первоначальную и обновленную древовидные структуры, которые далее рассматривают как входные объекты для задачи детектирования изменений [Flesca et al., 2007].

2. Индексирование древовидных структур, необходимое для учета порядка узлов слева направо в результирующем дереве и для отслеживания местоположения узла в иерархии дерева.

Индексирование проводится по правилам, приведенным в [Lim et al., 2001], и представляется в виде дополнительного атрибута в каждом узле древовидных структур.

3. Расчет критериев соответствия и поиск изменений проводится для узлов древовидных структур.

Данный этап необходим для исследования характеристики соответствия между двумя узлами и ее выражения через численные параметры.

4. Определение значимых для поиска изменений узлов.

На этом этапе происходит поиск значимых для пользователя узлов в полученной древовидной структуре, которые будут участвовать в алгоритме поиска изменений. К таким узлам можно отнести узлы дерева с текстовым содержанием, представляющим интерес для подписчика, они обычно представлены в виде листьев в результирующем дереве. Это обусловлено особенностями структуры веб-страниц, которые представлены на языке XML. Это определение не является ограничением для данного метода, так как его можно использовать и для внутренних узлов дерева. В рамках использования метода детектирования изменений для веб-страниц, такое допущение приемлемо.

5. Расчет параметров соответствия листьев древовидных структур, выбранных на предыдущем шаге.

В рассматриваемой математической модели, листья обоих деревьев сравнивают по критериям содержание, перечню атрибутов и положению узла в структуре дерева. Использование такой модели позволяет исследовать максимальное количество характеристик отдельного узла и находить наиболее точный показатель соответствия между узлами. Эти параметры будут рассчитываться попарно для всех выбранных узлов в старой и новой версии дерева.

6. Создание матрицы интегральных критериев соответствия для всех выбранных узлов в обеих версиях XML дерева.

На этом этапе завершается процесс исследования математической модели дерева и ее результатом является матрица интегральных критериев соответствия узлов дерева. Эта матрица показывает степень сходства между листьями деревьев в виде множества полученных значений.

7. Формулировка задачи линейного программирования на основе матрицы интегральных критериев соответствия.

На этом этапе происходит математическое моделирование задач поиска «хорошего соответствия» для листьев деревьев. Ограничениями этой задачи является необходимость нахождения целочисленного решения, поскольку параметр связности может иметь только два значения: 0 или 1. Результат решения этой задачи будет решением задачи хорошего соответствия начальной и обновленной древовидных структур. Если количество листьев в обоих деревьях не соответствует друг другу, то в процессе решения будет найдено, какие узлы были удалены или добавлены в новом дереве.

8. Решение задачи линейного программирования с помощью модифицированного метода генетического локального поиска.

В ходе решения задачи находят целочисленное решение, которое бы удовлетворяло всем поставленным условиям. Это решение позволит определить взаимное соответствие между узлами старой и новой версии.

9. Выполнение функции сравнения содержания в найденных соответствующих узлах на этапе решения задачи линейного программирования.

После нахождения пар соответствующих узлов в деревьях происходит сравнение текстового содержания в каждой паре. В данном случае можно использовать алгоритмы для сравнения неструктурированных текстов, которые представлены во многих работах, в том числе в [Myers, 1986]. Кроме сравнения текстового содержания соответствующих пар узлов, на этом этапе можно определить и изменение в структуре дерева относительно листьев дерева. Эти изменения могут соответствовать изменению местоположения узла с текстовым содержанием в обновленной древовидной структуре.

Приведенный метод детектирования изменений можно описать в виде следующей блок-схемы, представленной на рис. 1.



Рисунок 1 – Блок-схема метода детектирования изменений между древовидными структурами

3. Постановка задачи поиска хорошего соответствия между древовидными структурами

Сформулируем задачу определения «хорошего соответствия» в общем виде для древовидной структуры.

Дано:

$$\text{дерева } T_1 = T_1(a_i, n, R, AT, \text{con}(a_i)) \Big|_{i=1, \overline{n}}$$

$$\text{и } T_2 = T_2(a_j, n, R, AT, \text{con}(a_j)) \Big|_{j=1, \overline{m}},$$

где a_i – i -ый узел дерева,

n – количество узлов в дереве,

$R = \{r_i \mid i=1..m\}$ – совокупность родительских узлов в дереве T_1 , где m – количество родительских узлов, r_i – родительский узел i -го узла,

$AT = \{at_i \mid i=1..n\}$ – совокупность атрибутов узлов в дереве, at_i – атрибут i -го узла;

$\text{con}(a_i)$ – содержимое i -го узла.

$$\langle C, X \rangle = c_{11} \cdot x_{11} + c_{12} \cdot x_{12} + c_{13} \cdot x_{13} + c_{14} \cdot x_{14} + \\ + c_{21} \cdot x_{21} + c_{22} \cdot x_{22} + \dots + c_{(n-1)m} \cdot x_{(n-1)m} + c_{nm} \cdot x_{nm},$$

$$a_{11}x_{11} + a_{12}x_{12} + \dots + a_{1m}x_{1m} = b_1$$

...

$$a_{n1}x_{n1} + a_{n2}x_{n2} + \dots + a_{nm}x_{nm} = b_n$$

$$a_{(n+1)1}x_{11} + a_{35}x_{21} + \dots + a_{(n+1)(m+1)}x_{n1} \leq b_{n+1}$$

...

$$a_{(n+m)1}x_{1m} + \dots + a_{(n+m)(m+n)}x_{nm} \leq b_{n+n}$$

или в сокращенной записи $Ax_{ij} \leq B$, где матрица условий A и матрица ограничений B имеют значения:

$$A = \begin{pmatrix} 1 & 1 & 1 & \dots & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \dots & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & \dots & 1 & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 1 & \dots & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 & 0 & 1 \end{pmatrix}, B = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix},$$

где C – матрица интегральных критериев CS соответствия узлов, $C = \{c_{ij}\}_{n \times m}$,

$$\begin{aligned} c_{11} &= CS(A_1, B_1), c_{12} = CS(A_1, B_2), \\ c_{13} &= CS(A_1, B_3), c_{14} = CS(A_1, B_4), \\ c_{21} &= CS(A_2, B_1), c_{22} = CS(A_2, B_2), \dots, \\ c_{nm} &= CS(A_n, B_m), \end{aligned}$$

x_{ij} – параметр связности узлов A_i и B_j .

Найти: $\max_{x \in B^n: Ax \leq B} \langle C, X \rangle$.

При рассмотрении сложности решения поставленной задачи линейного программирования необходимо отметить, что данная задача предусматривает наличие решения только в булевых переменных, что требует решения задачи булева линейного программирования (БЛП). Эти задачи формулируются как задачи ЛП с дополнительным ограничением – принадлежности переменных множеству булевых переменных. В общем случае эта задача относится к задачам NP-сложности.

Для некоторых видов задач БЛП существуют точные методы решения: метод ветвей и границ, метод динамического программирования, метод Балаша; неточные: метод случайного поиска, локальной оптимизации, эвристические методы.

Вычислительная сложность имеющихся методов поиска решения задачи БЛП не удовлетворяет потребностей системы публикации/подписки из-за требования эффективного поиска соответствующих узлов для древовидных структур с большим количеством узлов, поэтому предложена модификация решения задачи БЛП с использованием генетических алгоритмов.

Генетические алгоритмы представляют собой итерационный процесс, конечной целью которого является получение оптимального решения.

Общую схему генетических алгоритмов проще понять, рассматривая задачу безусловной оптимизации. Которая формулируется как

найти:

$$\min \{F(x) \mid x \in X^n\}, X^n = \{0,1\}^n$$

Генетический алгоритм начинает работу по формированию начальной популяции x_0 – конечного набора допустимых решений задачи. Эти решения могут быть выбраны случайным образом, получены с помощью вероятностных «жадных» алгоритмов или другими методами. Выбор начальной популяции не имеет значения для сходимости процесса в асимптотике. Однако формирование «хорошей» начальной популяции (например, из множества локальных оптимумов) может заметно сократить время достижения глобального оптимума.

Основные операторы этого алгоритма: селекция, скрещивание и мутация.

Таким образом, с ненулевой вероятностью решение может перейти в любое другое решение, которое гарантирует в асимптотике получения точного решения задачи. Применение локального спуска позволяет генетическим алгоритмом сосредоточиться только на локальных оптимумах.

4. Модифицированный метод генетического локального поиска

Применим метод генетического локального поиска [Алексеева и др., 2007] для решения поставленной задачи.

Запишем матрицу C в следующем виде:

$$C = \begin{pmatrix} c_{11} & c_{12} & \dots & c_{1m} \\ c_{21} & c_{22} & \dots & c_{2m} \\ \dots & \dots & \dots & \dots \\ c_{k1} & c_{k2} & \dots & c_{km} \\ c_{n1} & c_{n2} & \dots & c_{nm} \end{pmatrix}.$$

На шаге 1 найдем допустимое решение для задачи булевого линейного программирования (БЛП). Для этого необходимо выполнить следующие действия.

Найти максимальное значение среди элементов матрицы C . Пусть максимальным элементом будет c_{k2} , тогда присвоим $x_{k2} = 1$. Из этого следует, что в матрице X :

$$x_{ki} = 0 \mid i \in [1, 2) \cup (2, n], x_{j2} = 0 \mid j \in [1, k) \cup (k, n].$$

Поскольку данные величины в матрице X найдены, то в матрице C необходимо уменьшить зону поиска максимальных значений, убрав элементы, которые не находятся во втором столбце и k -й строке. Эти действия необходимо повторять, пока не будут найдены все элементы X^k .

На основании указанных действий сформулируем алгоритм для шага 1:

1. Поиск максимального элемента матрицы C – элемент $c_{kl} = \max(C^*) \mid C^* \subset \{c_{11}, \dots, c_{nm}\}$.

2. Присвоить $x_{kl} = 1$. Из этого следует, что $x_{ki} = 0 \mid i \in [1, l] \cup (l, n], x_{j2} = 0 \mid j \in [1, k] \cup (l, n]$.

Если не получено X^k , то ограничить зону поиска максимального значения $C^* \subset \{c_{11} \dots c_{mm}\} \cap (\{c_{k1} \dots c_{km}\} \cup \{c_{l1} \dots c_{nl}\})$ и повторить шаг 1.

Как результат работы алгоритма получаем решение X^k . Это решение на шаге 2 необходимо проверить на оптимальность с помощью метода генетического локального поиска. Найдем окрестность $G(X^k)$, на которой необходимо оптимизировать значения функции.

В данной работе под окрестностью $G(X^k)$ подразумевается множество всех допустимых решений, полученных с помощью метода генетического локального поиска.

Рассмотрим метод адаптации генетического алгоритма к нахождению окрестности X^k для оптимизации целевой функции. В качестве пространства поиска выступает пространство решений X^n , в качестве функции пригодности – целевая функция $\langle C, X \rangle$.

На этапе селекции используем алгоритм «лучший и случайный», когда в решении X^k будут выбраны две строки: x_i^k при $c_{ki} = \max \{c_{ij} \mid i = 1..n, j = 1..m\}$ и случайная строка x_j^k .

В качестве оператора скрещивания выбран одноточечный оператор. Пусть $x_{il}^k = 1, x_{ip}^k = 0, l \neq p$ и $x_{ii}^k = 1, x_{iq}^k = 0, t \neq q$. Тогда после скрещивания: $x_{jt}^k = 1, x_{jp}^k = 0$ и $x_{il}^k = 1, x_{iq}^k = 0, t \neq q$. В процессе скрещивания два вектора «обменялись» единицами между собой.

В процессе мутации в каждой из строк единица может переходить на места, находящиеся в столбцах, которые содержат только 0. Среди всех вариантов выбирается наилучший по критерию максимума целевой функции.

После проверки, проведенной на шаге 2, будет получен оптимальный результат.

Сформулируем алгоритм для шага 2:

1. Для двух строк x_i^k и x_j^k в найденном на шаге 1 решении X^k , выбранных по алгоритму «полезный и случайный», заменим единицы на соответствующих местах.

2. Проводится мутация каждой строки по модифицированным правилам – с помощью поиска максимального значения целевой функции по

окрестности 1-замена для всех возможных перестановок в рамках строк x_i^k и x_j^k . Находим значение $f(X^{k+1})$ для всех $X_1^{k+1} \dots X_{l-1}^{k+1}$. Если $f(X^k) \geq f(X^{k+1})$, то $f(X^k)$ – локальный оптимум. Если же $f(X^k) < f(X^{k+1})$, то заменяем X^k на X^{k+1} и выбираем новую мутацию до достижения $m-n$ шагов для каждой строки.

3. При достижении шага n выбирается максимальное значение $f \max(X^n) = \max \langle C, x \rangle = \max \{f(X^k)\} \mid k = 1..n$. В противном случае переходим на шаг 1.

После работы алгоритма будет получено оптимальное решение задачи БЛП. Остановка алгоритма определяется количеством полученных локальных оптимумов, работа алгоритма останавливается при достижении n локальных максимумов.

Определим сложность работы алгоритма, оценивая численно временную сложность решения заданной задачи методом локальной оптимизации.

Ранее было принято без потери общности, что $n < m$. Пусть n – количество строк в матрицах C и X .

В алгоритме первого шага генетического локального поиска необходимо найти максимальное значение в каждой строке матрицы C . При этом на каждом шагу количество возможных вариантов будет уменьшаться на 1.

Таким образом, сложность задачи первого шага z_1 будет составлять разницу двух арифметических прогрессий S_m – суммы всех возможных вариантов перебора от 1 до m и S_{m-n} – суммы неиспользуемых вариантов от 1 до $(m-n)$:

$$\begin{aligned} R(z_1) &= S_m - S_{m-n} = \\ &= \frac{1+m}{2} \cdot m - \frac{1+(m-n)}{2} \cdot (m-n) = \\ &= \frac{n^2 + 2nm + n}{2}. \end{aligned}$$

Временная сложность второго шага обусловлена количеством перестановок в процессе мутации и составляет: $R(z_2) = S_{m-n} = 2(m-n)$.

Таким образом, общая временная сложность решения задачи методом линейной оптимизации составляет:

$$\begin{aligned}
 R(z) &= R(z_1) + R(z_2) = \\
 &= \frac{n^2 + 2nm + n + 2(m-n)}{2} = \\
 &= \frac{n^2 + 2nm - n + 2m}{2},
 \end{aligned}$$

$$R(z) \infty (n^2 + m^2) \leq 2n^2.$$

Предложенный алгоритм линейной оптимизации относится к неточным методам решения задачи БЛП, но он обеспечит нахождения локального максимума для поставленной задачи, который в большинстве случаев совпадает с глобальным максимумом, имея при этом более низкую временную сложность $O(n^2 + m^2)$.

5. Сравнение эффективности предложенного метода поиска соответствия между древовидными структурами и существующими аналогами

Для решения задачи поиска соответствия между XML документами использовался комбинаторный метод, сложность которого определяется как $O(n^2 + m^2)$, где n и m – количество листьев в начальной и обновленной древовидных структурах.

Для определения целесообразности такого подхода необходимо сравнить временную сложность поставленной задачи в этом подходе с временной сложностью алгоритмов, основанных на поиске минимальной последовательности редактирования, семантических алгоритмов и алгоритмов, основанных на хэш подписях.

Алгоритм Чанга и Шаша рассматривает два дерева для нахождения минимальной последовательности редактирования, может быть применен для XML документов. Условия этого алгоритма следующие. Иерархически структурированная информация может быть представлена как упорядоченные деревья – деревья, в которых потомкам каждого узла присвоен порядок. Рассматриваются деревья, в которых каждый узел имеет метку и значение, подразумевается, что каждый узел дерева имеет уникальный идентификатор. Минимальная временная сложность этого алгоритма определяется, как $O(p^2)$ [Wang et al., 2003], где $p = n + m$ – общее количество узлов в двух деревьях. Тогда сложность алгоритма равна: $O((n + m)^2)$.

Семантические алгоритмы рассматривают XML документ как структуру XML тегов. При сравнении учитывается иерархия синтаксиса языка и его закономерности. С помощью этого алгоритма можно определить только перемещения узлов, и невозможно определить удаление и вставку узла. Поэтому этот алгоритм не всегда находит корректное соответствие между узлами XML

документов. Сложность этого подхода [Lim et al., 2001]:

$$O(|X| \cdot |Y| \cdot (|X| \log |X| + |Y| \log |Y|)),$$

где X и Y – количество ветвей в семантической иерархии каждого из деревьев. Поскольку количество ветвей иерархии можно принять равным количеству листьев в каждом из деревьев, то сложность этого метода можно представить следующим образом: $O(m \cdot n \cdot (n \log n + m \log m))$.

В системе детектирования веб-страниц Коура рассматривается похожий с данной работой подход. Этот алгоритм рассматривает задачу нахождения соответствия между XML деревьями как задачу венгерского алгоритма. Сложность этой задачи в [Khoury et al., 2007] определена как $O((n + m)^3)$, что значительно больше, чем сложность предложенного алгоритма.

На рис. 2 показана сравнительная характеристика рассмотренных методов сравнения XML документов с помощью различных подходов.

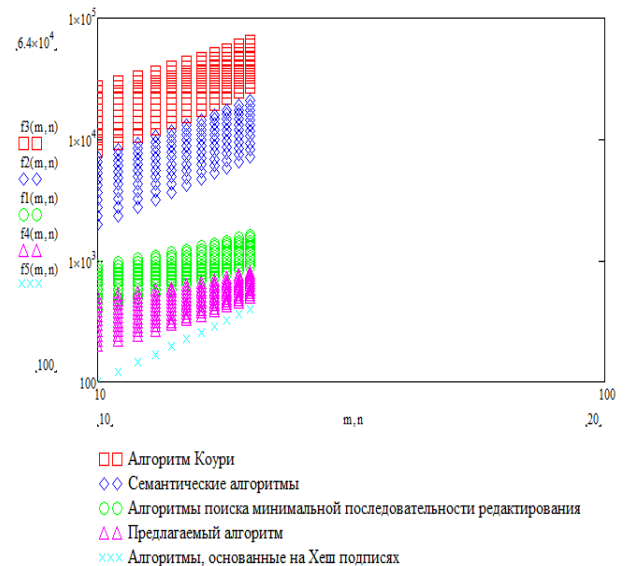


Рисунок 2 – Сравнительная характеристика рассмотренных методов

Найдем разницу между количеством операций, необходимых для выполнения предложенного алгоритма и самого эффективного из существующих, который удовлетворяет поставленным условиям – алгоритма Чанга и Шаша: $\Delta = (n + m)^2 - (n^2 + m^2) = 2nm$.

Эта разница будет существенная при увеличении значений n и m .

Ресурсом <http://www.websiteoptimization.com/> было проведено исследование, в котором было вычислено среднее значение количества элементов страницы HTML в среде Internet в 2007 году, которое составило 592 элемента, также было показано, что эта величина непрерывно растет. Например, количество HTML элементов на

странице выросла с 2006 по 2007 год от 281 до 592 элементов.

Пусть исходный документ HTML содержит 592 элемента, а в обновленном документе были убраны теги и их общее количество составит 100 элементов. Найдем выигрыш в эффективности в данном случае:

$$\eta = \frac{(n+m)^2 - (n^2 + m^2)}{(n+m)^2} = \frac{(592+100)^2 - (592^2 + 100^2)}{(592+100)^2} \cdot 100\% = 24,7\%.$$

При сравнении предложенного метода решения задачи поиска соответствия между XML документами с другими алгоритмами было показано, что предложенный алгоритм является эффективным с точки зрения временной сложности. Это достигнуто за счет решения задачи поиска соответствия между XML документами как задачи булевого линейного программирования и ее эффективного решения с помощью метода генетического локального поиска.

Заключение

Проведенные исследования позволяют сделать вывод о возможности использования метода генетического локального поиска в методе детектирования изменений, который обеспечивает удовлетворительное время детектирования изменений при сохранении достоверности полученных результатов.

Получил дальнейшее развитие метод генетического локального поиска для решения задачи булевого линейного программирования с учетом особенностей применения его в предметной области детектирования изменений в древовидных структурах, что позволило обеспечить удовлетворительную временную сложность решения задачи БЛП: при детектировании изменений в деревьях среднего размера (10-100 элементов) удалось уменьшить минимальное время детектирования изменений по сравнению с точными методами с 384 с до 6,72 мс.

Эффективность предложенных модификаций была доказана на основе анализа вычислительной сложности предложенных и существующих методов – время детектирования изменений в деревьях малого размера (до 10 элементов) было уменьшено в модифицированном методе Балаша с 181 мс до 0,22 мс, а в модифицированном методе динамического программирования с 142 с до 0,01 мс.

Библиографический список

[Алексеев и др., 2009] Алексеев О. М. Система публікування/передплати на науково-технічні інформаційні ресурси / Алексеев М. О., Молчанов Ю. М., Алексеев О. М. // Вісник Сумського державного університету, серія "Технічні науки", – Вип. 2. 2009. – С.7-14.

[Алексеева и др., 2007] Алексеева Е. В. Генетический локальный поиск для задачи о р-медиане с предпочтениями клиентов / Е. В. Алексеева, Ю. А. Кочетов // Дискретный анализ и исследование операций, 2007, серия 2, том 14, №1.

[Chakravarthy et al., 2004] Chakravarthy S. A learning-based approach for fetching pages in WebVigil / S. Chakravarthy, S. Sanka, J. Jacob // Proceedings of the 2004 ACM symposium on Applied computing. – 2004. – pp. 1725-1731.

[Chand et al., 2004] Chand R. XNET: a reliable content-based publish/subscribe system / R. Chand, P. Felber, S. Antipolis // Reliable Distributed Systems, 2004. – pp. 264-273.

[Chawathe, et al., 1997] Chawathe S. Meaningful change detection in structured data / S. Chawathe, Garcia-Molin // Proceedings of the ACM, SIGMOD International Conference on Management of Data, Tuscon, Arizona – 1997 - pp. 26-37.

[Chawathe, et al., 1998] Chawathe, S. Representing and querying changes in semistructured Data / Chawathe, S., Abiteboul, S, Widom, // Proceedings of the International Conference on Data Engineering, Orlando, Florida. – 1998. – pp. 4-13.

[Coruscio et al., 2002] Coruscio M. Formal Analysis of Clients Mobility in the Siena Publish/Subscribe Middleware / M. Coruscio, P. Inverardi, P. Pelliccione // Technical Report, Department of Computer Science, University of Colorado – 2002. – pp. 1-18.

[Flesca et al., 2007] Flesca E. Efficient and affective Web change Detection / E. Flesca, S. Masciari // Proceedings of the International Conference on Data Engineering, Tuscon, Arizona – 2007. – pp. 4-13.

[Jyoti et al., 2005] Jyoti J. CX-DIFF: A Change Detection Algorithm for XML Content and Change Visualization for WebVigil / J. Jyoti, A. Sachde, and S. Chakravarthy // Data and Knowledge Eng., vol. 52, no. 2, 2005. – pp. 209-230.

[Khan et al., 2003] Khan L. Change Detection of XML Documents Using Signatures / L. Khan, L. Wang and Y. Rao // Department of Computer Science University of Texas at Dallas, 2003.

[Khandagale et al., 2010] Khandagale H. A Novel Approach for Web Page Change Detection System / H. Khandagale, P. Halkarnikar // International Journal of Computer Theory and Engineering – 2010.

[Khoury et al., 2007] Khoury F. An Efficient Web Page Change Detection System Based on an Optimized Hungarian Algorithm / F. Khoury, R. El-Mawas, O. El-Rawas, E. Mounayar, H. Artail // IEEE Transactions on Knowledge and Data – Vol.19. – 2007. – pp.48-59.

[Lim et al., 2001] Lim S. An Efficient Algorithm to Compute Differences between Structured Documents / S. Lim, N. Yiu-Kai // Information theory, IEEE. – 2001. – pp. 171-180.

[Myers, 1986] Myers E. An O(ND) difference algorithm and its variations / E. Myers // Algorithmica, 1(2), 1986. – pp. 251-266.

[Wang et al., 2003] Wang Y. A fast change detection algorithm for XML documents / Y. Wang, D. DeWitt, X. Cai. // In International Conference on Data Engineering (ICDE'03). Citeseer. – 2003. – pp. 235-258.

[changedetection.com] <http://www.changedetection.com/>

[followthatpage.com] <https://www.followthatpage.com/>

[changedetect.com] <http://www.changedetect.com/>

CHANGES DETECTION OF XML DOCUMENTS

Globa L.S., Molchanov Y.N.

National Technical University of Ukraine 'Kyiv Polytechnic Institute', Kyiv, Ukraine

lgloba@its.kpi.ua

This paper presents a method for detecting changes in the XML-documents, which is based on genetic local search algorithm for the changes detection in the XML-documents to provide a satisfactory time of changes detection while saving the validity of the results. The proposed method provides a satisfactory time complexity: in changes detecting for the trees of medium size (10-100 members) there is reduced significantly the minimum changes detection time in comparison with exact methods.



OSTIS-2015

(Open Semantic Technologies for Intelligent Systems)

УДК 004.738.52

ИНТЕГРАЦИЯ КОРПОРАТИВНЫХ ДАННЫХ НА ОСНОВЕ WIKI-СИСТЕМ В УСЛОВИЯХ СЛАБОЙ СВЯЗАННОСТИ ИСТОЧНИКОВ

Галушка И.Н., Оксанич И.Г., Щербак С.С.

*Кременчугский национальный университет имени Михаила Остроградского,
г. Кременчуг, 39600, Украина.*

ilona.galushka@ya.ru

В работе рассмотрены технические аспекты совместного применения Wiki-систем с хранилищами триплетов RDF для организации электронного документооборота предприятий с территориально-распределенной структурой, проведен анализ существующих подходов и способов интеграции корпоративных данных и приведена соответствующая классификация. Разработаны способы интеграции корпоративных данных на основе концепции связанных данных предприятия, и идентификации объектов в корпоративных документах, как средства формирования терминологической базы предприятия.

Ключевые слова: сервисная шина предприятия; информационное пространство; связанные данные; модель предметной области; семантика; интеграция данных

Введение

Современные рыночные условия вынуждают предприятия искать новые пути для повышения своей рентабельности, например, путем внедрения высокоэффективных систем электронного документооборота. Подобные внедрения сталкиваются с необходимостью интеграции множества различных источников данных предприятия в единое информационное пространство, компоненты которого связаны между собой единым программным интерфейсом системы электронного документооборота. Зачастую источники между собой не связаны или имеют слабовыраженную связь, что приводит к необходимости решения ряда проблемных задач, нацеленных на повышение уровня связности данных в системе и интеграции существующих источников в согласованное информационное пространство связанных данных предприятия с единой точкой входа и развитыми унифицированными средствами "бесшовного" добавления новых источников.

В последнее время одним из популярных подходов к организации корпоративного электронного документооборота является использование Wiki-систем (англ. Enterprise Wiki), наиболее известной представительницей которых является MediaWiki. Эта система лежит в основе всемирно известного источника информации Википедия.

Популярность Wiki-систем обусловлена прежде всего удобным веб-ориентированным многопользовательском интерфейсом для создания и редактирования как структурированной так и не структурированной информации с поддержкой версииности и многоязычности. К сожалению, автоматическая обработка информации в Wiki-системах затруднена из-за отсутствия развитых средств для обеспечения машинного понимания и чтения (англ. Machine readable). Одной из проблем на этом пути является обеспечение терминологической базы для идентификации паттернов объектов предметной области (Про) предприятия в текстах и различных структурированных представлениях документов, например, таблицах, списках. В связи с чем, решение данной задачи в контексте интеграции различных корпоративных источников для организации эффективной работы информационного пространства связанных данных предприятия представляется нам актуальным и целесообразным.

На сегодняшний день информационное обеспечение предприятия часто строится на основе стихийной архитектуры, что подразумевает использование решений и приложений информационных систем от различных поставщиков. В такой архитектуре приложения интегрированы в негибкую инфраструктуру, в рамках которой не действуют унифицированные правила взаимодействия между системами, что приводит к необходимости создания интеграционных компонентов, которые должны быть достаточно адаптивными для

обеспечения эффективной и непрерывной работы предприятия [Завгородний, 2013].

В большинстве случаев предприятия имеют интеграционную архитектуру по топологии «точка-точка», внесение малейших изменений в которую может оказаться трудоемкой задачей, так как система сообщений и технологии, лежащие в основе, могут различаться.

Как правило, средние и большие предприятия состоят из территориально-удаленных подразделений, что требует использования эффективных средств коммуникации [Берко, 2009].

В последнее время, одним из перспективных направлений является построения эффективных систем электронного документооборота на основе слабосвязной архитектуры, что позволяет связать существующие задачи бизнес процессов предприятия единым программным обеспечением для минимизации затрат и увеличения степени интегрированности данных с возможностями организации открытого доступа к ним [Чистякова, 2014].

Целью данной работы является повышение эффективности систем электронного документооборота предприятий путем разработки модели и способа интеграции корпоративных данных на основе Wiki-систем и технологий связанных данных предприятия (англ. Linked Enterprise Data, LED).

В данной работе в качестве источников данных будем рассматривать корпоративные документы, расположенные в Wiki-системах, а в качестве средства обеспечивающего терминологическую базу будем использовать корпоративные онтологии. Ограничение выбранного набора средств решения задачи ни в коей мере не сужает общности исследования данной работы в связи с их унифицированной природой.

1. Интеграционные технологии современного предприятия

1.1. Классификация технологий интеграции корпоративных данных

Основной целью предприятия является эффективная реализация бизнес-процессов для обеспечения максимальной прибыли. Для достижения этой цели разработан ряд способов и инструментальных средств описания, проектирования и анализа бизнес-процессов, в рамках которых технологии интеграции являются одними из важнейших составляющих. Таким образом, становится актуальным вопрос выбора технологии интеграции среди множества существующих. Рассмотрим виды, уровни и способы интеграции информационных систем, и определим наиболее эффективные интеграционные технологии для предприятий с территориально-распределенной структурой.

В настоящее время выделяют несколько видов интеграции информационных систем (рис. 1) [Франгулова, 2010].



Рисунок 1 - Виды интеграции информационных систем

Информационно-ориентированная интеграция применяется, в основном, когда необходим обмен информацией между несколькими ИС. В процессе работы информационно-ориентированная интеграция использует обычно брокеры сообщений, связывающее программное обеспечение (ПО) (middleware), серверы репликации баз данных и другие технологии, целью которых является распространение информации между несколькими системами. Чаще всего данный вид интеграции используется при интеграции корпоративных приложений (Enterprise Application Integration, EAI) [Франгулова, 2010].

Технология сервисно-ориентированной интеграции опирается на слабосвязную архитектуру ИС, ориентированную на сервисы (Service Oriented Architecture, SOA). Эта технология применяется, когда необходимо совместное использование функций приложения и источников информации.

Сервисно-ориентированная архитектура, позволяет компоновать бизнес-процессы из компонентов, выполняющихся на разных платформах (корпоративных бинов J2EE, компонентов .NET, отдельных приложений), представлять в виде сервисов и повторно использовать в новых бизнес-процессах унаследованные компоненты [Гонтарь, 2013].

Процессно-ориентированная интеграция предоставляет возможность присоединиться к внутренним прикладным процессам каждого приложения, причем таким образом, чтобы не просто использовать его функции, а создать новый или мета-процесс, который и свяжет приложения [Франгулова, 2010]. Особенность использования этой технологии является возможность связывания большого числа разнородных информационных систем, используя при этом их встроенные функции.

В зависимости от подхода интеграции можно выделить следующие уровни: интеграция данных,

интеграция приложений, интеграция бизнес-процессов, интеграция на основе стандартов и интеграция платформ (рис. 2).



Рисунок 2 - Классификация интеграционных уровней и соответствующие им технологии

Интеграция бизнес-процессов (Business Process Integration, BPI) – основана на спецификации реализации и управления процессами обмена информацией между различными системами [Попов, 2013], что позволяет усовершенствовать операции интеграции и оптимизировать расходы в процессе использования ИС. Элементы BPI включают управление процессами, моделирование бизнес- и технологических процессов, которые охватывают различные задачи, процедуры, архитектуры, требования к входной и выходной информации, а также пошаговое разбиение каждого бизнес-процесса [Пушкар, 2010].

Интеграция приложений (англ. Application Integration) осуществляется путем объединения данных или функций одной системы с другой. Передача функций или данных, свойственных какому-либо приложению, в распоряжение другого приложения используется с той целью, чтобы их взаимодействие обеспечило бы выполнение определенной прикладной функции ИС [Аткин, 2010].

Интеграция данных (англ. Data Integration) основана на идентификации и каталогизации данных с целью их дальнейшего использования.

Успешная реализация интеграции бизнес-процессов и приложений на двух предыдущих уровнях зависит от того, как будут интегрированы в системе данные, принадлежащие разным источникам данных, в том числе баз и хранилищ данных. На этом уровне данные необходимо идентифицировать, каталогизировать и построить модель метаданных [Росинский, 2012].

Интеграция на основе стандартов (англ. Standards of Integration) основана на использовании стандартных форматов данных, таких как JSON, XML. В рамках такого подхода интеграция осуществляется на основе интеграционных схем на одном из языков описания данных, например, XML Schema [Арсеньев, 2001].

Интеграция платформ (англ. Platform Integration) касается процессов и инструментов, с помощью которых системы могут осуществлять безопасный и оптимальный обмен информацией. Обычно используется для завершения интеграции систем, формирования базовой архитектуры аппаратного и программного обеспечения и интеграции территориально-распределенных частей гетерогенной сети [Франгулова, 2010].

При межведомственной интеграции (англ. Interdepartmental Integration) заинтересованные в информационном обмене потребители заключают соглашения, определяющие состав данных, технологические, технические, организационные и экономические аспекты взаимодействия. В соответствии с регламентами ведомства передают часть своей информации в смежные ведомства, которые преобразовывают ее в соответствии с собственной терминологией и системой классификации. Часть информационных потоков являются однонаправленными, например, при формировании статистических отчетов), а часть – двунаправленными, ориентированными на взаимодействие запрос-отчет по запросу [Шаппелл, 2008].

Каждый подход имеет свои сильные и слабые стороны, но на практике используется комбинация различных подходов в зависимости от исходных условий и текущего уровня применения интеграционных средств и способов.

Проведём классификацию существующих способов интеграции.

По времени запуска:

1. Реального времени – если данные должны быть обновлены немедленно после изменений.
2. Отложенная – если процесс синхронизации данных запускается по какому-либо событию во времени или по расписанию.

По способу анализа информации:

1. По текущему состоянию – сравнение записей одной таблицы с записями другой, и на основании этого принимается решение о синхронизации,
2. Дельта-репликация – если в базе данных предусмотрен журнал вносимых изменений, и

алгоритм репликации переносит изменения по дельтам изменений, накопленным в журнале.

По направлению интеграции:

1. Односторонняя – если данные изменяются только в одном приложении, а в другой данные только хранятся и не подвергаются изменениям.

2. Многосторонняя – если данные могут изменяться и вводиться во всех приложениях.

По уровню интеллектуального анализа:

1. Синтаксическая интеграция. Основывается на внешнем сходстве объединяемых данных.

Например, при объединении двух таблиц мы предполагаем, что в поле «№ договора» все записи имеют схожий формат «Договор № 3». Однако если в одной таблице в этом же поле дополнительно указывается дата договора «Договор № 1 от 17.01.2015», а в другой таблице выделено отдельное поле «Дата», необходимо обеспечить интерпретацию данных из одного вида в другой.

2. Семантическая интеграция. Основывается на сравнении данных на содержательном уровне.

Данный подход предполагает передачу вместе с данными также и их описание – метаданные. Этот тип интеграции основывается на знании и учёте природы данных. Основой семантической интеграции стала реализация онтологического подхода. При этом связь между элементарными единицами данных осуществляется в соответствии с их определением в корпоративной онтологии [Берко, 2009].

С учетом вышесказанного для интеграции корпоративных данных применим подход, ориентированный на использование корпоративной онтологии, в которой расположена большая часть доменной терминологии (терминологии ПрО), используемой предприятием.

1.2. Способ интеграции корпоративных данных в условиях слабой связности источников

Интеграция корпоративных данных заключается в последовательном выполнении нижеперечисленных этапов:

1. Формирование первичной системы типов связанных данных предприятия в виде корпоративной онтологии

2. Формирование первичной онтологической базы знаний предприятия в виде наборов квад-четырёхэлементных структур для описания графов связанных данных, путем составления синонимических рядов терминов, используемых на предприятии, и установления соответствия с типами корпоративной онтологии.

3. Подготовка типовых корпоративных документов и формирование на их основе терминологической базы путем выделения наиболее часто встречающихся терминов и выявления известных на основе корпоративной онтологии с учетом базы стоп-слов, исключающих добавление незначимых терминов.

4. Идентификация объектов в корпоративных

документах и формирование схемы связанных данных корпоративного документа.

5. Формирование индекса связанных данных информационного пространства предприятия

Общая схема взаимодействия компонентов подсистемы интеграции системы электронного документооборота представлена на рис.3

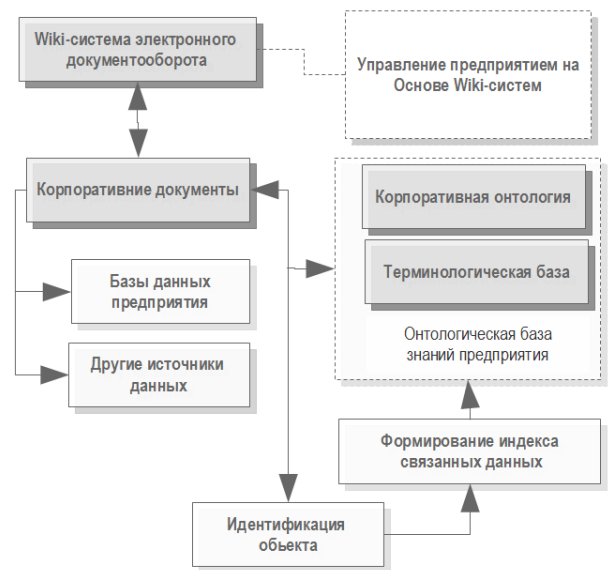


Рисунок 3 - Интеграция корпоративных данных

Рассмотрим более подробно составляющие этапы способа интеграции корпоративных данных.

На первом этапе администратор системы с специалистом предметной области формируют первичные знания о предметной области в виде системы типов связанных данных, определяя основные объекты и связи между ними.

На втором этапе формируется универсальное множество синонимов терминов, используемых в корпоративных документах, со спецификацией синонимических рядов.

На третьем этапе, спецификации объектов и отношений схемы связанных данных используются для определения соответствующих им терминов и структурных образований, идентификация которых может быть осуществлена. В работе в качестве таких образований рассматриваются внедренные в документы таблицы и списки.

Далее, с целью определения положений этапа идентификации объектов, рассмотрим понятие корпоративного документа и разработаем соответствующую ему математическую модель.

Определение 1. Корпоративный документ – это иерархически организованный документ, представляющий собой множество вложенных заголовков, каждый из которых состоит из множества текстовых фрагментов, подзаголовков, таблиц, изображений, объединенных общим контекстом.

Замечание 1. Корпоративные документы относятся в большинстве случаев к классу слабоструктурированных документов.

Замечание 2. С учетом ориентации работы на использование результатов исследований в wiki-ориентированных системах электронного документооборота, то расширим набор используемых структурных компонентов семантическими аннотациями терминов, поддерживаемыми расширениями Wiki-систем Semantic MediaWiki.

Формализуем понятие корпоративного документа.

Пусть Doc – корпоративный документ, H – множество вложенных заголовков, G – контекст документа, L – множество списков и таблиц документа, I – множество изображений, P – подзаголовки, W – семантические аннотации, то в соответствии с определением 1 с учетом замечания 1 и 2 корпоративный документ может быть представлен следующим выражением:

$$Doc = \langle H, G, L, I, P, W \rangle \quad (1)$$

Замечание 3. В работе под понятием множества во всех случаях без исключения подразумевается конечное множество элементов.

Замечание 4. Корневой заголовок документа, даже при его отсутствии рассматривается в работе как существующий для организации единой точки доступа к содержимому документа при его обработке.

Замечание 5. Списки в работе рассматриваются как одноколоночные таблицы, что не сужает ни в коей мере общности рассуждений.

Таким образом, математическую модель корпоративного документа, согласно определения 1, с учетом замечаний 1-5 представим выражением 1.

На основе выше представленной модели корпоративного документа разработаем способ идентификации объектов в корпоративных документах.

Пусть $A = \{a_1, \dots, a_n\}$ – терминологическая база предприятия, полученная в результате предобработки типовых документов предприятия и состоящая из n терминов, $B = \{b_1, \dots, b_k\}$ – набор из k -стоп-слов, причем $A \cap B = \emptyset$, $S = \{S_1, \dots, S_m\}$, где $S \subset S^U$ – синонимический ряд термина $a \in A$, S^U – множество всех синонимических рядов, тогда преобразование терминологических наборов в онтологическую базу знаний подразумевает установление соответствия между термином терминологического набора и типом онтологической базы знаний, что может быть выражено с помощью следующего отображения φ_1 :

$$\varphi_1 : A \xrightarrow{S} T^U \quad (2)$$

Определим спецификацию типа объекта t , являющегося элементом универсального множества

типов T^U , используемого в выражении 2, с помощью следующего выражения:

$$\forall t \in T^U : t = \langle S, R, I \rangle \quad (3)$$

где S – синонимический ряд термина, соответствующего типу объекта предметной области t , R – конечное множество отношений, характерных для данного типа, I – множество объектов типа t .

С учетом выражения 3 расширение синонимического ряда соответствующего термину объекта представим как процесс добавления различных терминов с соответствующими им синонимическими рядами к объектам типа t с помощью выражения 4:

$$S^t = S^t \cup [S_1 \cup \dots \cup S_p] \quad (4)$$

где S^t – синонимический ряд термина объекта t -го типа, причем $S^t \subset S \subset S^U$, S^U – универсальное множество синонимических рядов терминов предприятия.

В качестве программной реализации предложенных в статье модели и способов была использована система Semantic MediaWiki, а в качестве онтологической базы знаний предприятия использовался Openlink Virtuoso с встроенным RDF-хранилищем (англ. Resource Description Framework).

Заключение

В работе рассмотрены интеграционные технологии в контексте решения задач повышения рентабельности и эффективности электронного документооборота предприятия.

Рассмотрены технические аспекты совместного применения Wiki-систем с хранилищами триплетов RDF для организации электронного документооборота предприятий с территориально-распределенной структурой.

Получила дальнейшее развитие классификация подходов и способов интеграции корпоративных данных в рамках лоскутной автоматизации предприятий.

Предложена математическая модель корпоративного документа в виде графа связанных данных, которая в отличие от существующих поддерживает распределенное хранение и гипертекстовые семантические аннотации структурных компонентов, что позволяет обеспечить формальную основу для описания процесса интеграции корпоративных данных территориально-распределенных предприятий.

Получил дальнейшее развитие способ интеграции корпоративных данных в условиях структурной неопределенности и слабой связности источников, который в отличие от существующих позволяет добавлять новые источники данных без изменения общего алгоритма интеграции за счет

настройки критериев интерпретации структурных и семантических компонентов документов.

Предложен способ идентификации объектов в корпоративных документах на основе онтологического подхода, который позволяет формировать и расширять терминологическую базу в соответствии с системой типов, используемой на предприятии.

Рассмотрены практические аспекты реализации программных систем интеграции корпоративных данных на основе предложенных в работе моделей и способов.

Библиографический список

[Завгородний, 2013] Завгородний В. В. Информационная технология разработки специализированной СППР оперативного управления производством полупроводниковых изделий / В. В. Завгородний, И. В. Шевченко, В. Ф. Шостак, С. С. Щербак // Вісник Академії митної служби України. Серія: "Технічні науки". 2013 – № 1. – С. 69–76.

[Берко, 2009] Берко А.Ю. Способы та засоби семантичної інтеграції даних / А.Ю. Берко // Інформаційні системи та мережі. Вісник Національного університету "Львівська політехніка". 2009 – № 653. – С. 190–199.

[Чистякова, 2014] Чистякова И.С. Онтолого-ориентированная интеграция данных в семантическом вебе / И.С. Чистякова // Проблеми програмування. 2014. – № 2.–3. – С. 190–196.

[Франгулова, 2010] Франгулова Е. В. Классификация подходов к интероперабельности информационных систем / Е. В. Франгулова // Вестник АГТУ. Сер.: Управление, вычислительная техника и информатика. 2010. – № 2. – С. 176–180.

[Гонтарь, 2013] Гонтарь Н.А. Модель семантической сервис-ориентированной архитектуры / Н.А. Гонтарь // Наукові праці ДонНТУ. Сер.: "Інформатика, кібернетика та обчислювальна техніка". 2013 – № 1. (17) – С. 68–73.

[Попов, 2013] Попов В.А. Способ построения интегрированной системы управления предприятием на основе принципов непрерывного улучшения бизнес-процессов / В.А. Попов, А.В. Котляров // Авіаційно-космічна техніка і технологія. 2013 – № 2. (37) – С. 144–151.

[Пушкар, 2010] Пушкар А.И. Изменения бизнес-процессов предприятий по созданию и реализации информационных продуктов и услуг в интернет-среде / А.И. Пушкар, С.А. Назарова // Системи обробки інформації. 2010 – № 7 (88) – С. 167–173.

[Аткин, 2010] Аткин А. Интеграция ИТ: основные понятия и технологии / А. Аткин // Информационные технологии в экономике, управлени и образовании. 2010 – С. 284–289.

[Росинский, 2012] Росинский В.В. Обеспечение интеграции данных в корпоративных информационных системах на основе прогрессивных WEB-технологий / В.В. Росинский // Вісник ДУИКТ. 2012 – Т.10, №1. – С. 87–94.

[Арсеньев, 2001] Арсеньев Б. П., Яковлев С. А. Интеграция распределенных баз данных. – СПб.: Издательство «Лань», 2001. – 464 с.

[Шаппелл, 2008] Шаппелл Д. ESB – Сервисная шина предприятия: Пер. с англ. – СПб.: БХВ-Петербург, 2008. – 368 с.

[Лис, 2010] Лис К. П. Онтологическая интеграция данных моделирования для управления сервисно-ориентированной ИТ-инфраструктурой // Материалы 6-й международной конференции СпбГУЭФ. – СПб: Изд-во СпбГУЭФ. – 2010. – 62-67с.

[Андон, 2006] Андон П. Проблеми побудови сервіс-орієнтованих прикладних інформаційних систем в semantic web середовищі на основі агентного підходу / П. Андон, В. Дерещкий // Проблеми програмування. 2006 – № 2.–3. – С. 493–502.

[Глибовец, 2013] Глибовец А.Н. Семантическая паутина и WIKI-системы / А.Н. Глибовец, Н.Н. Глибовец, Д.Е. Покопцев, М.О. Сидоренко // Проблеми програмування. 2013 – №1. – С. 45–67.

[Андон, 2006] Андон П. Проблеми побудо- ви сервіс-орієнтованих прикладних інформаційних систем в semantic Web

середовищі на основі агентного підходу / П.Андон В. Дерещкий // Проблеми програмування. – 2006. – № 2-3. – (спец. вип.). – С. 493–502.

[Боркус, 2006] Боркус Владислав. Способы и инструменты интеграции корпоративных приложений: Отчет/ RC Group.– М.: RC Group, 2006.– 13 с.

[Гудов, 2006] Гудов А.М. Интеграция распределённых приложений при помощи системы электронного документооборота. / А.М.Гудов, С.Ю. Завозкин // Труды международной конференции "Вычислительные и информационные технологии в науке, технике и образовании". II том – Павлодар: ТОО НПФ "ЭКО", 2006.

[Росинский, 2010] Росинский В. В. Способы и средства интеграции в CRM-системах / В. В. Росинский // Системные технологии. – 2010. – №6(71). – С.197–207.

[Вавилов, 2011] Вавилов К. П. Web-интеграция корпоративных систем / К. П. Вавилов. // Информационные технологии моделирования и управления. – 2011. – №3(68). – С.341-347.

ENTERPRISE DATA INTEGRATION METHODS UNDER CONDITIONS OF LOW SOURCES RELATEDNESS

Galushka I.M, Oksanich I.G, Shcherbak S.S

Kremenchuk Mykhailo Ostrohradskyi National University,
ilona.galushka@ya.ru

Technological aspects of the integrated use of Wiki-systems and RDF triplestore for organization of enterprise e-document workflow with geographically distributed structure were considered, existing approaches and methods for integrating enterprise data were analyzed and relevant classification was given.

Main Part

We propose a model of corporate document as the main source of linked data in business processes. We have developed a method for enterprise data integration based on the linked enterprise data concept and propose a method for identifying objects in corporate documents as a means of forming enterprise term base.

Integration technologies in the context of efficiency increasing of enterprise e-document workflow were considered.

Technological aspects of integrated use of Wiki-systems and RDF triple stores for organization of enterprise e-document workflow with geographically distributed structure were considered.

The classification of approaches and methods of enterprise data integration within patchwork automation of enterprises got further development.

Conclusion

The mathematical model of corporate document in the form of linked data graph was proposed; as opposed to the existing models, it supports distributed storage and hypertext semantic annotations of structural components that allows providing formal base to describe a process of enterprise data integration of enterprises with geographically distributed structure.

Practical aspects of enterprise data software systems implementation based on the proposed models and methods are considered.



OSTIS-2015

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822:514

СЕМАНТИЧЕСКИЙ ПОДХОД К ИНТЕГРАЦИИ ОБЪЕКТОВ WEB OF THINGS

Рогушина Ю.В.* , Гладун А.Я.**

* *Институт программных систем НАН Украины, Киев, Украина*

ladamandraka2010@gmail.com

** *Международный научно-учебный центр информационных технологий и систем НАН Украины, Киев, Украина Киев, Украина*

glanat@yahoo.com

Развитие новых тенденций в формировании информационного пространства, в частности – Web of Things (WoT), требует разработки новых знание-ориентированных методов идентификации, поиска и исследования специфических для этого пространства информационных объектов. При создании методов и сервисов семантического поиска на основе онтологий в информационном пространстве Web of Things на базе Internet of Things предлагается использовать для этого элементы онтологического анализа.

Для того чтобы собрать все преимущества технологии Web-сервисов к пространству WoT, предложен подход для аннотирования описаний WoT-объекты при помощи онтологических моделей и семантических Web-сервисов.

Ключевые слова: Web of Things, Internet of Things, семантический Web-сервис, онтология, семантическое аннотирование.

Введение

В настоящее время наблюдается усложнение структуры и способов получения информации, доступной пользователям через Web. Все большая часть информации поступает не от людей, а от различных физических устройств, и в дальнейшем отправляется также различным устройствам, а не непосредственно пользователям.

Поэтому сейчас можно говорить о новом поколении Web, имеющем свою специфику, базирующемся на новых технологиях (в частности, связанных с Internet of Things) и требующем развития соответствующих способов и методов обработки.

1. Тенденции развития Web

Если технологической основой традиционного Web являлись Интернет-технологии, то базисом развития Web of Things является Internet of Things (IoT). WoT – это информация, передаваемая и получаемая через Web-среду от элементов, составляющих IoT.

Объект в WoT – это встроенное устройство, соединенное с физическим объектом. При этом сам физический объект не является сенсором или вычислительным устройством.

Например, в фонарный столб (физическое устройство) может быть встроен микроконтроллер с датчиками для управления освещенностью и мобильный телефон для связи. Другой пример – помещение в разумном доме, оборудованное датчиками температуры и средствами для связи с пользователем для получения указаний о желаемой температуре.

Таким образом, *объект в IoT* – это совокупность датчиков и встроенных устройств, соединенных с тем или иным физическим предметом, предназначенным для отличных от вычислений функций, и воспринимаемые с ним как единое целое.

Технология *Web вещей (Web of Things, WoT)* возникла с начала 2000-ых годов, как попытка создания Web-ориентированной сервисной платформы для «умных» вещей, т.е. предметов, снабженных различными датчиками или электронными устройствами для приема/передачи

данных и сигналов управления, связь между которыми поддерживается через IoT.

Web of Things можно рассматривать как новый этап развития Web, интегрирующий результаты его предыдущих поколений, в частности, Semantic Web и социального Web (таблица 1).

Таблица 1 – Эволюция поколений Web

	Описание	Технологии
Web 1.0	статические HTML-страницы	HTML, HTTP
Web 1.5	динамичный контент HTML	на стороне клиента (JavaScript, DHTML, Flash), на стороне сервера (CGI, PHP, Perl, ASP / .NET, JSP)
Web 2.0	Участие пользователей в обмене информацией, совместимость, ориентированный на пользователя дизайн, сотрудничество в Web, социальный Web; Semantic Web и семантические Web-сервисы	блоги, социальные сети, социальные теги, вики, подкасты, RSS каналы), URI, XML, RDF, OWL, SPARQL
Web 3.0	Web вещей, приборов и устройств реального мира, связанных полной интеграцией через Интернет	Стандарты IoT: URI, HTTP, Atom, REST

В дальнейшем будем понимать объект в WoT как

$$T_{WoT} = \langle Th_{ph}, \{s_i\}, i = \overline{1, n}, \{u_j\}, j = \overline{1, m} \rangle,$$

где

Th_{ph} – объект физического мира;

s_i – сенсоры, связанные с Th_{ph} ;

u_j – встроенные устройства, связанные с Th_{ph} .

Если технологической основой традиционного Web являлись Интернет-технологии, то базисом развития Web of Things является *Интернет вещей* (Internet of Things – IoT). WoT – это информация, передаваемая и получаемая через Web-среду от элементов, составляющих IoT.

Особенностями архитектуры современного IoT является наличие чрезвычайно большого количества разнообразных устройств и ресурсов, которые обмениваются информацией между собой и между прикладными программами, сообщают о состоянии ресурсов, обеспечивают наблюдение и измерение разных физических величин и т.п.

Пока трудно оценить перспективы развития WoT. Вероятно, некоторые приложения, такие как мобильные платежи и инициатива IBM, Smarter Planet, станет широко распространенным всего через несколько лет. Маркетинг также будет преобразован, так как потребители получают беспрепятственный доступ к цифровым продуктам от рекламы в физическом мире. Тем не менее, очевидно, что объем вычислений, связанных с обработкой данных от WoT, увеличится очень быстро и потребует технологий хранения и анализа потоковых данных.

При этом перспективным направлением представляется трансформация получаемых сведений в значительно более компактные знания и последующее использование именно таких «обработанных» знаний для решения прикладных задач.

WoT развивается в неразрывной связи с IoT и для повышения эффективности требует применения семантических, ориентированных на знания методов онтологического распознавания, выбора, поиска и управления объектами и сервисами доставки, хранения и обработки данных. В данной работе предлагается использование разработанной онтологии «умных» вещей для задач семантической интероперабельности (методы выравнивания онтологий), выбора сервисов и объектов в распределенном и гетерогенном пространстве Web of Things.

Отсутствие четкого и формального представления знаний в WoT может привести к двусмысленности в терминологии, а также препятствовать совместимости и в основном семантической совместимости.

Особенностями архитектуры современного Интернета вещей является наличие чрезвычайно большого количества разнообразных устройств и ресурсов, которые обмениваются информацией между собой и между прикладными программами, сообщают о состоянии ресурсов, обеспечивают наблюдение и измерение разных физических величин и т.п.

Распределенная природа и неоднородный характер IoT делает взаимодействие между "вещами" сложной задачей и требует методов, которые могут облегчить автоматизированную машинную обработку.

Идея IoT начала успешно развиваться благодаря развитию следующего набора информационных технологий: широкого распространения беспроводных сетей (Wi-Fi, 3G, LTE, WiMAX) и датчиков; активного перехода Интернет на IPv6; роста популярности облачных вычислений и появления технологий межмашинного взаимодействия (Machine to Machine, M2M) [Machine-to-machine, 2011].

С помощью M2M осуществляется доступ к удаленным объектам для сбора данных и

мониторинга состояния. M2M работает с протоколами TCP/IP, беспроводными сетями стандарта IEEE 802.11, технологиями сотовой связи и проводными сетями, такими как Ethernet. Эта технология незаменима в тех случаях, когда удалённое оборудование используется в труднодоступных местах или когда использование проводного соединения невозможно в принципе.

Набор технологий, разработанных в Semantic Web, таких как онтологии, семантические аннотации, связанные данные и семантические Web-сервисы могут быть использованы в качестве принципиальных решений в целях реализации WoT.

2. Постановка задачи

Для того, чтобы для обнаружения и использования объектов WoT использовать подходы и методы, разработанные для исследования семантических Web-сервисов, предлагается рассматривать каждую WoT-объект как семантический Web-сервис, функции которого соответствуют назначению физической составляющей WoT-объекты и связанных с ней датчиков и устройств.

Для семантического аннотирования WoT-объектов предлагается использовать как онтологии, специально разработанные для описания объектов WoT, IoT и Web-сервисов, так и произвольные онтологии предметных областей. При этом проблема исследования WoT сводится к задаче сопоставления онтологий, для которой на сегодня уже разработано некоторое количество алгоритмов и методов.

3. История возникновения Web of Things

Если проследить историю зарождения WoT, то можно отметить следующее. В 2002 году при разработке проекта по «умному городу» [Kindberg, 2002] было предложено связать физические объекты с Web-страницами, содержащими информацию о городе, и сервисами для пользователей.

Для этого были использованы инфракрасные интерфейсы или штриховые коды на объектах, позволяющие пользователям при взаимодействии с физическими объектами легко находить URI страниц, связанных с этими объектами.

Другая попытка использовать Web для связи с объектами реального мира заключалась в том, чтобы объединить «умные вещи» со стандартизированной архитектурой Web-сервиса, использующей такие традиционные стандарты, как SOAP, WSDL, UDDI [Guinard, 2010]. Но на практике это оказалось слишком сложно и неоправданно для простых объектов.

Поэтому вместо SOAP и WSDL, используемых

в Web-сервисах, была предложена технология "Web вещей" на основе вложенного протокола передачи гипертекста (HTTP) для серверов и Web 2.0 [Luckenbach, 2005].

Необходимые для этого Web-серверы с продвинутыми функциями (такими как альтернативные соединения или назначение сервера для уведомления о событиях) могут быть реализованы всего с 8 КБ памяти и без поддержки операционной системы. Кроме того, благодаря эффективной оптимизации стека протоколов TCP/HTTP, эти Web-серверы могут быть установлены на таких миниатюрных встроённых системах, как смарт-карты и др.

Так как внедрённые Web-серверы, которые используются в Интернете вещей, как правило, имеют в наличии значительно меньше ресурсов по сравнению с такими клиентами Web, как браузеры или мобильные телефоны, то асинхронный JavaScript и XML (AJAX) оказался хорошим способом передавать часть рабочей нагрузки от сервера к клиенту.

4. Физические компоненты WoT

Основная цель WoT заключается в том, чтобы обеспечить возможность людям общаться с объектами и физическими объектами, которые находятся вокруг них. Эти объекты могут также взаимодействовать друг с другом, а затем отправлять людям информацию на их смартфоны. Взаимодействие людей с физическими объектами осуществляется с помощью датчиков, которые могут получать информацию, передавать информацию и выполнять действия, требуемые человеком или компьютерами и объектами. Если поместить датчик в районе жилого дома или прикрепить его к физическому объекту или вещи, то он сможет отправлять информацию на телефон пользователя или на любой другой объект или объект.

Объектом в данном случае может быть любой объект в доме или офисе (например, дверь, светильник, окна, ключи от квартиры), устройство (кофеварка, стиральная машина, автомобиль) или же абстрактный объект, например, человек или животному.

Если поместить датчик на любую такую «объект» и иметь возможность общаться с ней, то она становится «умной» *вещью (smart thing)*.

На рисунке 1 представлена обобщённая схема взаимодействия пользователя с объектами через смартфон посредством Web-интерфейса.



Рисунок 1 - Объединение объектов через WoT

Если поставить датчик на объект, то он может отправлять сообщение, содержащее информацию о его состоянии, местоположении, включен он или выключен, открыт он или закрыт и т.д. Пользователь может использовать смартфон для отправки различных команд к своим объектам и указывать им, что делать. Например, при помощи датчиков можно узнать, когда ребенок возвратится домой из школы и выключен ли свет в гостиной, включить электроплиту, заблокировать или разблокировать входную дверь.

Важно, что при этом объекты могут взаимодействовать друг с другом: например, когда ребенок приходит со школы домой, то датчик на его рюкзаке автоматически посылает сообщение на смартфон об этом, а тот посылает сообщение о разблокировке замка на входной двери.

На рисунке 2 представлена иерархическая структура электронных устройств, используемых пользователями для подключения, мониторинга и управления окружающими объектами.



Рисунок2 - Иерархическая структура устройств, используемых для подключения объектов в IoT и WoT

Чтобы сообщество «умных» устройств или объектов было жизнеспособным, следует наделить каждую объект определенным интеллектом, помогающим ей распознать контекст своего

окружения (местоположение, собственное состояние и положение контролируемого объекта), а также наладить диалог с окружающими объектами и управляющим устройством [Vermesan, 2012].

Данные, собранные разными датчиками и устройствами по обыкновению являются многомодальными (температура, индикатор, звук, видео и т.п.). Разнообразие, изменчивость и разнотипность данных реального мира WoT вызывает многие проблемы с их обработкой, интегрированием и интерпретацией. Поэтому динамическая и ограниченная ресурсами природа WoT требует специальной архитектуры представления знаний и обработки, которая должна быть учтена технологиями семантического Web, такими как онтологический анализ, семантические метаданные, связанные данные (linked data), семантические Web-сервисы и социальные сети.

5. Технологические аспекты реализации Web of Things

Одним из первых прототипов Web of Things был проект, в котором датчики использовались для мониторинга и контроля за потреблением электроэнергии бытовых приборов [Web вещей, 2013], чтобы на домашнем компьютере пользователь мог визуально контролировать потребление электроэнергии.

Реализация функций человеко-машинного интерфейса при этом осуществлялась на основе интерфейса прикладного программирования RESTful API (Representational State Transfer - «Передача Репрезентативного Состояния») [Fielding, 2000].

Web-сервис REST реализует метод взаимодействия компонентов распределённого приложения в IoT, при котором вызов удаленной процедуры представляет собой обычный HTTP-запрос (обычно GET или POST; такой запрос называют REST-запрос), а необходимые данные передаются в качестве параметров запроса [Pautasso, 2008]. Этот способ является альтернативой более сложным методам, таким как SOAP, CORBA и RPC.

Другими словами REST означает концепцию построения распределённого приложения, при которой компоненты взаимодействуют наподобие взаимодействия клиентов и серверов в Web. Системы, поддерживающие REST, называются RESTful-системами.

Концепцию построения распределённого приложения на основе REST предполагает, что каждый запрос (REST-запрос) клиента к серверу содержит в себе исчерпывающую информацию о желаемом ответе сервера (желаемом репрезентативном состоянии), и сервер не обязан

сохранять информацию о состоянии клиента («клиентской сессии»).

Можно выделить ряд важных преимуществ REST Web-сервисов по сбору, доставке и передаче данных от/к умным объектам в Web of Things в отличие от традиционных SOAP Web-сервисов:

- надёжность (за счет отсутствия необходимости сохранять информацию о состоянии клиента, которая может быть утеряна);
- производительность (за счет использования специального кэша);
- масштабируемость; прозрачность системы взаимодействия, особенно необходимая для приложений по обслуживанию сети;
- простота интерфейсов; портативность компонентов; легкость внесения изменений;
- способность эволюционировать, приспосабливаясь к новым требованиям (на примере Web).

Таким образом, в архитектура REST очень проста в плане использования. По виду пришедшего запроса сразу можно определить, что он делает, не разбираясь в форматах (в отличие от SOAP, XML-RPC). Данные передаются без применения дополнительных слоев, поэтому REST считается менее ресурсоемким, поскольку не надо парсить (автоматически обрабатывать или разбирать с целью получения нужных данных) запрос чтоб понять что он должен сделать и не надо переводить данные из одного формата в другой.

Но важным моментом здесь остается информационная безопасность – недопустимо кому угодно позволять изменять информацию, то есть нужна еще авторизация и аутентификация. Эта проблема решается различными методами, например, при помощи различного типа сессий или просто HTTP Authentication.

6. Концептуальная модель архитектуры Web of Things

WoT является составной частью IoT: он предоставляет возможность мониторинга и управления объектами с помощью страниц WWW. На рис. 3 приведена структура организации WoT, из которой видно, что ключевую роль в нем играют два уровня: интеллектуальный уровень и уровень приложений, позволяющие использовать через Web как для изначально приспособленные к этому объекты (Web enabled things), так и для те объекты (non-Web enabled things), для которых необходимы соответствующие согласующие устройства (шлюзы), например, объекты, функционирующие по протоколам ZigBee или Bluetooth.

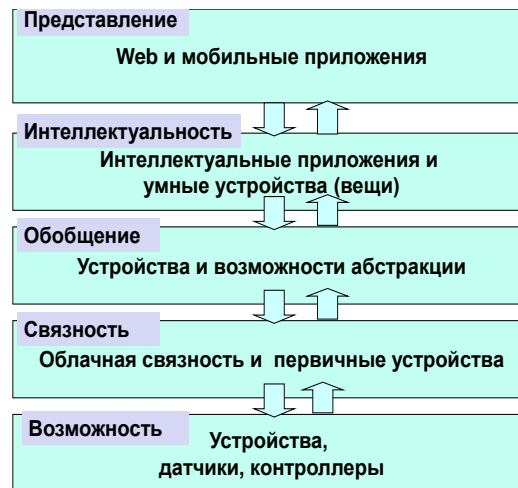


Рисунок 3 - Архитектура Web of Things как части Internet of Things

На рисунке 3 представлена пятиуровневая модель концептуальной архитектуры платформы [Guinard, 2011] Web of Things для взаимодействия человека с «умными» вещами. Уровень интеллектуальных приложений и умных вещей содержит базирующиеся на семантических технологиях (Semantic Web) программы, реализующие функции, поиска, сопоставления онтологий, логического вывода.

Для того, чтобы эффективно использовать преимущества, которые предоставляет сообщество "умных" вещей, нужно наделить каждую объект, входящую в это сообщество, определенным интеллектом (знаниями и правилами их использования), который должен помочь ей распознать контекст своего окружения (местонахождение, собственное состояние и положения объекта, контролирующего эту объект), а также наладить диалог с окружающими объектами и управляющим устройством [Gladun et al., 2009].

Онтологический подход становится тем ядром, который обеспечивает семантическое описание (виртуальную модель знаний) об этих объектах, которая может многократно использоваться разными приложениями.

Кроме того, использование онтологий для объектов WoT обеспечивает всю необходимую семантику для спецификации различных устройств, используемых в технологиях IoT/WoT, равно как и для спецификаций задач, решаемых непосредственно в приложениях WoT (ввод, вывод, логика управления), которые создаются на основе использования информации об этих устройствах.

Для создания целостной системы развертывания приложений WoT в определенной сфере деятельности необходима разработка формальной онтологической модели объектов (вещей), входящих в состав WoT.

Архитектура модели объектов WoT должна включать:

1. Анализ структуры знаний предметной области WoT, основных объектов и отношений:

- Использование мереологических (mereological) и онтологических методов для формирования терминологии домена WoT и структуры знаний;
- Структура OWL-онтологии.

2. Многократное использование существующих тезаурусов, таксономий и онтологий домена WoT:

- Семантический поиск релевантных объектов и анализ средств представления знаний и стандартов в WoT;
- Краткий обзор соответствующих онтологий и других структур знаний;
- Интеграция существующих таксономий и онтологий домена.

3. Архитектура методов для автоматизированного усовершенствования формальной онтологической модели WoT:

- Архитектура методов для автоматизированной добычи знаний (термины и отношения) из текстов естественного языка, которые касаются домена WoT;
- Методы автоматизированной лингвистической обработки текстов естественного языка;
- Усовершенствованная OWL-онтология .

4. Семантический поиск в домене WoT на основе > онтологии домена:

- Семантический поиск объектов WoT;
- Методы семантического поиска объектов WoT;
- Методы семантического поиска сервисов WoT (RESTfull);
- Рекомендации относительно использования WoT объектов.

Данные датчиков, которые связаны с разными событиями и ситуациями могут быть проанализированы и преобразованы в активное знание, которое позволяет нам лучше понимать физический мир и создать больше продуктов с добавленной стоимостью и сервисов, например, учет расстояний линий связи в smart grid может быть использован для того, чтобы лучше предусмотреть и балансировать потребляемой мощностью; анализируя комбинацию транспортных потоков, загрязнение, состояние погоды и перегрузки, можно создать знание на основе данных от датчиков, позволяя при этом обеспечить лучший трафик транспорта и муниципальное управление; контроль и обработка данных от датчиков и объектов, прикрепленных пациентам или людям преклонного возраста могут обеспечить лучшее медицинское обслуживание пациентов на расстоянии.

7. Архитектура WoT

Основная цель WoT – обеспечить наиболее эффективное принятие решений за счет более глубокого анализа наблюдения за окружающей средой.

Чтобы достигнуть этого, необходимо ввести несколько информационных уровней между сенсорами и блоком принятия решений (рис.4).

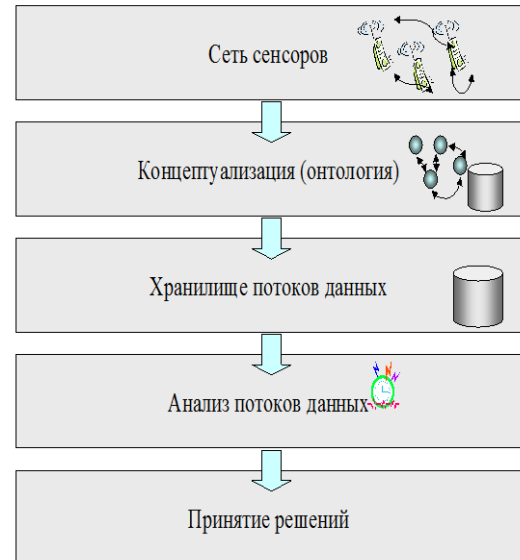


Рисунок 4 - Архитектура WoT

При этом возникает проблема концептуализации домена сенсоров. Ее решение связано с использованием онтологий и требует исследования семантической сети сенсоров (рисунок 5).

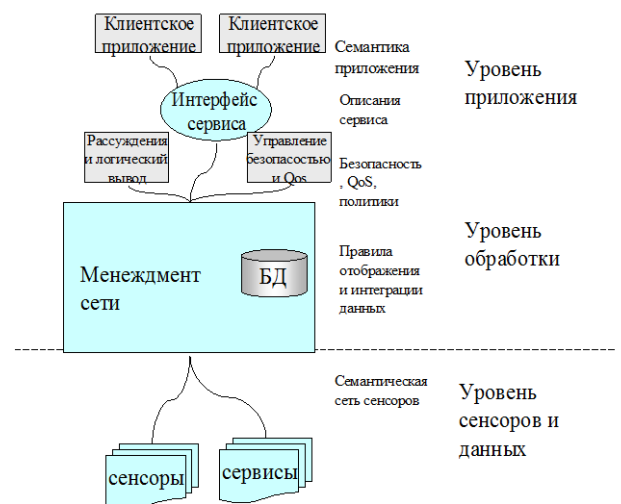


Рисунок 5 - Архитектура семантической сети сенсоров.

Некоторые из существующих онтологий покрывают домен сенсоров, но большинство из них – только его часть. Онтология W3C Semantic Sensor Network (SSN) представляет собой попытку покрыть весь этот домен полностью.

Таблица 2- Онтологии сенсоров.

онтология	Основные понятия
Avancha	Сенсор
CESN	Сенсор
CSIRO	Сенсор, процесс
Eid	Сенсор
Kim	Сенсор
Matheus	Система, сенсор
MMI	Сенсор, процесс
OntoSensor	Компонент, сенсор
OOSTethys	Компонент, система, процесс
SWAMO	Агент, процесс, сенсор

Кроме того, необходимые знания могут извлекаться из различных онтологий предметных областей, доступ к которым обеспечивают различные репозитории онтологий [Гладун и др., 2013].

Но, несмотря на наличие онтологий, извлечение полезных для пользователей сведений из потоков данных от сенсоров остается сложной задачей (рис.6).

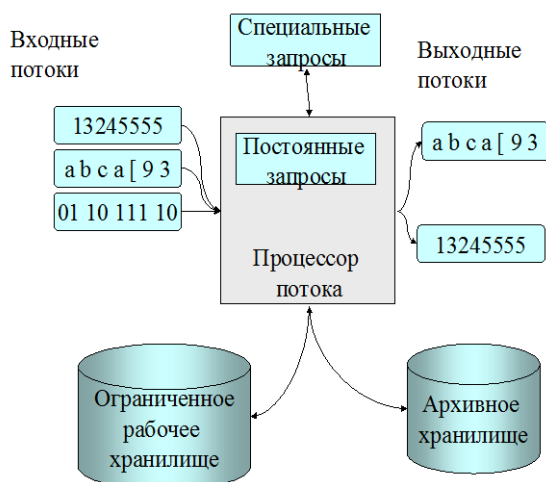


Рисунок 6 - Обобщенная архитектура извлечения знаний из потоковых данных

Это достаточно сложная задача, в которой можно выделить несколько отдельных проблем:

- поиск WoT-объектов, обладающих необходимыми возможностями пользователю;
- исследование свойств WoT-объектов, обладающих необходимыми возможностями;
- исследование потоков данных, передаваемых сенсорами и встроенными устройствами WoT-объектов, обладающих необходимыми возможностями;
- анализ свойств передаваемых потоков данных.

8. Средства вывода над онтологиями

Анализ современных методов и средств обработки (добычи) знаний и языков, для формирования запроса к онтологии позволил разработать подсистему интеллектуальных приложений на четвертом уровне модели WoT. Для добычи и обработки знаний, которые хранятся в онтологиях, используются блоки логического вывода (ризонеры, reasoners), которые играют ключевую роль в интеллектуальных системах, ориентированных на знание.

Существует множество реализаций процессоров логического вывода (reasoning engine) для OWL онтологии, которые различаются по возможностям, областях применения и качества выполнения задач. Обобщенный анализ позволяет разделить их на три группы в зависимости от метода реализации:

1. *Табличные DL-процессоры.* Традиционно были разработаны первыми для решения подобных задач. Имеют низкую производительность, но способны делать умозаключения на сложных онтологиях с множеством нетривиальных конструкций. До этого класса относятся ризонеры Pellet, RacerPro, FacT++, а также Hermit и SHER.

2. *Дизъюнктивные Datalog-процессоры.* Трансформируют онтологию в дизъюнктивную Datalog программу и используют методику дедуктивных баз данных и правило резолюций. Такие процессоры имеют удовлетворительное быстродействие при использовании некоторых оптимизаций, однако не поддерживают определенные OWL-конструкции, в частности кардинальные ограничения и номиналы. К этой группе принадлежит KAON2.

3. *Процессоры правил.* Используют системы обработки правил для умозаключений на онтологиях. Имеют высокую производительность, но могут обрабатывать лишь простые онтологии, лишенные многих важных конструкций. Представители этой группы: Sesame/OWLIM, Jena, Owljesskb.

Существуют два подхода к реализации логического вывода: на базе правил (с использованием алгоритмов forward-chaining и/или backward-chaining) и на базе семантического табло (semantic tableau). На базе правил реализованы Semantics, SDK и Owlrim, а на базе семантического табло - Pellet.

В качестве системы логического вывода для дескриптивной логики нами была избрана свободно распространяемая система Pellet версии 2.0.0. Система Pellet реализует логический вывод для дескриптивной логики класса ALCQNI (D), которая расширяет атрибутивный язык (AL) такими возможностями, как произвольное выражение отрицания, транзитивные отношения, инверсные отношения, иерархия отношений, количественные ограничения на отношения и некоторые конкретные домены.

На сегодняшний день система Pellet реализует наиболее выразительную дескриптивную логику с использованием высокопроизводительного алгоритма (tableau-based algorithm) логического вывода, который используется для обработки онтологий, описанных языком OWL DL.

OWL позволяет представлять знания об предметной области в виде онтологий, которые можно использовать и обрабатывать в разных приложениях; язык OWL запросов SPARQL позволяет создавать метаописания RDF и онтологий.

Разработанная WoT-онтология имеет цель поддержки процесса автоматизированного развертывания интеллектуальных приложений в гетерогенных средах WoT. WoT-онтология используется для поддержки семантического реестра WoT-объектов и позволяет:

- а) скрыть технологическую разнородность, которая характерна множеству гетерогенных объектов WoT;
- б) скрыть семантическую разнородность, присущую используемым гетерогенным онтологиям домена для того, чтобы семантически аннотировать данные умных объектов WoT.

Рассмотрим более детально первый этап этой задачи – поиск в пространстве WoT устройств, необходимых пользователю. При этом пользователь может специфицировать как функции или свойства самого физического объекта, связанного с соответствующей WoT-объектом, так и характеристики связанных с ним сенсоров и получаемых от них сведений либо предоставляемых ими сервисов.

При этом пользователь может применять термины из каких-либо доступных ему онтологий, описывающих пространство WoT и соответствующую предметную область (с ссылкой на сами онтологии). Но в описании IoT-объекты могут использоваться термины из других онтологий, относящихся к близким доменам.

Поэтому возникает проблема сопоставления онтологий. В общем случае эта задача крайне сложна и трудоемка, но при наличии ряда ограничений на сопоставляемые онтологии ее можно решить за приемлемое время.

9. WoT-объект как семантический Web-сервис

Web-сервисы являются наиболее современной попыткой реконструировать крупномасштабные распределенные вычисления. Они базируются на стандартах, которые действуют на синтаксическом уровне и не имеют возможностей для представления семантики. Семантика обеспечивает более качественные и масштабируемые решения для таких областей, как интероперабельность

сервисов, обнаружение и композирование сервисов и оркестровка процессов [Talantikite et al, 2009] .

Предлагается рассматривать WoT-объект как семантический Web-сервис. При этом основное внимание уделяется не использованию стандартов сервис-ориентированной архитектуры (что слишком сложно для WoT-объектов), а принципам создания и обработки семантической разметки на основе онтологий и поиску в пространстве этих описаний [Gladun et al., 2014] .

Web-сервисы представляются перспективным базисом для того, чтобы обеспечить решение для интероперабельности в разнородных средах. Они базируются на расширяемом языке разметки (XML), который представляет собой базовую технологию Web-сервисов. Тем не менее, описываются Web-сервисы (как правило, синтаксически) по таким стандартам, как UDDI, SOAP и WSDL.

В предлагаемом подходе данные используются для того, чтобы найти соответствие (matching) между терминами, используемыми в запросе о WoT, и терминами аннотации соответствующей WoT-объекты, представленной его поставщиком. Этот подход основан на архитектуре представления WoT-объектов как Web-сервисов путем добавления семантики в их описание, а также в запрос клиента [Рогушина и др., 2006] .

Web-сервисы предоставляют средства для того, чтобы разбивать на модули программное обеспечение таким образом, что его функциональность могла бы быть описана, обнаружена и развернута платформо-независимым образом по сети (например, интранет, экстранет и Интернет).

Предлагаемый подход к исследованию WoT основан на процессе аннотации, которая состоит в двух этапов [Bouchiha et al., 2012]:

- этапа категоризации, который позволяет классифицировать WoT-объект в соответствующий им домен;
- этапа сопоставления, который позволяет связать каждую сущность из WoT с соответствующей сущностью в онтологии домена.

Процесс аннотирования WoT-объектов опирается на методы сопоставления онтологий, которые, в свою очередь, используют некоторые меры подобия (similarity measures).

И категоризация, и сопоставление используют методы сопоставления онтологий – как специфичных для WoT, так и связанных с предметной областью [Гладун и др., 2006] . Например, при описании подключенной через IoT кофеварки может использоваться онтология бытовой техники, онтология WoT-сенсоров и персональная онтология пользователя,

описывающая, какие напитки он предпочитает в разное время суток.

Цель сопоставления всех этих онтологий – найти отношения между сущностями, выраженные в различных онтологиях, и определить, насколько они семантически близки. Очень часто эти отношения являются отношениями эквивалентности, которые обнаруживаются через меру сходства между сущностями онтологий (рис.7).

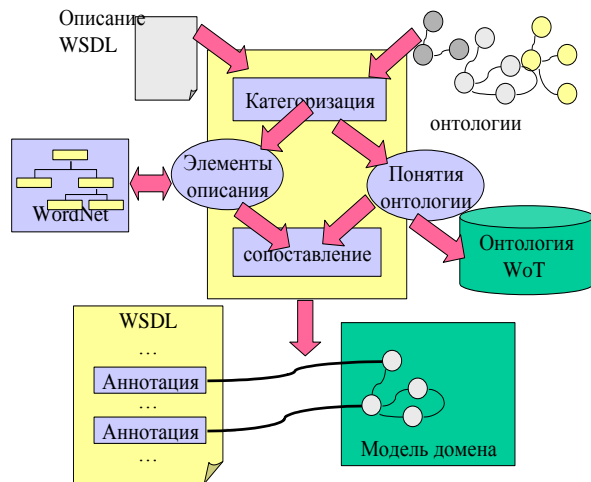


Рисунок 7 - Процесс аннотирования

Именно сопоставление онтологий является основой для поиска в пространстве объектов, семантически размеченных терминами различных онтологий [Рогожина, 2014].

Для завершения процесс сопоставления онтологий соответствия использует меры сходства (similarity measure) между объектами. Мера сходства направлена на количественную оценку того, насколько две сущности подобны. Формально она определяется следующим образом:

Сходство объектов можно определить следующим образом: для набора сущностей O сходство $\sigma : O \times O \rightarrow R$ – функция от пары сущностей из O в реальное число, выражающая сходство между двумя объектами так, что выполняются следующие свойства:

- 1) положительность

$$\forall x, y \in O, \sigma(x, y) \geq 0;$$

- 2) максимальность

$$\forall x \in O, \forall y, z \in O, \sigma(x, x) \geq \sigma(y, z);$$

- 3) симметрия

$$\forall x, y \in O, \sigma(x, y) = \sigma(y, x).$$

В качестве меры подобия можно использовать меру на основе Wordnet [Pedersen et al., 2004]. WordNet представляет собой лексическую базу данных. Таким образом, эти меры вычисляются, а затем нормализуются. Нормализация заключается в

инверсии значения меры для получения нового значения между 0 и 1. Значение 1 указывает, что между двумя структурами есть полная семантическая эквивалентность.

Меры подобия, зависящие от WordNet, можно разделить на три категории:

- (1) на основе длины пути между понятиями;
- (2) на основе информационного содержания;
- (3) на основе типа отношений между понятиями.

Когда доступен набор онтологий, сходство между двумя наборами вычисляется путем сравнения набора сущностей описания WoT-объекты и набора сущностей каждой онтологии. На основе таких мер системы решают, между какими онтологиям запустить алгоритм сопоставления. Выбранная онтология предметной области определяет категорию WoT-объекты. Этот процесс называется процесс категоризации WoT-объекты.

Онтология рассматривается как набор сущностей (понятий), и описание WoT-объекты – также как набор сущностей (типов данных, функций, типов сенсоров, интерфейса, операций, сообщений и т.д.).

Некоторые стратегии могут быть адаптированы для вычисления сходства между двумя наборами. Затем мы определяем стратегии отдельной связи (Single linkage), полной связи (Full linkage) и средней связи (Average linkage).

Важно отметить, что, так как онтологии базируются на дескриптивных логиках (DL), то при сопоставлении семантической разметки на основе онтологий можно применять методы логического вывода: для поиска Web-сервисов, или для WoT-объектов, представленных в виде Web-сервисов, отдельные сервисы описываются в терминах DL, а потом над ними осуществляется логический вывод [Di Noia et al., 2007].

Как правило, для логического вывода на основе DL используют отношения категоризации (subsumption) и выполнимости (satisfiability) для поиска соответствия потенциальных соответствий.

Выводы

Современный этап Web-технологий и, в частности, перспективы развития WoT, требуют формирования соответствующих методов и технологий для их эффективного использования. Представляется целесообразным применять для этого уже существующие и доказавшие свою полезность разработки, такие как Semantic Web и семантические Web-сервисы, модифицируя их в соответствии со спецификой использования объектов реального мира, связанных через IoT.

Разработанные алгоритмы и модели семантического распознавания информационных объектов на основе онтологического представления

знаний об этих объектах для дальнейшего их использования в интеллектуальных информационных технологиях и приложениях требуют дальнейшего развития и усовершенствования с учетом гетерогенности среды Web of Things.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

[Гладун и др., 2006] Гладун А.Я., Рогушина Ю.В. Онтологический подход к поиску веб-сервисов в распределенной среде Интернет // Информатика, Минск, № 4, 2006. – С.116-127.

[Гладун и др., 2013] Гладун А.Я., Рогушина Ю.В. Репозитории онтологий как средство повторного использования знаний для распознавания информационных объектов // Онтология проектирования, № 1 (7), 2013. – С.35-50.

[Рогушина, 2014] Рогушина Ю.В. Знание-ориентированные средства поддержки семантического поиска в Web. – LAP LAMBERT Academic Publishing. 2014. – 214 с. – ISBN 978-3-659-56520-5.

[Рогушина и др., 2006] Рогушина Ю.В., Гладун А.Я. Онтологическая модель интеллектуализации сервис-ориентированных вычислений в распределенной среде Интернет // Проблемы програмування, №2-3, 2006. – С.526-536.

[Bouchiha et al., 2012] Bouchiha D., Malki M. Semantic Annotation of Web Services // Proc.ICWIT, 2012.

[Di Noia et al., 2007] Di Noia T., Di Sciascio E., Donini F.M. Semantic Matchmaking as Non-Monotonic Reasoning: A Description Logic Approach // Journal of Artificial Intelligence Research 29, 2007. – P. 269-307.

[Fielding, 2000] Fielding Roy. Architectural Styles and the Design of Network-based Software Architectures (2000), Dissertation – <http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm>.

[Gladun et al., 2009] Gladun A., Rogushina J. Use of Semantic Web technologies in design of informational retrieval systems // in Book “Building and Environment”, 2009 Nova Scientific Publishing, New-York, USA. – P.89-103.

[Gladun et al., 2014] Gladun A., Rogushina J. Intelligent Techniques of User-Oriented Recognition of Objects from the Web Informational Resources // Journal of Emerging Technologies in Web Intelligence, Vol 6, No 3, 2014. – P. 348-353.

[Guinard, 2010] Guinard D, Fischer M, Trifa V (2010a) Sharing Using Social Networks in a Composable Web of Things. Proceedings of the 1st IEEE International Workshop on the Web of Things (WoT 2010) at IEEE PerCom, Mannheim, Germany.

[Guinard, 2011] Guinard Dominique A Web of Things Application Architecture – Integrating the Real-World into the Web. A dissertation submitted to ETH Zurich, 2011.

[Kindberg,2002] Kindberg T, Barton J, Morgan J, Becker G, Caswell D, Debaty P, Gopal G, Frid M, Krishnan V, Morris H, Schettino J, Serra B, Spasojevic M (2002) People, places, things: web presence for the real world. Mob Netw Appl 7:365-376.

[Luckenbach, 2005] Luckenbach T, Gober P, Arbanowski S, Kotsopoulos A, Kim K (2005) TinyREST – A protocol for integrating sensor networks into the internet. Proceedings of the Workshop on Real-World Wireless Sensor Network: SICS. Stockholm, Sweden.

[Machine-to-machine, 2011] Machine-to-machine (M2M) – the rise of machine// White Paper, Juniper, 2011.

[Pautasso, 2008] Pautasso Cesare; Zimmerman Olaf; Leymann Frank.. RESTful Web Services vs. Big Web Services: Making the Right Architectural Decision// in Proceedings of the 17th International World Wide Web Conference (WWW2008) - <http://www.jopera.org/docs/publications/2008/restws>.

[Pedersen et al., 2004] Pedersen T., Patwardhan S., Michelizzi J. WordNet: Similarity – measuring the relatedness of concepts // Proc. of the Nineteenth National Conference on Artificial Intelligence (AAAI-04), 2004. – P.1024-1025.

[Talantikite et al, 2009] Talantikite H. N., Aissani D., Boudjlida N. Semantic Annotations for Web Services Discovery and Composition // Computer Standards & Interfaces Journal, Elsevier, CSI, 31:1108-1117, 2009. – P.1108-1117.

[Vermesan, 2012] Vermesan Ovidiu and Friess Peter Internet of Things: Converging Technologies for Smart Environments and Integrated Ecosystems.– River Publishers, 2012.– 363 p.

[Web вещей, 2013] Web вещей визуализация энергопотребления – <http://www.webofthings.org/?s=energie+visible>.

SEMANTIC APPROACH TO THE WEB OF THINGS OBJECTS INTEGRATION

Rogushina J. *, Gladun A. **

* *Institute of Software Systems of the National Academy of Sciences of Ukraine, Kyiv, Ukraine*

ladamandraka2010@gmail.com

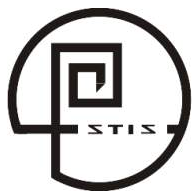
** *International Research and Training Center of Information Technology and Systems of National Academy of Sciences Ukraine, Kiev, Ukraine*

glanat@yahoo.com

Development of new trends in the formation of information space, in particular – Web of Things (WoT), requires the new knowledge-oriented methods of identification, retrieval and research of the specific to this space information objects. Use of ontological analysis For creating of methods and semantic search services based on ontologies in the Web of Things information space (on base of the Internet of Things) is proposed.

In order to integrate all the benefits of Web-services technology to the WoTspace, an approach to annotate WoT-things descriptions by ontological models and semantic Web-services is proposed.

Keywords: Web of Things, Internet of Things, semantic Web-service, ontology, semantic annotation.



OSTIS-2015

(Open Semantic Technologies for Intelligent Systems)

UDK 681.3.

РАЗРАБОТКА СРЕДСТВ ПЕРСОНИФИКАЦИИ ИНТЕЛЛЕКТУАЛЬНЫХ WEB-ПРИЛОЖЕНИЙ

Рогушина Ю.В.

Институт программных систем НАН Украины, Киев, Украина

ladamandraka2010@gmail.com

Предложен основанный на онтологическом анализе подход к персонификации интеллектуальных Web-приложений, который направлен на извлечение и использование знаний об информационных потребностях и особенностях восприятия информации пользователем. В качестве примера рассматривается семантический поиск в Web.

Ключевые слова: онтология, знания, семантический поиск, персонификация, самообучение.

Введение

В последнее время большинство прикладных разработок в сфере ИТ непосредственно связано с развитием интеллектуализации и персонификации различных приложений. В развитых интеллектуальных информационных системах (ИИС) поддерживается интеллектуальный адаптивный интерфейс с пользователем, который для более эффективного взаимодействия способен учитывать как контекст взаимодействия, так и особенности конкретного пользователя. Другим важным аспектом семантических технологий проектирования интеллектуальных систем является персонификация контента, предлагаемого различным пользователям.

Для этого возникает необходимость получения и обработки знаний – как о самих пользователях, так и о специфике той предметной области (Про), к которой относится решаемая пользователем при помощи ИИС конкретная задача.

Постановка задачи

Для того, чтобы разрабатывать эффективные средства персонификации интеллектуальных информационных систем, необходимо определить, какие именно сведения о пользователе могут улучшить функционирование ИИС; из каких источников можно извлечь эти сведения; с помощью каких методов доступные сведения могут быть преобразованы в пригодные для автоматической обработки знания; в какой форме их представить, чтобы облегчить их повторное использование; и как именно структурировать получаемую информацию

о пользователе с учетом предметной области решаемой задачи. Для решения этих вопросов предлагается использовать онтологическую модель информационного взаимодействия пользователя с ИИС, ориентированную на привлечение знаний из внешних онтологий, тезаурусов и распределенных ресурсов Web.

Источники персонифицированных сведений о пользователе ИИС

Рассмотрим подробнее, из каких источников ИИС может получать персонифицированные сведения о пользователях.

В первую очередь, это те сведения, которые пользователь сам явным образом сообщает о себе ИИС – при регистрации или затем в процессе взаимодействия с системой. С одной стороны, это наиболее простой для ИИС способ получить такие сведения, так как при этом основная часть усилий приходится на того, кто хочет воспользоваться ИИС. В таком случае разработчикам надо лишь предусмотреть, какая именно информация может быть полезна для адаптации ИИС к потребностям пользователя, и обеспечить механизм для ее ввода – соответствующие анкеты, тесты и т.п. к недостаткам этого способа можно отнести его трудоемкость для пользователя и недостаточную достоверность сведений, предоставляемых пользователем (например, пользователь может зависить свою квалификацию или неправильно понять вопрос).

Более объективные сведения о пользователе можно получить в процессе его тестирования. В частности, тесты позволяют определить психофизиологические особенности пользователя,

связанные со способностью к восприятию информации (например, экстраверт и интроверт по-разному оценивают различные формы представления информации), его скорость восприятия информации, способность к пониманию сложных информационных объектов (например, различные критерии легкости чтения) и эрудицию. Но при таком подходе пользователю придется потратить достаточно много времени для предоставления этих сведений. Это целесообразно только в том случае, если пользователь намерен одновременно взаимодействовать с ИИС либо если для него очень важно получить от ИИС результат, учитывающий его персональные особенности.

При персонификации ИИС важно правильно оценивать и учитывать для дальнейшей работы психофизиологические свойства пользователя, влияющие на его способность к восприятию информации

В большинстве ИИС эти свойства не используются, поэтому рассмотрим ее более детально. Сейчас существует много психологических тестов, позволяющих достаточно четко классифицировать личность (например, тест Айзенка, ММРІ), а затем на основе такой классификации охарактеризовать (с точки зрения формы представления информации, а не собственно контента) те ИР, которые в большей степени могут удовлетворить такого пользователя. Например, достаточно распространена классификация личностей по скорости возбуждения и торможения, в соответствии с которой выделяют четыре типа – холерик, сангвиник, флегматик и меланхолик, подразделение людей на экстравертов и интровертов и т.д. Социопсихофизиологические исследования обеспечиваются рядом компьютерных систем, которые используются для получения индивидуального адаптационного социопсихофизиологического портрета человека в различных сферах его жизнедеятельности и рассматривают проблемы создания для пользователя комфортного индивидуального адаптационно-информационного пространства [Кузьмина и др., 2012].

Такие индивидуально-типологические свойства, как сила, подвижность, уравновешенность нервных процессов, тип реагирования, эмоциональный фон, скоростные свойства анализаторных систем, особенности высших психических функций (восприятие, внимание, память, мышление, интеллект); поведение в стрессе, тип оптимальной работоспособности, могут в значительной мере определить выбор пользователя при восприятии информации – и, соответственно, при оценке предложенных ИР.

Например, для экстраверта положительными чертами ИР являются наличие иллюстраций, использование коротких простых предложений, эклектичный дизайн, применение ярких цветов для выделения важных элементов и т.д., а для интроверта такими положительными чертами

являются сложная структурированность текста, сложноподчиненные предложения, пастельная цветовая палитра. Поэтому можно предположить, что близкие по контенту ИР с различным оформлением по-разному будут оценены экстравертами и интровертами. Включение в число анализируемых при индуктивном выводе свойств психофизиологических параметров пользователя позволит находить более точные обобщающие правила.

Еще один источник персонифицированной информации – социальный Web. Сведения о пользователе могут быть извлечены из различных социальных сетей, Wiki-ресурсов, блогов и т.д. кроме того, ряд сведений о пользователе может быть извлечен из профилей пользователя в других ИИС. Это связано с тем, что даже при решении совершенно разных задач основные свойства личности пользователя (характер, темперамент, скорость реакции) остаются неизменными (или изменяются во времени пренебрежимо медленно), и потому допустимо повторное использование накопленных знаний в других системах. Кроме того, анализ того, какими именно приложениями оперирует пользователь и какие запросы к поисковым системам он выполняет, тоже может расширить представления ИИС о пользователе.

Но в различных ИИС похожие характеристики пользователя могут описываться различными терминами. Для обеспечения интероперабельности персональных данных можно использовать онтологический подход, интегрируя сведения на основе сопоставления онтологий различных ИИС.

Важным источником персонифицированных сведений о пользователе может послужить опыт его взаимодействия с ИИС. Именно в этих сведениях неявным образом содержатся различные предпочтения и требования пользователя, конкретизированные именно для данной ИИС. Именно поэтому извлеченные из таких сведений знания наиболее полезны для повышения эффективности работы ИИС. Например, если пользователь регулярно отказывается от каких-то рекомендаций ИИС, имеющих ряд общих свойств, то выявление этих свойств позволит в дальнейшем не предлагать ему такие рекомендации, что сэкономит и время пользователя, и вычислительные ресурсы ИИС. Так, если при покупке одежды в Интернет-магазине пользователь всегда отказывается от красных и зеленых товаров, то в результате анализа ИИС перестанет предлагать ему предметы таких цветов, даже если они соответствуют явно заданным условиям пользователя о размере и цене. Чтобы извлечение таких сведений осуществлялось эффективно, важно, с одной стороны, разработать в ИИС эффективные методы самообучения, а с другой – предусмотреть удачное структурирование сведений об обрабатываемых информационных объектах (если у товара отсутствует параметр «цвет», то даже самые совершенные методы самообучения не смогут

выделить правила, относящиеся к группе предметов зеленого цвета).

Именно онтологии могут послужить основой для такого структурирования. Значительную часть терминов, используемых для описания обрабатываемых в ИИС объектов, можно извлечь из онтологии соответствующей ПрО, а знания о связях между различными элементами – из онтологии задачи.

Основные элементы взаимодействия ИИС с пользователем

В наиболее общем виде элементы ИИС относительно взаимодействия с пользователем можно разделить на следующие составляющие:

- интерфейс ИИС с пользователем;
- контент, предоставляемый пользователю;
- хранилище, в котором накапливается опыт взаимодействия ИИС с пользователем;
- модель информационного взаимодействия ИИС с пользователем, обеспечивающая структурное представление знаний об основных элементах ИИС, необходимых для обеспечения потребностей пользователя.

Во всех этих составляющих для персонификации можно применять онтологическое представление знаний – как для внутренних знаний ИИС, закладываемых в нее при разработке, так и для тех знаний, которые импортируются ИИС из внешней среды или извлекаются ИИС из собственного опыта взаимодействия с пользователями.

Так как ориентация на конкретного пользователя и адаптация к его персональным возможностям и потребностям в значительной мере могут повысить эффективность работы ИИС, то целесообразно во всех этих составляющих использовать информационную модель пользователя и усовершенствовать ее на основе самообучения и поиска новых знаний.

Персонификация интерфейса ИИС

В [Курзанцева, 2010] отмечается, что сейчас много исследований (как украинских, так и иностранных ученых) посвящены по проблемам адаптации информационных систем к пользователям. Одним из наиболее распространенных методов решения этой проблемы является использование модели пользователя, причем увеличение сложности такой модели обеспечивает более высокую степень адаптации, что вызывает рост стоимости создания модели. В частности, такие подходы широко применяют при разработке поисковых и рекомендующих систем.

Модель стратегии построения интерфейса можно представить следующим образом: каждому пользователю предлагается единственная оптимальная конфигурация системы и сервисов, которая выводится из множества ответов

пользователя на вопросы интерфейса.

Если пользователь известен системе (режим 2), то подсистемой моделирования пользователя проводится анализ его последнего сеанса работы с системой и при соответствующих предусловиях происходит корректировка модели пользователя.

В [Яковлев и др., 2006] рассматривается использование онтологий при создании модели пользователя ИИС с адаптивным интерфейсом. Данный подход позволяет учесть совокупность понятий и связей между ними, имеющих место при взаимодействии пользователя с ИИС.

Решение задачи построения модели пользователя в предлагаемой постановке представляет собой выявление совокупности тех характеристик индивидуального пользователя (или классов пользователей), которые обеспечивают ИИС способностью приспосабливаться к потребностям, возможностям и целям пользователя. В целом при создании такой модели необходимо учитывать следующие процессы аспекты взаимодействия пользователя с ИИС: 1) накопление в ИИС персональной информации о пользователе; 2) наблюдение за пользователем в процессе работы; 3) определение персональных особенностей (свойств) пользователя; 4) прогнозирование действий пользователя на основе его свойств; 4) проверка предположений на согласованность; 5) использование в работе ИИС разработанных предположений.

При этом сама модель пользователя представлена в виде метаонтологии, содержащей такие понятия, как «информационная система», «требования к интерфейсу», «пользователь», «методы взаимодействия пользователя с системой», «методы формирования модели пользователя» и т.д.

На втором уровне в онтологии присутствуют свойства этих понятий. Например, для понятия «пользователь» это такие характеристики, как

- демографические характеристики;
- знания пользователя об ИИС;
- опыт пользователя при работе с ИИС;
- предпочтения пользователя в интерфейсе;
- цели и планы при решении задачи;
- заинтересованность в результатах;
- когнитивные особенности;
- специфические свойства пользователя (ограничения психологического, физического характера и т. д.).

Часто в качестве средства персонификации интеллектуальных информационных систем (ИИС) рассматривают их адаптивный интеллектуальный интерфейс.

В частности, в [Анохин, 2014] отмечается противоречивых подходов к организации адаптивного человеко-машинного интерфейса в современной литературе. Это обусловлено сложностью ИИС: с одной стороны, сложность

может стать причиной неожиданного и труднопредсказуемого поведения системы, а с другой стороны, именно благодаря своей сложности ИИС способна изменять свое поведение для адаптации к изменяющимся условиям.

Сложность системы представляет собой неформальное понятие, но можно выделить такие его признаки, как отсутствие формального математического описания; сложность наблюдения и управления вследствие большого числа процессов; независимость от субъекта; нестационарность, выражающаяся в динамическом изменении параметров; невозможность воспроизведения экспериментов.

Традиционно сложность систем описывают через их структурную и функциональную сложность: структурная сложность определяется связями между компонентами системы, а функциональная – динамическим поведением системы.

Наиболее простой мерой структурной сложности системы является количество входящих в нее элементов (если элементы системы образуют иерархическую структуру, то мера сложности может учитывать и количество уровней иерархии). Кроме того, в качестве меры сложности системы может служить и количество связей между ее элементами.

Такие меры достаточно просто использовать, если в основе ИИС лежит онтология: сложность системы зависит от количества терминов онтологии и количества отношений между ними.

Оценить функциональную сложность системы можно также на основании количества ее состояний. Однако определить количество реально достижимых состояний ИИС очень сложно.

Неотъемлемым признаком сложной системы, является адаптивность ее поведения, т.е. возможность целенаправленного изменения ее параметров и структуры в процессе функционирования. В [Rothrock et al., 2002] рассматриваются такие параметры интерфейса, которые могут изменяться в процессе адаптации, как содержание представляемой пользователю информации; форма представления информации и ведения диалога; распределение задач между человеком и машиной.

Адаптивный интерфейс ИИС позволяет системе самостоятельно, в результате анализа состояния и решаемых задач в текущей ситуации, настраивать отображение предоставляемой пользователю информации и доступных пользователю действий, соответствующих его текущим целям и возможностям. Текущая ситуация и решаемая задача являются факторами, порождаемыми объектом управления и средой деятельности.

Адаптивность интерфейса проявляется в фильтрации и упорядочении контента, предлагаемого пользователю, – как в зависимости

от предпочтений самого пользователя, так и под воздействием контекста и внешних факторов (например, продукт выгодно купить именно в данный момент) [Langley, 1999].

При описании формализованного адаптивного интерфейса возникает методологическая проблема, связанная с выбором критерия успешной адаптации, обеспечивающей более надежную и эффективную работу пользователя. В [Курзанцева, 2011] предлагается в качестве такого критерия использовать время поиска пользователем необходимого элемента, а в [Ходаков и др., 2003] – все усилия пользователя, включая моторные действия и запоминание информации.

Персонализация контента ИИС

Однако интеллектуальный адаптивный интерфейс является лишь одной из составляющих персонализации работы ИИС и касается в первую очередь выбора способов подачи информации пользователю. Другими важными компонентами персонализации является отбор интересных и полезных для пользователя сведений и выбор способов и методов их обработки.

Использование внешних, динамично обновляемых знаний в современных приложениях, особенно – в Web-ориентированных – является на сегодняшний день всем очевидной необходимостью.

Наиболее популярны при этом онтологии. Но их использование имеет как ряд достоинств, так и недостатков. Онтологии целесообразно использовать в ИИС с изменяющейся структурой БЗ – в тех случаях, когда разработчику заранее не известна не только структура знаний, но и свойства этой структуры, и поэтому из априорных сведений о них нельзя выбрать некое более простое средство представления знаний,

Достоинства онтологий широко известны и не требуют дополнительного обоснования:

- интероперабельности и возможность многократного использования в различных приложениях;
- теоретический базис в виде дескриптивных логик, обеспечивающий гарантии выводимости и разрешимости на онтологии;
- наличие большого количества уже созданных ранее онтологий;
- многочисленные языки представления, стандарты и программные средства для обработки онтологий.

Недостатки онтологий менее очевидны, но их необходимо в первую очередь учитывать при создании реальных приложений:

- в общем случае онтологии сложны в обработке и требуют большого времени и больших вычислительных ресурсов для установления выводимости или невыводимости того или иного факта на онтологии;

- значительная часть реальных онтологий (в отличие от «игрушечных» примеров) слишком громоздки для того, чтобы пользователь, не являющийся ее разработчиком, мог свободно вносить в нее изменения даже при достаточных знаниях о ПрО;

- проблема масштабируемости – при увеличении количества понятий в онтологии и связей между ними время обработки увеличивается настолько, что перестает удовлетворять потребностям реальных задач.

Кроме того, следует отметить, что в большинстве исследованиях, связанных с онтологическим представлением знаний, их полные выразительные возможности просто не используются, но так как их обработка все же предусмотрена, то сложность вычислений неоправданно увеличивается.

В то же время достаточно сложно спрогнозировать, какие именно знания о ПрО окажутся существенными при создании той или иной информационной системы, использующей соответствующую онтологию.

Поэтому представляется целесообразным перейти к двухуровневой модели знаний ПрО.

На первом уровне – модель ПрО – знания ПрО моделируются с помощью онтологии

$O_{ПрО} = \langle X_{ПрО}, R_{ПрО}, F_{ПрО} \rangle$, где X – множество концептов, R – множество отношений между концептами, F – функции интерпретации концептов из множества X и отношений из R (или набора каким-то образом интегрированных онтологий), на которые не накладывается практически никаких ограничений за рамками дескриптивных логик.

На втором уровне – модель задачи – используется только некоторое подмножество модели ПрО, представляющее собой ее проекцию на стоящую перед пользователем проблему. Например, ряд связей может быть отброшен или заменен каким-то одним отношением «между терминами существует связь», могут быть оставлены только термины, входящие в фиксированное подмножество и связанные с ними, оставлены только иерархические связи и т.п.

Таки образом формируется значительно меньшая и более простая в обработке онтология, которую уже можно более четко отнести к конкретной дескриптивной логике.

Частным случаем такой модели задачи является используемый при семантическом поиске онтологический тезаурус.

Онтологический тезаурус представляет собой функцию от онтологии ПрО.

При поиске информации, в частности, при поиске знаний в Web, большое значение приобретает персонификация поиска, т.е. учет личных особенностей пользователя,

осуществляющего поиск. Корреляции типа личности и предпочтениям, отдаваемым различным формам представления информации, посвящено много научных исследований.

Однако в них предпочтение отдается выявлению связей между свойствами пользователя и формальными характеристиками информационного ресурса.

Но интересно установить связи между свойствами пользователя и тем, каким структурам знаний он отдает предпочтение. Практически в любой ПрО, даже высоко формализованной, можно построить онтологии, отражающие ее структуру, существенно различающиеся как по количеству различных отношений между понятиями, в том числе иерархических, так и по количеству свойств каждого из понятий.

Методы пополнения знаний, используемых для персонификации

Одно из базовых направлений интеллектуализации систем обработки информации связано с их способностью к самообучению. Таким образом, представляется целесообразным включать в состав ИИС подсистему, предназначенную для извлечения правил и закономерностей из накопленного опыта. Пополнение знаний, используемых для персонификации, может осуществляться путем анализа опыта взаимодействия ИИС с пользователями при помощи различных методов Data Mining.

Для этого можно применять различные методы индуктивного и традуктивного извлечения знаний из данных, например, ID3, ACLS, CART и их различные модификации, ориентированные на обработку данных с различными свойствами (количественных, качественных, динамически изменяющихся и т.д.) [Rogushina et al., 2012].

При этом важно отметить, что подобные методы, как правило, предназначены для классификации, а не для кластеризации, и потому они позволяют выявлять закономерности только относительно тех понятий, связанных с пользователем, и их свойств, которые явным образом заданы при описании структуры знаний ИИС – например, при помощи набора соответствующих онтологий. Поэтому эффективность подобных методов самообучения в значительной мере зависит от удачного выбора таких онтологий.

Архитектура ИИС, обеспечивающая персонификацию

В наиболее общем виде архитектура ИИС, обеспечивающая персонифицированное взаимодействие с пользователями в информационном пространстве Web, описывает связи между информационными потребностями

пользователя, которые описываются через стоящую перед ним задачу, и теми информационными объектами (ИО) и информационными ресурсами (ИР), которые ИИС предоставляет пользователю в процессе решения этой задачи.

Во время работы с ИИС в ее базе данных накапливаются сведения о пользователях, и затем на основе их анализа формируются специфические правила, предназначенные для учета персональных особенностей и потребностей пользователя.

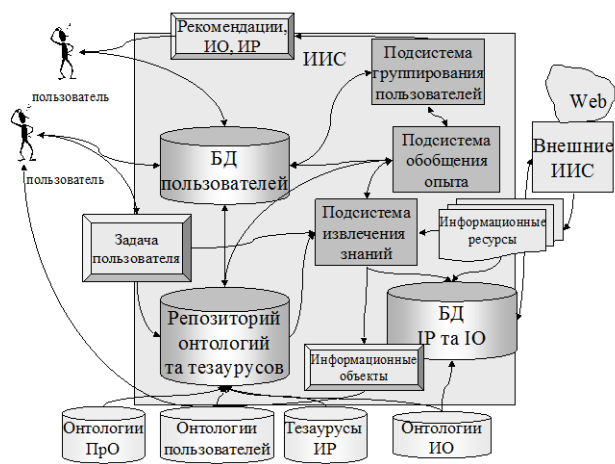


Рисунок – Обобщенная архитектура ИИС

Важным аспектом предложенной выше архитектуры является использование внешних источников знаний – репозиториев онтологий и тезаурусов, а также других ИИС, формирующих персонифицированные знания о пользователях в интероперабельной форме. Такой подход, в частности, применялся при разработке знание-ориентированной поисково-рекомендующей системы МАИПС, архитектура которой более подробно описана в [Рогущина, 2014].

Заключение

Предложенный в работе подход к персонификации семантических технологий проектирования ИИС обеспечивает более полный учет индивидуальных особенностей пользователя, а использование онтологий позволяет повторно использовать знания, накопленные ранее в других приложениях.

Библиографический список

- [Кузьмина и др., 2012] Кузьмина К.И., Оноприенко В.Н., Козак Н.С., Семик Т.М., Андон Т.А. Семейная медицина сегодня и проблема ее дальнейшей интеллектуализации с помощью информационных технологий и компьютерных систем // Теорія і практика управління соціальними системами, № 2, 2012. – С.56-67.
- [Курзанцева, 2010] Курзанцева Л.И. О построении интеллектуального адаптивного интерфейса на базе агентной технологии для компьютерных систем широкого назначения // Інформаційні технології та комп'ютерна інженерія, № No1 (17), 2010. – С.16-20.
- [Яковлев и др., 2006] Яковлев Ю.С., Курзанцева Л.И. // О применении онтологий для построения модели пользователя информационных систем // , Комп'ютерні засоби, мережі та системи. 2006, No 5. – С.109-116.
- [Анохин, 2014] Анохин А.Н. Адаптивный интерфейс для

операторов сложных систем, 2014. – <http://vspu2014.ipu.ru/proceedings/prcdn/6345.pdf>.

[Rothrock et al., 2002] Rothrock L., Koubek R., Fuchs F. et al. Review and reappraisal of adaptive interfaces: toward biologically inspired paradigms // Theoretical Issues in Ergonomics Science. 2002. Vol. 3, No. 1. P. 47-84.

[Langley, 1999] Langley P. User modeling in adaptive interfaces // Proceedings of the Seventh International Conference on User Modeling, Banff, Alberta, Canada, June 20-24, 1999. Springer, 1999. P. 357-370.

[Курзанцева, 2011] Курзанцева Л.И. Методика комплексного исследования адаптивного человеко-машинного интерфейса // Математичні машинні системи, № 4. 2011. – С.69-77.

[Ходаков и др., 2003] Ходаков В.Е., Ходаков Д.В. Адаптивный пользовательский интерфейс: проблемы построения // Автоматика. Автоматизация. Электротехнические комплексы и системы, № 1(11), 2003. – С. 45-57.

[Rogushina et al., 2012] Rogushina J., Gladun A. Ontology-based competency analyses in new research domains // Journal of Computing and Information Technology. V.20, N. 4, 2012. – P.277-293.

[Рогущина, 2014] Рогущина Ю.В. Знание-ориентированные средства поддержки семантического поиска в Web // Материали IV міжнародної науково-технічної конференції «Открытые семантические технологии проектирования интеллектуальных систем» OSTIS-2014, Минск, БГУИР, 2014. – С.339-352.

KNOWLEDGE-ORIENTED MEANS OF SEMANTIC SEARCH INTO THE WEB

Rogushina J.

Institute of software systems of National Academy of Sciences, Ukraine, Kiev, Ukraine

ladamandraka2010@gmail.com

For developing of effective personification means of intelligent information systems (IIS) it is necessary to describe what user information can improve the functioning of the IIS; the information sources; available methods; the form of knowledge presentation that facilitates its reuse; and the structure of user and domain knowledge. Ontological model of information interaction between user and IIS designed to attract knowledge from external ontologies, thesauri and distributed Web resources was proposed.

Keywords: semantic search, ontological model, thesaurus, inductive inference, recommending systems.



УДК 004.82

ОБЗОР И АНАЛИЗ МЕТОДОЛОГИЙ И МЕТОДОВ ПОСТРОЕНИЯ ОНТОЛОГИЙ

Хала Е.А.

** Международным научно-учебным центром информационных технологий и систем НАН и МОН Украины, г. Киев, Украина*

cecergreat@ukr.net

В работе выполнен обзор и сравнение методик создания онтологических описаний, приведены результаты их анализа. Приводится общая характеристика и недостатки автоматических методов построения онтологий. Описаны основные языки представления онтологий и наиболее значимые существующие онтологические ресурсы.

Ключевые слова: онтология, онтологические модели, онтологические системы, языки представления онтологий.

Введение

Понятие онтологии, заимствованное из философии, в настоящее время активно применяется в искусственном интеллекте и информатике. Все больше интеллектуальных задач, решение которых связано с обработкой знаний, решаются с применением онтологий [Гаврилова и др., 2000], [Куршев, 2007], [Найханова, 2008], [Нариньяни и др., 2001], [Россеева и др., 2000] и др. По всей вероятности в недалекой перспективе онтологии будут использоваться при решении очень многих интеллектуальных задач.

Самым распространенным определением онтологии на данный период является определение Т.Р. Грубера [Gruber, 1993], [Gruber, 1995], согласно которому онтология является точной спецификацией концептуализации. С этой точки зрения для каждой из баз данных или баз знаний, или систем, основанных на знаниях, или агентов знаний должны быть построены спецификации, основанные на некоторой концептуализации. Множества объектов и отношений между ними должны быть описаны в некотором словаре, в котором система, основанная на знаниях, представляет свои знания. К средствам создания онтологий относятся системы Ontolingua [Farquhar, 1997], Protege [Musen, 1998], OntoEdit [Sure, 2002], OilEd [Bechhofer, 2001], Web-Deso [Naichanova, 2005], Ontogrid [Загоруйко и др., 2005] и другие. В работах [Нариньяни и др., 2002] и [Овдей и др., 2004] приведены сравнительные оценки перечисленных средств с точки зрения их внешней и внутренней организации.

1. Классификация онтологий и языки описания онтологий

В проектировании онтологий условно можно выделить два направления, до некоторого времени развивавшихся отдельно. Первое связано с представлением онтологии как формальной системы, основанной на математически точных аксиомах. Второе направление развивалось в рамках компьютерной лингвистики и когнитивной науки. Там онтология понималась, как система абстрактных понятий, существующих только в сознании человека, которая может быть выражена на естественном языке.

Таким образом, существует два подхода к созданию и исследованию онтологий. Первый (формальный) основан на логике. Второй (лингвистический) основан на изучении естественного языка (семантики) и построении онтологий на больших текстовых массивах, так называемых корпусах.

В настоящее время данные подходы тесно взаимодействуют, и идет активный поиск связей, позволяющих комбинировать соответствующие методы. Поэтому иногда бывает сложно отделить лексические онтологии с элементами формальных аксиоматик от логических систем с включениями лингвистических знаний. Независимо от различных подходов можно выделить три основных принципа классификации онтологий [Соловьев, 2006]:

- по степени формальности;
- по наполнению, содержанию;
- по цели создания.

Ключевым моментом в проектировании онтологий является выбор соответствующего языка спецификации онтологий (Ontology specification language). Цель таких языков - предоставить возможность указывать дополнительную машинно-интерпретируемую семантику ресурсов, сделать машинное представление данных более похожим на положение вещей в реальном мире.

На сегодняшний момент выделяют три основных класса языков описания онтологии [Соловьев, 2006]:

- традиционные языки спецификации онтологии: Ontolingua, CycL и языки, основанные на дескрипционной логике (такие как LOOM), также языки, основанные на фреймах (OKBC, OCML, Flogic);
- более поздние языки, основанные на Web - стандартах (XOL, SHOE, UPML);
- специальные языки для обмена онтологией через Web: RDF(S), DAML, OIL, OWL.

2. Подходы и методологии построения онтологий

Подходы к построению онтологий можно условно разделить на классический и современный.

Под классическим подходом понимается построение онтологий в соответствии со

Таблица 1 – Сравнение методологий для построения онтологий

Methodologies	Type of development	Collaborative construction	Reusability support	Degree of application dependency	Life cycle recommendation	Strategies for identifying concepts	Methodology details	Interoperability support
TOVE	Stage based	No	Yes	Application semi independent	No	Middle out strategy	Some details	No
Enterprise model approach	Stage based	No	Yes	Application independent	No	Middle out strategy	Some details	No
METHONTOLOGY	Evolving prototype	No	Yes	Application independent	Yes	Middle out strategy	Sufficient details	No
KBSI IDEF5	Evolving prototype	No	Yes	Application independent	No	Not clear	Some details	No
Ontolingua	Modular development	Yes	Yes	Application independent	No	Not clear	Some details	Yes
Common KADS and KACTUS	Modular development	No	Yes	Application dependent	No	Top down strategy	Insufficient details	No
PLINIUS	Guidelines	No	No	Application independent	No	Bottom up strategy	Some details	No
ONIONS	Modular development / Guidelines	No	No	Application dependent	No	Not clear	Insufficient details	Yes
Mikrokosmos	Guidelines	No	No	Application dependent	No	Rule based strategy	Some details	No
MENELAS	Guidelines	No	No	Application dependent	No	Concepts Graphs (CG)	Insufficient details	No
SENSUS	does not mention any preference	Yes	Yes	Application semi independent	No	Bottom up	Some details	Yes
Cyc methodology	Evolving prototype	No	Yes	Application independent	No	Not clear	Some details	No
UPON	Evolving prototype	No	Yes	Application independent	Yes	Middle out strategy	Some details	No
101 method	Evolving prototype	No	Yes	Application independent	No	Developer's consent	Some details	No
On-To-Knowledge	Evolving prototype	No	No	Application dependent	Yes	Middle out strategy	Some details	No

стандартом онтологического анализа IDEF5 [ICE, 1994]. Современный подход к онтологическому анализу подразумевает, в основном, построение web-онтологий – онтологий в контексте семантической паутины и сводится к разработкам консорциума The World Wide Web Consortium [Андреева и др., 2009]. Среди методологий и методов построения онтологий можно выделить следующие:

- метод Усколда и Кинга [Uschold et al, 1998];
- методология Грюнингера и Фокса [Gruninger et al, 1995];
- подход Cyc [Lenat et al, 1989];
- метод KACTUS [Schreibe et al, 1995];
- метод SENSUS [Swartout et al, 1997];
- методология On-To-Knowledge [Staab et al, 2001];
- методология METHONTOLOGY [Fernandez et al, 2006].
- и др. методологии приведенные в таблице 1.

Рассмотренные методики имеют ряд общих характеристик – выбор основных понятий, сущностей и связей между ними, а затем их кодирование посредством различных инструментов. Некоторые методологии содержат этап, предшествующий выборке понятий – определение специфики предметной области, целей и сфер применения построенной онтологии.

Немаловажной частью методик являются алгоритмы и инструкции, по которым происходит выделение второстепенных сущностей и интеграции их в уже имеющуюся онтологию, а также метод построения основания будущей онтологии, используемой для последующего наращивания. Заключительным этапом является определение компетентности онтологии, проверка соответствия поставленным требованиям и решаемым задачам, а также сфер применения и использование построенной онтологии.

К сожалению ни одна из методологий не является полной, если сравнивать их со стандартом IEEE, а все предлагаемые решения не унифицированы. В настоящее время каждая группа применяет свою собственную методологию, что усугубляется недостаточной полнотой описания методологий. Поэтому, необходимы усилия по унификации методологий.

3. Методы автоматического построения онтологии

3.1. Представление онтологии в виде конечного автомата

Если представить онтологии в виде орграфа, где вершины – это множество предметных областей, а ребра – бинарное отношение между этими предметными областями, то с каждым таким орграфом будем ассоциировать конечный частичный детерминированный автомат без выходов [Кривый, 2008]. Представление онтологий в виде конечного автомата без выходов позволяет ввести операции на онтологиях. Операции на автоматах означают операции на регулярных языках, которые акцептируются этими автоматами [Кривый, 2008], [Бениаминов, 2003]. Причем множество операций (в случае надобности) можно расширять операциями на графах и алгеброй отношений. Поскольку каждая онтология является представлением некоторой совокупности отношений, то можно вводить операции реляционной алгебры.

3.2. Подход на основе лексико-синтаксических шаблонов

Данный подход был предложен в [Рабчевский, 2009] и относится к группе методов автоматического построения онтологий, использующих лингвистические средства. Сторонники подхода утверждают, что для построения онтологий следует активно использовать все уровни анализа естественного языка: морфологию, синтаксис и семантику. На основе лексико-синтаксических шаблонов выделяются онтологические конструкции [Анисимов, 2002], [Рабчевский, 2009].

Подход на основе лексико-синтаксических шаблонов не является специализированной на определенную предметную область и это его достоинством, однако лексико-синтаксические шаблоны как метод семантического анализа текстов

на естественном языке является не очень эффективным средством для автоматического построения онтологий.

3.3. Подход на основе системы продукций

Данный подход относится к группе методов автоматического построения онтологий, в основе которых лежат подходы из области искусственного интеллекта. В [Найханова, 2008] предлагается модель автоматического построения онтологий в виде системы продукций и применении генетического и автоматного программирования для создания необходимых моделей. Для создания методов автоматического построения онтологий автор разрабатывает модель генерации системы продукций (генетическое программирование), модель генерации преобразователей и модель генерации систем логического вывода (генетическое и автоматное программирование), модель аппарата активации продукций (автоматное программирование).

Применение продукционных правил обеспечивает следующие преимущества: простота и высокое быстродействие, модульность, удобство модификации, ясность, прозрачность, возможность постепенного наращивания. Среди недостатков можно выделить недостаточную семантическую связность между правилами.

3.4. Автоматическое построение онтологий по коллекции текстовых документов

В работе [Мозжерина, 2011] предлагается подход к решению проблемы автоматического построения онтологий, преимущественно основанный на статистических методах анализа текстов на естественном языке.

Для улучшения получаемой в результате работы системы онтологии, автор предлагает провести предварительную кластеризацию документов коллекции таким образом, чтобы в один кластер попадали тематически близкие документы, а дальнейшую работу проводить отдельно с каждым полученным кластером.

Также [Мозжерина, 2011] предполагает, что существующие статистические методы извлечения терминов из текстов на естественном языке могут показать лучшие результаты, если дополнить их определенными эвристиками, которые детальней описываются в работе автора.

Статистический подход является достаточно универсальными, но позволяет выделить только базовые отношения, необходимые для построения онтологии, что является его недостатком.

Заключение

Создание онтологий является перспективным направлением современных исследований по обработке информации, представляемой на естественном языке. В рамках работы выполнен

обзор методов онтологического инжиниринга и сформулирован обобщенный подход для построения онтологии, рассмотрены различные классификации онтологий. Уже сейчас существует ряд обширных онтологий, построенных как в рамках отдельных предметных областей, так и для незамкнутых областей знания. Наиболее перспективной является автоматизация создания онтологий, однако на данном этапе еще не разработаны эффективные процедуры, применение которых позволит сократить долю ошибок. Однако уже сейчас существует ряд приложений, успешно использующих онтологии в своей работе.

Библиографический список

- [Андреева, 2009] Андреева, Н.В. Выбор методов и средств онтологического анализа стандартов информационной безопасности / Н.В.Андреева, А.В.Любимов // VI Всероссийская между-зювская конференция молодых ученых. — СПб: СПбГУ ИТМО, 2009. Вып. 6. - С. 29-33.
- [Анисимов, 2002] Анисимов, А.В. Система обработки текстов на естественном языке / А.В.Анисимов, А.А.Марченко // Искусственный интеллект. — 2002. — № 4. — С. 157 — 163.
- [Бениаминов, 2003] Бениаминов, Е.М. Алгебраические методы в теории баз данных и представлении знаний / Е.М.Бениаминов. — М.: Научный мир, 2003 — 184 с.
- [Гаврилова, 2000] Гаврилова, Т.А. Базы знаний интеллектуальных систем / Т.А.Гаврилова, В.Ф.Хорошевский. — СПб: Питер, 2000. — 384 с.
- [Загоруйко, 2005] Загоруйко, Н.Г. и др. Система "Ontogrid" для построения онтологий / Н.Г.Загоруйко // Компьютерная лингвистика и интеллектуальные технологии. Тр. междунар. Конференции Диалог'2005. М., 2005. С. 146-152.
- [Крытый, 2008] Крытый, С.Л. Автоматное представление онтологий и операции на онтологиях / С.Л.Крытый, А.Н.Ходзинский — International Book Series. — N 1. — Algorithmic and Mathematical Foundations of the Artificial Inetelligence. — ITHEA: Sofia. -2008. — PP. 173-179.
- [Куршев, 2007] Куршев, Е.П. Исследование методов извлечения информации из текстов с использованием автоматического обучения и реализация исследовательского прототипа системы извлечения информации / Е.П.Куршев, Д.А.Кормалев, Е.А.Сулейманова, И.В.Трофимов // Математические методы распознавания образов: 13-я Всерос. конф.: сборник докладов. — М.: МАКС Пресс, 2007. — С.602-605.
- [Мозжерина, 2011] Мозжерина, Е.С. Автоматическое построение онтологий по коллекции текстовых документов / Е.С.Мозжерина // Электронные библиотеки: Перспективные Методы и Технологии, Электронные коллекции — RCDL 2011 — Воронеж, 2011 — С. 293 — 298.
- [Найханова, 2008] Найханова, Л.В. Методы и модели автоматического построения онтологий на основе генетического и автоматного программирования / Л.В.Найханова. - Красноярск, 2008. - 36 с.
- [Нариньяни, 2001] Нариньяни, А.С. Кентавр по имени ТЕОН: тезаурус+онтология / А.С.Нариньяни // Междунар. семинар Диалог'2001 по компьютерной лингвистике и ее приложениям. — Аксаково, 2001. — Т. 1. — С. 184—188.
- [Нариньяни, 2002] Нариньяни, А.С. ТЕОН-2: от тезауруса к онтологии и обратно / А.С.Нариньяни // Компьютерная лингвистика и интеллектуальные технологии: междунар. семинар Диалог'2002 — М.: Наука, 2002. — Т. 1. — С. 307-313.
- [Овдей, 2004] Овдей, О.М. Обзор инструментов инженерии онтологий / О.М.Овдей, Г.Ю.Проскудина // Журнал ЭБ. 2004 — №4
- [Рабчевский, 2009] Рабчевский, Е.А. Автоматическое построение онтологий на основе лексико-синтаксических шаблонов для информационного поиска / Е.А.Рабчевский // Труды XI Всерос. науч. конф. «Электронные библиотеки: перспективные методы и технологии, электронные коллекции». — Петрозаводск, 2009.
- [Соловьев, 2006] Соловьев, В.Д. Онтологии и тезаурусы:

учеб. пособие / В.Д.Соловьев, Б.В.Добров, В.В.Иванов, Н.В.Лукашевич; Казанский гос. ун-т, МГУ им. М.В. Ломоносова Казань. — М.: Казань, 2006. — 157 с.

[Bechhofer, 2001] Bechhofer, S. OilEd: A Reason-able Ontology Editor for the Semantic Web / S.Bechhofer, I.Horrocks, C.Goble, R.Stevens // Joint German/Austrian conf. on Artificial In telligence (KI'01). Lecture Notes in Artificial Intelligence LNAI 2174, Springer-Verlag, Berlin, pages.396-408, 2001.

[ICE, 1994] Information Integration for Concurrent Engineering (ICE). IDEF5 Method Report. — Knowledge Based Sys-tems, Inc., 1408 University Drive East College Station, Texas, USA. — September 21. — 1994.

[Farquhar, 1997] Farquhar, A. The Ontolingua server: A tool for collaborative ontology construction / A.Farquhar, R.Fikes, J.Rice // International Journal of Human-Computer Studies, 46(6), pages 707—728, 1997.

[Fernndez, 2006] Fernndez, M. METHONTOLOGY: From Ontological Art Towards Ontological Engineering / M. Fernndez, A.Gomez-Perez, N.Juristo // AAAI Technical Report SS-97-06, 2006.

[Gruber, 1993] Gruber, T.R. A translation approach to portable ontologies / T.R.Gruber // Knowledge Acqui-sition. 1993. —No 5(2). — С. 199-220.

[Gruber, 1995] Gruber, T. Towards principles for the design of Ontologies used for knowledge sharing / T.Gruber // International Journal of Human-Computer Studies. —1995. —No 43(5/6). —С. 907-928

[Gruninger, 1995] Gruninger, M. Methodology for the design and evaluation of ontologies / M.Gruninger, M.S.Fox // Proceeding of the Workshop on Basic Ontological Issues in Knowledge Sharing, IJCAI. 1995.

[Lenat, 1989] Lenat D. Building Large Knowledge Based Systems: Representation and Inference in the Cyc Project / D.Lenat, R.V.Guha // Addison-Wesley, 1989.

[Musen, 1998] Musen, M. Domain Ontologies in Software Engineering: Use of Protege with the EON Architecture / M.Musen, // Methods of Inform. in Medicine, pages 540-550, 1998.

[Naichanova, 2005] Naichanova, L.V. Knowledge presentation of inference system based on knowledge/ L.V.Naichanova // (IT@I ES'2005): Materials of the international scientific conference / edited by A.N. Tikhonov (chair.) and others; SIIT@T Informika. — Moscow: VIZCOM, 2005. — P. 63—66.

[Uschold, 1998] Uschold, M. The Enter-prise Ontology The Knowledge Engineering Review / M.Uschold, M.King, S.Moralee, Y.Zorgios // Vol. 13, Special Issue on Putting Ontologies to Use, 1998.

[Schreibe, 1995] Schreibe, G. The KACTUS View on the 'O' World / G.Schreibe, B.Wielinga, W.Jansweijer // In Proceedings of the National Dutch AI Conference. NAIC'95. 1995.

[Staab, 2001] Staab, S. Knowledge processes and ontologies / S.Staab, H.P.Schnurr, R.Studer, Y.Sure // IEEE Intelligent Systems - 2001. - V. 16. - № 1. - P. 26-34.

[Sure, 2002] Sure, Y. OntoEdit: Collaborative ontology development for the Semantic Web / Y.Sure, M.Erdmann, J.Angela, S.Staab, R.Studer, D.Wenke // In Proc. of the Inter. Semantic Web Conference (ISWC 2002), Sardinia, Italia, June 2002.

[Swartout, 1997] Swartout, B. Toward Dis-tributed Use of Large-Scale Ontologies / B.Swartout, P.Ramesh, K.Knight, T.Russ // Symposium on Ontological Engineering of AAAI. Mars, 1997.

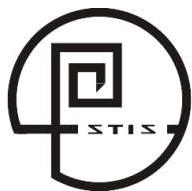
REVIEW AND ANALYSIS METHODOLOGIES AND TECHNIQUES OF ONTOLOGY BUILDING

Khala C.A.

International Research and Training Center of Information Technologies and Systems NAS and MES of Ukraine, Kiev, Ukraine

cecerongreat@ukr.net

The paper gives an overview and comparison of techniques for building ontological descriptions, the results of their analysis. Provides a general description and disadvantages of automatic methods for constructing ontologies. The basic ontology representation languages and the most significant existing ontological resources.



УДК 519.816

СПЕЦИФИКАЦИЯ ТЕМПОРАЛЬНЫХ ОТНОШЕНИЙ В СИСТЕМЕ «БИНАРНАЯ МОДЕЛЬ ЗНАНИЙ»

Плесневич Г.С., Нгуен Тхи Минь Ву

Национальный исследовательский университет «МЭИ», г. Москва, Россия

salve777@mail.ru

mimhvu.357@gmail.com

В работе кратко описывается концептуальный язык, содержащий средства для спецификации темпоральных отношений в онтологиях. Вместе с ним рассматриваются языки структурной и логической спецификации, поддерживающие систему «Бинарная Модель Знаний», которая предназначена для автоматизированного построения и анализа онтологий.

Ключевые слова: базы знаний, онтологии, языки для спецификации онтологий, темпоральные отношения.

ВВЕДЕНИЕ

Понятие времени присутствует во всякой деятельности агентов, требующей интеллекта. Работая с интеллектуальной информационной системой, агенты (естественные или искусственные) имеют дело с моделью реального мира, который часто является динамическим. Многие факты, содержащиеся в состоянии модели, представляют события, которые связываются друг с другом темпоральными отношениями («раньше», «позже», «одновременно», «в течение» и т.п.). Для описания темпорального знания в моделях используются соответствующие формальные языки.

В настоящей работе мы опишем (в общих чертах) язык для спецификации темпоральных отношений в системе «Бинарная Модель Знаний» (БМЗ), а также укажем метод логического вывода в этом языке. Система БМЗ предназначена для построения, анализа и интерпретации формальных онтологий [Plesniewicz, 2004], [Плесневич, 2005], [Плесневич, 2012].

1. О языках системы «Бинарная Модель Знаний»

Языки системы БМЗ относятся к классу концептуальных языков; их семантика основана на формальных понятиях.

Формальное понятие (в экстенциональном аспекте) строится из имен. Оно имеет следующие компоненты: (1) *имя понятия* C ; (2) *универсум понятия* U^C – множество всех имен, которые могут

обозначать примеры (экземпляры) понятия C ; (3) множество *точек соотнесения* Γ ; (4) для каждой точки соотнесения $\gamma \in \Gamma$ подмножество $E_\gamma^C \subseteq U^C$ – *множество всех примеров* понятия C в этой точке (точнее, имен этих примеров); (5) для каждой точки соотнесения $\gamma \in \Gamma$ заданное на множестве E_γ^C отношение эквивалентности \sim_γ^C – *корреференция* в этой точке; (6) для каждой точки соотнесения $\gamma \in \Gamma$ фактор-множество $\text{Ext}_\gamma^C = E_\gamma^C / \sim_\gamma^C$ – *экстенционал* понятия C в этой точке; (7) семейство $\text{Ext}^C = \{\text{Ext}_\gamma^C \mid \gamma \in \Gamma\}$ – *полный экстенционал* понятия C .

Замечания. 1) Корреферентные имена обозначают один и тот же объект моделируемой предметной области. 2) Точка соотнесения (другое название – «индекс») фиксирует множество примеров понятия. Например, для понятий, представляющих события или процессы, точками соотнесения могут быть моменты времени или временные интервалы. 3) Имя понятия может быть простым или составным. Составное имя понятия имеет структуру, определяющие связи примеров этого понятия с примерами понятий, имена которых входят в составное имя. Например, в терминологических (дескриптивных) логиках термины – это составные имена понятий [Baader et al., 2003].

Концептуальные языки можно классифицировать в соответствии с тем, какие из компонент формальных понятий они специфицируют. *Языки структурной спецификации* используются для определения универсумов понятий. *Языки логической спецификации* используются для спецификации экстенционалов независимо от точек соотнесения. *Языки модальной спецификации переходов* используются для

спецификации экстенционалы понятий с учетом их изменений при переходе от одних точек соотнесения к другим.

При моделировании объектов предметной области с помощью языков системы БМЗ мы предполагаем, что объекты характеризуются своими атрибутами. Атрибут рассматривается как функция, которая объектам сопоставляет другие объекты или значения – элементы типов данных.

В БМЗ входит язык ЯСС структурной спецификации. Два вида формальных понятий различаются в ЯСС: *классы* и *бинарные связи*. Рассмотрим примеры определения универсумов этих понятий в языке ЯСС. Предложения

Автомобиль[Марка:String, Двигатель, Габариты:
(Длина(мм): Integer, Ширина(мм): Integer,
Высота(мм): Integer, 'Колесная база (мм)':
Integer), 'Коробка передач': String],
Двигатель[Тип: Integer, Мощность(лс): Integer,
'Время разгона до 100 км/сек': Integer,
'Макс._скорость (км/час)': Integer].

определяют универсумы классов Автомобиль и Двигатель. В универсумы всегда входят так называемые *суррогаты* – стандартные имена для идентификации индивидуальных объектов. Мы рассматриваем их как элементы примитивного типа данных $Surr = \{\#1, \#2, \#3, \dots\}$. В универсумы также входят простые имена для индивидуальных объектов (представленные слитными строчками, начинающимися с малой буквы), а также кортежи. Так, универсум $U^{Автомобиль}$ включает все кортежи вида

[Марка: x , Двигатель : y , Габариты: z , 'Коробка
передач': u],

где $x \in String$, $y \in Surr$, $u \in String$ и z – элемент следующего производного типа данных

(Длина(мм): Integer, Ширина(мм): Integer,
Высота(мм): Integer, 'Колесная база (мм)':
Integer).

Вот конкретные примеры кортежей из универсумов $U^{Автомобиль}$ и $U^{Двигатель}$ (эти кортежи представляют информацию об автомобиле Audi 200):

[Марка: Audi200, Двигатель: #23, Габариты:
(Длина(мм): 4810, Ширина(мм): 1810,
Высота: 1420, 'Колесная_база (мм)': 2690),
'Коробка передач': 'МКПП']. (1.1)
[Тип: 5, Мощность(лс): 182, 'Время разгона до
100 км/сек': 8.1, 'Макс. скорость (км/час)': 230].

Бинарная связь в качестве своих примеров (экземпляров) имеет конкретные бинарные отношения между экземплярами двух понятий. Рассмотрим пример предложения ЯСС, задающего бинарную связь.

(Персона Владеет Автомобиль)[Дата: Date,
'Документы о регистрации покупки': String].

Здесь предполагается, что атрибут Дата имеет смысл «дата оформления владения». Значениями этого атрибута служат элементы типа данных Date, которые имеют формат dd.dd.dddd, где d обозначает цифру. Универсум $U^{Владеет}$ бинарной связи Владеет содержит все кортежи вида

[Персона: x , Автомобиль: y , Дата: z , 'Документы о
регистрации': u],

где $x \in Surr$, $y \in Surr$, z – элемент типа данных Date и u – элемент типа данных String.

Составляя онтологию для предметной области при помощи языков системы БМЗ, мы сначала пишем в языке ЯСС текст, определяющий универсумы понятий моделируемой предметной области. Этот текст, состоящий из предложений языка ЯСС, называется *структурной схемой*.

Прямой простейший способ спецификации экстенционалов понятий из структурной схемы состоит в записи конечного подмножеств универсумов этих понятий, а также в записи некоторых пар кореферентных элементов из этих подмножеств. Эти записи естественно представить в таблицах, соответствующих предложениям для данной структурной схемы. Например, для понятия Автомобиль мы определяем таблицу с именем Автомобиль и полями Surr, Por, Coref, Марка, Двигатель, Габариты, Колесная_база, Коробка_передач. (Здесь имя Surr выступает в роли атрибута.) В строке таблицы в поле Por (point-of-reference) записывается точка соотнесения, а в поле Coref помещается список кореферентных имен индивидов. Например, в таблицу Автомобиль мы можем поместить строку

(#7, 'текущая база данных', ['автомобиль № к 200
AE 77RUS', 'автомобиль А.П. Иванова'], Audi200,
#23, (Длина(мм): 4810, Ширина(мм): 1810,
Высота: 1420, Колесная_база(мм): 2690),
'МКПП'), (1.2)

в которой определены следующие кореференции:

#7 ~ 'автомобиль № к 200 AE (77RUS)' ~
'автомобиль А.П. Иванова' ~ t ,

где t обозначает кортеж (1.1). Также в таблицу Двигатель мы можем включить строку

(#23, [], 5, 182, 8.1, 230).

Строка (1.2) представляет объект с суррогатом #7, который является принадлежащим А.П. Иванову автомобилем марки Audi 200 с номером к 200 AE 77RUS. Автомобиль имеет габариты: длина 4810 мм, ширина 1810 мм, высота 1420 мм и колесную базу 2690 мм; он имеет коробку передач МКПП. Двигатель автомобиля обозначен суррогатом #23, который используется как ссылка к таблице Двигатель. Строка этой таблицы с суррогатом #23 содержит информацию о двигателе автомобиля. Точка соотнесения 'текущая база данных' говорит о том, что строка таблицы относится к текущей базе данных.

Суррогаты позволяют выполнять навигацию по таблицам при помощи операции «точка», получая информацию об объекте. Например, имеем

#7. Двигатель =
'автомобиль А.П. Иванова'. Двигатель =
'автомобиль № к 200 АЕ 77RUS'. Двигатель = #23,
#7. Двигатель . Мощность = #23.Мощность = 182,
'автомобиль А.Г. Иванова'. Двигатель .Тип = 5.

При прямом способе спецификации понятий действует предположение замкнутого мира CWA (closed world assumption) [Reiter, 1978]: таблицы для понятий полностью определяют их экстенционалы.

В сущности, структурная схема является схемой объектно-ориентированной базы данных, если действует CWA [Abiteboul et al, 1995]. Предложения языков логической и модальной спецификации из онтологии можно тогда рассматривать как ограничения целостности. Например, предложение

Автомобиль ISA Автомобиль(Двигатель . 'Макс. скорость(км/час)' ≤ 300) (1.2)

утверждает, что максимальная скорость любого автомобиля не превосходит 300 км в час. Его мы рассматриваем как ограничение, что записанный в базе данных автомобиль не должен иметь максимальную скорость, большую 300 км в час, т.е. в строках таблицы Автомобиль в поле Двигатель не должен стоять суррогат s такой, что $s.$ 'Макс. скорость(км/час)' ≤ 300).

При предположении открытого мира OWA (open world assumption) конечные таблицы лишь частично определяют экстенционалы понятий. Для более полного их определения можно в таблицы вставлять строки, представляющие некоторые контрпримеры понятий. Для этого в таблицу понятия мы включаем атрибут Deg (degree), значениями которого в строке служит 1 или 0 в зависимости от того, представляет ли строка пример или контрпример понятия. Записывая в таблице информацию о некотором объекте e , мы можем точно не знать, является ли он примером или контрпримером данного понятия C . Тогда можно воспользоваться нечеткой логикой, считая, что утверждение «объект e является примером понятия C » имеет некоторую степень истинности, выражаемую числом $r \in [0,1] = \{x / 0 \leq x \leq 1\}$. Таким образом, $e.$ Deg = r .

Но основным способом для спецификации экстенционалов понятий является использование предложений языков логической и модальной спецификации. В системе БМЗ имеется несколько таких языков. Один из этих языков, язык ЯДС-0, был ранее описан в работе [Плесневич, 2014]. Этот язык расширяет язык ALC дескриптивной логики [Schmidt-Schauss et al, 1991]. В ALC, как и во всяком «чистом» языке дескриптивной логики, утверждения имеют только вид включения понятий: $C \subseteq D$, где C и D – термины (вообще говоря, составные имена – описания понятий). В языке ЯЛС-0 имеются и утверждения другого вида, например, утверждение

EXIST Персона THAT Владеет SOME

Автомобиль THAT Имеет SOME Дефект.

В языках логической и модальной спецификации в системе БМЗ используются в качестве составных имен понятий термины вида $C(\alpha)$, где C – простое имя понятия, а α – *атрибутивное условие*, которые являются конъюнкциями простых атрибутивных соотношений. Термин $C(\alpha)$ обозначает понятие, примерами которого служат все объекты, удовлетворяющие условию α . Например, термин

Автомобиль(Двигатель . 'Макс. скорость' < 200;

Габариты . Длина(мм) < 4000;

'Коробка передач' = 'МКПП')

обозначает класс всех автомобилей с максимальной скоростью меньше 200 км в час, с коробкой передач МКПП и имеющих длину менее 4000 миллиметров.

В языках логической и модальной спецификации в БМЗ, как и в языках дескриптивной логики, мы можем давать новые простые имена понятиям, определяемым другим терминам. (Так что это простое имя и этот термин кореферентны.). Например, мы можем следующим образом определить класс, состоящий из владельцев автомобилей Ауди кроссовер, а также класс мини-автомобилей:

'Владелец автомобиля Ауди кроссовер' :=
Персона THAT Владеет Автомобиль(Марка IN
{'Audia Q3', 'Audia RS Q3', 'Audia Q5', 'Audia SQ5',
'Audia Q7'}),

Мини-автомобиль := Автомобиль(Габариты.

Длина(мм) < 3600; Габариты . Ширина < 1600).

Как и в языках дескриптивной логики, в языках логической спецификации в БМЗ используются предложения, определяющие включение понятий друг в друга. Эти предложения имеют вид C ISA D , где C и D – термины (простые или составные) имена понятий (классов или бинарных связей).

2. Темпоральная спецификация в системе «Бинарная Модель Знаний»

В системе БМЗ имеется язык ЯТС для темпоральной спецификации. В этом языке разработчик онтологии может формально представлять информацию о событиях, т.е. о понятиях, связанных со временем. Язык ЯТС расширяет язык ЯЛС-0 [Плесневич, 2014].

ЯТС близок к языкам темпоральной дескриптивной логики, в особенности, к языку, предложенному Альбрехтом Шмиделем [Schmiedel, 1990], [Artale et al., 1998], [Artale et al., 2000], [Artale et al., 2005].

Язык ЯТС содержит темпоральные переменные и темпоральные кванторы SOMETIME (когда-нибудь) и ALLTIME (все время), а также темпоральное отношение AT. В язык ЯТС входят фундаментальные бинарные отношения между временными интервалами: BEFORE (раньше), MEET

(встречает), DURING (в течение), START (начинает), OVERLAP (перекрывает), FINISH (заканчивает), EQUAL (равен). (См. смысл этих отношений указан в таблице 1.) В ЯТС имеются также имена для обратных отношений; AFTER = BEFORE*, MEET-BY = MEET*, CONTAIN = DURING*, START-BY = START*, OVERLAP-BY = OVERLAP*, FINISH-BY = FINISH*, где * обозначает операцию взятия обратного отношения. (Если ρ – бинарное отношение, то $x \rho^* y \Leftrightarrow y \rho x$.)

Таблица 1 – Фундаментальные отношения между интервалами

=====	BEFORE
=====	MEET
=====	DURING
=====	START
=====	OVERLAP
=====	FINISH
=====	EQUAL

Замечание. Вышеуказанные темпоральные отношения были введены Дж. Алленом в связи с представлением знания о временных интервалах [Allen, 1983]. (В статье Аллена эти отношения сокращенно обозначены *b, m, d, s, o, f, e, bi, mi, di, si, oi, fi.*)

Значениями темпоральных переменных и аргументов указанных бинарных отношений служат временные интервалы. Значениями второго аргумента темпорального отношения АТ также служат временные интервалы; значениями первого аргумента являются понятия или объекты.

Мы рассматриваем временные интервалы как точки соотнесения или как компоненты этих точек. Временные интервалы является элементами типа данных, определяемого разработчиком онтологии. В этот тип включаются такие константы, как ‘2015 г.’ и ‘15.01.2015 г.’, интервалы с концами, например, [‘15.09.2014’, ‘31.12.2014’], переменные для интервалов фиксированной длины, например, сек, мин, час, день. Для интервалов может указываться грануляция, например, сек gran мин и час gran день.

Имеются два вида термов в ЯТС: темпорально индексированные и ригидные. Интерпретация первых зависит от значения переменной now; вторые имеют постоянный экстенционал.

Мы покажем выразительные возможности языка ЯТС на примерах. Термин

(‘Владелец автомобиля’ АТ ‘2010 г.’)

задает класс, примерами которого служат все владельцы автомобилей в 2010 году. Следующее предложение языка ЯТС определяет класс с именем ‘Бывший владелец автомобиля’:

‘Бывший владелец автомобиля’ := Персона THAT Владеет(SOMETIME X) SOME Автомобиль; X BEFORE now.

Следующий термин определяет тех персон, которые в текущий момент (выраженный временным интервалом now) поменяли свой автомобиль Ауди 200 на автомобиль Тойота Ленд крузер.

Персона THAT Владеет(SOMETIME X) SOME Автомобиль(Марка = ‘Audi 200’); Владеет(SOMETIME Y) SOME Автомобиль(Марка = ‘Toyota land cruiser’); X START now; now FINISH Y; X MEET Y.

(Напомним, что точка с запятой обозначает конъюнкцию.) Факт, что А.П. Иванов владел когда-то автомобилем Ауди 200, в языке ЯТС можно записать так:

‘А.П. Иванов’ Владеет(SOMETIME X) SOME Автомобиль(Марка = ‘Audi 200’); X BEFORE now.

Язык ЯТС в качестве подязыка содержит формализм для представления темпоральных ограничений; он является расширением логики Аллена временных интервалов [Allen, 1983] и сетей темпоральных ограничений [Dechter et al., 1991]. Это расширение заключается в добавлении пропозициональных комбинаций, составленных из темпоральных ограничений, описанных в этих статьях, а также в ведении оценок для нечетких ограничений в логике Аллена.

3. О дедукции в языке темпоральных ограничений

В системе БМЗ предполагается реализовать изложенные в работах [Allen, 1983] и [Dechter et al., 1991] методы дедукции для темпоральных ограничений, а также некоторые расширения этих методов. В частности, будет реализован полученный нами метод дедукции для оценок нечетких ограничений. Мы рассмотрим этот метод на примере.

Очевидно, что в логике Аллена имеет место логическое следствие (см. рис.1)

$$A \text{ b } B, B \text{ m } C, A \text{ o } D, D \text{ o } C \models B \text{ d } D. \quad (3.1)$$

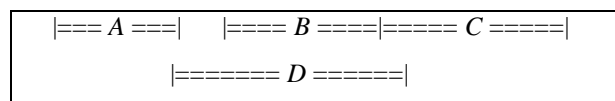


Рисунок 1 – Интервалы в логическом следствии (3.1)

Предположим, что предложения в (3.1) рассматриваются как атомы в нечеткой логике Заде, т.е. в произвольной заданной интерпретации «.» им приданы некоторые значения из $[0,1] = \{x / 0 \leq x \leq 1\}$. Пусть также в этой интерпретации истинны оценки

$$A \text{ b } B \geq a, B \text{ m } C \geq b, A \text{ o } D \geq c, D \text{ o } C \geq d, \quad (3.2)$$

т.е. верны неравенства « $A b B$ » $\geq a$, « $B m C$ » $\geq d$ и т.д. Тогда имеет место логическое следствие

$$A b B \geq a, B m C \geq b, A o D \geq c, D o C \geq d \vdash B d D \geq x \quad (3.3)$$

в том и только том случае, если не существует интерпретации, при которой оценки (3.2) истинны, но оценка $B d D \geq x$ ложна, т.е. $B d D < x$. Рассмотрим, как найти наибольшее число x , при котором справедливо логическое следствие (3.3).

Очевидно, что

$$\begin{aligned} A b B &\Leftrightarrow A^+ < B^-, A m B \Leftrightarrow A^+ = B^-, \\ A d B &\Leftrightarrow A^- < B^- \wedge A^+ < B^+, A s B \Leftrightarrow A^- = B^- \wedge A^+ < B^+, \\ A f B &\Leftrightarrow A^- < B^- \wedge A^+ = B^+, A e B \Leftrightarrow A^- = B^- \wedge A^+ = B^+, \\ A o B &\Leftrightarrow A^- < B^- \wedge B^- < A^+ \wedge A^+ < B^+. \end{aligned}$$

Здесь X^- обозначает левый конец интервала X , а X^+ – правый его конец. Так как конъюнкция в логике Заде интерпретируется нормой min , то, например, имеем

$$\begin{aligned} \langle A o B \rangle &= \langle A^- < B^- \wedge B^- < A^+ \wedge A^+ < B^+ \rangle = \\ &= \min(\langle A^- < B^- \rangle, \langle B^- < A^+ \rangle, \langle A^+ < B^+ \rangle). \end{aligned}$$

Отсюда следует состоятельность следующего правила вывода в логике оценок:

$$(A o B) \geq c \vdash (A^- < B^-) \geq c \text{ и } (B^- < A^+) \geq c \text{ и } (A^+ < B^+) \geq c.$$

В Табл.1 представлены правила вывода по методу аналитических таблиц для логики оценок, а на Рис.2 дано дерево вывода, доказывающее опровержением логическое следствие (3.3). Здесь также применялось еще одно правило вывода

$$(X < Y) > a, (Y < Z) > b \vdash (X < Z) > \max\{a, b\}.$$

Видно, что левая ветвь содержит оценки $(D^- < B^-) < x$ и $(D^- < B^-) \geq \max\{a, c\}$, которые при произвольной интерпретации дают двойное неравенство $\max\{a, c\} \leq \langle (D^- < B^-) \rangle < x$. Ясно, что оно будет не верно, если $x \leq \max\{b, d\}$. Следовательно, левая ветвь будет замкнутой, когда $x \leq \max\{a, c\}$. Аналогично получаем, что правая ветвь будет замкнутой, когда $x \leq \max\{b, d\}$. Отсюда следует, что $\min\{\max\{a, c\}, \max\{b, d\}\}$ является наибольшим значением x , при котором дерево будет замкнутым.

Таблица 1 – Правила вывода в нечеткой интервальной логике

$(A b B) \geq c \vdash (A^+ < B^-) \geq c$
$(A m B) \geq c \vdash (A^+ = B^-) \geq c$
$(A d B) \geq c \vdash (A^- < B^-) \geq c \text{ и } (A^+ < B^+) \geq c$
$(A s B) \geq c \vdash (A^- = B^-) \geq c \text{ и } (A^+ < B^+) \geq c$
$(A f B) \geq c \vdash (A^- < B^-) \geq c \text{ и } (A^+ = B^+) \geq c$
$(A e B) \geq c \vdash (A^- = B^-) \geq c \text{ и } (A^+ = B^+) \geq c$
$(A o B) \geq c \vdash (A^- < B^-) \geq c \text{ и } (B^- < A^+) \geq c$

и $(A^+ < B^+) \geq c$
$(A b B) \leq c \vdash (A^+ < B^-) \leq c$
$(A m B) \leq c \vdash (A^+ = B^-) \leq c$
$(A d B) \leq c \vdash (A^- < B^-) \leq c \text{ или } (A^+ < B^+) \leq c$
$(A s B) \leq c \vdash (A^- = B^-) \leq c \text{ или } (A^+ < B^+) \leq c$
$(A f B) \leq c \vdash (A^- < B^-) \leq c \text{ или } (A^+ = B^+) \leq c$
$(A e B) \leq c \vdash (A^- = B^-) \leq c \text{ или } (A^+ = B^+) \leq c$
$(A o B) \leq c \vdash (A^- < B^-) \leq c \text{ или } (B^- < A^+) \leq c$ или $(A^+ < B^+) \leq c$

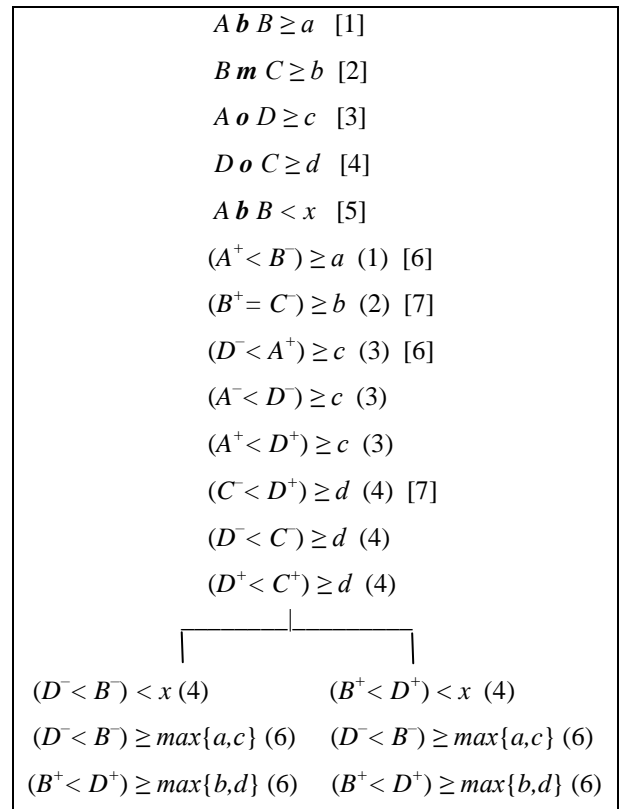


Рисунок 2 – Дерево вывода

Заключение

«Бинарная Модель Знаний» (БМЗ) – это система, предназначенная для построения и анализа онтологий. Эта система разрабатывается на кафедре прикладной математики Национального исследовательского университета «МЭИ». В статье сначала мы кратко описали язык ЯСС для структурной спецификации понятий онтологий, а затем описали (на базе ЯСС) язык ЯТС для спецификации темпорального знания в онтологиях. ЯТС расширяет ЯЛС-0 – один из языков логической спецификации в системе БМЗ. Для интерпретации ЯТС в системе БМЗ были разработаны методы логического вывода. Один из этих методов был кратко описан. Вторым автором была написана на языке Common Lisp программа-прототип, реализующая этот метод вывода.

Настоящая работа выполнялась при поддержке Российского Фонда Фундаментальных Исследований (проект № 14-07-00387).

Библиографический список

- [Abiteboul et al., 1995] Abiteboul S., Hull R., Vianu V. Foundation of databases. – Addison-Wesley, 1995.
- [Artale et al., 1998] Artale A., Franconi E. A temporal description logic for reasoning about actions and plans // Journal of Artificial Intelligence Research, no 9, 1998.
- [Allen, 1983] Allen J. Maintaining knowledge about temporal intervals // Communications of the ACM, vol. 26, no. 11, 1983
- [Artale et al., 2000] Artale A., Franconi E. A survey of temporal extensions of description logics // Annals of Mathematics in Artificial Intelligence, no. 1, 2000.
- [Artale et al., 2005] Artale and E. Franconi. Temporal description logics. In: Fisher M., Gabbay D. M., Vila L. (eds.) Handbook of Time and Temporal Reasoning in Artificial Intelligence. – Elsevier, 2005.
- [Baader et al., 2003] Baader F., Calvanese D., McGuinness D.L, Nardi D, Patten-Schneider P.F. The Description logic handbook: theory, implementation, and applications. – Cambridge University Press, 2003.
- [Dechter et al., 1991] Dechter R., Meiri I., Pearl J. Temporal constraint networks // Artificial Intelligence, vol. 49, 1991.
- [Lutz et al., 2008] Lutz C., Wolter F., Zakharyashev M. Temporal description logics: a survey. In Proceedings of the 15th International symposium on temporal representation and reasoning. – IEEE Computer Society Press, 2008.
- [Reiter, 1978] Reiter R. On closed world data bases. In: Callaire H., Minker J. (eds.) Logic and databases. – Perceus Publishing, 1978.
- [Schmiedel, 1990] Schmiedel A. A temporal terminological logic. In: Dietterich W., Swartout T.(eds). Proceedings of the 9th national conference on Artificial Intelligence AAAI-90.– MIT Press, 1990.
- [Schmidt-Schauss et al., 1991] Schmidt-Schauss M., Smolka G. Attributive concept description with complements //Artificial Intelligence, vol. 48, no. 1, 1991.
- [Plesniewicz, 2004] Plesniewicz G.S. Binary Data and Knowledge Model // In: Stefanuk, Kaijiri (eds.) Knowledge-based software engineering. – IOS, 2004.
- [Плесневич, 2005] Плесневич Г.С. Бинарные модели знаний //Интегрированные модели и мягкие вычисления в искусственном интеллекте. Сборник трудов III-го Международного научно-практического семинара (Коломна, 15-17 мая 2005 г.). – М.: Физматлит, 2005.
- [Плесневич, 2012] Плесневич Г.С. Формальные онтологии //Материалы международной научно-технической конференции OSTIS-2012 «Открытые семантические технологии проектирования интеллектуальных систем».– Минск: Изд-во БГУИР, 2012.
- [Плесневич, 2014] Плесневич Г.С. Один язык для спецификации онтологий //Материалы международной научно-технической конференции OSTIS-2014 «Открытые семантические технологии проектирования интеллектуальных систем».– Минск: Изд-во БГУИР, 2012.

SPECIFICATION OF TEMPORAL RELATIONS IN THE SYSTEM “BINARY MODEL OF KNOWLEDGE”

Plesniewicz G.S., Nguyen Thi Minh Vu

National Research University «MPEI», Moscow, Russia

salve777@mail.ru

mimhvu.357@gmail.com

In the paper a conceptual language with facilities for temporal relations specification are briefly described. Related to the language, languages for structural and logical specificatins are considered; they support the system «Binary Model of Knowledge» that intends for design and analysis of ontologies.

INTRODUCTION

The notion of time is ubiquitous in any agents activity that requires intelligence. Working with an intelligent information system, agents (natural or artificial) deal with a model of the real world that often is dynamic. For temporal knowledge specification appropriate languages are used. We will describe (in outline) a language for the specification of the temporal relations in the system “Binary Model of Knowledge” (BMK), and describe a method of logical inference for this language.

MAIN PART

The main part of the paper consists of three sections. In the first section, we describe the formal model of concept; it is used for defining denotative semantics of conceptual languages. Then we describe (in outline and by examples) the language LSS for structural specification in BMK.

In the second section we describe (in outline and by examples) the language LTS for temporal specification in BMK. The language is close to the Schmiedel’s language of temporal terminological (description) logic.

In the third section we consider the problem of disigning deduction methods for the language LTS. These methods are applied to statements for temporal intervals of events. We give an example of deduction for fuzzy statements.

CONCLUSION

In the present paper we have introduced a temporal conceptual language for supporting the system “Binary Model of knowledge” and have given an example of deduction in the language.



OSTIS-2015

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822:514

ОТ ТЕРМИНОЛОГИЧЕСКИХ СЕТЕЙ К ТОЛКОВЫМ СЛОВАРЯМ

Мальковский М.Г., Соловьев С.Ю.

Факультет ВМК МГУ имени М.В.Ломоносова, г.Москва, Россия

malk@cs.msu.su

soloviev@glossary.ru

Рассматривается задача преобразования терминологической сети в толковый словарь идеографического типа. Выявляются особенности и трудности поставленной задачи, а также описываются свойства терминологических сетей полезные для ее решения. Обсуждаются принципы организации специального программного обеспечения, поддерживающего интеллектуальную деятельность составителя словаря.

Ключевые слова: терминологическая сеть; толковый словарь; термин; определение.

Введение

Терминологическая сеть [Мальковский и др., 2012] есть продукт деятельности коллектива научных редакторов, занятых систематизацией терминов, представленных своими определениями-толкованиями. С формальной точки зрения терминологическая сеть представляет собой ориентированный граф, узлами которого являются термины, а дугами – экземпляры бинарных отношений из заранее фиксированного набора допустимых отношений. В терминологических сетях:

- каждая дуга $(A, B)_R$ есть упорядоченная пара узлов A и B , помеченная символом отношения R ; если для дуги не оговаривается тип отношения, то в записи такой дуги метка опускается;
- набор допустимых отношений обязательно содержит родовидовое отношение P , которому соответствуют дуги $(A, B)_P$, где A – вид, B – род;
- понятийным узлом называется узел, в который заходит хотя бы одна дуга;
- потомками понятийного узла B называются узлы A , связанные с B дугами (A, B) ;
- предками понятийного узла A называются узлы B , связанные с A дугами (A, B) ;
- каждый понятийный узел имеет уникальное имя, которое также служит наименованием понятия;
- как правило, наименование понятия есть общее наименование объектов, составляющих его объем: “Музыка”, “Музыкальные инструменты”, “Камерные музыкальные произведения” и т.д.

Выявление терминов, пригодных для образования понятийных узлов, а также

установление отношений между терминами составляет основу интеллектуальной деятельности научных редакторов. В своей работе редакторы используют определения терминов, справочную литературу и собственные знания.

В больших терминологических сетях выделяются так называемые терминологические кластеры, каждый из которых с определенными оговорками можно рассматривать как терминосистему [Шелов, 2003] некоторой проблемной области. Определяющей характеристикой кластера является его центр – выделенный понятийный узел, имя которого отождествляется с наименованием проблемной области. Типичный кластер содержит около 1000 определений терминов, сгруппированных вокруг 150-170 понятий. Количество бинарных отношений кластера достигает 1500 экземпляров, 600 из которых являются родовидовыми.

Переход от терминологической сети заданного кластера к толковому словарю идеографического типа¹ выполняется человеком. Конечная цель составителя словаря состоит в построении последовательности понятийных (тематических) гнезд, каждое из которых – в первом приближении – представляет собой группу взаимосвязанных терминов, раскрывающих через свои определения различные аспекты одного, особо выделенного термина (центрального термина понятийного гнезда; центра гнезда). По формальным признакам

¹ Толковые словари идеографического типа встречаются не часто, однако их уникальная способность служить одновременно и справочником, и учебником выгодно отличает их (см., например, [Годмен, 1988]) от прочей справочной литературы.

понятийное гнездо полностью подобно кластеру, отличие состоит лишь в назначении. Кластер фиксирует понятийный аппарат проблемной области, а гнездо есть часть понятийного аппарата, организованная в виде удобном для чтения. В словаре понятийное гнездо имеет конкретную печатную форму, содержащую наименование, оглавление и термины гнезда, а также схемы и отсылочные списки.

В качестве наименования гнезда, фигурирующим в его заголовке, используется имя центрального термина. Вопросы расположения терминов в гнезде выходят за пределы настоящего изложения, заметим лишь, что здесь не имеет смысла использовать алфавитный порядок, поэтому в интересах навигации все статьи словаря имеют единую нумерацию, и точная ссылка на статью состоит из термина и номера.

Как правило, понятийное гнездо содержит некоторое количество терминов, связанных родовидовыми отношениями с его центром. Традиционно [Гринева-Гринева, 2008] считается, что родовидовая структура является важнейшим компонентом терминологии. Более того, в развитых толковых словарях² родовидовые ссылки являются обязательными атрибутами словарных статей. В относительно компактных гнездах родовидовую иерархию удобно представлять схемой, что позволяет отказаться от атрибутивного метода указания связей между терминами гнезда.

1. Подход к построению словарей

При построении словаря перед его составителем стоят две трудные задачи. Первая задача состоит в необходимости преобразовать терминологическую сеть в структуру словаря. Вторая задача заключается в необходимости переосмыслить значительный объем, вообще говоря, излишне структурированной информации. Подход к разрешению этой проблемной ситуации состоит в создании интерактивного программного инструментария, способного нетривиально поддерживать деятельность составителя толковых словарей. Инструментарий позволяет оперировать так называемыми терминологическими блоками, которые следует рассматривать как прототипы будущих понятийных гнезд. В процессе построения словаря составитель изменяет количество и состав терминологических блоков, предписывая или запрещая построение тех или иных конкретных блоков. Формально терминологический блок представляет собой подсеть, в которой

- выделен один понятийный узел C (центр блока);
- прочие узлы A_1, \dots, A_n связаны с центром маршрутом $(A_i, A_{i1}) (A_{i1}, A_{i2}) \dots (A_{ik}, C)$;
- имеется множество узлов, связанных с центром родовидовыми связями (скелет блока);

² См., например, [Михальченко, 2006].

- потомки каждого понятийного узла либо (i) представлены полностью, либо (ii) полностью отсутствуют.

Понятийные узлы блока, отличные от центра и удовлетворяющие условию (i), называются внутренними узлами. Понятийные узлы блока, удовлетворяющие условию (ii), образуют сечение блока.

В определении терминологического блока зафиксированы желательные свойства понятийного гнезда: во-первых, блок не должен иметь пропусков и, во-вторых, блок должен обладать определенной системой подвидов. Кроме того, последнее свойство терминологических блоков позволяет однозначно доопределять частично заданные блоки минимальным набором узлов. В общем случае, для построения блока достаточно знать его центр и сечение. Базовый алгоритм построения совокупности терминологических блоков по известному терминологическому кластеру имеет следующий вид:

Этап 1. Считать центр кластера центром первого подлежащего построению блока.

Этап 2. Построить скелет блока как совокупность узлов, прямо или косвенно связанных с центром блока родовидовыми связями;

Этап 3. Построить блок (а) доопределением его скелета и (б) включением в его состав потомков тех узлов сечения, которые не имеют собственных подвидов.

Этап 4. Использовать узлы из сечения построенного блока в качестве центров новых терминологических блоков.

Предложенный алгоритм позволяет построить от 50 до 70 начальных терминологических блоков, которые способны сформировать у составителя словаря представления об оптимальных числовых характеристиках потенциальных понятийных гнезд: об общем объеме блока, об объеме его скелета³ и о количестве раскрытых понятий. Центры блоков-лидеров и блоков-аутсайдеров отмечаются предписывающими и запрещающими метками, которые используются при повторных вычислениях терминологических блоков. Алгоритм повторного построения терминологических блоков отличается от базового алгоритма двумя правилами обработки, имеющими безусловный приоритет.

Правило 1. Если в блок заносится узел, отмеченный предписывающей меткой, то этот узел заносится в сечение, и его потомки в дальнейшем построении блока участия не принимают.

Правило 2. Если в блок заносится узел, отмеченный запрещающей меткой, то потомки этого узла заносятся в блок автоматически.

Описанный процесс манипулирования

³ Есть основания предполагать наличие прямой зависимости между общим количеством терминов понятийного гнезда и количеством терминов в его родовидовой структуре.

терминологическими блоками с помощью меток практически неизбежно приводит к повторному использованию в различных блоках одних и тех же узлов. Составитель словаря вынужден противодействовать этому явлению; в противном случае в разных местах словаря появятся точные копии одних и тех же словарных статей. С точки зрения методов противодействия следует различать два случая возникновения узлов-дубликатов.

Случай 1 | Узлы сечений используются в качестве центров иных терминологических блоков. Ситуация штатная; выход из нее состоит в таком редактировании словарной статьи из узла-дубликата, при котором в тексте определения выделяется его анонс. Обработка такого рода известна, она прослеживается, например, в определениях из словаря селевых явлений [Перов, 1996]:

Отвал – насыпь из пустых горных пород, некондиционных полезных ископаемых, шлака. Отвал может размещаться в отрицательных формах рельефа или образовывать положительные. Отвал часто служит очагом зарождения или твердого питания антропогенных селей. (В этом определении анонс выделен курсивом.)

Полное определение используется лишь в одноименном гнезде, анонс же используется во всех остальных случаях. Дополнительно в анонс включается ссылка на гнездо:

Отвал – насыпь из пустых горных пород, некондиционных полезных ископаемых, шлака. [См. Отвалы]

Случай 2 | В заданном наборе блоков некоторые внутренние и непонятные узлы одновременно встречаются в двух и более блоках. В дальнейшем такие узлы называются спорными.

2. Подход к обработке спорных узлов

Появление спорных узлов объясняется наличием у них нескольких предков, попадающих в разные блоки. Проблему существования спорных узлов можно разрешить, если для каждого из них указать наиболее предпочтительного предка. Соответствующее решение принимает составитель словаря, а программный инструментариий обеспечивает комфортные условия для принятия адекватного решения. Подобное “разделение обязанностей” характерно для систем приобретения знаний [Осипов, 2013], в частности, для выбора наиболее предпочтительного предка наилучшим образом подходит метод сопоставления.

Программный инструментариий, оставляя окончательное решение на усмотрение составителя словаря, тем не менее, позволяет выполнить формальное исследование контекста того или иного спорного узла с целью выявления его необязательных предков. Исходя из особенностей терминологических сетей [Соловьев, 2008], можно

выделить, по крайней мере, пять эвристик для оценки контекстов.

Эвристика 1. Если все потомки B_1, \dots, B_k понятийного узла A имеют не менее двух предков, то все эти потомки B_1, \dots, B_k узла A могут считаться необязательными (рисунок 1).

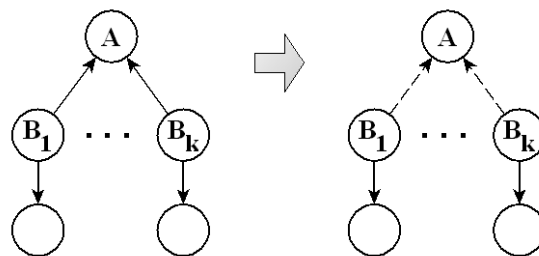


Рисунок 1 – Эвристика 1

Эвристика 2. Если узлы A, B, C связаны отношениями $(B, A), (C, A)$ и (B, C) , то потомок B узла C может считаться необязательным (рисунок 2).

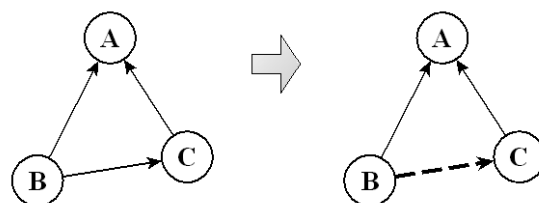


Рисунок 2 – Эвристика 2

Эвристика 3. Если узлы A, B, C, D связаны отношениями $(B, A), (C, A)_P, (D, B)_P$ и (D, C) , то потомок D узла B может считаться необязательным (рисунок 3).

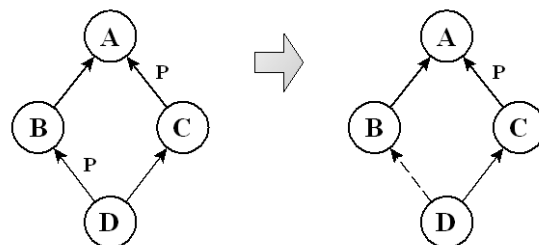


Рисунок 3 – Эвристика 3

Эвристика 4. Если узлы A, B, C, D связаны отношениями $(B, A), (C, A), (D, B)$ и (D, C) , то (а) потомок D узла B может считаться необязательным, (б) потомок D узла C может считаться необязательным, и (в) узел D может считаться потомком узла A (рисунок 4).

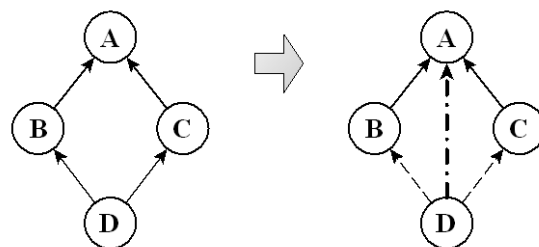


Рисунок 4 – Эвристика 4

Заметим, что эвристики 4 и 3 находятся в

соотношении “правило–исключение”, поэтому эвристика 4 применяется в случае, когда эвристика 3 неприменима.

Эвристика 5. Если среди предков понятийного узла A имеется ровно один узел B , для которого $h(B) = 1$, то потомок A узла B может считаться необязательным (рисунок 5). Здесь $h(B)$ – количество понятийных узлов из числа потомков узла B .

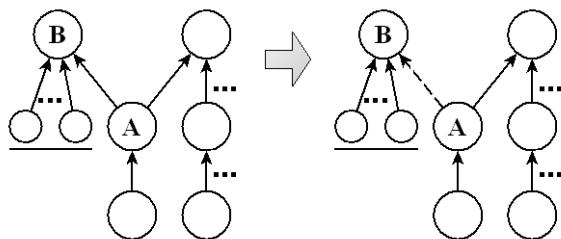


Рисунок 5 – Эвристика 5

При построении терминологических блоков необязательные потомки игнорируются, а в понятийном гнезде имена проигнорированных потомков заносятся в отсылочный список “См. также”, которым завершается печатная форма гнезда.

Заключение

Описанный подход навязывает определенную технологию творческому процессу составления толкового словаря. Ослабить сковывающие рамки технологии можно за счет гибкой реализации соответствующего программного инструментария, скажем, в рамках текстового процессора или системы поддержки творческих картотек. Вместе с тем, программная реализация должна допускать расширение базовых подходов. Например, в толковых словарях изредка встречаются понятийные гнезда с двумя центрами, требующие особых методов обработки. В список вопросов нуждающихся в теоретическом осмыслении, входят: обработка терминов, не имеющих определений, и структурирование терминов в гнезде.

Библиографический список

- [Годмен, 1988] Иллюстрированный химический словарь / А.Годмен. – М.: Мир, 1988.
- [Гринев-Гриневиц, 2008] Введение в терминографию / С.В.Гринев-Гриневиц. – М.: ЛИБРОКОМ, 2009.
- [Мальковский и др., 2012] Мальковский М.Г., Терминологические сети / М.Г.Мальковский, С.Ю.Соловьев // Материалы конференции OSTIS-2012. – Мн.: БГУИР, 2012. С. 77-82
- [Михальченко, 2006] Словарь социолингвистических терминов / Отв. ред. Ю.В.Михальченко. – М.: НИЦ НЯО, 2006.
- [Осипов, 2013] Лекции по искусственному интеллекту / Г.С.Осипов. – М.: ЛИБРОКОМ, 2013.
- [Перов, 1996] Селевые явления. Терминологический словарь / В.Ф.Перов. – М.: Изд-во Моск. ун-та, 1996.
- [Соловьев, 2008] Соловьев С.Ю. Схема и формула глоссария / С.Ю.Соловьев // Труды конференции КИИ-2008, т.2. – М.: ЛЕНАД, 2008. С.157-164
- [Шелов, 2003] Термин. Терминологичность. Терминологические определения / С.Д.Шелов. – СПб.: Филологический факультет СПбГУ, 2003.

FROM TERMINOLOGICAL NETWORKS TO THE EXPLANATORY DICTIONARIES

Malkovsky M.G., Soloviev S.Y.

Lomonosov MSU, Moscow, Russia

malk@cs.msu.su

soloviev@glossary.ru

In this article we

- consider the problem of converting existing terminological network into the ideographic explanatory dictionary;
- analyze the characteristics of the data and difficulties in solving the problem;
- describe the properties of terminological network which can be used to solve the problem;
- formulate the principles of software designed to support the dictionary compiler intellectual activity.

Introduction

In our problem the initial data is a cluster of terminological network. Cluster has some useful (for our purposes) numerical and structural characteristics.

Main Part

The compiler creates a dictionary as a set of conceptual slots. Slot describes some subconcept by listing its terms-aspects. Additionally, the slot has a name and contains scheme of relationships between terms. In its activities, the compiler of a dictionary faced with the task of forming slots and with the task of optimizing the number of cross-references between them.

Our idea is to develop a software tool that is able to support the activities of the compiler. The tool allows the compiler to build and adjust terminological blocks. The process ends when the terminological blocks can be converted into conceptual slots. When the compiler checks the blocks he should handle cross-references of two types. References of the first type define connection between the slots, these references require editing the definitions of certain terms. References of the second type are the horizontal connection between the terms, these links affect the contents of terminological blocks. Some heuristics to help handle the second type references.

Conclusion

In essence, the described approach imposes certain technology to the creative process of drawing up an explanatory dictionary. Loosen fetter frame technology can be due to the flexible implementation of appropriate software tools.



OSTIS-2015

(Open Semantic Technologies for Intelligent Systems)

УДК 004;621.398;681.5

ОБ ОДНОМ ПОДХОДЕ ТЕМАТИЧЕСКОГО МОДЕЛИРОВАНИЯ ТЕКСТОВ НА ЕСТЕСТВЕННОМ ЯЗЫКЕ

Чан Ван Ан

*Вьетнамский Государственный Технический Университет имени Ле Куи Дона,
г. Ханой, Вьетнам*

tavistu@gmail.com

В статье предлагается новый подход тематического моделирования текстов на естественном. Идея заключается в том, что, во-первых, создание метода представления текстов, где учитываются смысловые взаимосвязи между терминами, во-вторых, создание метода распределения терминов на соответствующие тематики. Смысловые взаимосвязи между терминами изображаются в виде графа, где, автором предлагается формула для определения рёбер между вершинами графа.

Ключевые слова: тематическое моделирование текстов; метод представления текстов; граф текста; латентно-семантический анализ.

Введение

В 1958 году Герхард Лисовски и Деонард Рост предложили работу по составлению каталога религиозных текстов на иврите, призванных помочь ученым определить значения терминов, которые были давно утрачены. Путём кропотливой ручной работы они собрали воедино все возможные контексты, в которых появлялся каждый из терминов. Следующей задачей было научиться игнорировать несущественные различия в формах слов и выделять те различия, которые влияют на семантику. Трудности, с которыми столкнулись Лисовски и Рост полвека назад, часто возникают и сегодня при автоматическом анализе текстов. Теоретически обоснованным и активно развивающимся направлением в анализе текстов на естественном языке, призванным решать перечисленные задачи, является тематическое моделирование коллекций текстовых документов [Коршунов, 2012].

Построение тематической модели может рассматриваться как задача одновременной кластеризации документов и слов по одному и тому же множеству кластеров, называемых темами. В терминах кластерного анализа тема — это результат би-кластеризации, то есть одновременной кластеризации и слов, и документов по их семантической близости. Обычно выполняется нечёткая кластеризация, то есть документ может принадлежать нескольким темам в различной степени. Таким образом, сжатое семантическое описание слова или документа представляет собой

вероятностное распределение на множестве тем. Процесс нахождения этих распределений и называется тематическим моделированием

Тематическое моделирование текстов играет важную роль в различных областях информационной технологии, как в организации баз знаний, поиске, реферировании коллекции документов и новостных потоков, фильтрации спама и т.д. При поиске, с помощью классификации текстовых документов, можно найти наиболее эффективные информации, сократить трудозатраты на поиск нужной информации.

1. Подходы тематических моделирований

1.1. Латентно-семантический анализ (LSA)

Существуют некоторые подходы тематических моделирований. Одним из них является латентно-семантический анализ (LSA) [Landauer, 1998] для выявления структуры семантических взаимосвязей между используемыми словами.

LSA позволяет выявлять значения слов с учетом контекста их использования путем обработки большого набора текстов. Принцип действия метода заключается в том, что сравнение множества всех контекстов, в которых слова или группы слов употребляются, и контекстов, в которых они не употребляются, позволяет сделать вывод о степени близости смысла этих слов или групп слов.

В качестве исходной информации LSA использует матрицу термины-на-документы.

Элементы этой матрицы содержат веса терминов в документах, назначенные с помощью выбранной весовой функции. В качестве примера можно рассмотреть самый простой вариант такой матрицы, в которой вес термина равен 1, если он встретился в документе (независимо от количества появлений), и 0 если не встретился (Рисунок 1).

	d_1	d_2	d_{n-1}	d_n
w_1	1	1	0	1
w_2	0	1	1	1
...
...
w_m	1	0	0	1
w_{m-1}	1	1	1	0

Рисунок 1 – Матрица “термы-на-документы”.

Где d_i – документы в коллекции;

w_i – термины, извлеченные из коллекции документов.

При применении данного метода, существуют следующие недостатки:

- матрица термы-на-документы обладает большим размером, из-за того, что, коллекция документов содержит много терминов;
- не учитываются смысловые взаимосвязи между терминами.

1.2. Метод извлечения терминов из текстов

В работе [Усталов, 2012] предложен метод извлечения терминов из текстов на русском языке при помощи графовых моделей: термины являются вершинами графа, связи между ними образуются путем последовательно сканирования текста заданным окном из $N \in [2, 10]$ слов. На каждой итерации для пары слов вычисляется величина связи $WC(w_1, w_2)$, обратно зависящая от расстояния между словами:

$$WC(w_1, w_2) = \begin{cases} 1 - \frac{d(w_1, w_2) - 1}{N - 1}, & \text{если } d(w_1, w_2) \in (0, N) \\ 0 & \text{если } d(w_1, w_2) \geq N, \end{cases}$$

где w_1 и w_2 - слова, $d(w_1, w_2)$ - расстояние между словами, N - размер окна.

Слова, для которых величина $WC(w_1, w_2)$ приняла нулевое значение, не включаются во множество вершин графа.

Основанием для вычисления величины $WC(w_1, w_2)$ служит наблюдения, что между двумя рядом стоящими словами часто существует семантическое отношение. Это необходимо для

обеспечения связности представления текста в виде графа. Чем выше расстояние $d(w_1, w_2)$, тем ниже вероятность существования такого отношения.

При обработке графа, работа [Усталов, 2012] ведётся исключительно с одиночными именами существительными и прилагательными. Объединение этих слов в словосочетания будет выполнено на этапе сборки словосочетаний.

Данный метод обладает большим недостатком:

- учёт смысловых взаимосвязей между терминами только для тех, что находятся только в одном окне;
- не учитывается частота встречаемости пары терминов, которые находятся в одном предложении, хотя такие термины имеют смысловую взаимосвязь.

2. Новый подход к тематическому моделированию с предлагаемым методом представления текстов

Две выше работы являются наиболее близкими к данной работе. Идея заключается в том, что, во-первых, создание метода представления текстов, где учитываются смысловые взаимосвязи между терминами, и, во-вторых, создание метода распределения терминов на соответствующие тематики.

Пусть D является входной коллекцией документов (множество текстовых документов). После нормализации коллекции документов получается множество нормальных предложений, и формируется $W = \{w_i \mid i = 1..n\}$ – множество в них терминов. Каждый документ $d \in D$ представляет собой последовательность терминов $(w_{1d}, w_{2d}, \dots, w_{md})$ из множества W , $(m \leq n)$. Термин может повторяться в документе несколько раз.

По мнению Воронцова К. В. [Воронцов, 2012], тема - это набор терминов, неслучайно часто совместно встречающихся в относительно узком подмножестве документов. Темы описывают содержание коллекции документов. Пусть $T = \{t_i \mid i = 1..k\}, k < n$ – множество тем в коллекции документов D . Каждая тема создается путем группирования некоторых терминов из множества терминов $C - t_i = \{w_j \mid j = 1..h\}, h < n$. Причем, каждый термин находится в одной теме. Это значит:

$$\begin{cases} \bigcup_{i=1}^{|T|} t_i = W \\ \bigcap_{i=1}^{|T|} t_i = \phi \end{cases}$$

Тогда коллекция документов рассматривается в виде $D = \langle W, T \rangle$.

При классификации документов, нужно решить следующие задачи:

- извлечение терминов из коллекции документов,
- группирование терминов по тематикам.

Такой процесс описывается следующей схемой (рисунок 2):



Рисунок 2 – Схема процесса группирования терминов по тематикам.

Как видно из этой схемы, в начале, осуществляется нормализация документов с целью получения слов в виде инфинитива. После чего предлагается сделать шаг извлечения терминов из нормальных документов, вместе с тем, выполняется разбиение предложений, это важный шаг для вычисления величины связей между терминами. Особенностью предлагаемой схемы является представление коллекции документов в виде графа, где, вершины графа является терминами, между ними существуют связи, которые являются ребрами графа. Как видно из схемы, после шага построения графа связей между терминами, выполняется процедура группирования терминов по тематикам.

В работе [Гринева, 2009] предложен метод извлечения ключевых терминов. Такой метод состоит из следующих шагов:

- извлечение терминов-кандидатов;
- объединение всех синонимов.

При извлечении терминов-кандидатов исходная коллекция документов разбивается на лексемы. С

помощью словаря объединяются все термины, близкие по смыслу и значению.

По списку полученных терминов на предыдущем шаге, строится семантический граф, вершинами которого являются термины документа, наличие ребра между двумя вершинами означает тот факт, что термины семантически связаны между собой, вес ребра является численным значением семантической близости двух терминов, которые соединяет данное ребро. Граф предлагается в виде $G = \langle W, E \rangle$, где W - множество терминов коллекции документов D , а E - связи между ними. Предлагается, что чем ближе по смыслу, тем меньше по расстоянию.

Для определения связей между терминами, предлагаются следующие гипотезы:

- учет взаимосвязи между терминами;
- термины, которые находятся в одном предложении, имеют связь по смыслу;

Для оценки величины связи предлагается параметр $P(w_i, w_j)$, значение данной оценки тем меньше, чем больше смысловая взаимосвязь между терминами w_i и w_j ;

частота встречаемости пары терминов w_i и w_j тем больше, чем больше смысловая взаимосвязь между терминами w_i и w_j .

Из этих гипотез автором предлагается формула для определения ребер между вершинами графа:

$$P(w_i, w_j) = \begin{cases} \prod_{k=1}^F \frac{\min[s_k(w_i, w_j)]}{L_k}, & \text{если } w_i \text{ и } w_j \\ & \text{в одном предложении} \\ 0, & \text{если } w_i \text{ и } w_j \text{ ни раз} \\ & \text{не находятся в одном предложении} \end{cases} \quad (1)$$

где:

F - частота встречаемости пары терминов w_i и w_j в предложениях;

$s_k(w_i, w_j)$ - позиционная мера пары терминов w_i и w_j в k -ом предложении;

L_k - длина k -ого предложения.

Как отмечено в работе [Усталов, 2012],

$s_k(w_i, w_j) = |p_k(w_j) - p_k(w_i)|$, при $w_i \neq w_j$, где $p_k(w_i)$ - порядковый номер термина w_i в k -ом предложении. Чем выше расстояние

$s_k(w_i, w_j)$, тем ниже вероятность существования такого отношения.

Если в k -ом предложении существуют несколько пар терминов c_i и c_j , то выбирается пара, которая имеет самую маленькое значение позиционной меры - $\min[s_k(w_i, w_j)]$.

3. Анализ результата

По данному методу проведем анализ следующих текстов:

Первый текст:

В статье предлагаются модели и средства для интерактивного взаимодействия в процессе обучения неродному языку.

Описывается четыре вида интерактивного взаимодействия: интерактивное взаимодействие между членами курса при обучении теоретических основ курсов, интерактивность при взаимодействии между пользователями по принципу «социальной сети», интерактивность при проведении практики, интерактивное взаимодействие между учащимися и средствами системы при самообучении.

Второй текст:

В статье предлагается математическая модель процесса обучения неродному языку, как процесса происходящего с одновременным присутствием двух противоположностей: наличием родного языка и неродного языка. Системы с такими свойствами называются амбивалентными.

Для реализации обучающей среды предлагается технологическая схема обучения.

Третий текст:

В статье предлагается математическая модель амбивалентной системы обучения неродному языку, в которой отражаются такие особенности процесса обучения как появление отката и окостенения.

На основе применения численных методов таких, как метод золотого сечения и метод Тьюки определяются моменты времени начала этих явлений. Предложенное в данной статье решение этой задачи позволит увеличить эффективность процесса обучения неродному языку.

После нормализации, (в данной работе рассматриваются только существительные и прилагательные), получается следующий нормальный фрагмент с семантическим графом:

Первый текст:

Статья| математический| модель| процесс| обучение| неродной| язык|, процесс| противоположность| родной| язык| неродной| язык|. Система| свойство| амбивалентный|.

Реализация| обучающий| среда| технологический| схема| обучение|.

Второй текст:

Статья| математический| модель| процесс| обучение| неродной| язык|, процесс| противоположность| родной| язык| неродной| язык|. Система| свойство| амбивалентный|. Математический| модель| дифференциальный| уравнение| Колмогоров|. Модель| обучение| основа| уравнение| зависимость| уровень| знание| обучаемый| начало| уровень| знание|, интенсивность| процесс| изучение| неродной|

Третий текст:

Статья| математический| модель| амбивалентный| система| обучение| неродной| язык| особенность| процесс| обучение| появление| откат| окостенение|. Основа| применение| численный| метод| метод| золотой| сечение| метод| Тьюки| момент| время| начало| явление|. Предложенный| данный| статья| решение| задача| эффективность| процесс| обучение| неродной| язык|.

Используя формулу (1) получаем следующий семантический граф терминов:

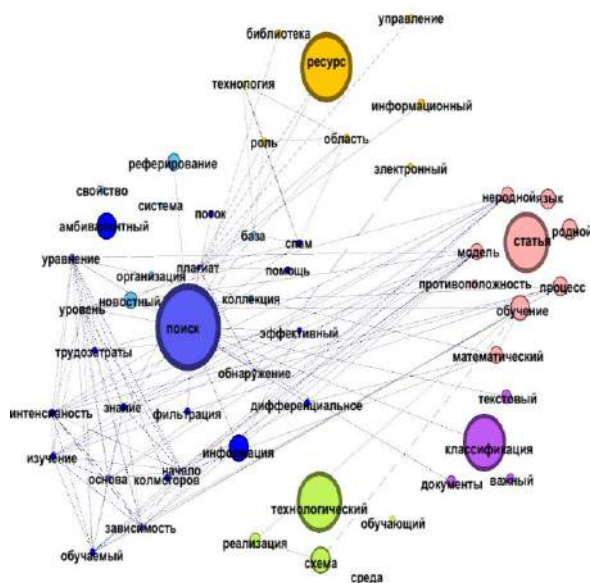


Рисунок 3 - Семантический граф терминов.

После получения семантического графа терминов выполняется шаг автоматического обнаружения кластеров терминов. При решении данной задачи был выбран метод K-medoid [Гринева, 2009] для распределения набора объектов графа (терминов) по k группам, таким образом, чтобы конечная сумма несхожести между каждым объектом была минимальна. То есть, используется критерий абсолютной погрешности. По такому методу, для выше приведенного примера, если данная коллекция документов распределяется по 4 тематикам, то получается следующие таблицы 1 и 2 распределения терминов.

Таблица 1. Распределения терминов на 4 тематик

Тематика 1	Тематика 2
неродной	момент
уровень	начало
знание	явление
интенсивность	время
процесс	метод
язык	золотой
обучение	сечение
система	тьюки
амбивалентный	численный
свойство	применение
эффективность	основа
особенность	
появление	
откат	
окостенение	
противоположность	
родной	
дифференциальный	
уравнение	
зависимость	
обучаемый	
изучение	
колмогоров	

Таблица 2. Распределения терминов на 4 тематик

Тематика 3	Тематика 4
технологический	статья
реализация	модель
обучающий	математический
среда	решение
схема	задача

При распределении на 10 тематик получается таблица 3, 4, 5, 6 и 7:

Таблица 3. Распределения терминов на 10 тематик

Тематика 1	Тематика 2
процесс	момент
модель	время
обучение	
появление	
откат	
окостенение	
особенность	
противоположность	
изучение	

дифференциальный	
колмогоров	
эффективность	
задача	

Таблица 4. Распределения терминов на 10 тематик

Тематика 3	Тематика 4
технологический	статья
реализация	решение
обучающий	математический
среда	
схема	

Таблица 5. Распределения терминов на 10 тематик

Тематика 5	Тематика 6
тьюки	начало
золотой	уровень
метод	зависимость
сечение	уравнение
	обучаемый
	явление

Таблица 6. Распределения терминов на 10 тематик

Тематика 7	Тематика 8
применение	амбивалентный
численный	свойство
основа	система

Таблица 7. Распределения терминов на 10 тематик

Тематика 9	Тематика 10
интенсивность	неродной
знание	родной
	язык

Как видно из полученных таблиц, термины в одной тематике имеют сильные смысловые взаимосвязи, такие термины характеризуют содержательный смысл тематик, в которых они находятся.

Заключение

В статье был представлен новый подход тематического моделирования текстов на естественном, которое является перспективным инструментом для обработки больших коллекций документов.

Данный подход позволяет автоматически систематизировать и реферировать электронные архивы большого масштаба, который человек не в

силах обработать, даёт возможность использовать ресурсы Интернет не только как корпус текстов, но и как полноценную базу данных.

Библиографический список

[Коршунов, 2012] Коршунов, А. В. Тематическое моделирование текстов на естественном языке / А. В. Коршунов, А. Г. Гомзин // Труды Института системного программирования РАН, 2012, Том 23, С. 215-242.

[Landauer, 1998] Landauer, T. An introduction to latent semantic analysis / T. Landauer, P. Foltz, and D. Laham // In Discourse Processes, 1998, v. 25, pp. 259-284.

[Усталов, 2012] Усталов, Д. А. Извлечение терминов из русскоязычных текстов при помощи графовых моделей/Д. А. Усталов// Теория графов и приложения. Материалы конференции. С. 62-69.

[Воронцов, 2012] Воронцов, К.В. Регуляризация, робастность и разреженность вероятностных тематических моделей / К.В. Воронцов, А.А. Потапенко// Компьютерные исследования и моделирование, 2012, Том 4, № 4.

[Гринева, 2009] Гринева, М. Анализ текстовых документов для извлечения тематически сгруппированных ключевых терминов / М. Гринева, М. Гринев // Труды Института системного программирования РАН, 2009, Том 16, С. 155-165.

ABOUT AN APPROACH OF TOPIC MODELING FOR TEXT IN NATURAL LANGUAGE

Tran Van An *

* *Le Quy Don University of Science and Technology, Ha Noi, VietNam*

tavistu@gmail.com

Introduction

The paper proposes a new approach of topic modeling for text in natural language. Firstly, the creation a method for submission of texts. Secondly, the creation of the distribution terms for each relevant topics. Semantic relationships between terms are represented as a graph. The author proposes a formula for define the edges between the vertices of the graph.

Main part

Constructed semantic graph whose vertices are the terms of the document. Terms semantically related to each other, the weight of an edge is the numerical value of the semantic proximity of the two terms, which connects the given edge.

After receiving the semantic graph of terms is performed step automatic detection of clusters of terms. In solving this problem was chosen method of K-medoid for the distribution of a vertex set of the graph (terms) in groups.

Conclusion

The paper was presented new approach of topic modeling for text in natural language, which is a promising tool for processing large document collections. This approach supports you to automatically organize and abstracted electronic archives of a large scale, which people can not handle, makes it possible to use Internet resources not only as corpus, but also as a full-fledged database.



OSTIS-2015

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822:514

MOTIVATION OF MODELING COMMONSENSE REASONING PROCESS

Naidenova X. A.

*S. M. Kirov Military medical academy
Saint Petersburg, Russia*

ksennaidd@gmail.com

In this article, the role of classification reasoning in human commonsense (plausible) thinking is considered. Modeling this mode of thinking is a key problem for constructing Intelligent Computer Systems.

Key words: classification reasoning; plausible reasoning; machine learning; conceptual thinking.

Introduction

The symbolic methods of machine learning work on objects with symbolic, Boolean, integer, and categorical attributes. With this point of view, these methods can be considered as the methods of mining conceptual knowledge or the methods of conceptual learning.

In this article, we concentrate on the supervised conceptual learning methods. Until **now**, the theory of logical inference does not include classification reasoning as its inalienable component, although precisely the classification reasoning constitutes an integral part of any mode of reasoning. Furthermore, the current models of commonsense reasoning do not include classification too. However, the role of classification in inferences is enormous. Classification, as a process of thinking, performs the following operations:

- 1) forming knowledge and data contexts adequate to a current situation of reasoning;
- 2) reducing the domain of searching for a solution of some problem;
- 3) generalizing or specifying object descriptions;
- 4) interpreting logical expressions on a set of all thinkable objects;
- 5) revealing essential elements of reasoning (objects, attributes, values of attributes etc.);
- 6) revealing the links of object sets and their descriptions with external contexts interrelated with them. This list can be continued.

We believe that conceptual learning is a special class of methods based on mining and using conceptual knowledge the elements of which are objects, attributes (values of attributes), classifications (partitions of objects into disjoint blocks), and links between them. These links are expressed by the use of implications:

“object \leftrightarrow class”, “object \leftrightarrow property”, “values of attributes \leftrightarrow class”, and “subclass \leftrightarrow class”.

We understand commonsense (or plausible) reasoning as a process of thinking based on which the causal connections between objects, their properties and classes of objects are revealed. In fact, commonsense reasoning is critical for the formation of conceptual knowledge or ontology in the contemporary terminology.

Studying the processes of classification within the framework of machine learning and knowledge discovery led to the necessity of reformulating the entire class of symbolic machine learning problems as the problems of finding approximations of a given classification of objects. This reformulation is based on the concept of a good diagnostic test (GDT) for the given classification of objects [Naidenova, 1992]. A good classification test has a dual nature. On the one hand, it is a logical expression in the form of implication or functional dependency; on the other hand, it generates the partition of a set of objects equivalent to the given classification of this set or the partition that is nearest to the given classification with respect to the inclusion relation between partitions.

If we take into account that implications express relations between concepts (the object \leftrightarrow the class, the object \leftrightarrow the property, the property \leftrightarrow the class), we can assume that schemes of inferring and applying implications (rules of the “if-then” type) form the core of classification processes, which, in turn, form the basis of commonsense reasoning. Deductive steps of commonsense reasoning imply using known facts and statements of the “if-then” type to infer consequences from them. To do it, deductive rules of reasoning are applied, the main forms of which are modus ponens, modus tollens, modus ponendo tollens and modus tollendo ponens. Inductive steps imply applying data and existing knowledge to infer new implicative

assertions and correct those that turned out to be in contradiction with the existing knowledge. These steps rely on inductive rules of reasoning represented by inductive canons stated by British logician John Stuart Mill: the Method of Agreement, the Method of Difference, the Joint Method of Agreement and Difference, the Method of Concomitant Variations and the Method of Residues.

The task of inferring all good diagnostic tests is formulated as searching for the best approximations of a given classification (a partitioning) on a given set of objects. A whole class of machine learning problems, namely, symbolic supervised learning can be reduced to inferring good classification tests from a given dataset (contexts). Good classification tests serve as left parts of implicative assertions, functional dependencies, and association rules.

The concept of a good classification (diagnostic) test underpins our approach to commonsense reasoning.

The analysis of algorithms of searching for all good diagnostic tests in terms of constructing Galois lattice allowed us to decompose algorithms into sub-problems and operations that represent known deductive and inductive modes (*modus operandi*) of commonsense reasoning. Each step of constructing a classification lattice can be interpreted as a mental act. These mental acts can be found in any reasoning: stating new propositions, choosing the relevant part of knowledge and/or data for further steps of reasoning, involving a new rule of reasoning (deductive, abductive, inductive, *trductive*, etc.).

The inferences of lattice construction engage both inductive and deductive reasoning rules. The implicative dependencies (implications, interdictions, rules of compatibility) generated in a process of good tests construction are used immediately in this process for pruning the search space with the aid of deduction.

Note that reasoning begins with using mechanisms for restricting the search space: 1) for each set of values (objects), to avoid constructing all its subsets, 2) for each step of reasoning, to choose a set of values (objects) without which solutions cannot be constructed. For this goal, admissible and essential values (objects) are determined. The search for the admissible or essential values (objects) uses inductive diagnostic rules.

Reasoning requires a lot of techniques related to increasing its efficiency such as valuation, anticipation, making hypotheses, probable reasoning, generalization, and specification. One of the important techniques is decomposition of the main problem into sub-problems. It implies using the following operations: choosing sub-problems, ordering sub-problems (ordering arguments, attributes, objects, variables, etc.), optimizing sub-problem selection, and some others. The most familiar examples of sub-problem ordering are so called tree-like scanning and level wise scanning methods. Some interesting variations of selecting sub-problems are the choice of a more flexible sub-problem, for example, one with minimal difference from a previous sub-problem and a sub-problem with minimal possible

number of new solutions [Zakrevskij, 2013]. Intermediate results of reasoning are used for decreasing or locally bounding the number of sub-problems. Furthermore, it is required, in some cases to use equivalent transformations of data structures. As a whole, reasoning can be considered as gradually extending and narrowing the context of reasoning.

1. Commonsense reasoning in Intelligent Computer Systems

We shall consider the intelligent computer system as a system capable to communicate with the users by means of commonsense reasoning on conceptual knowledge rather than by means of special formal query languages. One of the main principles, which is posed in the foundation of intelligent computer systems, says: "Knowledge is a Means of Data Organization and Management".

The inseparability of data from knowledge with respect to their interacting is manifested in the fact that knowledge governs the process of inputting data in databases. First, there is a mechanism (or it must exist) of recognizing the fact that an inputted portion of information was already earlier perceived or already known, and revealing data not having appeared earlier or not corresponding to what earlier was known. For example, if it was known that birds have wings and fly, but information appears, that X is a bird, has wings, but it does not fly, then "knowing system" must ask how it is necessary to change the knowledge. The formation of knowledge cannot be without this ability to ask. Probably, the computer knowledge base must know how to pose these questions and to obtain the answers to modify the knowledge Base.

In the process of analyzing new data, the necessity also occurs to generate an appropriate context of reasoning. We believe that knowledge must serve for managing the processes of data entering and organization and data must aim at developing knowledge.

The queries to intelligent computer systems can be of the following types:

The factual queries when the answers can be obtained directly from the data;

The conceptual queries when the answers can be obtained via the knowledge.

Consequently, the intelligent system must be capable of recognizing the type of query. The conceptual queries must be interpreted (understood) via the knowledge. Furthermore, answering conceptual questions requires communication between data and knowledge. An intelligent system works like a thinking individual as follows:

PERCEPTION PHASE or ENTERING THE QUERY;
COMPREHENSION or UNDERSTANDING THE QUERY (pattern recognition phase);

FULFILLMENT OF ANSWER TO THE QUERY (commonsense reasoning phase);

QUERYING THE USER if it is necessary and RETURNING to the phase of PERCEPTION.

Entering data/knowledge can have different goals: “IT IS NECESSARY TO KNOW” is a simple message of the user;

“ENTERING NEW DATA WITH ASSIMILATIONS OF THEM by the INTELLIGENT SYSTEM”; it implies the implementation of a dialog interface and a supervised or unsupervised learning process; the result of this process is the upgraded knowledge.

The incorporation of a commonsense reasoning mechanism into data-knowledge processing is becoming an urgent task in intelligent computer systems and conceptual data-knowledge design.

We take into account that data are the source of conceptual knowledge and that knowledge is the means of data organization and management. The following processes are based on commonsense reasoning.

1. Entering and eliminating data:

a) Entering data: by the user or by querying from the side of an intelligent system;

b) Eliminating data: by the user or by an intelligent system (for example, “freezing” data-knowledge).

Entering data by the user implies solving a pattern recognition task. In fact, entering data means enlarging and correcting knowledge such that it will be consistent with the current situation (data).

Eliminating data implies eliminating knowledge inferable from this data. This is the deductive phase of commonsense reasoning.

2. Deductive and inductive query answering requires commonsense reasoning in the form of a dialog between a user and an intelligent system or/and between an intelligent system and an ontology. This reasoning includes:

a) Pattern recognition of the meaning of query (what is required: fact, example, sets of examples, concept, dependency, or classification?);

b) Forming the context of a query (domain of reasoning);

c) Pattern recognition of conceptual level of a query:

Factual level;

Conceptual level with a certain degree of generalization.

There can be the following variants of answering questions:

Reply is in the context of reasoning;

Reply is inferred from the context of reasoning;

Reply requires entering or inferring new knowledge.

Answering questions is connected with extending data about the situation (query) consistent with the system’s knowledge or enlarging the context of reasoning and involving inductive steps of inferring (machine learning) new knowledge.

3. Knowledge optimization is a task for the intelligent system itself, consequently, it requires unsupervised conceptual learning (self-learning) based on unsupervised conceptual clustering (or object generalization) and interpreting the results of clustering (or generalization) via the system’s or ontological knowledge.

4. Automated development of intelligent systems with the incorporated commonsense reasoning mechanisms is currently not supported by any programming language or programming technology. This technology must include:

The possibility to specify concepts (objects) with their properties and inferential links between them;

The possibility to induce some constituent elements of the intelligent system’s knowledge from data by the use of learning mechanisms;

The possibility to incorporate the mechanisms of commonsense reasoning in intelligent systems.

2. The system of Interrelating Classification Operations

2.1. The Operations of Addition and Multiplication Given on the Set of Classes

Two operations are given on the set of classes’ names: the addition operation $+$ and the multiplication operation \circ . To add classes A and B means to define class D of all objects possessing the common properties of classes A and B : $d = a \cap b$, where a, b, d are the set of properties of objects of classes A, B, D , accordingly.

We call $I(x)$, where x is the set of properties of some set of objects the interpretation of x in the power set 2^G , where G is the set of all objects to be considered.

For example, A - the class of ‘blue wooden beads’, B - the class of ‘white wooden beads’, D - the class of ‘wooden beads’: $I(\text{wooden beads}) = I(\text{blue wooden beads}) \cap I(\text{white wooden beads})$ and at the same time $d = \text{‘blue wooden beads’} \cap \text{‘white wooden beads’} = \text{‘wooden beads’}$.

It is insufficient to have only addition operation to deals with classes. How could one form the set of objects possessing at the same time the properties of different classes, for example, “water transport”, “mountain landscape”, throat-microphone, “snow-slip”, “tragicomedy” and so forth. We need in multiplication operation.

To multiply classes A and B means to define class D of all objects having all the properties of class A and all the properties of class B , that is $d = a \cup b$. For example, A - ‘a person who has a child’, B - ‘a person who is a man’, D - ‘father’: $I(\text{father}) = I(\text{a person who has a child}) \cap I(\text{a person who is a man})$ and $d = \text{‘has a child’} \cup \text{‘is a man’}$.

For the completeness of operation’s definition, we shall consider the cases of empty interpretation and empty description. It is possible that the multiplication operation has not a result because of obtaining empty interpretation. In this case, the description obtained is said to be contradictory and to be equal to the special symbol α - ‘inconsistent description’.

Also, it is possible that the result of addition operation is a class with empty description. It means that the objects of the class obtained have no common property. In this case, the description of this class is said to be equal to the special symbol ω .

2.3. The Operations of Subtraction and Division Given on the Set of Classes

One of the important aspects of mental operations is their reversibility [Piaget & Inhelder, 1959]. Addition operation has subtraction as its reverse operation. ($A = D - B$). Reverse operation with respect to multiplication operation is division ($A = D : B$). If subtraction is easy to understand (it is the dissociation of classes), then for division operation it is not the case. Consider the meaning of division operation. For example, a child saw a fox at the picture but he said that it is a dog. According to a child, a dog and a fox are very similar. However, an adult does not agree with the child, he begins to explain: it is not a dog, it does not bike, a fox is wild, it lives in the forest, steals hens, a dog does not do this, it lives at home with people, and it guards hens, eats meals of people and so on. Division operation is necessary for differentiating two concepts. Let's the concept Z be equal to $DOG + FOX$, z be the common property for dog and fox. To divide concepts is to find a property y such that the union of y and z results in the property $c = y \cup z$ corresponding only with the set of dogs and only with this set: $I(c) = I(DOG)$.

2.4. The Operations of Generalization and Specification Given on the Set of Objects' Descriptions

Two operations are given on the sets of objects' descriptions: the operation $*$, or the generalization operation, and the operation \bullet , or the specification or refinement operation. The first one produces for any pair of descriptions $C(o_1)$, $C(o_2)$ their maximal common part $C(o_1) \cap C(o_2)$, the second one produces the minimal description including (containing) $C(o_1)$ and $C(o_2)$, that is $C(o_1) \cup C(o_2)$.

If the result of generalization operation is equal to ω , then it means that object o_1 is unlike o_2 . The specification operation is not defined in case of $C(o_1)$ and $C(o_2)$ are inconsistent ($I(C(o_1)) \cap I(C(o_2)) = \text{empty}$). Then the result of this operation will be equal a special symbol α . It means that there is no object which possesses $C(o_1)$, $C(o_2)$ at a time.

Boldyrev, N.G. (1974) advanced a formalization of object description procedures as algebra with two binary operations of refinement and generalization defined by an axiom system including lattice axioms. This work turned to be basic for developing the classification theory in the scope of algebraic lattices.

2.5. The Set-Theoretical Operations Given on the Set of all Subsets of Objects

The set-theoretical operations of union \cup and intersection \cap are given on the set of all subsets of objects and, consequently, on the set of all classes of objects.

Let X be a set of objects. A non-empty class $L(X)$ of subsets of X such that the union and the intersection of two sets belonging to $L(X)$ also belong to $L(X)$ is an example of a lattice, called a *lattice of sets* or *set lattice* (Rasiova, 1974).

The coordination of classification operations means that the operations on classes' names, on conceivable objects and on objects' descriptions are performed simultaneously and they are in agreement with one another.

The coordinated classification operations generate logical implicative assertions. These assertions can be understood if classifications are performed as the system of coordinated operations. The classification operations are connected with understanding the operations of quantification: "not all c are a ", "all b are c ", "no b are c ", "some c are b ", "some b are not a " and so on. The violation of the coordinated classification operations implies the violation of reasoning. Piaget & Inhelder (1959) have shown that

1) Classification reasoning is a result of gradual development of a person;

2) Appearing the ability to apply formal logical operations is connected with spontaneous appearing the ability to coordinate mental operations;

3) A key problem of the development of operational classification in mind is the problem of understanding the inclusion relation. If understanding this relation is not achieved by a person, then it is impossible for him to understand both the classification and quantification operations.

3. Reasoning as Searching for the Equivalence Relationships on a Set of Expressions

In mind, we operate only on words or names (common or proper). A name can be the name of an object, the name of a class of objects and the name of a classification or collection of classes. A class of objects can contain only one object hence the name of an object is a particular case of the name of a class. Analogically, the name of a class is a particular case of the name of a classification. In the knowledge bases, names of objects and of classes of objects become names of attribute values, and names of classifications become names of attributes.

The equivalence relations on names serve as the foundation of commonsense reasoning. For example the expression "Whale is a mammal" is true because of the fact that the property C exists such that the following expressions occur simultaneously: " $Whale \leq C$ " and " $mammal = C$ ", interpreted on the set of all thinkable animals. By the law of transitivity, we have " $whale \leq mammal$ ".

In the thinking, some concepts are determined via others with the aid of equivalence relations between their names, for example, "father is a man having a child" is the word representation of the dependency on classification names (" $father = man$ ") and (" $to\ have\ a\ child$ "), interpreted on the set of all thinkable men. The knowledge of equivalence relations on the names of classifications makes it possible to draw the conclusions, which would be impossible without this knowledge.

Assume that in one of the agencies the unemployed

persons of young age and the unemployed women are registered. Can one infer that, in this agency, unemployed young women are registered too (in other words, whether the interpretation of expression “(unemployed) and (young) and (women)” is not empty set)? No, it is impossible to achieve this conclusion. But if it is known that “unemployed \leq young”, then it is possible to assert that there are “unemployed young women” in the agency [Laurent, & Spyrtos, 1988].

With the use of the + and \circ operations on names of classes, of the * and \bullet operations on descriptions of objects we produce the expressions which are interpreted with the use of the set-theoretical operations on the set of all subsets of objects, but the mechanisms of generating expressions do not exhaust reasoning. The main point of reasoning is finding the equivalences between the expressions constructed over the names of classes or properties. Two expressions are equivalent if their interpretations are equal. Already the act itself of constructing a class requires establishing the identity between the name of the class and the expression on the object properties defining this class.

The equivalent expressions are simply the different names of the same interpretation. For example, we can meet in the crosswords the following definitions of the concepts requested: “it is the same that destiny”, “the title of prince of royal dynasty in Spain and Portugal”, “a fodder or food plant with fleshy root”. Defining the equivalence of words (expressions) underlies not only constructing concepts and their definitions via object properties but also updating conceptual knowledge, diagnostic reasoning, supplementing imperfect descriptions of objects (classes, situations) with new properties, inferring causal dependencies between properties, and functional dependencies between classifications.

Defining a class of objects via the properties of objects is the traditional task of learning concepts from examples. The relationships between properties are implications in the form $a b c \rightarrow p$, where a, b, c – properties of objects or values of corresponding attributes, p – the name of class of object.

The relationships between classes are defined by the relationships between properties of object belonging to these classes. These relationships generate hierarchical structures on the names of classes coordinated with the structures of properties and the structures of appropriate sets of objects used for interpreting considered classes and properties. The inference of these relationships is also one of the machine learning problems. Moreover, this inference is basic to creating the methodology of ontology construction.

The relationships between classifications are expressed formally by functional dependencies between attributes of objects. The inference of these relationships can be considered as hierarchical knowledge integration.

The principle concept of the GTA is the concept of classification. To give a target classification of objects, we use an additional attribute KL not belonging to U . A target attribute partitions a given set of objects into

disjoint classes the number of which is equal to the number of values of this attribute.

In case of inferring implicative dependencies, we have two classes: 1) the objects in description of which the target value k appears (positive examples); 2) all the other objects (negative examples).

Let $M = \{\cup \text{dom}(attr), attr \subseteq U\}$, where $\text{dom}(attr)$ is the set of all values of $attr$. Let $X \subseteq M$. Let G be the set of objects considered, $G = G_+ \cup G_-$, where G_+ and G_- the sets of positive and negative objects, respectively; let $P(X) = \{\text{all the objects in description of which } X \text{ appears}\}$. We call $P(X)$ the interpretation of X in the power set 2^G . If $P(X)$ contains only positive objects and the number of these objects more than 2, then we call X a description of some positive objects or a test for positive objects.

We define a good test or good description for a subset of positive objects as follows.

Definition 1. A set $X \subseteq M$ of attribute values is a good test for a subset of positive objects if it is a test and no such subset $Y \subseteq M$ exists, so that $P(X) \subset P(Y) \subseteq G_+$.

In [Naidenova, 1992], it is proven that this problem is reduced to searching for causal dependencies in the form $X \rightarrow v$, $X \subseteq M$, v is the name of the given class of positive examples.

Now consider a given classification K and the attributes of U from the point of view of the partitions which are induced by the attribute values on the set G . The good test for a given classification is defined on the base of partition model for relations [Naidenova, 1982; 2012].

We shall use $Kl(A)$ to denote the partition by a collection A of attributes on the set of objects, and we use $Kl(A) \otimes Kl(B)$ to denote the product of two partitions $Kl(A)$ and $Kl(B)$. The relation of partial order over a set of partitions is introduced in the standard way: $Kl(A) \leq Kl(B)$ iff $Kl(A) \otimes Kl(B) = Kl(A)$ [Ore, 1942].

Let K be an additional attributes (or a set of attributes) the values of which partition a given set of objects into disjoint classes.

Definition 2. A subset A of U is a test for a given classification K of a given set of objects, if the following condition is satisfied: $Kl(A) \leq Kl(K)$ ($Kl(X) \otimes Kl(K) = Kl(X)$).

Definition 3. A test A in U for a given classification K for a given set of objects is said to be good if the following condition holds: $(\forall Y \subseteq U) (Kl(Y) \leq Kl(K)) (Kl(A) \leq Kl(Y) \Rightarrow Kl(Y) = Kl(A)$.

In [Naidenova, 1982], it is proven that this problem is reduced to searching for functional dependencies in the form $A \rightarrow K$, $A \subseteq U$, $K \notin U$.

By constructing the all good tests for classes of objects or for classifications of objects, we obtain the expressions of two kinds: $A_1 \cup A_2 \cup \dots \cup A_m \rightarrow K$ and $X_1 \cup X_2 \cup \dots \cup X_m \rightarrow v$. Thus, the Good Tests Analysis allows constructing, simultaneously, the hierarchical structures of object classifications and the descriptions

of object classes in terms of their properties. This knowledge can be used for realizing human commonsense reasoning.

4. Reasoning as a Special Kind of Computing

Reasoning can be reduced to solving the equations of the following types:

- a) "Call things by their correct names": $y = \varphi(x)$, $x = a$, $y = ?$
- b) Find the interpretation x satisfying the equality of expressions φ_1 , φ_2 : $\varphi_1(x) = \varphi_2(x)$ or $a = \varphi_1(x)$;
- c) "Approximate $\varphi()$ with the use of $\varphi_1(), \varphi_2(), \dots, \varphi_k()$ and the set of given operations.

In these equations, x is a sub-domain of reasoning and φ is an expression on the names the interpretation of which is equal to or included in x . Given y , it is necessary to find an interpretation x satisfying the equation $y = \varphi(x)$; Given x it is necessary to find y as an expression the interpretation of which is equal to or included in x .

Example 1. $y = \text{"an infectious children's disease * name of 5 letters * the fist letter is "m" * the last letter is "s"}(x)$ ". To find x is to find the name of the concrete children's disease: $x = \text{mumps}$.

Example 2. $y = \varphi(\text{voltmeter})$; *voltmeter* belongs to a class of electric instruments, since "*voltmeter* \leq *electric instrument*". Furthermore, this instrument serves for measuring tension. Thus, $y = \text{electric instrument * measuring tension}$ ".

Example 3. $y = \varphi(\text{quilted jacket})$; $y = \text{"warm clothing *working clothing* wadded jacket *jacket without the collar"}$.

Example 4. " $y_1 + y_2 * \text{blue} = \varphi(\text{good weather})$ ". Let us define this expression so that it would not contradict with the observed true situation: "*the sun + the sky * blue*".

In resolving equations, the passages from the expressions to their interpretations and from some expressions to the others through the known dependences between them are performed.

For example, assume that the expression "*birthday in the piggery*" is given. It is necessary to find another equivalent expression, which consists of one word (this example is taken from a crosswords).

The concept "*birthday*" defines the region of reasoning or the region of interpretation "*the living beings*" ("*the living beings* \leq *birthday*"). Note that the region of interpretation is expressed by using words, i., e., by using its name. Thus, we pass from the properties to the names of their interpretations. Since the discussion deals with the piggery, then the region of reasoning is "*the living beings born in the piggery*".

The contraction of the region of reasoning occurs by means of the multiplication of properties "*the living beings * born * in the piggery*".

But "*the living beings born in the piggery*" = "*piggy*", thus, we pass to the search for equivalent

expression for $y = \text{"the birthday * piggy"}$. It is now clear that $y = \text{"farrow"}$.

Conclusion

In this chapter, we examined the classification as the system of interconnected structures of objects, properties and classes together with the operations, with the aid of which these structures are built.

We showed that the structural connections between classes and properties of objects, between different classes, and between properties of objects make it possible to build logical expressions on the names of objects, classes and properties, interpreted on the set of all subsets of conceivable objects.

The tasks of machine learning deal with mining the classification connections making it possible to establish equivalence relations between the logical expressions, utilized in the processes of commonsense reasoning.

Bibliography List

[**Boldyrev, 1974**] Boldyrev, N. G. Minimization of Boolean Partial Functions with a Large Number of "Don't Care" Conditions and the Problem of Feature Extraction / N.G. Boldyrev // Discrete Systems. The Proceedings of International Symposium. – Riga, Latvia: Publishing House "Zinatne". P. 101-108.

[**Laurent & Spyrtos, 1988**] Laurent, D., & Spyrtos, N. Partition Semantics for Incomplete Information in Relational Databases / D. Laurent and N. Spiratos // Haran Boral, & Perke Larson (Eds). The Proceedings of ACM – SIGMOD'88. – ACM Press. P. 66-73.

[**Piaget & Inhelder, 1959**] Piaget, J., & Inhelder, B. La Genèse des Structures Logiques Élémentaires Classifications et Sériations / J. Piaget & B. Inhelder // – Neuchâtel: Delachaux & Niestlé.

[**Rasiowa, 1974**] Rasiowa, E. An Algebraic Approach to Non-Classical Logic, Studies in Logic, Vol. 78 / E. Rasiowa. – Amsterdam-London: North-Holland Publishing Company.

[**Naidenova, 1982**] Naidenova, X. A. Relational model for analyzing experimental data / The Transaction of Acad. Sci. of USSR, Series Technical Cybernetics, 4, 103-119.

[**Naidenova, 1992**] Naidenova, X. A. Machine learning as a diagnostic task / I. Arefiev, (ed.), Knowledge-Dialogue-Solution //Materials of the Short-Term Scientific Seminar, (pp.26-36). Saint Petersburg, Russia: State North-West Technical University.

[**Naidenova, 2012**] Naidenova, X.A. Good Classification Tests as Formal Concepts. /F. Domenach, D.I. Ignatov, and J. Poelmans (eds) // LNAI 7278, 211-226.

[**Ore, 1942**] Ore, O. Theory of equivalence relations / Trans. of the AMS, 9, 573-627.

ОБОСНОВАНИЕ МОДЕЛИРОВАНИЯ ПРОЦЕССОВ ПРАВДОПОДОБНЫХ РАССУЖДЕНИЙ

Найденова К. А.

Военно-медицинская академия им. С.М. Кирова
Санкт Петербург, Россия

ksennaidd@gmail.com

В статье рассматривается роль классификационных рассуждений в человеческих правдоподобных рассуждениях. Моделирование этого типа мышления является ключевой проблемой для конструирования интеллектуальных компьютерных систем.

Ключевые слова: классификационные рассуждение; правдоподобные рассуждения; машинное обучение; концептуальное мышление.



OSTIS-2015

(Open Semantic Technologies for Intelligent Systems)

УДК 004.89

ЭВРИСТИЧЕСКИЙ МЕТОД УДОВЛЕТВОРЕНИЯ ОГРАНИЧЕНИЙ НА ОСНОВЕ ИХ МАТРИЧНОГО ПРЕДСТАВЛЕНИЯ

Зуенко А.А. *, Очинская А.А.**

* *Институт информатики и математического моделирования КНЦ РАН, г. Апатиты, Россия*
zuenko@iimm.ru

** *Кольский филиал Петрозаводского государственного университета, г. Апатиты, Россия*
ochinsk@yandex.ru

В статье описан разработанный авторами метод эвристического поиска, предназначенный для решения задач удовлетворения ограничений. Метод опирается на применение матрицеподобных структур алгебры кортежей, которые позволяют представлять и эффективно обрабатывать ограничения с конечными доменами. В отличие от большинства аналогов сложность метода зависит не от количества значений в доменах переменных и суммарного числа переменных в исследуемой системе ограничений, а определяется количеством уравнений и средним числом используемых в них переменных.

Ключевые слова: эвристический поиск, задача удовлетворения ограничений; алгебра кортежей.

Введение

В современном программировании можно выделить несколько основных парадигм: императивное или алгоритмическое программирование, логическое программирование, функциональное программирование и др. Важное место в этом ряду занимает программирование в ограничениях (constraint programming) [Рассел и др., 2006], [Bartak, 1999].

Программирование в ограничениях как самостоятельное научное направление сложилось в конце 60-х – начале 70-х годов прошлого века. Первыми приложениями были задачи обработки изображений и параметрического моделирования пространственно-двумерных сцен. С тех пор направление существенно эволюционировало, охватывая новые классы задач, начиная от решения sudoku и ребусов и заканчивая решением систем линейных уравнений и задач искусственного интеллекта. Например, в [Ченцов, 2013] рассматриваются задачи маршрутизации перемещений с ограничениями в виде условий предшествования, причем допускаются также ограничения других типов, диктуемые особенностями прикладных задач. В частности, описывается задача, связанная со снижением облучаемости при работе в радиационных полях, а также задача листовой резки деталей на станках с числовым программным управлением.

Парадигма удовлетворения ограничений является обобщением методов пропозициональной логики на случай, когда переменные могут принимать не одно из двух фиксированных значений (“истина”/”ложь”), а любые значения из некоторой конечной области определения.

Программирование в ограничениях отличается от традиционного логического программирования, в первую очередь, тем, что решение задачи рассматривается не как доказательство истинности/ложности некоторого утверждения, а сводится к поиску вектора тех значений переменных, которые удовлетворяют условиям задачи. В большинстве случаев при решении задач удовлетворения ограничений (Constraint Satisfaction Problem – CSP) требуется получить не все возможные решения задачи, а найти хотя бы одно из них. Система программирования в ограничениях, как правило, стремится сократить перебор вариантов с целью минимизации отката в случае неуспеха. Для ускорения вычислительных процедур могут быть использованы различные методы, в частности эвристические.

Задачи удовлетворения ограничений относятся к классу задач комбинаторного поиска. Самый общий подход к решению подобных задач основан на обходе дерева поиска. Процессу поиска придается рекурсивный характер, при котором на каждом шаге текущая задача заменяется несколькими подзадачами, рассматриваемыми затем в некотором

порядке, причем менее перспективные из них могут отбрасываться. Таким образом, при обходе дерева поиска переплетаются процессы декомпозиции возникающих ситуаций и их редуцирования. Вполне оправдано стремление к декомпозиции на возможно меньшее число подзадач и к редуцированию, по возможности, более глубокому. Поэтому основные методы решения задач удовлетворения ограничений могут быть разбиты на два класса [Ruttka, 1998]. Первый класс содержит различные варианты алгоритмов поиска в глубину с возвратами, которые строят решение путем расширения частичного присваивания шаг за шагом, используя различные эвристики и применяя разумные стратегии возврата из тупиковых вершин. Ко второму классу относятся алгоритмы распространения ограничений, которые исключают из пространства поиска некоторые элементы, не входящие в решение, обеспечивая снижение размерности задачи.

В данной работе предлагается моделировать ограничения в виде матрицеподобных структур алгебры кортежей (АК) [Кулик и др., 2010], [Зуенко и др., 2011], а также рассматривается разработанный авторами метод вывода на ограничениях, использующий оригинальные эвристики для эффективного поиска решений задачи CSP. В отличие от большинства аналогов применение АК-объектов позволяет обойтись без предварительной бинаризации системы ограничений. По сравнению с ранее предложенными одним из авторов методами вывода в структурах АК [Зуенко, 2013], [Зуенко, 2014] предлагаемый вниманию метод строит дерево поиска, сопоставляя его уровням не переменные (атрибуты), а уравнения системы ограничений.

Далее приведем некоторые сведения из алгебры кортежей в объеме, необходимом для дальнейшего изложения.

1. Представление ограничений в виде структур алгебры кортежей

В [Кулик и др., 2010] приводятся основы АК и демонстрируется ее применение для унификации представления и обработки различных видов данных и знаний, а также решения различных задач логического и логико-вероятностного анализа. Близкий подход применяется также в [Zakrevskij, 2012] для решения задач распознавания образов и упрощения баз знаний.

Конечные предикаты в АК сжато описываются с помощью двух типов структур: C -систем и D -систем. Конкретные экземпляры этих структур называются АК-объектами. Системы ограничений с конечными доменами, обычно, удобно представлять в виде D -систем, а корни системы ограничений (решения) искать в виде C -систем. В АК имеются алгоритмы преобразования D -систем в C -системы и наоборот.

Рассмотрим D -систему A , содержащую n D -кортежей (строк) и m атрибутов (столбцов):

$$A[X_1 X_2 \dots X_m] = \begin{bmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,m} \\ a_{2,1} & a_{2,2} & \dots & a_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n,1} & a_{n,2} & \dots & a_{n,m} \end{bmatrix},$$

где $a_{k,l}$ – компоненты D -системы A , удовлетворяющие ее схеме, среди которых могут встречаться и пустые компоненты.

Известно [Кулик и др., 2010], что D -система эквивалентна пересечению n C -систем, каждая из которых сопоставляется одной из строчек исходной D -системы. Множество C -систем, полученных в результате преобразования каждого D -кортежа из A в C -систему, обозначим как $\{C_i\} = \{C_1, C_2, \dots, C_n\}$. Причем, количество непустых строк в каждой C -системе совпадает с количеством непустых компонент соответствующей строки D -системы, то есть, если $a_{k,l} = \emptyset$, то строка с номером l в C -системе с номером k будет пуста. С учетом вышеизложенного имеем:

$$A[X_1 X_2 \dots X_m] \equiv \bigcap_{i=1}^n C_i, \text{ где}$$

$$C_i[X_1 X_2 \dots X_m] = \begin{bmatrix} a_{i,1} & * & \dots & * \\ * & a_{i,2} & \dots & * \\ \vdots & \vdots & \ddots & \vdots \\ * & * & \dots & a_{i,m} \end{bmatrix}.$$

Заметим, что строки C -систем $\{C_1, C_2, \dots, C_n\}$ содержат по одной нефиктивной компоненте. Другими словами, каждая компонента i -ой строки D -системы A определяет некоторую строку в C -системе $C_i \in \{C_1, C_2, \dots, C_n\}$.

Приведем конкретный пример. Пусть имеется D -система P , заданная в $S = \{a, b, c, d\}^3$:

$$P = \begin{bmatrix} \{a,c\} & \{d\} & \{b,d\} \\ \emptyset & \{a,d\} & \{a,c\} \\ \{b,c\} & \emptyset & \{b\} \end{bmatrix}.$$

Рассмотрим преобразование D -системы в C -систему:

$$P = \begin{bmatrix} \{a,c\} & * & * \\ * & \{d\} & * \\ * & * & \{b,d\} \end{bmatrix} \cap \begin{bmatrix} \emptyset & \emptyset & \emptyset \\ * & \{a,d\} & * \\ * & * & \{a,c\} \end{bmatrix} \cap \begin{bmatrix} \{b,c\} & * & * \\ \emptyset & \emptyset & \emptyset \\ * & * & \{b\} \end{bmatrix}.$$

Вычисляя пересечение первых двух C -систем (пустые C -кортежи здесь и далее для экономии места не показаны), получим:

$$\begin{bmatrix} \{a,c\} & \{a,d\} & * \\ \{a,c\} & * & \{a,c\} \\ * & \{d\} & * \\ * & \{d\} & \{a,c\} \\ * & \{a,d\} & \{b,d\} \end{bmatrix} \begin{matrix} (1,2) \\ (1,3) \\ (2,2). \\ (2,3) \\ (3,2) \end{matrix}$$

Компоненты вектора чисел означают номера перемножаемых строк первой и второй матриц.

Теперь найдем пересечение полученной промежуточной C -системы с третьей C -системой:

$$\begin{bmatrix} \{c\} & \{a,d\} & * \\ \{a,c\} & \{a,d\} & \{b\} \\ \{c\} & * & \{a,c\} \\ \{b,c\} & \{d\} & * \\ * & \{d\} & \{b\} \\ \{b,c\} & \{d\} & \{a,c\} \\ \{b,c\} & \{a,d\} & \{b,d\} \\ * & \{a,d\} & \{b\} \end{bmatrix} \begin{matrix} (1,2,1) \\ (1,2,3) \\ (1,3,1) \\ (2,2,1) \\ (2,2,3) \\ (2,3,1) \\ (3,2,1) \\ (3,2,3) \end{matrix}$$

Компоненты вектора чисел, по-прежнему, означают номера строк перемножаемых матриц (в данном случае перемножаются три матрицы).

Принимая во внимание особенности преобразования D -систем в C -системы, можно заранее по виду исходной D -системы A посчитать количество строк результирующей C -системы в наихудшем случае (когда не появляется пустых строк). Для этого достаточно перемножить количества непустых компонент, находящихся в каждой строке D -системы A . Для нашего примера имеем: $l = 3*2*2=12$.

2. Метод лексикографического перебора

При последовательном перемножении C -систем промежуточная C -система может оказаться слишком большой, и ее трудно зафиксировать в памяти компьютера. Рассмотрим метод, позволяющий не хранить в памяти большие объемы промежуточных результатов. По сути, предлагаемый метод является развитием метода, предложенного в [Закревский, 2003]. Его отличие состоит в том, что в перемножении (пересечении) участвуют не дизъюнктивные нормальные формы (ДНФ) булевых функций, а дизъюнктивные нормальные формы конечных предикатов. К тому же, перемножаемые ДНФ имеют определенный вид, а конечная цель перемножения – преобразование конъюнктивной нормальной формы конечного предиката (D -системы) в его ДНФ (C -систему).

Метод состоит в лексикографическом переборе комбинаций строк, принадлежащих различным пересекаемым C -системам $\{C_1, C_2, \dots, C_n\}$, причем

множество C -систем $\{C_1, C_2, \dots, C_n\}$ заранее упорядочено. Рассмотрим метод более детально.

Ранее при перемножении C -систем каждой строке результирующей C -системы приписывался определенный вектор чисел a , т.е. линейно упорядоченное множество компонент a_i , где $i = 1, 2, \dots, p$, p – количество перемножаемых на данном шаге C -систем. Каждая из компонент принимала значения из некоторого линейно упорядоченного множества A_i – множества индексов непустых строк C -системы C_i .

Теперь пусть имеется произвольная D -система P размерности $n \times m$. Пронумеруем строки (кортежи) и столбцы (атрибуты) этой D -системы. Строке с номером i сопоставим линейно упорядоченное множество A_i – множество номеров столбцов, содержащих непустые компоненты в данной строке. Рассмотрим векторы вида $a = (a_1, a_2, \dots, a_n)$, где $a_i \in A_i$ – номера непустых компонент i -ой строки D -системы. Заметим, что размерность вектора равна количеству кортежей D -системы (количеству перемножаемых C -систем). Каждый такой вектор определяет либо одно из решений системы ограничений, представленной в виде D -системы, либо пустой кортеж. Действительно, если пересечь строки a_1 из C_1 , a_2 из C_2 , ..., a_n из C_n , то получим некоторую строку результирующей C -системы или пустую строку.

Будем считать, что $a < a'$, если существует такое i , что $a_i < a'_i$ и $a_j = a'_j$ для любого j , такого, что $j < i$, причем $i = 1, 2, \dots, n$.

Для упорядоченного таким способом множества всевозможных значений вектора a введем следующее обозначение: $\text{lex}(A_1, A_2, \dots, A_n)$.

На основе лексикографического упорядочения значений вектора a , то есть множества потенциальных решений, и строится дерево поиска.

Лексикографический перебор может быть сокращенным, когда перебираются не все значения вектора a , а лишь часть из них. Однако порядок при этом не нарушается: каждое последующее из рассматриваемых значений вектора a должно быть больше предыдущего. Именно сокращенный перебор и будет нас интересовать, поскольку нет необходимости рассматривать некоторые участки полной лексикографически упорядоченной последовательности значений вектора a , в которых заведомо не содержится решений поставленной задачи.

На очередном шаге рассматривается некоторое частичное значение вектора a , где определены лишь значения компонент a_1, a_2, \dots, a_i , а значения других компонент $a_{i+1}, a_{i+2}, \dots, a_p$ остаются неопределенными. Неопределенность будем отмечать символом «-». Если окажется, что пересечение компонент, соответствующих указанным в векторе a индексам, на данном шаге пусто, то очевидно, что всякое расширение рассматриваемой комбинации путем подбора новых

компонент из оставшихся кортежей D -системы, так же даст пустое пересечение. Поэтому можно не перебирать все возможные варианты таких расширений, а изменить текущее частичное значение вектора a , выбрав следующее возможное значение для компоненты a_i .

Может оказаться, что компонента a_i уже обладает последним из возможных значений. В таком случае следует сократить рассматриваемую частичную комбинацию, выбросив из нее последний элемент (значение i уменьшается на единицу), и выбрать следующее возможное значение для предпоследней компоненты.

Если же пересечение компонент, соответствующих рассматриваемой частичной комбинации, не пусто, то дополняем ее индексом первой непустой компоненты следующего кортежа D -системы, и переходим к рассмотрению полученной новой комбинации.

Очевидно, что решение может быть найдено лишь на том шаге, когда $i=p$.

Рассмотрим, как строится дерево поиска для примера, разобранный в предыдущем подразделе.

Запишем возможные значения вектора a , то есть $\text{lex}(A_1, A_2, A_3)$. Учитывая, что $A_1 = \{1,2,3\}$, $A_2 = \{2,3\}$, $A_3 = \{1,2\}$, имеем: (1, 2, 1), (1, 2, 3), (1, 3, 1), (1, 3, 3), (2, 2, 1), (2, 2, 3), (2, 3, 1), (2, 3, 3), (3, 2, 1), (3, 2, 3), (3, 3, 1), (3, 3, 3).

Сам описанный процесс поиска решений задачи CSP для данного примера может быть представлен с помощью таблицы 1, где знаком (“+”) отмечены решения задачи CSP.

Таблица 1 – Лексикографический перебор решений CSP

значение вектора a	частичное/полное решение задачи CSP
1 --	[{a, c} * *]
1 2 -	[{a, c} {a, d} *]
1 2 1	[{c} {a, d} *]+
1 2 3	[{a, c} {a, d} {b}]+
1 3 -	[{a, c} * {a, c}]
1 3 1	[{c} * {a, c}]+
1 3 3	\emptyset
2 --	[* {d} *]
2 2 -	[* {d} *]
2 2 1	[{b, c} {d} *]+
2 2 3	[* {d} {b}]+
2 3 -	[* {d} {a, c}]
2 3 1	[{b, c} {d} {a, c}]+
2 3 3	\emptyset
3 --	[* * {b, d}]
3 2 -	[* {a, d} {b, d}]
3 2 1	[{b, c} {a, d} {b, d}]+
3 2 3	[* {a, d} {b}]+
3 3 -	\emptyset

Из строк таблицы, помеченных знаком “+”, получаем ту же итоговую C -систему, что и в предыдущем подпункте.

3. Динамическое формирование дерева поиска

Часто нет необходимости искать все возможные решения поставленной задачи, а достаточно найти лишь одно из них. В этом случае можно воспользоваться описанным ранее методом с учетом того, что перебор будет прекращен при первом найденном решении. Однако для поиска одного корня D -системы метод лексикографического перебора недостаточно эффективен. Дело в том, что в рассмотренном методе дерево поиска формируется статически: каждому уровню дерева заранее приписывается некоторая строка D -системы (уравнение системы ограничений) и заранее установлен порядок просмотра компонент выбранной строки D -системы. Другими словами, метод лексикографического перебора относится к классу методов “слепого” поиска и обладает всеми недостатками методов данного класса.

Как уже упоминалось во введении, предлагаемый в работе метод строит дерево поиска, сопоставляя его уровням не переменные (атрибуты), а уравнения системы ограничений. Каждый конкретный узел в пределах уровня соответствует некоторой непустой компоненте выбранной строки.

В настоящей статье предлагается для выбора приемника текущего узла дерева поиска использовать эвристики. Это позволяет отложить исследование менее перспективных ветвей дерева поиска, отдав предпочтение более перспективным с точки зрения получения скорейшего решения. Сам перебор при этом престаёт быть лексикографическим, так как каждое последующее рассматриваемое значений вектора a может как угодно отличаться от предыдущего.

Текущее состояние решаемой задачи CSP полностью характеризуется частичным решением, записанным в виде набора пар “атрибут – усеченный домен атрибута”, а также остатком D -системы, получаемым в ходе процедуры распространения ограничений на основе правил редукции, подробно представленных в работе [Зуенко, 2014].

Здесь, ввиду требований к объему работы, правила редукции не приводятся, а основное внимание уделяется эвристикам, применяемым в процессе поиска. Одни эвристики служат для выбора строки D -системы, другие для выбора компоненты строки, включаемой в решение.

Э1. Выбираем строку D -системы с наименьшим числом компонент.

Э2. В случае неоднозначности выбора, производимого согласно Э1, выбираем уравнение (строку), которое содержит наименьшее число корней. Причем, подсчет числа корней выполняется в предположении, что количество переменных уравнения равно числу непустых компонент соответствующей строки D -системы, а не общему числу атрибутов D -системы.

Э3. Выбираем ту компоненту, которая при предварительной проверке наименее ограничивает диапазон возможных значений для других атрибутов.

Э4. В случае неоднозначности выбора, производимого на основе Э3, находим компоненту, которая позволяет элиминировать из D -системы наибольшее количество строк.

Так как значения вектора a при данном подходе не будут лексикографически упорядочены, то требуется дополнительная переменная для хранения уже рассмотренных значений вектора a . Это позволяет в случае обнаружения тупиковой ветви дерева поиска совершать возврат без повторного рассмотрения значений вектора a .

Рассмотрим особенности применения этого метода на примере. Пусть имеется D -система, заданная в $S = X_1 \times X_2 \times X_3 \times X_4 = \{1, 2, 3, 4\}^4$:

1	{2,3}	∅	{1,2,3}	∅
2	{2}	{1,2,5}	{4}	∅
3	∅	{1,2,4}	{3,4}	{3,5}
4	{1,2}	{3,4,5}	∅	{3}
5	∅	∅	{1,5}	{4}
6	∅	{2,3}	{1,2,3}	{1}
7	{2,4,5}	∅	{3,5}	{3,5}
8	{5}	∅	{2,3,4}	{1,3,4}
9	{1,3}	∅	{3,4,5}	∅

Вспользуемся эвристикой Э1. Строки с номерами 1, 5, 9 имеют по две непустых компоненты. Оценим число корней уравнений, задаваемых этими строками, как того требует Э2. Строка №1 может быть представлена как следующая ортогональная C -система в подпространстве $S_1 = X_1 \times X_3 = \{1, 2, 3, 4\}^2$:

$$\begin{bmatrix} \{2,3\} & * \\ \{1,4\} & \{1,2,3\} \end{bmatrix} = \begin{bmatrix} \{2,3\} & \{1,2,3,4\} \\ \{1,4\} & \{1,2,3\} \end{bmatrix}.$$

Для данной C -системы число корней подсчитывается очевидным образом:

$$k_1 = 2*4 + 2*3 = 14.$$

Для пятой и девятой строк исходной D -системы имеем, соответственно: $k_5 = 10$; $k_9 = 14$. Значит, выбираем строку №5 и начинаем просматривать ее компоненты. Сначала берем компоненту №3, полагая ее новым доменом атрибута X_3 . Но при таком выборе нельзя, воспользовавшись правилами редукции из работы [Зуенко, 2014], сузить области возможных значений ни для одного атрибута, отличного от третьего. Аналогично, обстоят дела и при проверке компоненты №4. Следовательно, применение эвристики Э3 не позволяет осуществить выбор компоненты.

Теперь применим Э4. При выборе компоненты №4 из D -системы, помимо самой выбранной строки

№5, уходит строка №8. Действительно четвертая компонента восьмой строки поглощает четвертую компоненту пятой строки: $\{4\} \subseteq \{1,3,4\}$. Учитывая то, что компонента $\{4\}$ становится новым доменом атрибута X_4 , то четвертые компоненты обеих этих строк становятся полными, а строки D -системы, содержащие полные компоненты могут быть удалены [Кулик и др., 2010]. Итак, мы описали один шаг поиска, остальные шаги выполняются по аналогии.

Процесс поиска решения отображен в таблице 2. Каждый шаг поиска представлен отдельной строкой, где записаны два вектора: вектор a и вектор частичного решения задачи CSP. В отличие от таблицы 1, теперь на каждом шаге поиска для вектора a может быть конкретизирована более чем одна позиция. Как и ранее, позиции вектора a соответствуют строкам исходной D -системы, но значения, находящиеся в этих позициях, теперь интерпретируются как номера компонент, которые становятся полными на текущем шаге поиска в результате осуществления выбора.

Таблица 2 – Перебор уравнений (ограничений) с использованием эвристик

значение вектора a	частичное/полное решение задачи CSP
---- 4 ---4-	[* * * {4}]
3 ---43-4-	[* * {1, 2, 3} {4}]
3-3-4334 3	[* * {3} {4}]
31314334 3	[{2} * {3} {4}]

В каждой строке таблицы 2 в точности одна позиция вектора a выделена жирным шрифтом. В этой позиции содержится номер выбранной на текущем шаге поиска компоненты D -системы. Сама позиция указывает на строку, где эта компонента находится.

Заключение

Предложены оригинальные эвристики, отличающиеся от принятых в теории удовлетворения ограничений. Оказалось, что динамически управляя выбором строк и компонент D -системы на каждом шаге поиска, то есть формируя дерево поиска на основе анализа текущего состояния задачи, можно существенно снизить вычислительную сложность процедуры поиска решений задачи CSP, сводя к минимуму вероятность выполнения возвратов.

Работа выполнена при поддержке РФФИ (проекты № 13-07-00318-а, № 14-07-00205-а, 14-07-00256-а, 14-07-00257-а), Президиума РАН (проект 4.3 Программы № 15), ОНИТ РАН (проект 2.3 в рамках текущей Программы фундаментальных научных исследований).

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

[Закревский, 2003] Закревский, А. Д. Логические уравнения / А. Д. Закревский //– 2-е изд., стереотип. – М. : Эдиториал УРСС, 2003. – 96 с.

[Зуенко и др., 2011] Зуенко, А.А. Интеграция баз данных и знаний интеллектуальных систем на основе алгебраического подхода / А.А. Зуенко, Б.А. Кулик, А.Я. Фридман // Открытые семантические технологии проектирования интеллектуальных систем = Open Semantic Technologies for Intelligent Systems (OSTIS-2011): материалы Международ. научн.-техн. конф. (Минск, 10-12 февраля 2011 г.) – Минск: БГУИР, 2011. – С.59-70.

[Зуенко, 2013] Зуенко, А.А. Матрицеподобные вычисления в задачах удовлетворения ограничений / А.А. Зуенко // Шестая Всероссийская мультиконференция по проблемам управления, 30 сентября – 5 октября 2013 г., г. Геленджик, с. Дивно-морское: материалы мультиконференции в 4 т. – Ростов-на-Дону: Изд-во Южного федерального университета, 2013. -Т.1. – С.30-34.

[Зуенко, 2014] Зуенко, А.А. Вывод на ограничения с применением матричного представления конечных предикатов / А.А. Зуенко // Искусственный интеллект и принятие решений, 2014. – Вып. 3. – С.21-31.

[Кулик и др., 2010] Кулик, Б.А. Алгебраический подход к интеллектуальной обработке данных и знаний / Б.А. Кулик, А.А. Зуенко, А.Я. Фридман. – СПб.: Изд-во Политехн. ун-та, 2010. – 235 с.

[Рассел и др., 2006] Рассел, С. Искусственный интеллект: современный подход. 2-е изд. / С. Рассел, П. Норвиг // пер. с англ.; ред. К.А. Птицына. – М.: Изд. дом «Вильямс», 2006. -1408 с.

[Ченцов, 2013] Ченцов, А.Г. Экстремальные задачи маршрутизации: теория и приложения /А.Г. Ченцов // Шестая Всероссийская мультиконференция по проблемам управления, г. Геленджик, с. Дивноморское, 30 сентября – 5 октября 2013 г.: материалы мультиконференции: в 4 т. – Ростов-на-Дону: Изд-во Южного федерального университета, 2013. -Т.3. – С.213-220.

[Bartak, 1999] Bartak R. Constraint Programming: In Pursuit of the Holy Grail // Proceedings of the Week of Doctoral Students (WDS99), Part IV. – Prague: MatFyzPress, 1999. P. 555–564.

[Ruttkey, 1998] Ruttkey Zs. Constraint satisfaction a survey // CWI Quarterly.1998. V. 11. P. 163–214.

[Zakrevskij, 2012] Arkadij Zakrevskij. Integrated Model of Inductive-Deductive Inference Based on Finite Predicates and Implicative Regularities. In: “Diagnostic Test Approaches to Machine Learning and Commonsense Reasoning Systems”, IGI Global, P 1-12.

HEURISTIC METHOD OF CONSTRAINT SATISFACTION BASED ON MATRIX REPRESENTATION OF CONSTRAINTS

Zuenko A.A. *, Ochinskaya A.A. **

* *Institute for Informatics and Mathematical Modelling, Kola Science Centre of RAS, Apatity, Russia*

zuenko@iimm.ru

** *Kola Branch of Petrozavodsk State University, Apatity, Russia*

ochinsk@yandex.ru

This paper describes a heuristic search method developed by the authors for solving of constraint satisfaction problems. The method relies on the using of matrix structures of n-tuple algebra (NTA). The structures allow to represent and to effectively handle constraints with finite domains. The computational complexity of the method is determined by the number of logical equations and the average number of variables in equations.

Introduction

There are several major paradigms in modern programming: the algorithmic programming, logic programming, functional programming, and etc. The constraint programming takes an important place in this series.

In this paper we propose to model constraints in the form of matrix structures of n-tuple algebra: *C*-systems and *D*-systems. A method developed by the authors for constraint inference uses the original heuristics for effective search of solutions of the constraint satisfaction problem.

Unlike most analogues using of NTA-objects provide an opportunity to solve constraint satisfaction problem without reducing constraint set into the set of binary relations. Compared with the constraint inference method that previously proposed by the one of the author, the new method builds a search tree by matching its levels not with variables (attributes), but with the equation of the system constraints.

Main Part

The state of the constraint satisfaction problem is completely characterized by a partial solution, written as a set of pairs of "attribute – domain of attribute ", as well as the remainder of *D*-systems, received during the constraint propagation procedure.

In this article we propose to use heuristics for choosing of successor the current node of the search tree. It allows to defer the analysis of less promising branches of the search tree, giving preference more promising ones to get a rapid decision. Some heuristics are used to select the line of *D*-system, and others are designed to select components of this line that will be included into the decision.

The process of finding solutions is displayed in a special table. Each step of the search is presented as a separate line with two vectors: vector and the vector of the partial solution of the problem.

Conclusion

Our heuristics are original and differ from heuristics of constraint satisfaction theory. It is clear that we can significantly reduce computational complexity of search procedure by dynamic building of search tree basing on the analysis of the current state of a constraint satisfaction problem. As a result, the probability of backtracking during the search is minimized.



OSTIS-2015

(Open Semantic Technologies for Intelligent Systems)

УДК 519.711.74

СЕТЕВАЯ МОДЕЛЬ РАСПРОСТРАНЕНИЯ АКТИВНОСТИ В СРЕДЕ ГЕТЕРОГЕННЫХ АВТОМАТОВ

Жилякова Л.Ю.

*Институт проблем управления им. В.А. Трапезникова РАН,
г. Москва, Россия*

zhilyakova.ludmila@gmail.com

В работе описываются принципы распространения нескольких видов активности в сети, вершины которой представлены автоматами. Автоматы неоднородны – они имеют различную внутреннюю структуру и отличаются множеством состояний. Задана формальная модель с m видами активности и $2^m - 1$ классами автоматов, которые ее реализуют. Построена модель пороговых взаимодействий в социальной сети с двумя видами активности и пятью типами агентов с разными порогами активации.

Ключевые слова: ресурсная сеть, сеть автоматов, социальная сеть, активность в сетях.

Введение

Работа посвящена формальному описанию пороговой сетевой модели, в которой распространяется несколько видов целочисленного ресурса (фишек). Вершины сети представлены автоматами, имеющими различное количество специализированных слотов, каждый из которых способен хранить один тип ресурса. В особых случаях, описанных отдельно, некоторые слоты могут принимать и генерировать разные типы ресурса в зависимости от типа автомата и от его текущего состояния.

Предлагаемая модель является разновидностью моделей рассеяния на графах. Это большой класс моделей, описывающих распространение активности в разнообразных предметных областях. Описанию математического аппарата и обзорам приложений этих моделей посвящено большое количество работ (см., например, [Blanchard et al., 2011] и [Lovasz et al., 1995]).

В более узком смысле, модель можно назвать модификацией ресурсной сети, предложенной О.П. Кузнецовым и описанной в [Кузнецов и др., 2010], [Жилякова, 2013а-с] и др. Однако у нее имеется ряд основополагающих отличий от ресурсной сети. Первое из них – целочисленность. В этом смысле модель наследует свойства целочисленной пороговой игры выстреливания фишек (chip-firing game), описанной в [Biggs, 1999a, 1999b], [Bjorner et al., 1991, 1992], а также в работах Л. Ловаса, П. Уинклера и других исследователей.

Таковыми пороговыми моделями, в частности, описываются явления самоорганизующейся критичности «лавина» или «абелева куча» [Bak et al., 1988] [Bak, 1996] [Dhar, 1999], [Speer, 1993] и др.

С другой стороны, в приведенных выше моделях рассматривается распространение однородного ресурса (делимого или неделимого). В них состояние каждой вершины сети определяется только количеством ресурса в ней на данном такте времени. В модели, предлагаемой нами, вершины могут содержать несколько видов ресурса и при этом находиться в ряде состояний. В зависимости от состояния вершина может по-разному реагировать на приход ресурсов разного вида. Такие модели наиболее близки к генным сетям [Ананько и др. 1999]. Генная сеть задается ориентированным помеченным графом, вершины которого трактуются как некие биологические элементы, имеющие разную степень концентрации, изменяющуюся во времени. При этом сеть может быть разнородной в том смысле, что различные ее фрагменты, называемые регуляторными контурами, могут функционировать в соответствии с различными правилами. Процессы в генных сетях могут быть описаны не только с помощью аппарата дифференциальных уравнений, но и дискретными автоматными моделями [Евдокимов и др., 2011]. Вершинам сети сопоставляются некоторые функции, которые описывают изменение концентрации во времени. Функционирование регуляторного контура генной сети определяется изменением показателей концентрации в вершинах в зависимости от структуры всей сети и свойств этих функций. Таким образом, динамику сети

можно рассматривать как последовательность состояний, переходы между которыми определяются ее структурой. В [Евдокимов и др., 2011] для алгоритмического исследования дискретных моделей генных сетей, в том числе для поиска неподвижных точек автоматных отображений «нерегулярной природы», используются логические (булевы) уравнения.

В настоящей работе предлагаются целочисленные мультиресурсные сети автоматов, не сводимые к булевым уравнениям. Каждой вершине сопоставлен набор слотов и набор функций, в соответствии с которыми изменяется количество того или иного ресурса. Кроме того, на типах ресурсов заданы отношения, в зависимости от которых слоты могут увеличивать или уменьшать (и в пределе прекращать) активность.

1. Описание автоматной модели

1.1. Основные определения

Сеть задается ориентированным графом $G = (V, E)$, $|V| = n$.

Кроме того, задано множество типов (цветов) фишек $C = \{\circ, \bullet, \ominus, \dots, \omin�\}$. $|C| = m$. Будем называть эти цвета 1, 2, ... m . Тогда $C = \{1, 2, \dots, m\}$.

Пусть вершины сети $i \in V$ – автоматы, содержащие некоторое количество слотов для хранения фишек. Слот типа k может хранить фишки типа k , $k \in [1, \dots, m]$.

Вершины сети представлены на рисунке 1.

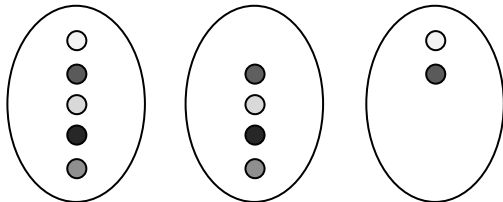


Рисунок 1 – Автоматы с разными наборами слотов

Слоты в вершине на каждом такте времени t могут иметь фишки своего типа или оставаться пустыми.

Однако пустые слоты потенциально способны иметь фишки. Если вершина не имеет слота для данного цвета, то эти фишки она принять не может.

Количества фишек в каждом слоте задают состояние вершины на такте t . Состояние вершины – упорядоченный вектор длины m , в котором для каждого типа с имеющимся слотом указывается количество фишек (если фишек нет, то 0); для несуществующих у данной вершины типов – значения типа null.

Количество фишек типа c_i в вершине i – соответственно $q_{ii}(t)$; состояние вершины – вектор $Q_i(t)$ длины m .

Состояние сети – матрица $(Q(t))_{n \times m}$.

Вершины обмениваются фишками в дискретном времени t по ребрам, имеющим различные пропускные способности.

Слоты могут иметь разную глубину. Глубина слота – это порог активности вершины. Если слот имеет нулевую глубину, вершина становится активной, получив единственную фишку соответствующего цвета.

Пропускные способности – целочисленные переменные величины, изменяющиеся в дискретном времени. Каждое ребро имеет m пропускных способностей: по каждому типу они могут различаться. Если ребро не может проводить фишки определенного типа, пропускная способность его по этому типу равна нулю.

Фрагмент сети представлен на рисунке 2.

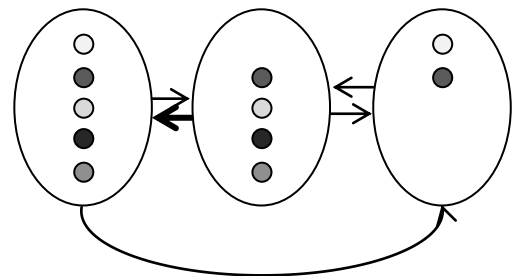


Рисунок 2 – Фрагмент сети с ребрами различных типов («желтое ребро» обозначено желтой фишкой на нем; черные ребра – нейтральны, они могут проводить фишки всех типов)

1.2. Отношения между типами фишек

Между типами фишек существуют разные отношения, поэтому ребра могут существовать не только между вершинами, имеющими одинаковые слоты. Т.е. сеть не распадается на m одноцветных сетей, наложенных друг на друга.

1. *Отношение эквивалентности:* $\bullet = \bullet$. Каждый тип эквивалентен сам себе. Вершины с одинаковыми слотами могут обмениваться фишками одинакового типа, если они смежны.

2. *Отношение заменимости:* $\bullet \rightarrow \circ$ ($l \rightarrow r$). Если в вершину v_i , связанную с вершиной v_j ребром e_{ij} , на такте t пришли фишки типа l , а в вершине j нет слота l , но имеется слот r , то она получит ресурс типа l , «чужой» для нее с некоторой вероятностью p_{lrij} , зависящей от типов l и r и от пропускной способности ребра. Если в вершине более одного подходящего слота, возможно несколько вариантов поведения, реализуемых в разных моделях:

- каждый слот получит фишки со своей вероятностью (может быть, все, а может быть, ни один);
- между слотами есть конкуренция, и победитель забирает всё.

Фишки, попадая не в свой слот, изменяют цвет на цвет слота.

3. *Отношение склеивания:* $\bullet + \circ = \bullet$ ($r + l = k$). Если в вершине нет слотов типа r и l , но есть слот

типа k , то получив по фишке типов r и l , она положит в свой слот фишку k .

4. *Отношение расщепления:* $\circ + \bullet = \bullet + \circ$ ($r + l = k + s$).

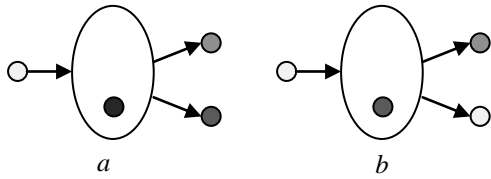


Рисунок 3 – Два типа отношения расщепления

Если в слоте l есть фишка, и к нему приходит фишка r , то фишка l исчезает, а на выход передаются фишки k и s (рисунок 3a). При этом возможны случаи, когда $k = l$ (т.е. фишка на входе совпадает с одной из фишек на выходе) (рисунок 3b). «Сложение» слева от знака равенства некоммутативно. Так, если выполняется $r + l = k + s$, равенство $l + r = k + s$ может не выполняться, так как оно соответствует конфигурации, представленной, на рисунке 4, которая не эквивалентна конфигурации на рисунке 3a.

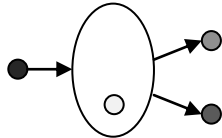


Рисунок 4 – Двойственное расщепление

5. *Отношение катализа:* в вершину приходят фишки типа r , и это запускает внутреннюю реакцию, в результате которой возникают фишки в слоте l .

Блокирующие и отрицательные отношения

6. *Отношение блокировки:* $\bullet \geq \circ$ ($r \geq l$). Не все типы сравнимы по этому отношению (как и по отношению заменимости), но если один тип превосходит другой, то вершина, имеющая фишки большего типа выше порогового значения (активная по большему типу), не может активизироваться по меньшему типу, даже если смежные с ней вершины активны по меньшему типу. Это отношение является частичным порядком.

7. *Отношение вытеснения:* $\bullet \lceil \circ$ ($r \lceil k$). Если вершина v_i , имеющая слоты типов r и k , на такте t получила фишки типа r , на такте $t+1$ из слота k исчезнут все фишки, которые там были.

Это отношение может быть симметричным и несимметричным.

2. Функционирование сети

2.1. Генерирование фишек

Некоторые вершины сети способны генерировать фишки определенного типа (рисунок 5).

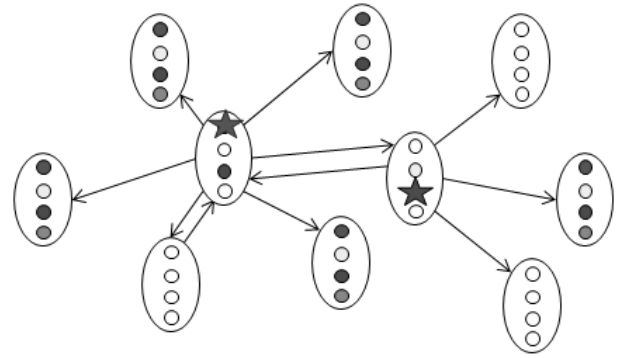


Рисунок 5 – Сеть с вершинами-источниками

В сети автоматов генерация фишек определенного цвета происходит в выделенных вершинах не на каждом такте. Генерация ресурса происходит стохастически – с некоторой вероятностью p .

2.2. Распространение фишек

Если количество фишек данного типа в вершине v_i в момент t не превосходит глубину слота, то на данном такте вершина по этому типу не активна. Если в вершине на такте t количество фишек определенного типа превысило пороговое значение, она активизируется по этому типу.

Активность в сети может быть *постоянной* и *переменной*.

Постоянная активность – активность такого рода, когда тип фишек остается неизменным, то есть фишки не вступают в отношения заменимости, склеивания, расщепления и катализа. Постоянная активность, в свою очередь, может быть чистой, смешанной (по нескольким не конкурирующим типам).

Переменная активность – это активность, в которой хотя бы раз произошла смена цвета фишек.

Модель параллельна. Все вершины, способные выстрелить в момент времени t , выстреливают.

Глубина слота для каждого типа фишек задается петлей того же типа.

По способу распространения существует два типа фишек: тиражируемые и нетиражируемые фишки.

Тиражируемые фишки сверх порогового значения передаются по исходящим ребрам по следующему правилу. В каждое исходящее ребро уходит $\min\{q_{ij}(t) - r_{iil}, r_{ijl}\}$, где $q_{ij}(t)$ – количество фишек типа l на такте t , r_{iil} – пропускная способность петли типа l (т.е. пороговое значение или глубина слота), r_{ijl} – пропускная способность ребра типа l , соединяющего вершину v_i с вершиной v_j . В вершине не остается фишек; r_{iil} фишек возвращаются в нее назад по петле, и вершина находится в пред-активном состоянии по этому типу.

Для *не тиражируемых фишек* выполняется закон сохранения. Они отдаются в ребра в

очередности, соответствующей пропускным способностям: в каждое ребро поочередно отдается по полной пропускной способности, начиная с наибольшей. Процесс завершается, когда фишки сверх порогового значения заканчиваются. Очередность может изменяться для конкретных моделей.

Также в зависимости от модели в сети могут быть только тиражируемые или только не тиражируемые фишки.

3. Модель социальной сети с разными типами агентов

Пусть $V = \{v_1, v_2, \dots, v_n\}$ – множество агентов, входящих в социальную сеть, описываемую ориентированным графом $G = (V, E)$, где $E \subseteq V \times V$ – множество взвешенных ребер. Агенты в сети влияют друг на друга – наличие дуги e_{ij} от вершины v_i к вершине v_j соответствует влиянию i -го агента на j -го, а ее вес $r_{ij} \in \mathbb{N}$ обозначает степень влияния.

Вершины сети неоднородны: каждая вершина представляет собой автомат, который может находиться в счетном множестве состояний. На каждом шаге дискретного времени t вершина выбирает одно из трех действий: два вида активности и бездействие. Активности считаются антагонистическими по отношению друг к другу.

В сети передаются фишки двух типов. Третий тип фишек отвечает за внутреннее состояние вершин. Будем называть эти типы 1, 2 и 3. Типы 1 и 2 – два вида противоположной активности (революционная/реакционная, созидательная/разрушительная и т.п.). Тип 3 – «внутренние фишки», которые являются показателем степени когнитивного консонанса/диссонанса вершины, то есть того, насколько активность агента, представленного вершиной, соответствует его внутренним принципам. Фишки типа 3 – виртуальные внутренние фишки. Их количество может быть не только положительным, но и отрицательным, а также нецелым. Чем больше фишек этого типа имеет вершина, тем в большем согласии с собой находится агент. Наличие когнитивного диссонанса выражается в отрицательных значениях у вершины фишек типа 3.

3.1. Внутренняя структура агентов и их состояния

Агенты в сети представлены автоматными вершинами, имеющими три слота: по одному для каждого типа фишек. Состояние вершины – количество фишек каждого типа. Вершины могут обмениваться фишками только первых двух типов.

Будем подразумевать, что каждый такт времени в сети состоит из двух полутактов. На первом полутакте вершины принимают фишки из других вершин; в конце первого полутакта вершина изменяет свое состояние (рассчитывает величину когнитивного диссонанса). На втором полутакте в

зависимости от этого состояния вершина либо молчит, либо активируется по заданному типу.

Обозначим через $c_{ji}(t)$ количество фишек типа i , $i = 1, 2, 3$, в вершине v_j на первом полутакте t . Это количество соответствует состоянию, в котором вершина уже получила фишки типов 1 и 2, вышедшие из смежных вершин на предыдущем такте, и пересчитала количество фишек типа 3, но еще не выполняла никаких действий на этом такте.

Пусть $c_{ji}^{in}(t)$ – количество фишек типа i , пришедших в вершину v_j , $c_{ji}^{out}(t)$ – количество фишек типа i , вышедших из вершины v_j на такте t , $i = 1, 2$. Тогда:

$$c_{j1}(t) = c_{j1}(t-1) - c_{j1}^{out}(t-1) + c_{j1}^{in}(t);$$

$$c_{j2}(t) = c_{j2}(t-1) - c_{j2}^{out}(t-1) + c_{j2}^{in}(t).$$

Когнитивный диссонанс рассчитывается по формуле:

$$c_{j3}(t) = c_{j3}(t-1) - [|x_{j1}|c_{j1}^{in}(t) + |x_{j2}|c_{j2}^{in}(t)], \quad (1)$$

где $x_{j1}, x_{j2} \in [-1, 1]$.

$c_{j3}(0) = c_{j3}^* > 0$ – показатель начального «довольства жизнью».

Величины x_{j1}, x_{j2} – индивидуальные параметры агента, по которым он оценивает два вида активности.

Своей активностью агента v_j на такте t будем называть ту из них, для которой выполняется:

$$i_j(t) = \arg \max \{x_{j1}c_{j1}^{in}(t), x_{j2}c_{j2}^{in}(t)\}$$

Если x_{j1}, x_{j2} имеют разные знаки, свой тип фишек остается неизменным:

$$i_j(t) = i_j = \arg \max \{x_{j1}, x_{j2}\}$$

Если для агента x_{j1}, x_{j2} одного знака, он может поддаться «очарованию толпы» и последовать за большинством.

3.2. Порог активности

Вершины становятся активными по типу 1 или по типу 2, если на такте t величина $c_{j3}(t)$ достигнет сверху порогового значения $T_{ja} < 0$.

$$T_{ja} = -k_j(1 - |x_{j1}|)(1 - |x_{j2}|) \quad (2)$$

где $k_j > 0$ – некоторый коэффициент пропорциональности.

При этом считается, что вершина активизируется в конкретный момент только по *своему* типу, и фишки противоположного типа изменяют свой цвет.

Если вершина на данном такте не достигла порога активации, она складывает пришедшие фишки в слотах соответствующих цветов.

Если вершина активна на такте t , все ее фишки сверх порогового значения пропадают. Величина $c_{j3}(t)$ принимает значение по умолчанию:

$$c_{j3}(t) = c_{j3}(0) = c_{j3}^* > 0.$$

Если вершина не получает фишек на такте t , ее слоты 1 и 2 теряют по одной фишке данного типа. Так происходит постепенное забывание и угасание активности.

3.3. Типы агентов

Любая вершина задается четверкой: $v_j = \{x_{j1}, x_{j2}, c_{j3}^*, k_j\}$, а ее состояние определяется набором $c_i(t)$, $i = 1, 2, 3$. По значениям параметров $x_{j1}, x_{j2}, c_{j3}^*, k_j$ каждая вершина представляет агента одного из попарно непересекающихся классов:

$$V = V_{R1} \cup V_{R2} \cup V_A \cup V_C \cup V_P,$$

где V_{R1} – агенты-революционеры;

V_{R2} – агенты-реакционеры;

V_A – рациональные агенты;

V_C – агенты-конформисты;

V_P – пассивные агенты.

Каждый класс характеризуется своим набором состояний и правилами выбора действия.

3.3.1. Революционеры и реакционеры

Количество вершин в классах V_{R1} и V_{R2} относительно мало. Это те пассионарные агенты, которые начинают или поддерживают любую активность «своего цвета».

Эти вершины имеют фишки в слотах только двух типов: 1 и 3 – для революционеров; и 2 и 3 для реакционеров.

Для агентов-революционеров выполняется:

$$x_{j1} = 1, x_{j2} = -1, x_{j4} = 0, c_{j3}^* = 0, k_j = 0.$$

Первый полутакт (прием фишек и расчет диссонанса):

$c_1(t) = c_1^{in}(t) + c_2^{in}(t)$ – фишки типа 2 трансформируются в тип 1;

$$c_3(t) = -[c_1^{in}(t) + c_2^{in}(t)].$$

Второй полутакт (после выстреливания фишек):

$$c_1(t + 1/2) = 0; c_3(t + 1/2) = 0.$$

Из формулы (2) порог активности: $T_{ja}(t) = 0$. Вершина активна, если она обладает хотя бы одной фишкой типа 1.

В каждое исходящее ребро на такте t вершина v_j посылает ресурс, равный $\min(c_{j1}(t), r_j^{out})$.

Считается, что за такое «выстреливание» вершина расходует все фишки типа 1.

Реакционеры действуют аналогично, выполняя при этом действие типа 2.

3.3.2. Рациональные агенты

Рациональные агенты склонны оценивать свои действия. Для каждого из них значения параметров: $x_{j1}, x_{j2}, c_{j3}^*, k_j$ могут варьироваться, задавая разных агентов внутри одного типа.

$x_{j1}, x_{j2} \in [-1, 1]$; $c_{j3}^*, k_j > 0$, где x_{j1}, x_{j2} задают устойчивость агента к окружающей внешней активности, c_{j3}^* – уровень «устойчивости недеяния».

3.3.3. Конформисты

Конформисты также задействуют все типы слотов, однако значения параметров у них фиксированы. Они не отдадут предпочтения виду активности.

$$x_{j1} = x_{j2} = 0.5, c_{j3}^* > 0.$$

$$c_{j3}(t) = c_{j3}(t-1) - 0.5k_j[c_{j1}(t) + c_{j2}(t)].$$

Их готовность к действию отличается постоянными порогами, задаваемыми величиной T_{ja} . Если их диссонанс перешел пороговую величину, они действуют по типу i , такому, что

$$i = \arg \max \{c_{j1}^{in}(t), c_{j2}^{in}(t)\}.$$

3.3.4. Пассивные агенты

Пассивные агенты всегда имеют $T_{ja} = -\infty$, и никакая активность соседних вершин не в состоянии вовлечь эти вершины в активность.

3.4. Функционирование сети

Каждой вершине поставлен в соответствие набор ее характеристик.

Состояние вершины будем называть «преднагруженным» по типу i , ($i = 1, 2$) на такте t , если достаточно хотя бы одной фишки этого типа, чтобы активироваться (по этому типу).

Задается матрица начального состояния $C(0) = (c_{ij}(0))_{n \times 3}$, где первые два столбца $C_1(0), C_2(0)$ – векторы с неотрицательными компонентами. Задается вектор начального возбуждения $C_{1a}(0)$, который активирует некоторые преднагруженные по типу 1 вершины и некоторых агентов-революционеров. Далее по сети распространяются активности одного или двух видов в соответствии с правилами распространения и параметрами вершин.

Заключение

В работе описаны основные организационные признаки сети гетерогенных автоматов со счетным множеством состояний, в которой распространяется несколько видов целочисленного ресурса (фишек). Заданы отношения на ресурсах в общем виде. Рассмотрен пример применения данного аппарата к описанию распространения активности в социальной сети. Функционирование и свойства модели социальной сети, распространение в ней двух видов активности, зависимость распространения и загухания активности от доли вершин того или иного типа, изменение топологии под влиянием этих процессов и многие другие вопросы планируется исследовать в дальнейших работах.

Библиографический список

[Ананько и др., 1999] Ананько Е.А., Колпаков Ф.А., Подколдная О.А., Игнатъева Е.В., Горячковская Т.Н., Степаненко И.Л., Колчанов Н.А. Генные сети. 1999. – [Электронный ресурс] – URL: http://www.bionet.nsc.ru/ICIG/session/1999/rus/part1/1_18.pdf (дата обращения 01.12.2014).

[Базенков и др., 2013] Базенков Н. И., Губанов Д. А. Обзор информационных систем анализа социальных сетей / Управление большими системами. Выпуск 41. М.: ИПУ РАН, 2013. С.357-394.

[Бреер и др., 2014] Бреер В.В., Новиков Д.А., Рогаткин А.Д. Стохастические модели управления толпой // Управление большими системами. 2014. № 52. С. 85 – 117.

[Губанов и др., 2010] Губанов Д.А., Новиков Д.А., Чхартишвили А.Г. Социальные сети: модели информационного влияния, управления и противоборства. – М.: Физматлит. – 2010. – 228 с.

[Жилякова, 2013а] Жилякова Л.Ю. Эргодические циклические ресурсные сети. I. Колебания и равновесные состояния при малых ресурсах / Управление большими системами. Выпуск 43. М.: ИПУ РАН, 2013. С. 34 – 54.

[Жилякова, 2013б] Жилякова Л.Ю. Эргодические циклические ресурсные сети. II. Большие ресурсы / Управление большими системами. Выпуск 45. М.: ИПУ РАН, 2013. С.6-29.

[Жилякова, 2013с] Жилякова Л.Ю. Управление предельными состояниями в поглощающих ресурсных сетях // Проблемы управления, 2013, № 3. С. 51 – 59.

[Евдокимов и др., 2011] Евдокимов А.А., Кочемазов С.Е., Семенов А.А. Применение символьных вычислений к исследованию дискретных моделей некоторых классов генных сетей // Вычислительные технологии. 2011. Т. 16. № 1. С. 30-47.

[Кузнецов и др., 2010] Кузнецов О.П., Жилякова Л.Ю. Двусторонние ресурсные сети – новая потоковая модель // Доклады Академии Наук, 2010, том 433, № 5. – С. 609 – 612.

[Семенов и др., 2013] Семёнов А.А., Кочемазов С.Е. О дискретно-автоматных моделях конформного поведения / Управление большими системами. Выпуск 46. М.: ИПУ РАН, 2013. С.266-292.

[Bak et al., 1988] Bak, P., Tang, C. and Wiesenfeld, K. Self-organized criticality, *Physical Review A* 38. 1988, P. 364 – 374.

[Bak, 1996] Bak, P. *How Nature Works: The Science of Self-Organized Criticality*. New York: Copernicus. 1996. (Рус. пер. Как работает природа: Теория самоорганизованной критичности. – М.: УРСС: Книжный дом «Либроком», 2013. – 276 с.)

[Biggs, 1999a] Biggs, N.L. Chip-Firing and the Critical Group of a Graph // *Journal of Algebraic Combinatorics* 9 (1999), pp. 25–45. Kluwer Academic Publishers. Netherlands. 1999.

[Biggs, 1999b] Biggs, N. The Tutte-polynomial as a growth function // *J. Algebraic Combinatorics* 10. 1999, pp. 115–133.

[Björner et al., 1991] A. Björner, L. Lovasz and P. Shor, Chip-firing games on graphs, *Europ. J. Comb.* 12 (1991), 283–291.

[Björner et al., 1992] A. Björner and L. Lovasz, Chip-firing games on directed graphs, *J. Algebraic Combinatorics* 1 (1992), 305–328.

[Blanchard et al., 2011] Blanchard, Ph., Volchenkov, D. *Random Walks and Diffusions on Graphs and Databases: An Introduction* (Springer Series in Synergetics). Springer-Verlag – Berlin–Heidelberg. 2011.

[Dhar, 1999] Dhar, D. The abelian sandpile and related models // *Physica A: Statistical Mechanics and its Applications*. Volume 263, Issues 1–4, 1 February 1999, pp. 4 – 25.

[Kempe et al., 2003] Kempe D., Kleinberg J., Tardos E. Maximizing the Spread of Influence through a Social Network / Proceedings of the 9-th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2003. P. 137-146.

[Li et al., 2013] Li Y., Chen W., Wang Y., Zhang Z.-L. Influence diffusion dynamics and influence maximization in social networks with friend and for relationships. In. Proc. 6-th ACM Int. Conf. Web Search and Data Mining. P. 657-666. 2013.

[Liu et al., 2012] Liu H., Tang L., Yu P.S. Modeling blogger influence in a community. *Social Network Analysis and Mining*, June 2012, Volume 2, Issue 2, pp 139-162.

[Lovasz et al., 1995] L. Lovasz and P. Winkler. Mixing of Random Walks and Other Diffusions on a Graph // *Surveys in Combinatorics*, 1995 (ed. P. Rowlinson), London Math. Soc. Lecture Notes Series 218, Cambridge Univ. Press, 119–154.

[Speer, 1993] Speer, E. R. Asymmetric abelian sandpile models // *Journal of Statistical Physics*. April 1993, Volume 71, Issue 1-2, pp. 61 – 74.

NETWORK MODEL OF ACTIVITY PROPAGATION AMONG HETEROGENEOUS AUTOMATA

Zhilyakova L. Yu.

*Trapeznikov Institute of Control Sciences, Russian
Academy of Sciences, Moscow, Russia*

zhilyakova.ludmila@gmail.com

In work the main principles of propagation of several different kinds of activity in a network with complex vertices are described. The vertices of a network are heterogeneous automata (finite or infinite) – they have a different internal structure and different set of states. The formal model with m kinds of activity and $2^m - 1$ class of automata is defined and described. A model of threshold interactions in social networks with two types of activity and five types of agents with different activation thresholds is built.

Introduction

The work is devoted to the formal description of the threshold network model, which applies several types of integer resource (chips). Vertices of the network represented automata having a different number of specialized slots, each of which is capable of storing one type of resource. The proposed model is a special kind of diffusion on graphs – a large class of models describing the propagation of activity in a variety of domains. But our model has a number of features that make it original and give it greater expressive power. Some of them are the existence of many kinds of interacting resources and the complex structure of vertices.

Main Part

A network is based on an oriented graph $G = (V, E)$, $|V| = n$. There are m types of integer resources (chips) in the network. Let vertices $i \in V$ be automata, each of them has some number of slots for chips $k \in [1, \dots, m]$. The state of automaton is defined by the quantity of every kind of chips. So, the state of a vertex is described by integer vector of length m . The types of chips are not independent, because of that the heterogeneous network is not a simple combination of several homogeneous ones. There are such kinds of relationships between some types of chips as equivalence, substitutability, gluing, splitting, catalysis, locking, and displacement.

The social network with some of described properties is presented. There are two kinds of opposite activities in this network and one kind of chips responsible for internal state of automata.



УДК 656:[681.5:004]

СИСТЕМА ПРИОРИТЕТНОГО ДВИЖЕНИЯ НА ПЕРЕКРЕСТКАХ «ЗЕЛЕНАЯ ВОЛНА» ДЛЯ ТРАНСПОРТНЫХ СРЕДСТВ ОПЕРАТИВНОГО НАЗНАЧЕНИЯ

Сокоян А.Л., Шуть В.Н.

*Брестский государственный технический университет,
г. Брест, Республика Беларусь*

lexing2008@yandex.ru

lucking@mail.ru

В работе рассматривается система «Зеленая волна», которая позволяет решить проблему безопасного движения транспортных средств оперативного назначения с включенными проблесковыми маячками при движении через регулируемые перекрестки.

Ключевые слова: приоритетное движение через перекресток, транспортные средства оперативного назначения, безопасность дорожного движения.

Введение

Внезапное появление транспортного средства (ТС) оперативного назначения с включенными проблесковыми маячками может создать опасные ситуации для близлежащих транспортных средств, заставляя водителей маневрировать и экстренно реагировать для освобождения пути.

Некоторые водители волнуются и создают конфликтные ситуации, которые могут привести к аварии с участием транспортного средства оперативного назначения, блокированию полосы движения, увеличению времени прибытия оперативного транспорта к месту назначения.

Использование системы «Зеленая волна» позволяет обеспечить транспортным средствам оперативного назначения беспрепятственный проезд перекрестков, сокращая путаницу для водителей других автомобилей, улучшая время реагирования на чрезвычайные ситуации, повышая эффективность оперативной службы, устраняя конфликтные ситуации на перекрестках.

Несмотря на то, что водитель автомобиля с закрытыми окнами и включенной музыкой обычно замечает приближение оперативного транспорта только когда тот находится непосредственно сзади, самый большой процент столкновений происходит на перекрестке, а не при обгоне автомобиля [1]. Статистика показывает, что встречных столкновений очень мало. Это опровергает миф, что выезд на полосу встречного движения является основной опасностью для движения ТС оперативного назначения [1].

1. Существующие подходы

1.1. Системы на основе акустических детекторов

Системы такого типа используют акустический детектор (рисунок 1). Когда транспортное средство оперативного назначения подъезжает на определенном расстоянии к светофорному объекту, детектор улавливает звук сирены и отправляет сигнал на контроллер светофора о необходимости включения зеленой фазы в данном направлении [2]. Такая система может использоваться автономно или в сочетании с другими системами.

Преимуществом является недорогая интеграция в существующие светофоры и отсутствие необходимости оснащать транспортные средства оперативного назначения дополнительными устройствами, поскольку сирены уже установлены.

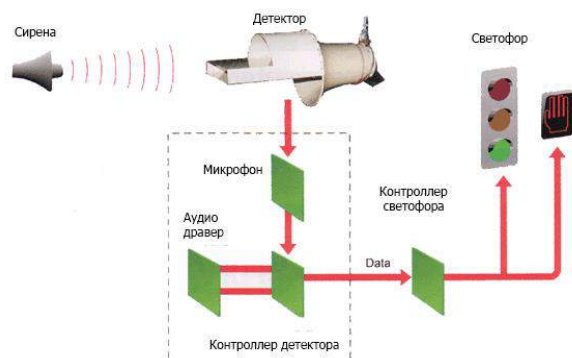


Рисунок 1 – Схема акустического детектора

Основным недостатком этого метода является то, что звуковые волны легко отражаются зданиями или крупными автомобилями, находящимися вблизи светофорного объекта. В результате, звуковая волна может быть распознана акустическим детектором другого направления движения.

Акустические датчики иногда могут быть слишком чувствительными, распознавая сирену от очень далеко едущего автомобиля оперативного назначения слишком рано.

1.2. Системы на основе акустического детектора и видеокамеры

Избежать недостатков отражения волн звука и слишком раннего реагирования акустического детектора позволяет интеграция в систему видеокамеры с возможностью распознавания светового сигнала транспортного средства оперативного назначения (рисунок 2). Примером такой системы является The SONEM 2000 от компании Traffic Systems, LLC.

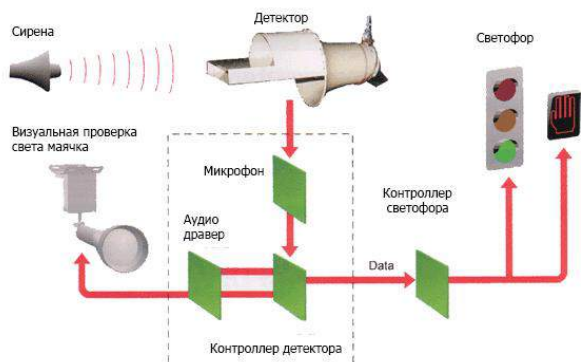


Рисунок 2 – Схема акустического детектора + видеокамера

Недостатком подобных систем является их удорожание, а также появление ситуаций, когда ТС оперативного назначения подъезжает к перекрестку, но не попадает в видимую область камеры, в связи с помехой, погодными условиями и т.д.

1.3. Системы на основе прямой видимости

Транспортные средства оперативного назначения оснащаются излучателями, которые посылают узко-направленный сигнал перед авто в сторону светофора. В качестве сигнала может выступать инфракрасный импульс или видимый свет. Каждый светофор должен быть оборудован приемником сигнала (рисунок 3).

Таким оборудованием могут быть оснащены не только транспортные средства оперативного назначения, но и автобусы, троллейбусы (общественный транспорт) для быстрого проезда перекрестков. Но для того, чтобы не было коллизий, когда к светофору приближается общественный транспорт и транспортное средство оперативного назначения, для каждого типа транспортных средств резервируется своя частота диапазона.

К недостаткам системы можно отнести обязательное требование нахождения излучателя в

транспортном средстве и приемника на светофоре на линии видимости, возможные сбои при изменении атмосферных условий, условий освещения (прямые солнечные лучи могут препятствовать детектированию сигнала).



Рисунок 3 – Приемник сигнала

1.4. Системы на основе глобальной навигации

В транспортные средства оперативного назначения интегрируется навигационный модуль (GPS/ГЛОНАСС или др.) для определения текущих координат расположения и модуль GSM для передачи данных о расположении ТС на сервер управления светофорными объектами.

В момент, когда водитель включает проблесковые маячки, активизируется передача координат навигационного модуля на центральный сервер.

Сервер определяет текущее расположение ТС оперативного назначения и включает зеленую фазу на светофоре по направлению движения ТС.

2. Существующие внедрения

Внедрение системы в городском округе Фэйрфакс (США, штат Вирджиния) позволило уменьшить время отклика на чрезвычайную ситуацию. Система позволила транспортным средствам оперативного назначения первыми пройти через большие загруженные перекрестки с меньшим количеством конфликтных ситуаций. При этом экономилось от 30 до 45 секунд на перекрестке.

Внедрение системы в городе Plano (США, штат Техас) резко сократило количество транспортных происшествий - в среднем с 2,3 аварий на перекрестках в год до менее одной аварии на перекрестках каждые пять лет.

Кроме того, в связи с сокращением задержек на перекрестках, экстренные службы города Plano могут достичь тех же показателей времени реагирования что и раньше, но уже с меньшим количеством пожарных станций и станций скорой медицинской помощи. Это обеспечивает значительную экономию средств бюджета. Город сохранил время отклика свыше 90%, тем самым

достигнут первый класс в рейтинге пожаротушения – наиболее высокий по шкале от 1 до 10, что привело к уменьшению стоимости страхования зданий.

Система установленная в Сент-Пол (США, штат Миннесота) позволила транспортным средствам полиции, пожарным спасателям и скорой медицинской помощи добраться до места аварии быстрее и с уменьшенной вероятностью аварии на перекрестке. В последующие годы после развертывания системы значительно сократились расходы на выезд по вызову ТС оперативного назначения.

В 1978 году, в городе Денвер, департамент безопасности тестировал систему прерывания сигнала светофора для транспортных средств оперативного назначения [3]. Исследование проводилось в течение 90 дней в районе, в котором расположены три пожарных станции и 75 регулируемых перекрестков. Пожарные записывали время в пути, необходимое для прохода типичных маршрутов до и после установки системы. Данные показали, что с внедрением системы время отклика уменьшилось от 14% до 23%, с экономией приблизительно 70 секунд на маршруте протяженностью от трех до шести регулируемых перекрестков.

В США в период с 1996 по 2006 более 25% аварий с участием транспортных средств оперативного назначения происходило на перекрестках [4].

В 1977 году был произведен анализ аварийности пожарных автомобилей в городе Сент-Пол до внедрения системы и после внедрения на 285 из 308 перекрестков. В течение десяти лет наблюдения количество аварий с транспортными средствами пожарной службы снизилось от 8 до 3.3 (в среднем) в год.

3. Описание разработки

Функционирование системы впереди бегущая «зеленая волна» для ТС оперативного назначения основано на том, что каждое транспортное средство экстренного назначения оснащено GPS/GSM терминалом.

При включении проблесковых маячков водителем транспорта оперативного назначения автоматически активируется передача данных GPS о расположении транспортного средства по GSM каналу на сервер организации, осуществляющей управление светофорными объектами (СМЭП ГАИ). Сервер СПЭМ ГАИ вычисляет светофорные объекты впереди движения транспортного средства и подает на светофорные объекты запрещающий сигнал для всех полос и всех направлений движения.

3.1. Что дает система

ДТП с участием транспортных средств служб оперативного назначения имеют следующие последствия:

- гибель людей;
- потери здоровья выживших и необходимость их лечения и реабилитации;
- значительная задержка времени реагирования на экстренный вызов. Транспортное средство не приедет на вызов, вместо него необходимо отправить другое ТС;
- затраты на восстановление (если это возможно) или покупку нового специализированного автотранспортного средства.

Преимущества, которые дает система «Зеленая волна» благодаря блокированию всего перекрестка:

- исключение возможности ДТП при пересечении перекрестка на красный свет;
- уменьшает время реагирования за счет исключения притормаживания ТС оперативного назначения на перекрестках;
- останавливает автомобили во всех направлениях, в том числе попутные. Что дает возможность эффективно маневрировать. Водителю ТС оперативного назначения проще маневрировать в ситуации, когда попутные автомобили не движутся;
- освобождает полосу встречного движения от транспортных средств, что позволяет осуществить выезд на пустую встречную полосу и развить максимальную скорость;
- не подставляет под угрозу жизнь и здоровье сотрудников служб оперативного назначения и других автомобилей.

3.2. Преимущества разработки

В момент подъезда транспортного средства оперативного назначения к светофорному объекту у всех аналогов системы включается зеленая фаза для направления движения транспортного средства оперативного назначения. В предлагаемой системе сигнал на светофорном объекте во всех направлениях и для всех участников дорожного движения становится красным. Тем самым исключаются любые помехи (пешеходы, автомобили, движущиеся попутно и т.д.) для движения оперативного транспорта.

Автономные светофорные объекты, не обладающие центральным управлением с сервера СМЭП, могут быть оборудованы акустическими детекторами. Поскольку при приближении транспортных средств оперативного назначения во всех направлениях должен гореть красный сигнал светофора, отпадает необходимость решать проблему с отражением звуковой волны сирены, и нет необходимости увеличивать стоимость системы, внедряя видеокамеры и модули распознавания цвета сигнала проблескового маячка.

Благодаря такому подходу, сокращается количество акустических детекторов, необходимых для оснащения одного перекрестка.

4. Разработка системы

4.1. Требования к системе

Система должна обеспечивать передачу данных с GPS приемника транспортного средства по GSM каналу на сервер, осуществляющий управление светофорными объектами. На сервере должна находиться информационная база, содержащая глобальные координаты расположения всех светофорных объектов.

В системе можно выделить администратора системы, обладающего доступом к созданию и редактированию информационной базы системы.

Функционирование происходит по следующим этапам:

- администратор сервиса единожды вносит информацию обо всех светофорных объектах города, обо всех транспортных средствах оперативного назначения;
- на всех автотранспортных средствах экстренного реагирования устанавливается GPS/GSM терминал;
- при включении проблесковых маячков автоматически активизируется передача данных с GPS приемника на сервер по GPRS каналу;
- сервер распознает транспортное средство экстренного реагирования и блокирует ближайший один или несколько светофорных объектов (в зависимости от скорости движения и расстояния между светофорами) на пути следования ТС;
- транспортное средство проезжает регулируемый перекресток;
- сервер вводит светофорный объект в штатный режим функционирования.

Передача данных между ТС оперативного назначения и сервером происходит с использованием шифрования.

Разработку системы можно разбить на составляющие:

- создание подсистемы администрирования информационной базой данных;
- создание подсистемы редактирования светофорных объектов на карте (подсистема редактор карты);
- создание подсистемы анализа GPS данных и управления светофорными объектами;
- оснащение ТС оперативного назначения GPS/GPS терминалами.

В рамках всего программного комплекса должны быть реализованы следующие процессы:

- редактирование карты администратором: администратор сервиса имеет доступ к редактору

карты. Редактор карты позволяет наносить на карту светофорные объекты;

- редактирование информационной базы: администратор сервиса добавляет в информационную базу новые светофоры, новые транспортные средства оперативного назначения, может редактировать информацию о светофорах, о транспортных средствах оперативного назначения;
- добавление/редактирование администраторов информационной базы данных;
- анализ полученных данных от ТС, распознавание транспортного средства, расчет впередиидущих светофорных объектов
- блокировка впередиидущих светофорных объектов;
- ввод светофорных объектов в штатный режим функционирования.

Соответственно этим процессам можно выделить несколько требований, предъявляемых к системе:

- администратор системы наделяется своими правами только после прохождения авторизации в панели администратора;
- администратор должен иметь возможность управления пользователями системы;
- администратор должен иметь возможность добавлять, изменять и удалять светофорные объекты, транспортные средства оперативного назначения в информационной базе системы;
- администратор может просматривать список всех светофорных объектов, список всех транспортных средств;
- администратор должен иметь возможность наносить на карту светофорные объекты;
- автоматическая подстройка карты под размер окна браузера при загрузке страницы;
- система должна обеспечивать корректное отображение в современных браузерах: Opera 10.5+, Chrome 5+, Firefox 5.0 +, IE 8+, Safari.

4.2. Разработанная система

Для доступа в панель администратора пользователь проходит авторизацию с использованием адреса электронной почты и пароль. Скриншот страницы авторизации приведен на рисунке 4.

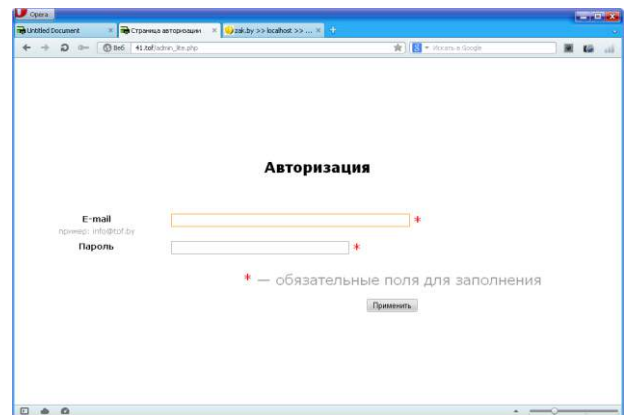


Рисунок 4 – Авторизация в систему

В панели администратора администратор может вносить изменения в информационную базу: добавлять, редактировать, удалять светофорные объекты (рисунки 5).

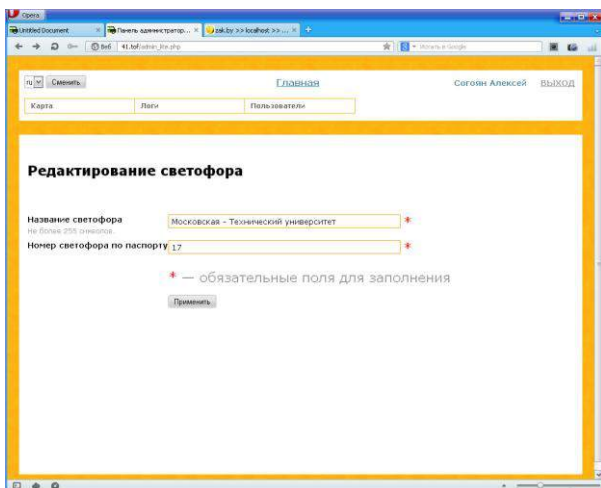


Рисунок 5 – Редактирование светофорного объекта

Администратор в любой момент времени может просмотреть список всех светофорных объектов, которые присутствуют в информационной базе данных системы. Система отображает список светофоров в виде записей (рисунок 6). Каждая запись содержит номер записи, название светофора, номер по паспорту светофорного объекта, ссылку на страницу редактирования, ссылку для удаления.

№	Название светофора	№ паспорта	Изменить	Удалить
1	Московская - Гарькина ТКЗ	1952		
2	Московская - Советской Конституции	11		
3	Московская - Технический университет	17		
4	Московская - Патриарший проезд	19		
5	Московская - Звезда	21		
6	Московская - Дирець Профсоюзная	22		
7	Московская - Обя Больница	33		
8	Московская - ЦМТ	39		
9	Московская - Гарькина ТКЗ	52		
10	Московская - БМПС	57		
11	Московская - Стафеева	58		
12	Московская - Пионерская	7		
13	Московская - Богданкина	70		
14	Московская - Звезд	8		

Рисунок 6 – Просмотр списка светофорных объектов

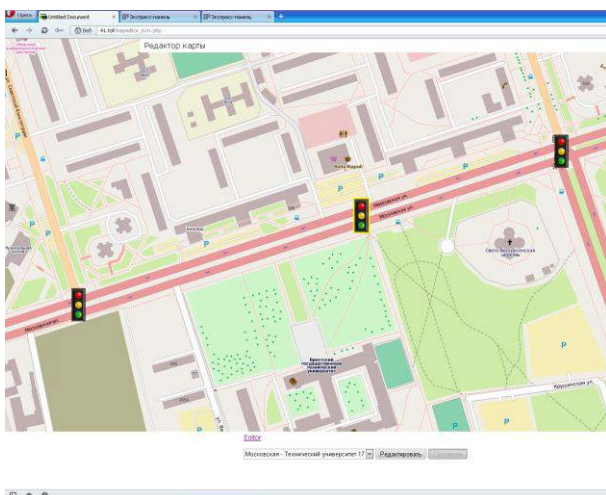


Рисунок 7 – Нанесение светофорных объектов на карту

Администратор имеет возможность наносить на карту светофорные объекты с помощью редактора карты (рисунок 7). Таким образом происходит привязка светофора к глобальным координатам навигационной системы. Карта города на странице редактора автоматически подстраивается под размер окна браузера при загрузке. Отображение карты и панели администратора протестировано в браузерах Opera 10.5+, Chrome 5+, Firefox 5.0 +, IE 8+, Safari.

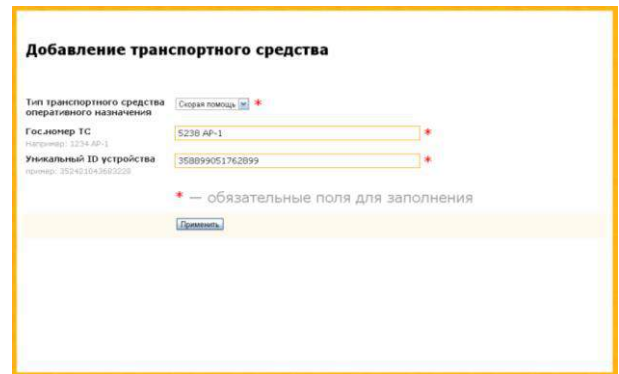


Рисунок 8 – Добавление транспортного средства оперативного назначения

Реализована возможность добавления и редактирования транспортного средства оперативного назначения в информационной базе системы (рисунок 8).

№	Тип ТС оперативного назначения	Гос.номер	Уникальный ID GPS устройства	Изменить	Удалить
1	Скорая помощь	S238 AP-1	358899051762899		
2	Скорая помощь	7854 AB-1	303613100748844		
3	Скорая помощь	4578 AB-1	359710042725841		
4	Милиция	1256 EB-1	359710041825630		
5	Милиция	9841 EB-1	359710041925632		
6	МЧС	1695 AA-1	869158006622605		
7	МЧС	1696 AA-1	869158006622605		
8	МЧС	1698 AA-1	868204057790197		
9	ГМЛ	2147 BB-1	897846515871634		
10	ГМЛ	7898 BB-1	104563782594315		

Рисунок 9 – Просмотр списка транспортных средств оперативного назначения

Просмотреть все транспортные средства в системе можно на странице списка транспортных средств оперативного назначения (рисунок 9).

Администратор имеет возможность просмотреть список пользователей системы (рисунок 10).

№	Имя пользователя	E-mail	Основная группа	Изменить	Удалить	Состояние
1	Согови Алексей	info@tob.by	Администраторы			●
2	User128	user128@yandex.ru	Администраторы			●
3	Levint Energizer	levint2008@yandex.ru	Администраторы			●

Рисунок 10 – Список пользователей

Администратор имеет возможность добавлять/редактировать новых пользователей (рисунок 11)

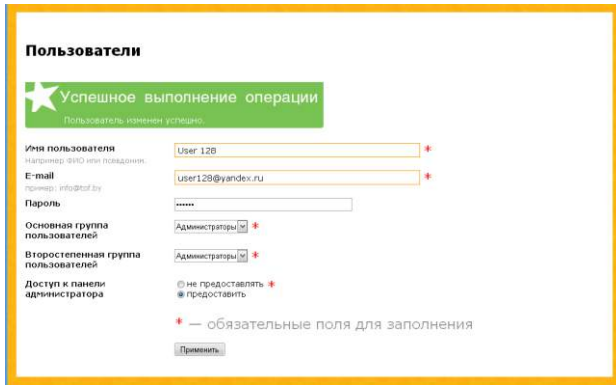


Рисунок 11 – Редактирование пользователя системы

Администратор может видеть журнал действий пользователей (рисунок 12).

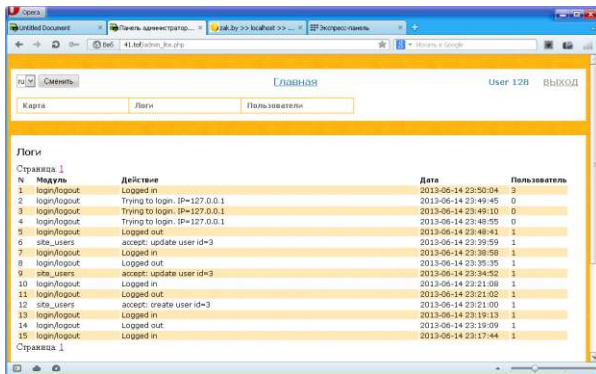


Рисунок 12 – Журнал действий пользователей

Заключение

Внедрение системы приоритетного проезда перекрестков для транспортных средств служб оперативного назначения позволяет повысить безопасность движения, сократить время реагирования на вызов. Благодаря предложенному подходу, который заключается в том, что на перекрестках, через которые движется транспортное средство с включенными маячками, во всех направлениях на светофорах начинает гореть красный сигнал светофора, значительно сокращается риск столкновения с другими транспортными средствами и пешеходами. Для автономных светофоров такой подход позволяет внедрить акустические датчики с минимальными финансовыми затратами.

Работа выполнена при поддержке Европейского гранта «Grant Agreement Number 2013-4550/001-001» по проекту Be-Safe – Белорусская сеть безопасных дорог 544181-TEMPUS-1-2013-1-IT-TEMPUS-JPCR.

Библиографический список

[Robert, 1989] Robert Elling, NREMT-P, Dispelling Myths on Ambulance Accidents, - Journal of Emergency Medical Services (JEMS), July 1989

[A CROSS-CUTTING STUDY, 2006] Traffic Signal Preemption for Emergency Vehicles A CROSS-CUTTING STUDY - 2006, U.S. Department of Transportation, Washington, USA.

[City of Denver Department of Safety, 1978] Time Study on the Effectiveness of the Opticom Traffic Control System (Year 1978), report prepared for the City of Denver by the Denver Department of Safety, FHWA Report No. D-ORTS/78.5.

[U.S. DOT, 2003] Fatality Analysis Reporting System (FARS) Web-Based Encyclopedia Queries for Emergency Use Crash Statistics. <http://www-fars.nhtsa.dot.gov>.

THE SYSTEM PRIORITY TRAFFIC AT INTERSECTIONS "GREEN WAVE" FOR EMERGENCY VEHICLES

Sogoyan A.L., Shuts V.N.

*Brest State Technical University,
Brest, Republic of Belarus*

lexing2008@yandex.ru

lucking@mail.ru

In this paper we consider a system of "Green Wave", which solves the problem of the safe movement of emergency vehicles with included flashing light when driving through controlled junctions.

Introduction

The sudden appearance of an emergency vehicle with flashing light creates a dangerous situation for other vehicles, forcing drivers to react urgently and maneuver for the release of the road.

Some drivers get confused and create conflicts which could lead to an accident involving an emergency vehicle blocking traffic lanes, increase the operational time of arrival of transport to their destination.

Main Part

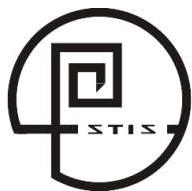
Using the "Green Wave" allows emergency vehicles free passage intersections, reducing confusion for drivers of other vehicles, improving response time to emergencies, to increase the effectiveness of operational life, eliminating conflicts at intersections.

Despite the fact that the driver of the car with the windows closed and turned on the music usually notice the approach of an emergency vehicle only when it is located directly behind the car, the largest percentage of the collision occurs at an intersection.

Conclusion

Implementation of the system of priority directions at intersections for emergency vehicles will improve safety, reduce response time to call. Thanks to this approach, which consists in the fact that at intersections in all directions at the traffic lights change from flashing red light, significantly reduced the risk of collision with other vehicles and pedestrians. For stand-alone traffic lights, this approach allows to introduce acoustic sensors with minimal cost.

This work was supported by the European Grant Agreement Number 2013-4550 / 001-001po project Be-Safe - Belarusian Road Safety Network 544181-TEMPUS-1-2013-1-IT-TEMPUS-JPCR.



OSTIS-2015

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822:514

ПРОГНОЗ РАЗВИТИЯ ИНФОРМАТИЗАЦИИ ОТЕЧЕСТВЕННОГО ОБРАЗОВАНИЯ

Роберт И.В.

*ФГБНУ «Институт информатизации образования Российской академии образования»,
г. Москва, Россия*

rena_robert@mail.ru

В статье описано современное состояние информатизации образования как области научно-педагогического знания. Выявлены содержательные составляющие методологии научной области «информатизация образования». Представлены подходы к осуществлению прогноза развития научно-практических зон, возникающих в традиционных науках и в междисциплинарных исследованиях в связи с развитием информатизации образования, и возникающими при этом научно-педагогическими проблемами. Описаны ближнесрочный и дальнесрочный прогноз развития информатизации образования как области научно-педагогического знания.

Ключевые слова: информатизация образования; информационные и коммуникационные технологии; когнитивно-информационное взаимодействие; конвергенция педагогической науки и информационных и коммуникационных технологий; конвергенция педагогической науки и наукоемких технологий; нано-, инфо-, когнитивные технологии; технология «Виртуальная реальность» трансфер-зона; трансфер-интегративная область научного знания.

1. Современное состояние информатизации отечественного образования как области научно-педагогического знания

1.1. В отечественных научных разработках реализацией возможностей **информационных и коммуникационных технологий (ИКТ)**, в сфере образования (Ваграменко Я.А., Вострокнутов И.Е., Козлов О.А., Мартиросян Л.П., Лавина Т.А., Роберт И.В., Рудинский И.Д., Тихонов А.Н. и др.) занимается отрасль педагогической науки – **информатизация образования** (целенаправленный процесс обеспечения сферы образования методологией, теорией, технологией и практикой разработки и оптимального использования средств ИКТ, ориентированный на реализацию целей обучения, развития индивида, включающий в себя подсистемы обучения и воспитания). Информатизация образования рассматривается в настоящее время и **как область научно-педагогического знания**, которая ориентирована на обеспечение сферы образования методологией, технологией и практикой решения следующих проблем и задач:

– философско-методологические, научно-педагогические, социально-психологические, медицинские, нормативно-технологические и технические предпосылки развития образования в условиях массовой сетевой коммуникации и глобализации современного информационного общества;

– методология и теория отбора содержания образования, разработка методов и организационных форм обучения, воспитания, соответствующих задачам развития личности обучаемого и его социализации в современных условиях информационного общества массовой сетевой коммуникации и глобализации;

– методология разработки моделей инновационных и развитие существующих педагогических технологий применения средств ИКТ в здоровьесберегающих условиях на различных уровнях образования;

– выявление и предотвращение возможных рисков и негативных последствий психолого-педагогического, социо-культурного и медицинского характера при использовании средств ИКТ в образовательных целях;

– создание методических систем обучения, ориентированных на развитие интеллектуального потенциала обучаемого, на формирование умений самостоятельно приобретать знания, осуществлять

информационную деятельность и информационное взаимодействие образовательного назначения;

- разработка и использование электронного образовательного ресурса, инструментальных программных средств и систем автоматизации и управления образовательным процессом, обработки учебного эксперимента как реального, так и "виртуального";

- создание и применение средств автоматизации психолого-педагогического тестирования, диагностирующих методик контроля и оценки уровня знаний обучаемых, их продвижения в учении, установления интеллектуального потенциала обучающегося;

- оценка педагогико-эргономического качества педагогической продукции, функционирующей на базе ИКТ;

- автоматизация и управление технологическими процессами в образовании;

- интеллектуальные информационные системы образовательного назначения.

1.2. В связи с изложенным спектром исследований, инициируемых самим процессом использования средств ИКТ в сфере образования, **информатизация образования рассматривается, как трансфер-интегративная область научного знания**, так как обеспечивает: во-первых, трансфер (от лат. *transfero* – переношу, перемещаю), то есть перенос (перемещение) определенных научных идей или научных проблем в другую научную область, в которой в связи с этим зарождается (образуется) новая, доселе не существующая, научно-практическая зона, адекватно существенным признакам данной науки и практики её реализации; во-вторых, интегративная (от лат. *integration* – объединение), то есть объединяющая в единое целое определенные части (зоны), которые зародились (образовались) в определенной науке и практики ее реализации в связи с феноменом трансфера. При этом под **трансфер-зоной** будем понимать некоторую инновационную область научного знания и его практической реализации, которая возникла в определенной традиционной науке в связи с необходимостью решения научных проблем, привнесенных в эту науку в результате развития информатизации образования.

Рассмотрим каждую из трансфер-зон которые «зародились» в традиционной педагогической науке в виде определенных научно-практических зон, существенные признаки которых позволяют отнести их к педагогике.

А. Дидактика в условиях информатизации образования рассматривается как теория обучения, цели которого отражают запросы на подготовку члена современного информационного общества массовой глобальной сетевой коммуникации, содержание которого отражает кардинальные изменения, происходящие в науке, технике, производстве, а методы которого адекватны современным методам познания научных, социальных закономерностей и реализуют

дидактические возможности ИКТ. Перечислим составляющие трансфер-зоны:

- совершенствование педагогических теорий в аспекте изменения парадигмы учебно-информационного взаимодействия, осуществляемого между обучающим, обучающимся и интерактивным источником учебной информации, функционирующем на базе ИКТ;

- совершенствование предметных методик, реализующих дидактические возможности ИКТ, в условиях изменения парадигмы информационного взаимодействия между обучающим, обучаемым и интерактивным источником учебной информации;

- создание методических систем обучения, ориентированных на реализацию дидактических возможностей ИКТ и использование интерактивного информационного ресурса, в том числе сетевого;

- теория информационно-предметной среды со встроенными элементами технологии обучения, ориентированная на изменение парадигмы учебно-информационного взаимодействия, осуществляемого между обучающим, обучаемым (обучающимся) и интерактивным источником учебной информации, реализующим дидактические возможности ИКТ.

Б. Теория и практика предотвращения возможных негативных воздействий педагогического характера при использовании обучаемым (обучающимся) средств ИКТ в образовательной или досуговой деятельности предполагает их рассмотрение в процессе индивидуальных, групповых, коллективных занятий, в процессе информационного взаимодействия учебного или досугового назначения в условиях информационного взаимодействия.

В. Методология разработки стандартов в области использования ИКТ в профессиональной деятельности педагогических кадров предполагает:

- создание стандартов в области владения средствами ИКТ в профессиональной деятельности учителя школы, библиотекаря, школьного психолога, школьного врача;

- создание стандартов в области владения средствами ИКТ в профессиональной деятельности преподавателей СПО, ВПО по уровням и профилям подготовки;

- создание стандартов в области владения средствами ИКТ в профессиональной деятельности администрации образовательного учреждения по уровням и профилям подготовки;

- создание стандартов в области владения средствами ИКТ научными и научно-педагогическими кадрами в процессе научно-исследовательской деятельности по различным профилям научных специальностей.

Г. Методология разработки стандартов в области использования обучаемым ИКТ в учебной деятельности (общего среднего образования, по уровням и профилям, а также профессионального образования, по среднему и высшему уровням) предполагает:

– создание стандартов в области владения средствами ИКТ в процессе освоения различных учебных предметов (предметных областей);

– создание стандартов в области владения средствами ИКТ в процессе осуществления информационной деятельности и информационного взаимодействия (в том числе, сетевого) учебного назначения.

Таким образом, на основе выявления направлений интеграционных процессов, объединяющих в единое целое определенные научно-практические зоны, которые образовались в определенной традиционной науке и (или) практики ее реализации в связи с феноменом трансфера, можно осуществлять *прогноз развития научно-практических зон, возникающих в традиционных науках в связи с возникающими проблемами информатизации образования*, в том числе в связи с использованием информационных и коммуникационных технологий в сфере образования, и на этой основе осуществлять *ближнесрочный и долгосрочный прогнозы развития информатизации образования как области педагогического научного знания*.

2. Ближнесрочный прогноз развития информатизации образования как области научно-педагогического знания

На основании вышеизложенных подходов к осуществлению прогноза развития информатизации образования как области педагогического научного знания остановимся на *основных направлениях фундаментальных исследований в области информатизации образования в аспекте прогноза* развития последнего на ближнесрочную перспективу.

2.1. В контексте решения психолого-педагогических, медико-социальных и нормативно-правовых проблем направление **«Информатизация образования, интеллектуального развития и социализации современного человека»** выявляет и теоретически обосновывает философско-методологические, социально-педагогические и медико-психологические условия функционирования информационно-образовательного пространства непрерывного образования и пути его создания и использования. Вводится и обосновывается терминологический понятийный аппарат, описывающий определения и их толкование относительно информационно-образовательного пространства в контексте содержательной сути философской категории «пространство».

Выявляются теоретико-методологические основы подготовки педагогических и управленческих кадров как координаторов информатизации образования, интеллектуального развития и социализации современного человека в условиях функционирования информационно-образовательного пространства. Проектируется сетевая модель методической системы подготовки педагогических и управленческих кадров в области применения информационных и коммуникационных технологий в профессиональной деятельности учителей-предметников, администрации и управленческого состава образовательных учреждений, в том числе ИКТ-компетенции, дифференцированные по различным позициям.

Создается научно-методическая база, в состав которой включены педагогико-эргономические и медико-психологические требования к созданию высокотехнологичной здоровьесберегающей информационно-образовательной среды образовательного учреждения различного уровня и профиля. Особое значение отведено созданию теории и технологии научно-методического обеспечения реализации педагогических инноваций в условиях ее функционирования.

Разрабатывается система психологической, методической и медико-социальной поддержки когнитивно-информационного взаимодействия при проектировании и реализации педагогических инноваций в условиях функционирования высокотехнологичной информационно-образовательной среды. Модели сетевого взаимодействия между участниками образовательного процесса в информационно-образовательной среде, реализующей дистанционные образовательные технологии.

Обосновываются теоретические модели и учебно-методическое обеспечение информационной безопасности личности в условиях социально-экономической, культурной дифференциации и глобальной, массовой сетевой коммуникации современного общества.

Остановимся на более подробном описании каждой из отмеченных выше позиций.

2.1.1. Философско-методологические, медико-психологические, социально-педагогические основания создания и развития информационно-образовательного пространства.

В настоящее время широко используется словосочетание «образовательное пространство» без теоретико-методологической подосновы, то есть без обоснованного понимания его содержательной сути. В этой связи остановимся на трансформации содержательной сути словосочетания «образовательное пространство» в контексте понятия философской категории «пространство». Это выражается в том, что в современной научно-педагогической литературе словосочетание «образовательное пространство» приобретает

новые характерные черты, присущие философской категории «пространство».

Перечислим эти характерные черты в контексте терминологии педагогической науки.

1) **Позиционирование элемента** (субъекта, объекта, процесса) **на основе установленного набора параметров, описывающих конкретный элемент, принадлежащий пространству.**

Позицией **субъекта образовательного пространства** (например, сотрудника образовательного учреждения) можно считать его служебный статус, описываемый должностными характеристиками, учитываемыми, в том числе, знания и умения в области использования средств ИКТ в своей профессиональной деятельности. В свою очередь, **набором параметров**, описывающих позицию конкретного элемента, принадлежащего образовательному пространству, можно считать **набор программно-аппаратных средств и систем, научно-педагогических и инструктивно-методических материалов, необходимых для функционирования технико-технологического и информационно-методического обеспечения рабочего места сотрудника образовательного учреждения.**

2) **Наличие системы параметров, описывающих позицию элемента** (субъекта, объекта, процесса), **принадлежащего пространству.**

Системой параметров, описывающих позицию субъекта образовательного пространства (например, сотрудника образовательного учреждения), можно считать совокупность программно-аппаратных средств и систем, научно-педагогических и инструктивно-методических материалов. Наличие этих позиций обеспечивает функционирование образовательного пространства, взаимосвязанных между собой, влияющих на функционирование каждого из них, и имеющих формальное описание в рамках определенного концепта технико-технологического и информационно-методического обеспечения рабочего места сотрудника образовательного учреждения.

3) **Наличие аксиоматики, описывающей «поведение» элемента** (субъекта, объекта, процесса), **принадлежащего пространству.** Аксиоматикой, описывающей «поведение» **субъекта образовательного пространства** (например, сотрудника образовательного учреждения), можно считать исходные положения (перманентно-стабильные), на основе которых разрабатываются служебные (или квалификационные) характеристики сотрудника образовательного учреждения в области его служебного, научно-образовательного, административного статуса. Кроме того, условия и основания применения им научно-педагогической, учебно-методической литературы, программно-методического обеспечения, обязательного для его

профессиональной деятельности и определяющее легитимность его служебного статуса.

4) **Возможность изменения позиции элемента** (субъекта, объекта, процесса), **принадлежащего пространству, с последующим его описанием в той же системе параметров.**

Несмотря на то, что **профессиональная деятельность субъекта образовательного пространства** (например, сотрудника образовательного учреждения) позиционируется и описывается, адекватно его служебному статусу и должностным обязанностям (квалификационным характеристикам) и заданному изначально научно-педагогическому, учебно-методическому, программно-методическому обеспечению его профессиональной деятельности, его «поведение» может изменяться в зависимости от выдвижения новых целей и задач, поставленных перед данным сотрудником, по определенным «п» направлениям. Причин для этого достаточно много в современном интенсивно изменяющемся социуме: инновационные программы обучения, появление новых технологий, активно внедряющихся в образование и т. пр.

Таким образом, в данном контексте представления об образовательном пространстве в психолого-педагогическом понимании данного термина рассматривается на основе философской категории «пространство». Так, понятие «пространство» в психолого-педагогическом контексте представляется посредством описания (словесного, формализованного) субъекта, объекта, процесса по совокупности определенных параметров, которые равнозначны по концепту и могут изменяться «п» направлениям.

Приведем в качестве примера достаточно часто применяемое словосочетание «образовательное пространство школы (вуза)». В вышеозначенном понимании у любого сотрудника (субъект образовательного пространства) образовательного учреждения на его рабочем месте, оснащенном современными средствами ИКТ с выходом в информационную сеть (локальную, глобальную), имеются равные возможности по определенным параметрам, равнозначным по концепту технико-технологического обеспечения (аппаратно-программное оснащение, доступ к информационному ресурсу, возможность осуществления информационного взаимодействия с другими пользователями, с интерактивным источником информации и т. пр.) и информационно-методического обеспечения (научно-педагогические и учебно-методические материалы, электронные учебные средства, прикладное программное обеспечение, инструментальные средства разработки приложений и т. пр.). При этом профессиональная деятельность сотрудника образовательного учреждения позиционируется и описывается адекватно его служебному статусу и должностным обязанностям (или квалификационным характеристикам). Вместе с

тем, деятельность (поведение) субъекта образовательного пространства может изменяться, адекватно целям и задачам, поставленным перед данным сотрудником образовательного учреждения, по определенным «п» направлениям.

В этой связи *теория и технология создания и использования педагогических инноваций в условиях функционирования информационно-образовательного пространства, реализованного на базе ИКТ*, предполагает проведение и реализацию следующих фундаментальных исследований:

- развития понятия «образовательное пространство» в контексте философской категории «пространство»;
- научно-педагогические, технологические и медико-психологические требования к формированию и функционированию информационно-образовательного пространства образовательного учреждения;
- педагогико-эргономические и технико-технологические требования к информационным системам, обеспечивающим создание педагогических инноваций и управление технологическими процессами в образовании;
- профессиональные компетенции преподавателя образовательного учреждения в области создания педагогических инноваций на базе ИКТ.

Выявление тенденций развития дидактики в условиях функционирования информационно-образовательного пространства, реализованного на базе ИКТ, служит основой разработки педагогико-технологических подходов к созданию модели методической системы, обеспечивающей интеллектуальное развитие и социализацию учащихся в условиях функционирования информационно-образовательного пространства. При этом выявляются механизмы осуществления социального партнерства общеобразовательной школы и педвуза в области самообразовательной и проектно-исследовательской деятельности

2.1.2. Теоретико-методические основания подготовки педагогических и управленческих кадров в области информационных и коммуникационных технологий.

Научно-педагогическое и организационно-методическое обеспечение подготовки педагогических и управленческих кадров в области применения средств ИКТ в профессиональной деятельности в условиях двухуровневого образования предполагает стандартизацию в области применения ИКТ в педагогической и организационно-управленческой деятельности сотрудников образовательных учреждений общего среднего и профессионального образования. Создание стандартов в области владения средствами ИКТ предполагается в: профессиональной деятельности учителя школы, библиотекаря, школьного психолога, школьного врача;

профессиональной деятельности преподавателей СПО, ВПО по уровням и профилям подготовки; профессиональной деятельности администрации образовательного учреждения по уровням и профилям подготовки; процессе научно-исследовательской деятельности научно-педагогических кадров.

Теория и технология создания методической системы непрерывной подготовки педагогических и управленческих кадров (по уровням и профилям) как координаторов модернизации образования и социализации современного человека в контексте формирования профессиональных компетенций в области ИКТ направлена на разработку:

- целей, содержания, организационных форм и методов обучения;
- методических подходов к использованию педагогической продукции, функционирующей на базе ИКТ;
- научно-методического и технологического обеспечения мониторинга уровня педагогических ИКТ-компетенций выпускников педагогических вузов и педагогов;
- создание информационных моделей квалиметрического оценивания уровня подготовленности обучающихся и степени овладения ими ИКТ-компетенциями в соответствии с требованиями ФГОС нового поколения.

Таким образом, разработка научно-педагогического обеспечения подготовки педагогических кадров в области создания информационно-коммуникационной предметной среды, разработки авторских сетевых информационных ресурсов и организации научно-исследовательской, управленческой, методической и культурно-просветительской деятельности в условиях функционирования информационно-образовательного пространства, основывается на реализации теоретических положений информатизации образования [5] и научно-педагогических и организационно-методических подходов к созданию интенсивных обучающих систем и типовых учебных аппаратно-программных комплексов.

2.1.3. Система психологической, методической и медико-социальной поддержки пользователя при когнитивно-информационном взаимодействии со средствами информационных и коммуникационных технологий.

В аспекте предотвращения возможных негативных последствий использования средств ИКТ в образовательной деятельности важным компонентом фундаментальных научных исследований является создание психолого-педагогического и медико-социального обеспечения безопасности когнитивно-информационного взаимодействия пользователя с интерактивным источником образовательного ресурса. При этом под *когнитивно-информационным взаимодействием* будем понимать информационное взаимодействие,

реализованное на базе ИКТ, между индивидами или между индивидом (ми) и интерактивным источником образовательного ресурса (в том числе сетевого), основанное на личностном восприятии информации (в том числе «концентрированной», информационно-емкой) адекватно персонифицированной познавательной системе, созданной самим индивидом, или адекватно «персональным конструктам» (в терминологии когнитивной психологии). Восприятие индивидом информационно-емкой информации, как вербальной, так и аудиовизуальной, основано, прежде всего на его личностных особенностях. Вместе с тем, целесообразно формирование умений:

- словесного описания содержательной сути информации (концептуализация информации);
- формализации информации на базе графов, фреймой, логических цепочек, алгоритмов;
- символизации информации в виде пиктограмм, символов;
- графической интерпретации содержательной сути информации в виде графиков, диаграмм;
- онтологизации информации на базе введения персонифицированной системы понятий, их подмножеств и подвидов.

На этой основе возможно создание требований к условиям осуществления когнитивно-информационного взаимодействия, а также создание моделей его реализации. Особое значение приобретает разработка механизмов диагностики мотивации реализации когнитивно-информационного взаимодействия. Параллельно этому должны также разрабатываться теоретико-методологические подходы к созданию научно-методической базы формирования компетентности преподавательского корпуса в этой области.

Результаты вышеозначенных теоретических разработок послужат базой для создания медико-психологических рекомендаций по использованию педагогических инноваций в условиях когнитивно-информационного взаимодействия обучающего, обучаемого и интерактивного средства обучения (или интерактивного информационного образовательного ресурса). Разработанные на этой основе научно-методические рекомендации по осуществлению когнитивно-информационного взаимодействия участников образовательного процесса, реализуемого в информационно-образовательной среде, будут ориентировать обучающего и обучающихся на безопасное и комфортное взаимодействие.

Прикладное значение приобретает создание и использование программно-методического обеспечения Интернет-радио и Интернет-телевидения, ориентированного на социализацию когнитивно-информационного сетевого взаимодействия пользователей, а также учебно-методическое и информационное обеспечение формирования позитивной направленности

Интернет-среды средствами сетевого взаимодействия школьников, учителей и студентов.

2.1.4. Психолого-педагогические основы проектирования и реализации педагогических инноваций в высокотехнологичной здоровьесберегающей информационно-образовательной среде

В связи с появлением высокотехнологичных зданий для образовательных учреждений открываются принципиально новые технико-технологические, психолого-педагогические и методические возможности реализации высокотехнологичной здоровьесберегающей информационно-образовательной среды. Современная реализация высокотехнологичной информационно-образовательной среды учебного заведения предполагает создание совокупности условий для осуществления информационной деятельности и информационного взаимодействия между организаторами учебного процесса, обучаемыми, обучающимися и интерактивными аппаратно-программными комплексами, функционирующими на базе ИКТ для обеспечения системной автоматизации процессов учебно-воспитательной деятельности, управления учебной деятельности, обработки результатов обучения, мониторинга технического состояния оборудования образовательного назначения. При этом в настоящее время не разработаны ни методологические подходы к формированию информационно-образовательной среды учебного заведения, оснащенного высокотехнологичным оборудованием, ни условия организации здоровьесбережения в ней.

Вышеизложенное определяет необходимость разработки методологических, социально-психологических и педагогико-эргономических требований к функционированию высокотехнологичной здоровьесберегающей информационно-образовательной среды учебного заведения.

Реализация педагогико-эргономических и медико-психологических требований к высокотехнологичной здоровьесберегающей информационно-образовательной среде и научно-методическое обеспечение реализации педагогических инноваций наилучшим образом реализуема на базе систем «Виртуальная реальность».

2.1.5. Научно-методическое обеспечение информационной безопасности личности в условиях современного общества.

Концепция информационной безопасности личности предполагает рассмотрение особенностей различных возрастных категорий пользователя как социального ноосубъекта, способного воспринимать и реализовывать футур-инновации в условиях социально-экономической, культурной дифференциации, массовой коммуникации и

глобализации современного общества. Учебно-методическое обеспечение информационной безопасности личности предполагает: выявление устойчивых поведенческих алгоритмов, механизмов, средств информационной защиты человека в условиях глобальной массовой сетевой коммуникации современного общества и разработку комплексных методик формирования устойчивых состояний личности как социального субъекта, обеспечивающих ее информационную безопасность и способы активного противодействия негативным воздействиям информационно-агрессивной среды Интернет. В этой связи необходимо создание методической системы формирования компетенций у студентов педагогических вузов в области информационной безопасности личности, жизнедеятельность которой осуществляется в условиях современного общества информатизации и глобализации.

3. Дальнесрочный прогноз развития информатизации образования как области научно-педагогического знания

На основе анализа технических и технологических инноваций, инициирующих развитие информационного общества массовой сетевой коммуникации и глобализации, процесса конвергенции наук и технологий, а также интенсивного развития нано-, инфо-, когнитивных технологий выявлены **основные направления долгосрочного прогноза информатизации образования**. Остановимся на их описании.

3.1. *Развитие теории обучения и различных подходов к обучению* (личностно-ориентированный, деятельностный, компетентностный подходы к обучению; проблемное обучение, алгоритмизация обучения и др.) обусловлено следующими факторами:

- изменением парадигмы учебного информационного взаимодействия (в том числе реализацией сетевой парадигмы), при котором интеллектуально активными становятся обучающийся, обучающий и интерактивный источник учебной информации в условиях функционирования информационно-образовательного пространства;

- использованием электронных баз и банков данных учебно-методических материалов, в том числе «банков данных по эксперименту», «банков данных проб и ошибок», «библиотек методических решений» и пр.;

- реализацией различных видов учебной деятельности в условиях использования ИКТ (информационной деятельности по поиску, сбору, обработке, применению учебной информации, а также деятельности по моделированию, формализации, продуцированию учебного материала, в том числе, в электронном виде);

- совершенствованием педагогических технологий, ориентированных на самостоятельную учебно-информационную деятельность и социализацию сетевого взаимодействия как с пользователями, так и с интерактивным электронным ресурсом образовательного назначения.

3.2. Важным направлением перспективных фундаментальных исследований в области информатизации образования является **конвергенция наук и технологий** [1], а именно – **конвергенция педагогической науки и наукоемких технологий**.

Учитывая словарное значение слова конвергенция (от английского convergence – приближение, схождение, уподобление; или от латинского convergens – совпадающий или convergere приближаться, сходиться), определим **конвергенцию** как схождение, сближение или сходство, совпадение каких-то признаков или свойств независимых друг от друга объектов, процессов, явлений. При этом определим **конвергентный** – как характеризующийся конвергенцией.

Будем рассматривать **педагогическую науку** как науку о специально организованной целенаправленной и систематической деятельности педагога, направленной на обучение, воспитание, передачу социального опыта ученику с использованием определенных форм и методов передачи содержания образования. Современные информационные и коммуникационные технологии (ИКТ) рассматриваются в данном контексте как практическая часть научной области информатики, представляющая собой совокупность средств, способов, методов автоматизированного сбора, обработки, хранения, передачи, использования, продуцирования информации для получения определенных, заведомо ожидаемых, результатов. При этом информационная технология, реализация которой осуществляется с помощью средств микропроцессорной, вычислительной («компьютерной») *техники* отличается следующими характерными особенностями:

- реализация возможностей современных программных, программно-аппаратных и технических средств и устройств, средств и систем передачи, транслирования информационных ресурсов, информационного обмена;

- использование специальных формализмов (логико-лингвистических моделей) для представления декларативных и процедурных знаний в электронной форме; при этом логико-лингвистическое моделирование резко расширяет возможности решения задач для трудно или совсем неформализуемых областей знаний и сфер деятельности;

- обеспечение прямого (без посредников) доступа к диалоговому режиму при использовании

профессиональных языков программирования и средств искусственного интеллекта;

- обеспечение простоты процесса взаимодействия пользователя с компьютером, исключение необходимости регулятивного сопровождения.

Реализация всех вышеперечисленных позиций основана на характерных особенностях ИКТ, отмеченных выше.

Определим *конвергенцию педагогической науки и информационных и коммуникационных технологий* как приближение, схождение, уподобление педагогических технологий и ИКТ, а также их взаимное влияние друг на друга, возникновение сходства в функциях педагогической науки и ИКТ, а также в структурах педагогических технологий и ИКТ.

Процесс конвергенции педагогической науки и технологий инициирует развитие информатизации образования за счёт взаимного влияния друг на друга различных областей психолого-педагогической науки и информационных и коммуникационных, а также когнитивных технологий. При этом перспективные фундаментальные научные исследования ориентированы на создание теоретико-методологических оснований к познанию закономерностей развития информатизации образования на основе выявления условий взаимного влияния и проникновения информационных и коммуникационных технологий в педагогические технологии и обратно, а также к выявлению сходства в функциях и структурах информационных и коммуникационных технологий и педагогических технологий.

На этой основе прогнозируется создание информационно-коммуникационных предметных сред со встроенными элементами технологии обучения по каждому учебному предмету (предметной области), позволяющих предоставить в распоряжение обучающегося и обучающего инструмент визуализации объектов данной предметной области, инструмент измерения и исследования закономерностей для осуществления самостоятельного «микрооткрытия» изучаемой закономерности. Следствием вышеизложенного становится *создание методических систем обучения в условиях функционирования информационно-коммуникационных предметных сред со встроенными элементами технологий обучения (по каждому учебному предмету или предметной области)*, обеспечивающих: имитацию различных, поддающиеся описанию, операционализаций; создание виртуальных моделей, имитирующих динамику поведения изучаемых объектов или развития процессов с последующим анализом и прогнозом тенденций их изменения; конструирование виртуальных миров с использованием библиотек готовых виртуальных объектов.

3.4. Развитие научно-методического обеспечения подготовки, переподготовки и повышения квалификации педагогических и управленческих кадров в области использования программно-аппаратных и информационных комплексов образовательного назначения прогнозируется в условиях функционирования высокотехнологичной информационно-образовательной среды учебного заведения. В данном контексте разработка дидактических и программно-технологических оснований информационного взаимодействия обучающихся с робототехническими системами и устройствами, реализующими технологию виртуальной реальности, в условиях функционирования высокотехнологичной информационно-образовательной среды учебного заведения определяет развитие профессионального образования и профильной подготовки в среднем звене образования.

В качестве инновационных средств подготовки педагогических и управленческих кадров прогнозируется система сетевых инструментальных средств разработки электронного образовательного ресурса, контент которого предполагает реализацию «встраиваемых» возможностей информационных технологий (вычислительных, поисковых, аналитических, моделиформирующих), что позволяет осуществлять:

- имитацию реальных (учебных, профессиональных) действий, поддающихся операционализации и моделированию, с последующим обеспечением тренировки данного вида деятельности;

- имитацию динамики развития изучаемых или исследуемых объектов, процессов с возможностью анализа и прогноза тенденций их изменения или развития с последующим обеспечением информационного взаимодействия на уровне обмена информацией (данными параметров, визуальными образами или символами);

- имитацию информационного взаимодействия с виртуальными объектами, с возможностью привлечения информации в области аккумулированного опыта осуществления деятельности (учебной, профессиональной);

обеспечение информационного взаимодействия с виртуальными объектами определенной предметной области, адекватно ее закономерностям.

3.5. Методическое и технологическое обеспечение создания и применения тренажеров, позволяющих имитировать и моделировать любые, поддающиеся описанию, операционализации, максимально приближающие деятельность пользователя к реальной учебной и (или) профессиональной деятельности прогнозируется на основе технологии «Виртуальная реальность». Такие тренажеры обеспечивают:

- создание «виртуальных миров», которые выступают по отношению к реальному миру как модели, позволяющие имитировать динамику поведения изучаемых или исследуемых объектов или развития процессов;

- создание моделей (с определенным, чаще всего лимитированным, числом качественных характеристик, таких, как размер, цвет, вес, потенциал движения и т.д.) виртуальных объектов и окружающей их виртуальной среды, позволяющих имитировать реальные объекты, динамику протекания определенных реальных процессов с последующим анализом и прогнозом тенденций их изменения или развития;

- реализацию стереоскопической визуализации трехмерных объектов (в области математики, биологии, архитектуры, искусства и т. пр.) и абстрактных данных в виде графиков, диаграмм, матриц, таблиц, схем, структур и пр.;

- использование библиотек готовых виртуальных объектов или моделирование объектов из ранее созданных отдельных их частей, в том числе с использованием звуковых возможностей программного обеспечения;

Заключение

Таким образом, на основе анализа фундаментальных и прикладных научных исследований в области наук от образования, технических и технологических инноваций, инициирующих развитие информационного общества массовой сетевой коммуникации и глобализации, анализа процесса конвергенции наук и технологий, а также интенсивного развития нано-, инфо-, когнитивных технологий были спрогнозированы (на краткосрочную перспективу) и представлены выше основные направления развития информатизации образования.

Библиографический список

[Ковальчук М.В., 2011] Ковальчук М.В. Конвергенция наук и технологий – прорыв в будущее // Российские нанотехнологии. Том 6.- 2011.- №1-2.- С.13-23.

Концепция долгосрочного социально-экономического развития Российской Федерации на период до 2020 года. Государственная Программа Российской Федерации «Информационное общество (2011-2020 годы)».

Основы политики Российской Федерации в области развития науки и технологий на период до 2020 года и дальнейшую перспективу. (Утверждено Президентом Российской Федерации 11. января 2-12 г., Пр-83).

Перечень критических технологий Российской Федерации (в части информационно-телекоммуникационные системы) (утвержден Указом Президента Российской Федерации, № 899 от 7.07.20011 г.).

[Роберт И.В., 2014] Роберт И.В. Теория и методика информатизации образования (психолого-педагогический и технологический аспекты). 3-е изд. – М.: ИИО РАО, 2014. – 354 с.

[Роберт, Т.А., 2010] Толковый словарь слов и словосочетаний понятийного аппарата информатизации образования / Составители: И.В. Роберт, Т.А. Лавина. М.: ИИО РАО, 2010.

NATIONAL EDUCATION INFORMATIZATION DEVELOPMENT PROGNOSIS

Robert I.V.

FSBSI «Education informatization institute of the Russian academy of education»

rena_robort@mail.ru

This article describes present state of education informatization as a scientific and pedagogic area of knowledge. Substantial parts of «education informatization» knowledge area methodology are highlighted. Approaches to conducting a development prognosis of research-and-practice zones emerging in traditional and interdisciplinary sciences due to education informatization development are presented. Short-term and mid-term development prognosis of education informatization as a scientific and pedagogic area of knowledge are made.

Introduction

Education informatization is a purposeful process of providing education sphere with methodology, theory, technology and practice of development and optimal usage of ICT oriented towards education goals implementation, individual development including education and guidance subsystems. It is also viewed as a scientific and pedagogic knowledge area oriented towards providing education sphere with methodology, technology and practice of solving a wide range of problems, including:

– various prerequisites for education development considering mass network communication and information society globalization;

– methodology and theory for selecting educational content, development of organizational forms and methods of education and guidance according to modern society requirements;

– design and development methodologies for new and existing pedagogic technologies of ICT usage in health-saving environment across various levels of education;

– detection and prevention of possible risks and negative consequences of varied nature due to ICT usage in education;

Traditional pedagogics spawns a number of transfer zones. Transfer zone is an innovative area of scientific knowledge and its practical implementation emerged due to need for solving scientific problems introduced into that science as a result of education informatization development. An example of one such zone would be a theory and practice of negative pedagogical impacts prevention.

Due to existence of the transfer phenomena in modern science we can do a short-term and long-term development prognosis of research-and-practice zones emerged in traditional sciences in connection with emerging education informatization problems.

Main Part

The short-term prognosis of education informatization is as follows.

A theory and technology of pedagogical innovation in an ICT-based educational spaces assumes a wide range of fundamental research, including: development of the “educational space” term, formulating requirements for its functioning, technological requirements for informational systems in education, professional pedagogical skills in creating an ICT-based educational innovations.

A theory and technology for creating a methodological system of continuous professional training for pedagogical and management personnel for ICT-based educational space is oriented towards development of education forms and methods, methodological approaches to creating and using an ICT-based pedagogical products, scientific, pedagogical and technical ICT skill monitoring tools, creating informational models for skill quality control, including ICT skills.

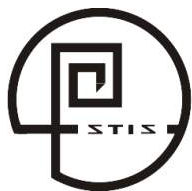
Methodological recommendations on cognitive interaction among educational process participants will be developed. They will lead educators and trainees towards safe and comfortable interactions. Furthermore, a set of requirements for high-tech health-saving educational environment functioning needs to be developed. Personal information security in education also needs further research.

The main points of long-term prognosis of education informatization are as follows:

- education theory advancements, development of new teaching approaches;
- convergence of pedagogical science and high-end technologies;
- development of methodological support tools for training, retraining and skill improvement in teaching and management;
- methodological and technical support tools for creating and applying training facilities that model real-world professional and/or educational activities as realistically as possible.

Conclusion

Based on analysis of fundamental and applied scientific research in education, technical and technological innovations, initiating development of mass network communication and globalization society, science and technology convergence analysis, and rapid development of nano-, info- and cognitive technologies, primary directions for education informatization development were predicted for the short and long term.



OSTIS-2015

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822:514

ИНТЕЛЛЕКТУАЛЬНЫЕ ТЕХНОЛОГИИ ОБУЧЕНИЯ В ИНФОРМАЦИОННО-ОБРАЗОВАТЕЛЬНОМ ПРОСТРАНСТВЕ

Тельнов Ю.Ф., Трембач В.М.

*Московский государственный университет экономики, статистики и информатики (МЭСИ),
г.Москва, Россия*

ytelnov@mesi.ru

vtrembach@mesi.ru

Рассмотрен подход к созданию современных интеллектуальных обучающих систем на основе многократно используемых компонентов - учебных объектов и образовательных сервисов. Представлена архитектура интеллектуальной обучающей системы с многократно используемыми компонентами на основе онтологической и многоагентной организации информационного образовательного пространства. Описана интеллектуальная технология обучения с помощью различных классов образовательных сервисов.

Ключевые слова: интеллектуальная обучающая система, информационно-образовательное пространство, многократно используемые компоненты, учебные объекты, образовательные сервисы, интеллектуальная технология

ВВЕДЕНИЕ

Развитие сетевых технологий и результаты исследований в области ИИ дали возможность создания интеллектуальных обучающих систем, которые позволяют адаптировать учебный процесс к конкретному обучающемуся в информационно-образовательном пространстве (ИОП) [Богословский и др., 2000; Зенкина и др., 2014]. Сформировалось и в дальнейшем укрепляется понимание того, что современные обучающие системы должны создаваться с использованием возможностей вычислительных сетей, средств телекоммуникации и интеллектуальных технологий. Построение архитектур интеллектуальных обучающих систем (ИОС) связано с реализацией компетентностного подхода в образовании, заключающегося в переносе акцента с предметного обучения на формирование результатов в соответствии с требованиями конкретной профессии [Тельнов, 2014а; Зенкина и др., 2014].

Использование основных положений компетентностного подхода в непрерывном образовании означает формирование разнообразных образовательных программ для различных категорий обучающихся по различным видам и формам обучения. Данная задача может быть решена за счет введения адаптивности в процесс обучения. Основная проблема заключается в планировании и реализации образовательной

траектории обучающегося к цели на основе управляемого усвоения им учебного материала в обучающей системе. Индивидуальную образовательную траекторию обучающегося [Зенкина и др., 2014] можно определить как персональный путь реализации личностного потенциала обучающегося в образовательном процессе.

Согласно психолого-педагогической литературе, ценность индивидуальной образовательной траектории обучающегося заключается в том, что она позволяет на основе оперативно регулируемой самооценки, активного стремления к совершенствованию собственных знаний и умений, пополнить знания при проектировании своей учебной деятельности с целью отработки методов и техники самостоятельной работы в различных формах учебно-познавательной деятельности. При этом очень важно наличие у каждого обучающегося своей индивидуальной задачи по проектированию индивидуальной образовательной траектории, которая формируется из повторно используемых компонентов – учебных объектов, хранящихся в общем репозитории.

Для совместного использования учебных объектов преподавателями и студентами применяются программные сервисы, которые могут рассматриваться как специальные виды объектов, представляемые в общем реестре (репозитории) образовательных сервисов ИОП [Тельнов, 2014б]. В

качестве основных классов образовательных сервисов предлагается выделить:

- Формирование образовательной программы с учетом возможности построения индивидуальной траектории обучения.
- Формирование учебно-методических комплексов под индивидуальную траекторию обучения.
- Выбор и выполнение тестовых заданий с учетом их адекватности формируемым компетенциям и оценки соответствующего уровня.
- Формирование и выполнение практических заданий (проектной деятельности), включая постановку практической задачи с учетом профиля обучающегося и отбор проектных решений по прецедентам.

Целью настоящей статьи является формализация интеллектуальной технологии формирования индивидуальных траекторий обучения на основе повторно-используемых компонентов (учебных объектов и образовательных сервисов). В качестве инструментальных средств реализации интеллектуальной технологии предлагается применять методы онтологической и многоагентной организации информационного-образовательного пространства.

1. Онтологическая организация хранения учебных объектов и сервисов в репозитории

Учебный объект - относительно новое понятие в образовательных технологиях. По определению комитета по стандартам обучающих технологий IEEE (IEEE LTSC), учебный объект – это "любой объект, цифровой или нецифровой, который может использоваться многократно, на который можно делать ссылки при использовании соответствующей обучающей технологии". Главными характеристиками учебного объекта являются возможность многократного использования и разметка метаданными.

Учебные объекты делают возможным создание независимых компонентов образовательного контента, которые обеспечивают реализацию образовательных целей. Существуют многие подходы к их классификации, структуре, описаниям составных элементов. IMS и SCORM обеспечивают детальные стандарты и рекомендации для учебных объектов. Стандарт SCORM [SCORM, 2004] определяет структуру учебных материалов и интерфейс среды выполнения. Благодаря этому, учебные объекты могут быть использованы в различных системах электронного дистанционного образования. SCORM описывает эту структуру с помощью нескольких основных принципов, спецификаций и стандартов, основываясь при этом на других уже созданных спецификациях и стандартах электронного и дистанционного образования.

Составной частью SCORM являются метаданные учебных объектов (Learning Object Metadata, LOM). Цель этого стандарта – облегчение поиска, рассмотрения, оценки и использования учебных объектов для учеников, учителей или автоматических программных процессов. Определяя общую концептуальную схему данных, данный стандарт обеспечивает связывание учебных объектов.

Стандарты для метаданных определяют минимальный набор атрибутов, необходимый для организации, определения местонахождения и оценки учебных объектов. Значимыми атрибутами учебных объектов являются:

- тип объекта,
- имя автора объекта,
- имя владельца объекта,
- сроки распространения и
- формат объекта и др.

Использование метаданных учебных объектов является центральным звеном в ИОС, которые могут учитывать индивидуальные требования к обучению.

Одним из возможных подходов к организации хранения учебных объектов теоретического, практического, тестового, процедурного и пр. назначения может быть предложен подход, основанный на системе метаданных Дублинского ядра DC (Dublin Core) [DC, 2013] и являющийся перспективным средством формирования описательных метаданных для широкого класса учебных объектов, которые могут быть представлены в онтологии учебных объектов. Согласно рекомендации RFC [Система, 2005], все элементы Дублинского ядра можно разбить на три группы: элементы, относящиеся к содержанию ресурса - Content (Title, Subject, Description, Type, Source, Relation, Coverage); элементы, описывающие цифровой ресурс с точки зрения интеллектуальной собственности - Intellectual Property (Creator, Publisher, Contributor, Rights); элементы, относящиеся к конкретному экземпляру ресурса - Instantiation (Date, Format, Identifier, Language). В дополнение к перечисленному составу атрибутов необходимо добавить ссылки на коды формируемых компетенций по различным направлениям подготовки высшего образования с указанием необходимого уровня освоения компетенций.

Аналогично учебным объектам задается описание программных сервисов как объектов специализированных реестров, по сути таких же репозиторийев. В добавление к перечисленной метаинформации ключевые слова могут быть дифференцированы по ролевым признакам, уточняющим семантику использования сервиса [3]:

- VD – вид деятельности (проектирование, разработка, тестирование, применение и др.),
- G – цели (коды компетенций и уровень овладения),

- I – вход (исходные данные),
- O – выход (результат),
- Act – актер (исполнитель вида деятельности),
- Source – ссылка на источник знаний (учебный объект).

Метаописание учебных объектов и сервисов представляются в онтологиях учебных объектов и сервисов и совместно с онтологиями предметной области и компетенций позволяет повысить адаптивность и интеллектуальность их применения в различных целях при формировании индивидуальных траекторий обучения

2. Архитектура интеллектуальной обучающей системы

Для осуществления образовательной деятельности можно выделить задачи, которые решаются ИОС на всех этапах формирования компетенций обучающихся. Такими задачами являются:

- регистрация пользователей;
- получение доступа к персональной среде;
- просмотр базы знаний с обеспечением целостности данных, исключением ошибок ввода, облегчением ввода данных, автоматизацией обработки описаний на множестве объектов и поиском;
- просмотр, наполнение и редактирование репозитория с широкими возможностями в оформлении учебного материала, большим набором мультимедийного наполнения, простотой и удобством, как создания новых учебных статей, так и их редактирования, с обеспечением коллективного доступа, наличием механизма ревизии описаний;
- ввод текущих оценок компетенции;
- оценка уровня знаний;
- контроль получения знаний;
- формирование индивидуальных траекторий – планирование индивидуальной программы обучения;
- реализация индивидуальных программ обучения с использованием индивидуальной среды обучения.

Для решения выделенных задач в интеллектуальной обучающей системе на основе агентно-ориентированного подхода можно выделить следующие компоненты:

- интерфейс обучающегося;
- модуль оценки знаний обучающегося;
- подсистема формирования индивидуальных планов обучения;
- система управления базой знаний;
- индивидуальная среда обучения;
- репозиторий учебных объектов.

Такая ИОС ориентирована на индивидуальную работу с обучающимися. Интерфейс обучающегося является по своей сути аппаратно-программной

сущностью, обеспечивающей обучаемому возможность работы со всеми имеющимися в системе сервисами. Этот модуль позволяет формировать и хранить требуемые компетенции, текущие компетенции обучающегося и сформированные индивидуальные программы обучения. Модуль оценки знаний дает обучающемуся возможности определять свой текущий уровень компетенций, контролировать процесс отработки индивидуальной программы обучения. Подсистема формирования индивидуальных программ обучения осуществляет планирование последовательности учебных объектов в зависимости от требуемой компетенции и имеющихся у обучающегося знаний, умений, навыков. Система управления базой знаний обеспечивает функции системы управления базами данных и функции, связанные с формированием и использованием знаний в работе с обучающимся. Индивидуальная среда обучения должна обеспечить возможность работы со всеми учебными объектами в рамках спланированной последовательности. Репозиторий учебных объектов должен обеспечить создание, хранение и использование учебных объектов различной природы.

Разработка современных ИОС связана с рядом проблем, которые сдерживают эффективные технологии проектирования интеллектуальных систем. Для их преодоления необходимы определенные условия и наработки. Одним из условий должно стать создание эффективной технологии проектирования ИОС. В работе [Голенков и др., 2013] для решения этой задачи предлагается методика компонентного проектирования, которая является фактором зрелости любых технологий и которая основывается на постоянно расширяемых библиотеках многократно используемых компонентов (типовых технических решений). Для этого требуется решение многих вопросов, некоторые из них представлены ниже:

- обеспечение совместимости (интегрируемости) компонентов интеллектуальных систем на основе унификации представления этих компонентов;
- создание библиотек многократно используемых (типовых) компонентов интеллектуальных систем и уточнения типологии таких компонентов (предметные онтологии, многократно используемые фрагменты баз знаний, машины вывода, интерфейсные компоненты и т.д.);
- создание средств компьютерной поддержки синтеза интеллектуальных систем из имеющихся компонентов и некоторые другие.

Требуется также независимость процесса обновления базы знаний интеллектуальной системы от процесса обновления моделей, а также методов обработки знаний от процесса обновления средств технической реализации [Трембач, 2012].

Необходимость решения этих вопросов связана с тем, что в каждой предметной области множество задач и необходимых решений может несколько изменяться. Для решения в ИОС задач образования, ее системе управления базой знаний (СУБЗ) необходимы следующие блоки:

- блок для формирования и реализации планов обучения,
- блок для обучения на собственном опыте и успешных решениях,
- блок оценки знаний обучающихся.

Блок для формирования и реализации планов обучения позволяет использовать персональную среду обучения, которая обеспечивает формирование новых компетенций каждым обучаемым с учетом его текущих знаний, умений, навыков.

Работу с актуальными знаниями, их соответствие действительности обеспечивает блок для обучения на собственном опыте и успешных решениях. В этом блоке реализуются методы машинного обучения.

Для полноценного функционирования персональной среды обучаемого необходим блок оценки знаний. Он обеспечивает мониторинг обработки индивидуальной траектории обучения.

Архитектура ИОС, обеспечивающая реализацию рассмотренных решений, показана на рис. 1.

Данная архитектура включает ядро ИОС – систему управления базами знаний ИОС. На уровне интерфейсов СУБЗ находится блок формирования и поддержки агентов обучающегося. С помощью многократно используемых компонентов (типовых технических решений) формируется агент для каждого обучающегося. В сформированном агенте обучающегося используются следующие элементы: интерфейс обучаемого с персональной средой; формирователь (планировщик) индивидуальной траектории; модуль (агент) для оценки знаний обучаемого на всех этапах формирования компетенций; модуль персональной среды обучения для реализации индивидуальной траектории обучения; модуль формирования репозитория персональной среды обучения и поддержания его в актуальном состоянии.

Для работы СУБЗ ИОС могут потребоваться многократно используемые компоненты для решения задач машинного обучения (накопления опыта), актуализации содержимого базы знаний ИОС, распознавания типовых ситуаций в обучении, валидации баз знаний и другие [Трембач, 2013], а также набор онтологий предметной области, компетенций, учебных объектов и образовательных сервисов [Тельнов, 2014а].

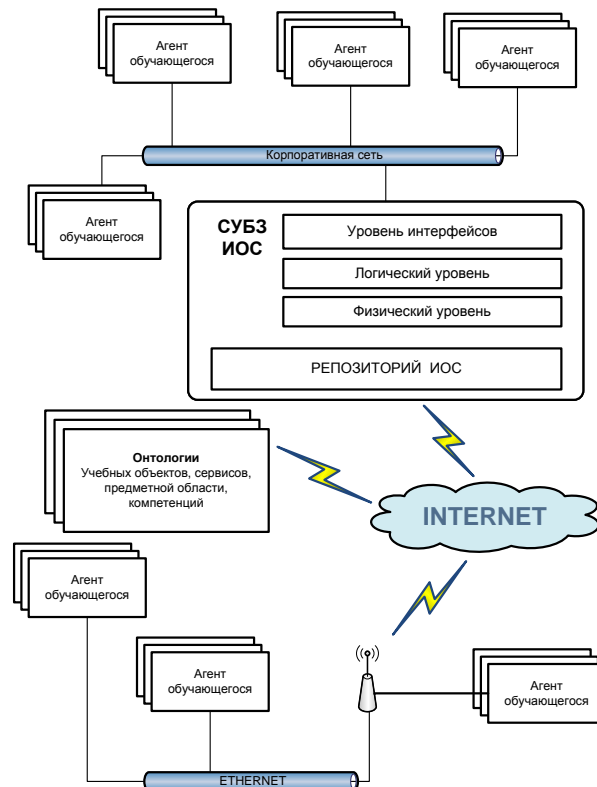


Рисунок 1 – Архитектура интеллектуальной обучающей системы

Рассмотрим основные технологические процессы индивидуального обучения на основе представленной архитектуры ИОС.

3. Технология формирования индивидуальной траектории обучения студентов

Для обучающихся по образовательным программам важно обеспечить своевременную поставку контента, определить последовательность изучения учебного материала и доступа к дополнительным источникам (научным, учебным, методическим материалам). Таким образом, необходимым является формирование последовательности учебных объектов в привязке к конкретным профессиональным компетенциям, формируемым в рамках образовательной программы, которые составляют учебно-методические комплексы.

Схема процесса формирования учебно-методического комплекса, основанная на использовании репозитория учебных объектов, онтологий компетенций и предметной области, представлена на рисунке 3 и включает следующие технологические операции:

- формирование и обновление модели обучающегося;
- генерация сценария обучения в соответствии с моделью обучающегося;
- формирование учебно-методического комплекса в соответствии с индивидуальной траекторией обучения.

Для учебно-методического обеспечения участников образовательного процесса должна поддерживаться модель обучающегося, определяющая его принадлежность к той или иной категории. Так, объективные характеристики обучающегося должны быть получены из истории обучения (направление подготовки, профиль, полученные знания, умения, навыки, освоенные компетенции) и регистрационных данных. Также можно выделить ряд характеристик, описывающих предпочтения обучающегося:

- характер профессиональной работы,
- вид деятельности,
- роль в виде деятельности (пользователь, исполнитель, аналитик, руководитель, исследователь),
- квалификация,
- вид полученного образования,
- цель изучения (ознакомление, практическая работа, теоретическое обобщение).

Программный сервис, обеспечивающий формирование модели обучающегося, должен непрерывно отслеживать его активность, изменение поведения, навыков, уровня знаний с целью повышения релевантности результатов поиска научно-образовательного контента в информационно-образовательном пространстве.

На основе полученной модели обучающегося формируется сценарий обучения. При этом выполняется проверка соответствия выбранного вида деятельности модели обучающегося направлению подготовки, профилю, цели изучения и уровню компетенции. При этом компетентностные характеристики направления подготовки, профиля, уровня образования должны быть сопоставлены процессам и задачам, решаемым субъектами соответствующих видов профессиональной деятельности. Ролевые характеристики субъектов отражаются в онтологии предметной области, а соответствие между целями обучения и процессами (задачами) определяется на основе моделей компетенций. Если соответствие есть, то сценарий обучения считается сформированным, в противном случае определяют другой вид деятельности для обучения или проводится повышение квалификации по недостающим знаниям (углубление знаний в смежных учебных курсах).

Затем определяются связанные элементы вида деятельности и предоставление соответствующих учебных объектов. Для формирования перечня изучаемых учебных объектов определяется необходимость углубления изучения вида деятельности. Если требуется углубленное изучение материала, то осуществляется декомпозиция вида деятельности на составные процессы и дальнейшее их сопоставление с моделью обучающегося. Если же углубление изучения материала не требуется, рассматривается необходимость расширения изучения вида деятельности. При расширении

изучения вида деятельности определяются подвиды (подклассы) деятельности и основных понятий, осуществляется их сопоставление с моделью обучающегося и отбор объектов для изучения.

В ходе сопоставления с моделью обучающегося должна быть предоставлена возможность вызвать сервисы тестирования и провести серию уточняющих тестов на понимание основных понятий, а затем, интерпретировав результаты тестирования, предоставить дополнительные материалы. В случае успешного тестирования системой формируется список понятий для изучения – сценарий обучения.

После формирования сценария обучения происходит обращение к конкретным учебным объектам, которые выдаются обучающемуся в заданной последовательности. Для этого программный сервис по гиперссылкам в онтологии целей обучения и онтологии предметной области осуществляет переход в онтологию учебных объектов и далее к конкретным учебным объектам. Результаты изучения материалов влияют на список учебных объектов, формируемый в дальнейшем. Они интерпретируются программным сервисом и отражаются в модели обучающегося.

ЗАКЛЮЧЕНИЕ

Представленная в статье интеллектуальная технология обучения в интегрированном информационно-образовательном пространстве на основе сервисной архитектуры позволяет в значительной степени сократить затраты и повысить актуальность научно-образовательного контента за счет интеграции усилий участников образовательного процесса на основе применения интеллектуального инструментария программных сервисов. Таким образом, использование информационно-образовательного пространства при формировании профессиональных компетенций студентов обеспечивает, с одной стороны, полноту и всесторонность охвата научно-образовательным контентом потребностей обучающихся, а с другой стороны, повышает гибкость и адекватность формирования индивидуальных траекторий обучения применительно к конкретным профилям подготовки специалистов.

Работа выполнена при финансовой поддержке РФФИ (проект № 13-07-00917а, проект № 14-07-00880а).

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- [Богословский и др., 2000] Богословский В.И., Извозчиков В.А., Потемкин М.Н. Наука в педагогическом университете: Вопросы методологии, теории и практики. - СПб., 2000.
- [Гаврилова и др., 2006] Гаврилова Т.А., Кудрявцев Д.В., Горовой В.А. Модели и методы формирования онтологий // Научно-технические ведомости Санкт-Петербургского государственного политехнического университета. 2006. № 46.
- [Гаврилова и др., 2006] Гаврилова Т.А., Кудрявцев Д.В., Горовой В.А. Модели и методы формирования онтологий //

Научно-технические ведомости Санкт-Петербургского государственного политехнического университета. 2006. № 46.

[Голенков и др., 2013] Голенков В.В., Гулякина Н.А. Открытый проект, направленный на создание технологии компонентного проектирования интеллектуальных систем. – В кн Междунар. научн.-техн. конф. . Открытые семантические технологии проектирования интеллектуальных систем = Open Semantic Technologies for Intelligent Systems (OSTIS-2013): материалы III Междунар. научн.-техн. конф. (Минск, 21-23 февраля 2013г.) / – Минск : БГУИР, 2013. – с. 55-77.

[Зенкина и др., 2014] Зенкина С.В., Трембач В.М., Некоторые подходы к представлению действительности для решения задач обучения специалистов в современной образовательной среде, // Научно-практический журнал «Открытое образование», МЭСИ, №4, 2014, с. 39-49

[Тельнов, 2014а] Тельнов Ю.Ф. Принципы и методы семантического структурирования информационно-образовательного пространства на основе реализации онтологического подхода // Вестник УМО. Экономика, статистика, информатика, 2014, № 1. – С. 187-191.

[Тельнов, 2014б] Тельнов Ю.Ф. Модель многоагентной системы реализации информационно-образовательного пространства // Четырнадцатая национальная конференция по искусственному интеллекту с международным участием КИИ-2014 (24-27 сентября 2014 г., г. Казань, Россия): Труды конференции. Т.1. – Казань: Изд-во РИЦ «Школа», 2014. – С. 334 – 343.

[Трембач, 2013] Трембач В.М. Системы управления базами эволюционирующих знаний для решения задач непрерывного образования: Монография. - М.: МЭСИ, 2013 - 255 с.

[Трембач, 2012] В.М. Трембач, Основные этапы создания интеллектуальных обучающих систем // Программные продукты и системы, №3, 2012, с. 148-152.

[SCORM, 2004] SCORM - сборник спецификаций и стандартов для систем дистанционного обучения, URL: <http://www.edu.ru/db/portal/e-library/00000053/SCORM-2004.pdf>

[DC, 2013] <http://dublincore.org/documents/dces/>

[Система, 2005] Система метаданных дублинского ядра // новости XML-технологий, URL: <http://xmlhack.ru/archives/2005/06/000111.html>

INTELLIGENT TECHNOLOGIES OF TUTORING IN INFORMATION-EDUCATIONAL SPACE

Telnov Yu.F., Trembach V.M.,

Moscow State University of Economics, Statistics and Informatics, Moscow, Russia

ytelnov@mesi.ru

vtrembach@mesi.ru

The approach to the creation of modern intelligent tutoring systems based on reusable components - learning objects and educational services. Presents the architecture of intelligent tutoring systems with reusable components based on the ontological and multi-agent organization information educational space. Described intelligent technology training using different classes of educational services.

INTRODUCTION

Competence-based approach in continuous education requires creation of a wide range of educational programs. This can be addressed by introducing adaptability in the learning process. The main problem is the planning and implementation of student's personal educational path based on a controlled learning process. Individual educational trajectory [Zenkina, etc., 2014] can be defined as a way

of leveraging person's learning potential. It stimulates one's active commitment to knowledge and skill improvement and consists of reusable learning objects stored in a shared repository. Software services are used for sharing of learning objects among teachers and students. They may be thought of as special kinds of objects in the Central repository of IES educational services [Telnov, 2014b].

The purpose of this paper is the formalization of the intelligent technologies for-the formation of individual learning paths based on reusable components (learning objects and educational services).

MAIN PART

Knowledge base of information-educational space requires reusable components [Golenkov, etc., 2013] for machine learning, IES knowledge base content management, typical training situations recognition, knowledge base validation, etc. [Trembach, 2013], and a set of domain ontologies, competencies, learning objects and educational services [Telnov, 2014a].

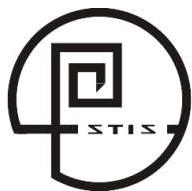
A learning object is "any entity, digital or non-digital, which can be used repeatedly, which you can link with the use of appropriate learning technologies". Learning objects allow the creation of independent components of educational content. IMS and SCORM provide detailed standards and guidelines for educational facilities, e.g., Learning Object Metadata. Using metadata of learning objects is a Central part of IOS, which can take into account individual training requirements. It can be based on Dublin core metadata [DC, 2013] and is a promising mean of creating metadata for a wide class of learning objects.

The following components are needed to solve problems in agent-oriented intelligent learning system: the student interface, the assessment module, the individual learning trajectory generator subsystem, knowledge base management system, individual learning environment, learning objects repository.

CONCLUSION

The intelligent technology of formation of the individual trajectory learning path in the integrated information educational space based on service architecture greatly reduce the cost of its creation and to improve the relevance of scientific and educational content through the integration of the efforts of the participants of the educational process based on the use of intelligent software services. Thus, the use of information and educational space in the formation of professional competences of students provides comprehensive educational content coverage and increases flexibility and adequacy of educational programs and the necessary educational content in accordance with the specific training profiles.

Work is financially supported through RFFR (project No. 13-07-00917a, project No. 14-07-00880a).



OSTIS-2015

(Open Semantic Technologies for Intelligent Systems)

УДК 007.52:004.81

ОПРЕДЕЛЕНИЕ УРОВНЯ УСВОЕННЫХ ЗНАНИЙ ПО ОБУЧАЮЩЕМУ КУРСУ, ПРЕДСТАВЛЕННОМУ СЕМАНТИЧЕСКОЙ СЕТЬЮ

Янковская А.Е. *, Шурыгин Ю.А. **, Ямшанов А.В. **, Кривдюк Н.М. **

** Томский государственный архитектурно-строительный университет,
Томский государственный университет,
Томский государственный университет систем управления и радиоэлектроники,
Сибирский государственный медицинский университет,
г. Томск, Россия,
ayyankov@gmail.com*

*** Томский государственный университет систем управления и радиоэлектроники,
г. Томск, Россия
kcup@kcup.tusur.ru
yav@keva.tusur.ru
knm@kcup.tusur.ru*

Рассматривается новый подход к построению диагностических тестов по учебной дисциплине, представленной в виде формализованных знаний, заданных семантической сетью. Также предлагается новый подход для разработки учебного курса в виде удобном как для естественного представления, так и для формализации знаний. Описывается архитектура интеллектуальной обучающе-тестирующей системы, основанной на предлагаемом подходе. Обсуждаются проблемы и преимущества применения рассматриваемых подходов. Кроме того, предлагается новый подход к оценке уровня усвоенных знаний и инструментарий для визуализации этой оценки, построенный на основе 3-симплекса. Рассматривается отображение результатов тестирования и динамики обучения респондента с применением 3-симплекса, приводятся иллюстрирующие примеры. Приводятся преимущества применения когнитивного средства 3-симплекса при создании интеллектуальной обучающе-тестирующей системы и перспективы дальнейшего развития модуля визуализации.

Ключевые слова: интеллектуальная обучающе-тестирующая система, семантическая сеть, смешанные диагностические тесты, средства когнитивной графики, 3-симплекс, программный модуль.

ВВЕДЕНИЕ

В 70-х годах XX столетия начался переход к использованию вычислительных систем. В настоящее время переход стал массовым в различных областях применения. Этот процесс затронул и сферу образования. Массовая компьютеризация повлияла на то, как мы обучаем и как мы учим. Всё больше учебных дисциплин и образовательных предметов используют информационные технологии с применением всемирной паутины. Всего за несколько лет MOOCs (Massive Open Online Courses - массовые открытые онлайн-курсы) приобрели миллионную аудиторию студентов со всего мира [Koller, 2012, Костюк, 2014]. Появилось множество новых подходов в образовании к обучению, среди которых весьма

перспективным подходом является смешанное образование и обучение (Blended Education and Training) [Bliuc, 2007, Yankovskaya, 2013]. Кроме того, в настоящее время имеется множество других индикаторов, указывающих на существенные изменения в традиционной системе образования. Использование новых информационных технологий не только позволяет «оцифровывать» традиционные образовательные процессы и практики, но и создавать принципиально новые подходы – учить по-другому. Умение использовать новые возможности информационных технологий становится востребованным для современного преподавателя и серьезным конкурентным преимуществом для образовательных учреждений в борьбе за выживание и таланты.

При этом в борьбе за таланты приходится играть в рамках установленных доступных ресурсов. В связи с этим, весьма актуален вопрос достижения консенсуса высокого качества образования и оптимизацией временных и стоимостных затрат, используемых на это образование. Многие из существующих попыток ухода от индивидуального и группового обучения между обучающимися и преподавателем в сторону конвейерного и доступного обучения вызывают отторжение, как у преподавателей, так и у обучаемых (респондентов).

Значимыми проблемами, влияющими на это отторжение, являются:

1. Отсутствие или недостаток личного контакта между разработчиком курса, преподавателем и непосредственно обучаемым.

Следует отметить, что достаточно часто преподаватель сам является разработчиком курса. Однако мы придерживаемся более общей модели, когда разработчик и преподаватель являются разными лицами.

2. Отсутствие удобного средства контроля за обучением.

Почти все подходы по оценке качества обучения, используемые в подобных системах (например, тесты) [Титов, 2010], имеют ряд недостатков: носят псевдовероятностный характер и напоминают скорее лотерею, чем проверку знаний, поскольку проверяют знания лишь на уровне определений, а не навыков оперировать ими; позволяют легко манипулировать системой проверки и получать лучшие оценки результатов обучения, чем реальный уровень знания.

Современный уровень развития информационных и коммуникационных технологий позволяет создать значительный фундамент для модернизации существующих образовательных систем, начиная со школ до ВУЗов и центров профессиональной подготовки и переподготовки.

Одной из отличительных черт современного этапа развития образовательных систем является поиск и применение педагогами-исследователями эффективных способов использования достижений в различных проблемных областях: информатики, кибернетики, синергетики, теории искусственного интеллекта, распознавания образов и других. Для применения большинства из уже имеющихся достижений необходима строгая формализация знаний проблемной области, лежащих в основе учебной дисциплины.

Ниже опишем проблемы и способы их решений, связанные со следующими задачами:

1. Тестирование, как способ проверки степени усвоения знаний.

2. Методологический аппарат для построения базы данных и знаний по диагностическим тестам и вспомогательные механизмы, облегчающие процесс тестирования.

3. Оценка результатов прохождения тестов и их визуальная интерпретация с применением графических, включая когнитивные, средств визуализации.

В настоящее время, тестирование знаний респондентов проводится, в основном, на основе тестов, для которых не важен порядок вопросов предъявляемых тестом (безусловных тестов) с использованием среды Интернет в режиме off-line или on-line. Как правило, в обучающе-тестирующих системах весьма слабо развита объяснительная компонента результатов тестирования, а также организация диалога преподавателя с респондентом. Кроме того, проверка тестирования по конечному результату (ответу), тем более на основе меню и с использованием безусловных тестов, не всегда возможна, но всегда примитивна [Янковская, 2011].

1. Основные понятия и определения

Приведем основные понятия и определения используемые далее.

Респондент – человек, проходящий обучение, например, студент ВУЗа.

Преподаватель – человек, проводящий или сопровождающий процесс обучения респондента.

Разработчик – человек, разрабатывающий курс обучения и/или курс тестирования по рассматриваемой дисциплине.

Обычный текст (ОТ) – текст в общепринятом понимании, например, данная статья.

Текст со специальной разметкой (ТСР) – обычный текст с вкраплениями служебной информации, наиболее ярким примером которого является вики-разметка.

«Обычная оценка» (ОЦ) – оценка результатов тестирования респондента в общепринятом понимании, например, количество баллов по 100 бальной или 5-ти бальной шкале.

Карта действий респондента (КДР) – максимально подробный журнал действий респондента, получаемая и используемая внутри интеллектуальной обучающее-тестирующей системы.

Структурированные знания (СЗ) – представление учебного курса в виде, пригодном для анализа интеллектуальной обучающее-тестирующей системой.

Проверяющий тест указывает на факт наличия ошибки, но не указывает на саму ошибку.

Под образом понимается совокупность различных результатов тестирования, одинаково оцениваемые по выбранной шкале (например, 5-и бальная, 100 бальная, 10-и бальная).

Диагностический тест (ДТ) – совокупность признаков, различающих любые пары объектов, принадлежащих разным образам.

Безусловный диагностический тест – тест, в котором не важен порядок вопросов предъявляемых тестом.

Условный диагностический тест – тест, в котором каждый последующий вопрос зависит от результатов, полученных на предыдущих вопросах.

Смешанный диагностический тест, представляет оптимальное сочетание безусловных и условных составляющих [Yankovskaya, 1996].

2. Архитектура интеллектуальной обучающе-тестирующей системы

Архитектура интеллектуальной обучающе-тестирующей системы не создается с нуля, а является продолжением исследований, приведенной в публикации [Янковская, 2014]. Ниже на рис. 1 представлена модифицированная архитектура системы.

По сравнению с предложенной ранее архитектурой, изменён подход к созданию курса обучения/тестирования. Если ранее создание учебного курса и создание курса тестирования были несвязанными задачами, то в модифицированной структуре эти задачи объединены. Идея объединения состоит в представлении учебной дисциплины в виде достаточно близком к обычному тексту, но в то же время удобном для формализации зависимостей между знаниями. Для этого предлагается формат представления текста со специальной разметкой и специальный инструментарий, который предназначен для того чтобы:

1. Строить сетевую модель знаний между терминами, излагаемыми в рамках учебной дисциплины.
2. Проверять корректность учебного курса по изучаемой дисциплине для построенной модели.
3. Сопровождать разработку курса обучения, показывая потенциально непроработанные или изолированные от другого материала области знаний.

4. Сопровождать разработку курса тестирования, осуществляя автоматические проверки и показывая области учебной дисциплины, еще не покрытые курсом обучения.

5. Удалять всю служебную разметку и преобразовать текст со специальной разметкой к обычному тексту.

Подробное описание модели представления обучающего курса приводится в следующем разделе.

Представим последовательность обучения (тестирования) респондента по модифицированной архитектуре интеллектуальной обучающе-тестирующей системы:

1. Респондент проходит обучение по интересующей его дисциплине или ее части. Содержательно учебный курс по соответствующей дисциплине может быть представлен обычным текстом, но при этом иметь дополнительное интерактивное и мультимедийное содержание.

2. На основе пройденного материала такого, для которого не важна последовательность задаваемых вопросов, формируется безусловная составляющая теста. Респондент отвечает на содержащиеся в тесте вопросы из безусловной составляющей теста.

3. Осуществляется переход к условной составляющей теста. При этом после каждого ответа на вопрос, система определяет, какой вопрос задается следующим из условной составляющей теста.

Условная составляющая теста содержит: 1) вопросы другого уровня сложности, относящиеся к знаниям, уже проверенным пройденной частью теста; 2) вопросы, проверяющие насколько хорошо респондент может использовать новые знания, объединяя уже проверенные.

4. Во время прохождения теста в базу данных записываются все шаги, которые совершает респондент. Такая детализация не является необходимой для вычисления оценки респондента, но может быть весьма полезна для исследователя или преподавателя. На основе этой последовательности шагов и решений респондента формируется карта действий респондента (КДР).



Рисунок 1 – Архитектура обучающе-тестирующей системы

5. После прохождения респондентом всех вопросов КДР проецируется в набор оценочных коэффициентов, определяющих насколько хорошо респондент справляется с различными заданиями на основе следующих способностей: 1) запоминание и воспроизведение материала в неизменённом виде; 2) воспроизведение материала в изменённом виде; 3) извлечение новых знаний на основе изученного материала; 4) решение задач и т.д.

После завершения обучения и тестирования респонденту показывается сеть проверенных знаний, интерпретация КДР и вычисленные коэффициенты в виде «обычной оценки». Далее, если у респондента есть фрагмент учебного курса, который он не изучил или не смог успешно пройти тестирование, то он может продолжить обучение, вернувшись к пункту 1.

6. При успешном завершении процедуры тестирования по всему учебному курсу считается, что учебная дисциплина успешно изучена и респондент усвоил весь предлагаемый учебный курс (с возможными допущениями). При этом результатом прохождения всего учебного курса будет оценка, отражающая усредненные показатели за все этапы тестирования, пройденные респондентом.

Поскольку общепринятой в России является 5-ти бальная шкала оценки результатов, которая весьма груба, то для лучшего отражения уровня знаний и навыков респондента предлагается использовать вышеописанный набор оценочных коэффициентов, который определяет, насколько хорошо респондент справляется с различными классами задач. Так как такая оценка является более сложной для интерпретации, чем общепринятая, то целесообразно применение дополнительных инструментов, например, когнитивных графических средств на базе 3-симплекса. О применении данного инструмента излагается далее в разделе "Применение 3-симплекса для интерпретации результатов обучения".

3. Модель представления учебной дисциплины

Существует множество различных способов представления знаний в интеллектуальных системах, из которых наиболее часто используются способы представления знаний, приведенные ниже [Поспелов, 1990]:

1. Продукционная, основанная на представлении знаний в виде правил "если-то".
2. Семантическая, основанная на представлении знаний в виде семантической сети, узлы которой соответствуют понятиям и объектам, а дуги – связям между объектами.
3. Фреймовая, основанная на представлении знаний в виде структур данных, для некоторого концептуального объекта.
4. Логическая, основанная на представлении знаний в виде совокупности формул, определенных в конечном алфавите.

При этом для представления учебной дисциплины некоторые модели являются более удобными, нежели другие и различные авторы имеют разные предпочтения по модели, наиболее подходящей для представления учебного курса.

Тем не менее, формализация уже имеющегося учебного курса в любой из упомянутых выше моделей является весьма трудоемкой задачей, сопоставимой с написанием нового учебного курса. Кроме того, разработка нового курса также потребует значительных затрат, для того чтобы представить информацию как в виде удобном для изучения респондентом (в виде обычного текста), так и в виде удобном для анализа интеллектуальной системой (в виде структурированных знаний).

В качестве консенсуса между достаточностью структурированности знаний и простотой их поддержки предлагается следующий подход:

1. Весь учебный курс хранится в виде текста со специальной разметкой (ТСП). Пример такого текста изображен на рис. 2.
2. При необходимости получения обычного текста, из исходного текста удаляется вся служебная разметка и остается только обычный текст. Пример такого текста изображен на рисунке 4.

Подсистема управления процессами

=====

{@Процесс} {!это} {последовательная программа} располагающая {\программным\ окружением}, достаточным для своего выполнения.

{@Программное окружение} {!образуют} {системные программы} и {системные структуры}, обслуживающие данную программу.

{@Диспетчеризация процессов} это процесс {!распределения} {квант времени} между {процессами}, существующими в системе и находящимися в состоянии {\состояние процесса\ готов}.

{@Важнейшими аппаратными ресурсами ВС} {!являются} {ЦП} и {ОП}.

{@Управление |ЦП| и |ОП|} {!осуществляется} как программно - {подпрограммами ядра ОС|=программное управление ЦП и ОП|}, так и аппаратно - {аппаратурой ЦП|=аппаратное управление ЦП и ОП|}.

При этом {@программное управление ЦП и ОП} {!использует} и {!дополняет} соответствующее {аппаратное управление \ЦП и ОП\}.

Рисунок 2 – Пример исходного текста со специальной разметкой

При необходимости получения структурированных знаний, из исходного текста удаляются все фрагменты, не защищенные служебной разметкой, все идентификаторы приводятся к начальной форме, строится семантическая модель знаний в виде семантической сети. Пример такой сети изображен на рис. 5.

Как уже говорилось, текст со специальной разметкой представляет собой обычный текст с вкраплениями служебной информации. Весь текст разбивается на утверждения, в каждом из которых есть:

1. Объект – терм, который определяется в данном утверждении. В служебной разметке он выглядит как “{@Объект}”.

2. Один или несколько субъектов. Это термы, которые участвуют в утверждении, но не является объектом. В служебной разметке они выглядят как “{Субъект}”.

3. Один или несколько основных глаголов, определяющие связь между объектом и субъектом (субъектами). В служебной разметке они выглядят как “{!Определяет}”.

Вид семантической сети определяется следующими правилами:

1. Она представляет собой ориентированный граф со связями без имен и весов.

2. Вершины могут быть одним из двух типов: 1) терм, отражающий какое-то понятие из предметной области; 2) связь, показывающая отношение между одним или несколькими термами.

3. Вершины, представляющие собой термы, имеют вид прямоугольника. Будем использовать для их обозначения понятие "вершина-терм".

4. Вершины, представляющие собой связи, имеют вид прямоугольника со скругленными углами. Будем использовать их для обозначения понятие "вершина-связь".

5. Различные вершины-термы не могут быть соединены между собой напрямую без вершин-связей.

б. Связь, исходящая из вершины-связи к вершине-терму, показывает, что данный терм определяется данной связью (является объектом). Связь, входящая из вершин-термов в вершину-связь, показывает, что данные термы, учувствуют в определении объекта (являются субъектами).

После формирования данной сети, необходимо покрыть все связи сети набором тестов. При этом тестовый вопрос может быть задан как к одной связи, так и к нескольким. На рис. 3 показаны примеры отображения таких вопросов, обозначенных (1) и (2) соответственно.

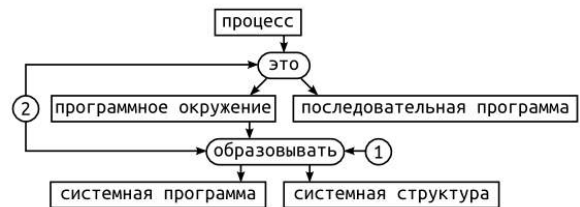


Рисунок 3 – Пример покрытия сети тестами

Примерами подобных вопросов, например, могут быть:

1. Вопрос с несколькими правильными ответами: "Что образовывает программное окружение процесса?" и ответами: 1) операционная система; 2) системная программа; 3) пользователь; 4) интернет; 5) системная структура.

2. Вопрос с одним правильным ответом: "Без чего не может существовать процесс?" и ответами: 1) пользователя; 2) программного shell'a; 3) устройства ввода-вывода; 4) системной структуры; 5) вытесняющего планировщика задач.

С другой стороны одна связь может быть покрыта как одним вопросом, так и несколькими. Хотя это и противоречит составлению теста с точки зрения минимально достаточного количества вопросов, это может иметь смысл, поскольку сокращает вероятность жульничества респондентом и переводит диалог между респондентом и обучающей системой на уровень близкий к естественному.

Подсистема управления процессами

Процесс - последовательная программа располагающая окружением, достаточным для своего выполнения. Окружение программы образуют системные программы и системные структуры, обслуживающие данную программу. Диспетчеризация процессов это процесс распределения квант времени между процессами, существующими в системе и находящимися в состоянии готов.

Важнейшими аппаратными ресурсами ВС являются ЦП и ОП. Управление ЦП и ОП осуществляется как программно - подпрограммами ядра ОС, так и аппаратно - аппаратурой ЦП. При этом программное управление ЦП и ОП использует и дополняет соответствующее аппаратное управление.

Рисунок 4 – Пример выходного обычного текста

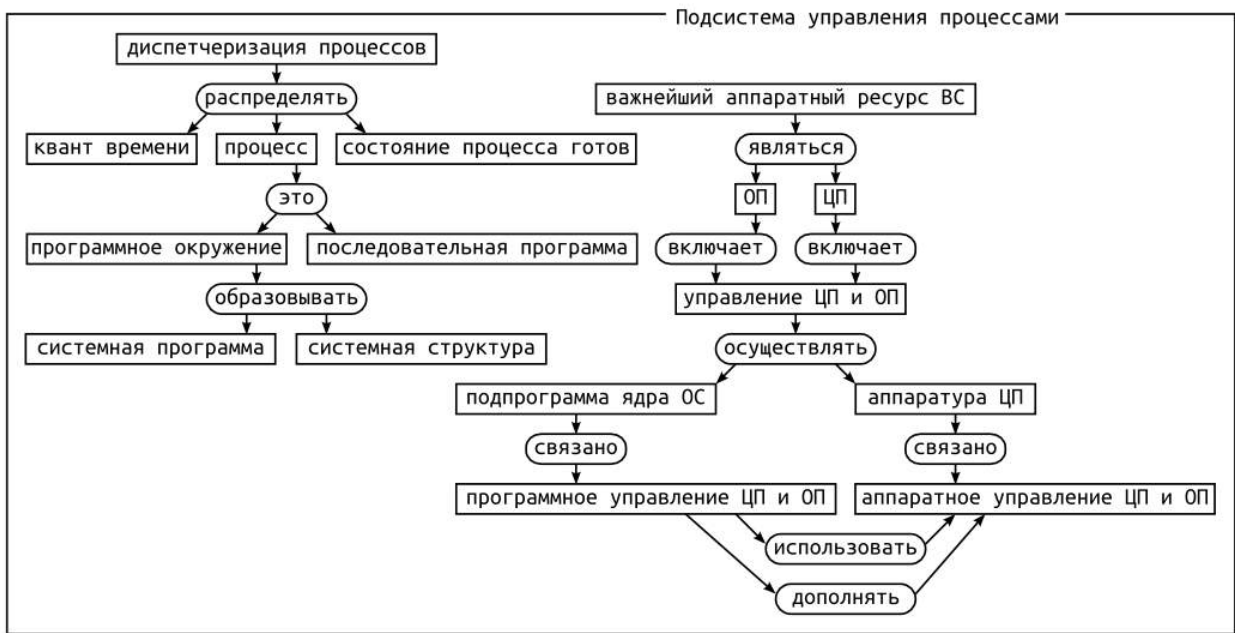


Рисунок 5 – Пример выходной сети знаний

4. Математические основы отображения исследуемого объекта в 3-симплексе

В основе принятия и обоснования решений в обучающе-тестирующих системах лежит следующая теорема, предложенная в публикациях [Янковская, 2000, Yankovskaya, 2004].

Теорема: для любого набора одновременно не равных нулю чисел a_1, a_2, \dots, a_{n+1} , где n – размерность правильного симплекса, можно найти одну и только одну такую точку, что $h_1:h_2:\dots:h_{n+1} = a_1:a_2:\dots:a_{n+1}$, где h_i ($i \in \{1, 2, \dots, n+1\}$) – расстояние этой точки до i -ой грани [Янковская, 1991, Кондратенко, 1992].

При $n=3$ коэффициент a_i ($i \in \{1, 2, 3, 4\}$) представляет собой степень условной близости исследуемого объекта к i -му образу.

Поскольку 3-симплекс обладает свойством постоянства суммы расстояний (h) из любой точки до его граней и свойством сохранения отношений $h_1/a_1=h_2/a_2=h_3/a_3=h_4/a_4$, то расстояния h_1, h_2, h_3, h_4 вычисляются на основе коэффициентов a_i ($i \in \{1, 2, 3, 4\}$) и операции нормализации из следующих соотношений

$$\left\{ \begin{array}{l} h = \sum_{i=1}^4 h_i \\ h = \alpha \sum_{i=1}^4 a_i \end{array} \right. , \quad (2)$$

$$\frac{h_1}{a_1} = \frac{h_2}{a_2} = \frac{h_3}{a_3} = \frac{h_4}{a_4}$$

по формуле

$$h_i = \frac{h \cdot a_i}{\sum_{i=1}^4 a_i}, \text{ при } i \in \{1, 2, \dots, 4\}, \quad (3)$$

где α – коэффициент масштабирования.

Данная теорема использовалась в более чем тридцати прикладных интеллектуальных системах и в трех инструментальных средствах выявления различного рода закономерностей и принятия диагностических, организационно-управленческих и классификационных решений в целях принятия и обоснования принимаемых решений.

Далее при изложении будем использовать термин 3-симплекс, опустив слово правильный.

5. Применение 3-симплекса для интерпретации результатов обучения

Поскольку в ходе тестирования происходит достаточно подробный сбор информации о процессе прохождения теста, то целесообразно применение дополнительных инструментов, которые облегчают преподавателю задачу анализа этой информации. Задачу анализа информации удобно декомпозировать как минимум на две подзадачи:

1. Разработка инструментария для анализа процесса прохождения теста, позволяющего упростить интерпретацию карты действий респондента (КДР).
2. Разработка инструментария для анализа результата прохождения теста, упрощающего интерпретацию набора оценочных коэффициентов для различных классов задач.

В связи с рамками публикации будет рассмотрен только инструментарий для упрощения анализа

оценочных коэффициентов, а вопросы, связанные с инструментарием для упрощения анализа КДР, будут рассмотрены в следующих публикациях.

Для визуализации и обоснования динамики результатов прохождения теста также как и ранее предлагается использовать 3-симплекс, но с небольшими изменениями.

Переходя от одномерной оценки (например, 5-ти бальной), отражающей величину усвоенного материала к многомерной оценке, основанной на наборе оценочных коэффициентов, становится нецелесообразным использование 3-симплекса, отображающего образы 5-и бальной шкалы оценки результатов тестирования: неудовлетворительно (2), удовлетворительно (3), хорошо (4), отлично (5).

Один из 4-х образов, сопоставлен грани 3-симплекса, соответствует образу респондента, который не проявляет успеха ни в одной из областей знаний по набору оценочных коэффициентов. Это необходимо для того, чтобы на 3-симплексе было возможным осуществить разделение объектов с одинаковыми отношениями расстояний от объекта до граней 3-симплекса, но с различными абсолютными показателями оценочных коэффициентов. Для примера, респондент с показателями 2-1-2-2 и респондент с показателями 6-3-6-6, без введения дополнительного образа, будут представлены в виде одной точки, но обладать отличающимися в 3 раза абсолютными показателями.

Приведем пример использования 3-симплекса для области разработки программных систем.

В рамках разработки клиент-серверной программной системы с мультимедийными возможностями целесообразен перевод набора оценочных коэффициентов в следующие показатели: 1) решение задач требующих большой сосредоточенности; 2) решение нетривиальных задач; 3) быстрая обучаемость и знание большого количества технологий.

На 3-симплексе вышеизложенные показатели отображены, соответственно, оранжевым, желтым и зеленым цветами (рис.6).

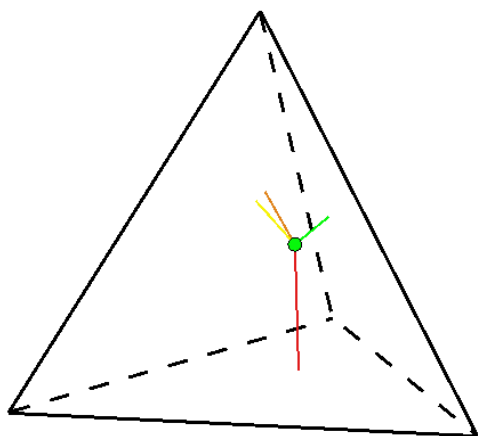


Рисунок 6 – Пример использования 3-симплекса

Исходя из вышеизложенного по изображенному на рис. 6 3-симплексу можно утверждать, что мы рассматриваем специалиста немного выше среднего уровня с примерно одинаковым распределением способностей, но небольшим перевесом в сторону увлечения изучения и применения различных технологий (3-го показателя).

Имея такую информацию о каждом исполнителе, можно эффективно планировать ход разработки программы.

Заключение

В статье отражена актуальность проводимых исследований и уровень развития современных интеллектуальных обучающе-тестирующих систем. Показана целесообразность внедрения смешанного обучения с использованием смешанных диагностических тестов, представляющих собой оптимальное сочетание безусловной и условной составляющих.

Кратко описаны проблемы и способы их решения для задач: проверки степени усвоения знаний; создания методологического аппарата, а также базы данных и знаний; оценки результатов прохождения тестов как по фрагменту учебного курса, так и по всему учебному курсу.

Приведена модифицированная архитектура интеллектуальной обучающе-тестирующей системы. Дано описание модели представления учебного курса в интеллектуальной обучающе-тестирующей системе как для респондента в виде обычного текста, так и для системы в виде текста со специальной разметкой, который может быть представлен в виде семантической сети.

Указаны способы формирования вопросов для тестирования по семантической сети с указанием примеров.

Представлены ранее созданные математические основы отображения исследуемого объекта в 3-симплексе. Приводится применение 3-симплекса для интерпретации результатов обучения в виде удобном для анализа показателей обучения или качества составления учебного курса.

Описаны оценочные коэффициенты, отражающие используемые респондентом способности для решения различного класса задач:

- запоминание и воспроизведение материала в неизменённом виде;
- воспроизведение материала в изменённом виде;
- извлечение новых знаний на основе изученного материала;
- решение задач и т.д.

Описана идея системы оценок, основанная на сопоставлении существенных показателей по выбранной дисциплине, представленной набором оценочных коэффициентов.

Дальнейшие исследования направлены на программную реализацию описанного инструментария и развитие предложенного подхода.

Также дальнейшие исследования направлены на создание инструментария для визуализации и анализа карты действий респондента, по изучению выбранной учебной дисциплины, представленной в интеллектуальной обучающее-тестирующей системе.

Работа выполнена при финансовой поддержке грантов РФФИ (проекты № 13-07-98037р_сибирь-а, № 13-07-00373а и № 14-07-00673), а также частично гранта РГНФ (проект № 13-06-00709).

Библиографический список

[Кондратенко, 1992] Система визуализации TRIANG для обоснования принятия решений с использованием когнитивной графики / С.В. Кондратенко, А.Е. Янковская // Тезисы докладов III конференции по Искусственному интеллекту. Том, Тверь, 1992, 152-155

[Костюк, 2014] Массовые открытые онлайн курсы – современная концепция в образовании и обучении / Костюк Ю.Л., Левин И.С., Фукс А.Л., Фукс И.Л., Янковская А.Е. // Вестник Томского государственного университета. – 2014. – № 1 (26). – С. 89-98.

[Поспелов, 1990] Данные и знания / Поспелов Д.А. // Искусственный интеллект. Справочник в 3-х кн. Кн. 2. Модели и методы: Справочник/ под ред. Д. А. Поспелова. Будущее искусственного интеллекта. 1990, 7-13.

[Титов, 2010] Информатика: экспресс-подготовка к интернет-тестированию: учебное пособие / В.М. Титов, О.Н. Рубальская, О.В. Маленкова и др.; под ред. О.Н. Рубальской - М.: Финансы и статистика; ИНФРА-М, 2010, -240 с.

[Янковская, 1991] Преобразование пространства признаков в пространство образов на базе логико-комбинаторных методов и свойств некоторых геометрических фигур / А.Е. Янковская // Распознавание образов и анализ изображений: новые информационные технологии. Тез. докладов I Всесоюзной конференции. Часть II. – Минск, 1991. – С. 178-181.

[Янковская, 2000] Отражение образного мышления специалиста в интеллектуальной распознающей системе патогенеза заболевания / Янковская А.Е., Тетенов Ф.Ф., Черногорюк Г.Э // Компьютерная хроника.–2000. – №6. – С. 77-92.

[Янковская, 2011] Янковская А.Е. Смешанные диагностические тесты – новая парадигма создания интеллектуальных обучающих и контролирующих систем // Материалы Всероссийской научно-практической конференции "Новое качество образования в новых условиях". – Томск: ТОИПКРО, 2011. – Т.1.– с. 195-203. (ISBN 978-5-903029-30-3).

[Янковская, 2014] Средства когнитивной графики в интеллектуальных обучающе-тестирующих системах / Янковская А.Е., Ямшанов А.В., Кривдюк Н.М. // Открытые семантические технологии проектирования интеллектуальных систем Open Semantic Technologies for Intelligent Systems (OSTIS-2014), Материалы IV междунар. науч.-техн. конф. – Минск : БГУИР, с 303-308, 2014.

[Bliuc, 2007] Research Focus and Methodological Choices in Studies into Students' Experiences of Blended Learning in Higher Education / A. M. Bliuc, P. Goodyear, R.A. Ellis // Internet and Higher Education, 10, (2007), 231-244.

[Koller, 2012] What we're learning from online education / Koller D. // TED-talks, august 2012. http://www.ted.com/talks/daphne_koller_what_we_re_learning_from_online_education.html

[Yankovskaya, 1996] Design of Optimal Mixed Diagnostic Test With Reference to the Problems of Evolutionary Computation / Yankovskaya A. // Proc. of the First International Conference on Evolutionary Computation and Its Applications, Moscow, EVCA'96, 1996. P. 292-297.

[Yankovskaya, 2004] Optimization of a set of tests selection satisfying the criteria prescribed / A.E. Yankovskaya, V.I. Mozheiko

// 7th International Conference on Pattern Recognition and Image Analysis: New Information Technologies (PRIA-7-2004). Conference Proceedings, Vol. I, St. Petersburg: SPbETU 2004, 145-148

[Yankovskaya, 2013] Application Mixed Diagnostic Tests in Blended Education and Training / Yankovskaya A.E., Semenov M.E. // Proceedings of the IASTED International Conference Web-based Education (WBE 2013) February 13 - 15, 2013 Innsbruck, Austria. – 2013. pp. 935-939.

APPLICATION OF COGNITIVE GRAPHICS TOOLS BASED ON THE 3-SIMPLEX IN INTELLIGENT TRAINING-TESTING SYSTEMS

Yankovskaya A.E. *, Shurygin Y.A. **, Yamshanov A.V. **, Krivdyuk N.M. **

* *Tomsk State University of Architecture and Building
National Research Tomsk State University
Tomsk State University of Control Systems and Radioelectronics
Siberian State Medical University
Tomsk, Russia
ayyankov@gmail.com*

** *Tomsk State University of Control Systems and Radioelectronics
Tomsk, Russia
kcup@kcup.tusur.ru
yav@keva.tusur.ru
knm@kcup.tusur.ru*

In the paper, a new approach to construct diagnostic tests for a learning course, represented in structured knowledge form (semantic network), is given. Also, a new approach to develop a learning course in representation, which is useful for usual examination and for knowledge formalization, is proposed. The architecture of intellectual training-testing system based on these approaches is presented. Then, we discuss the advantages and problems of applying these approaches. Besides, the method to estimate numerical rating of learned lessons and special tool for solution this task based on the 3-simplex are suggested. The displaying results of testing and the dynamics of learning with using the 3-simplex are considered. In conclusion, we discuss the advantages of using the 3-simplex in creating intelligent training-testing systems and the prospects for further development of the visualization module.



OSTIS-2015

(Open Semantic Technologies for Intelligent Systems)

УДК 004.853

WOLFRAM MATHEMATICA - СРЕДСТВА И ТЕХНОЛОГИИ РАЗРАБОТКИ ИНТЕЛЛЕКТУАЛЬНЫХ ОБУЧАЮЩИХ СИСТЕМ

Таранчук В.Б.

Белорусский государственный университет,
г. Минск, Республика Беларусь

taranchuk@bsu.by

В работе описаны новые возможности и рекомендации применения технологий Wolfram Research для создания и сопровождения интеллектуальных обучающих систем. Приведены примеры из практики подготовки учебных материалов дисциплины «Компьютерная графика».

Ключевые слова: интерактивные образовательные ресурсы; система компьютерной алгебры *Mathematica*; формат вычисляемых документов; компьютерная графика.

Введение

Важным направлением развития современного образования является повышение эффективности использования информационных технологий. В настоящее время аппаратное и программное обеспечение компьютера предоставляют разные возможности создания и использования электронных документов с компонентами интеллекта, динамической интерактивности. Такие документы имеют ряд преимуществ перед печатными изданиями. Актуальной является задача определения требований к содержанию электронных документов, способам их подготовки, типовым правилам визуализации информации, что в свою очередь предполагает решение ряда технических вопросов.

В настоящей работе описаны основные возможности и рекомендации применения технологий компании Wolfram Research, в частности, системы компьютерной алгебры *Mathematica* и формата вычисляемых документов CDF для создания и сопровождения интеллектуальных обучающих систем; приведены примеры из практики подготовки учебных материалов дисциплины «Компьютерная графика».

1. Базовый инструментарий

1.1. О системе *Mathematica*

В середине XX века на стыке математики и информатики возникло и интенсивно развивается фундаментальное научное направление компьютерная алгебра – наука об эффективных

алгоритмах вычислений математических объектов. Направление компьютерная алгебра представлено теорией, технологиями, программными средствами. К прикладным результатам относят разработанные алгоритмы и программное обеспечение для решения с помощью компьютера задач, в которых исходные данные и результаты имеют вид математических выражений, формул. Основным продуктом компьютерной алгебры стали программные системы компьютерной алгебры – СКА (Computer Algebra System, CAS).

Программных комплексов, выполняющих символьные вычисления, достаточно много; систематически выходят обновления и описания возможностей новых версий. С обзором СКА по состоянию на 2008 г. можно ознакомиться в книге [Дьяконов, 2009], текущее состояние и основные функциональные возможности описаны в [Таранчук, 2013]. Большинство СКА не только применимы для исследования различных математических и научно-технических задач, но и содержат все составляющие языков программирования – де факто являются проблемно ориентированными языками программирования высокого уровня. Широкое распространение в настоящее время имеют следующие СКА: *Derive*, *Maxima*, *Axiom*, *Reduce*, *MuPAD*, *Mathcad*. Особое место занимает система компьютерной математики *MATLAB*. Лидерами СКА являются *Mathematica* и *Maple* – мощные системы с собственными ядрами символьных вычислений, оснащенные интеллектуальным пользовательским интерфейсом и обладающие широкими графическими и редакторскими возможностями. Эти две системы по факту являются кроме прочего интерактивными

математическими энциклопедиями, в которых можно изучать описания, постановки задач, методы решения, выполнять упражнения.

Система компьютерной алгебры *Mathematica* компании Wolfram Research является одним из наиболее мощных и широко применяемых интегрированных интеллектуальных программных комплексов мультимедиа-технологии. В системе реализованы и доступны пользователям практически все возможности аналитических преобразований и численных расчётов, поддерживается работа с базами данных, графикой и звуком. *Mathematica* даёт пользователю возможности анализировать, манипулировать, иллюстрировать графиками все функции чистой и прикладной математики. Система обеспечивает расчеты с любой заданной точностью; построение двух- и трёхмерных графиков, их анимацию, формирование геометрических фигур; импорт, обработку, экспорт изображений, аудио и видео ([Таранчук, 2013]). *Mathematica* классифицируется, как платформа для разработки, полностью интегрирующая вычисления в процесс от начала до конца. Отмечаются уникальные возможности системы в использовании средств информационных технологий, научно-методического обеспечения образовательного процесса и научных исследований в высших учебных заведениях.

1.2. Формат вычисляемых документов (CDF)

Начиная с версии 8, пользователи *Mathematica* получили возможность создания интерактивных книг, отчётов, программных приложений в CDF формате [CDF, 2015]. Такие документы с помощью бесплатной программы CDF Player можно свободно распространять и работать с ними, в том числе в виде веб-объектов всех популярных браузеров. CDF документы можно создавать с инструментами интерактивности (меню, кнопками, указателями, бегунками, динамическими локаторами), с возможностями представления результатов в математической нотации, визуализации шагов вычислений и иллюстрирования графиками всех типов (1D, 2D, 3D, анимация), импорта и экспорта результатов во все общепринятые форматы данных и графики. Реакцией на команды пользователя через инструменты интерактивности являются обеспечиваемое использованием встроенной вычислительной подсистемы формирование и обновление контента. В документах формата CDF можно размещать текст, таблицы, изображения, аудио и видео, предусмотрено также использование печатной вёрстки и технических обозначений. Если предварительно необходимо запрограммировать, сгенерировать в *Mathematica*, то можно выполнять аналитические преобразования, вычисления, импорт и экспорт данных, графическую визуализацию; поддерживаются компоновки документа с разбивкой на страницы, со структурной детализацией; режим слайд-шоу, разные способы формирования и просмотра результатов в режиме реального времени. Важно, что формат CDF делает

набор математических выражений семантически точным. В дополнение к качественной верстке, пригодной для публикаций, формулу можно вводить полностью набранной типографским способом и использовать для вычислений, доступно также указание формата вывода результатов: математическая нотация, формат языка программирования. Оформление документа можно контролировать, используя каскадные таблицы стилей. Документ, первоначально созданный в одном стиле, можно преобразовать в множество форм: отчет, статья, учебник, презентация, инфографика или приложение. *Mathematica* предоставляет создателям документов несколько сотен опций для форматирования и стилистического оформления, возможно немедленное обновление стилей динамического и статического контента.

1.3. Проект Wolfram Demonstrations

Компанией Wolfram Research создан и регулярно обновляется систематизированный каталог свободно распространяемых онлайн-интерактивных демонстраций - программных приложений-проектов ([Wolfram Demonstrations Project, 2015]). По состоянию на январь 2015 г. в каталоге размещены и доступны посетителям сайта более 9890 демонстраций по разным разделам науки, техники, жизни. Целями проекта являются: демонстрация возможностей системы *Mathematica*; расширение круга пользователей разработок Wolfram; освоение приёмов программирования в системе для разработки приложений; уменьшение вычислительных затрат для широкой аудитории пользователей. Включённые в коллекцию модули с интерактивным интерфейсом динамически иллюстрируют решения задач, различные процессы и понятия в широком диапазоне областей: математика, естественные науки, техника, экономика и т.д.; охватывают различные уровни знаний от элементарной школьной математики до сложных тем, например, таких как квантовая механика или модели биологических организмов.

Все включаемые в каталог демонстрационные примеры имеют непосредственно связанный с графикой или визуализацией пользовательский интерфейс, который динамически пересчитывается в ответ на такие действия пользователя, как передвижение ползунка, нажатие на кнопку или перетаскивание графического элемента ([Таранчук, 2014]). Каждая демонстрация имеет описание представляемой идеи. Все модули коллекции доступны для скачивания в формате системы *Mathematica* NB и формате вычисляемых документов CDF.

2. Примеры реализаций

Основные компоненты, применяемые средства создания и сопровождения интеллектуальных обучающих систем отметим на примерах подготовки электронных интерактивных учебных материалов дисциплины «Компьютерная графика».

Специфика преподавания этого предмета состоит в том, что в каждой теме изучается не только теоретический, но требуется сопровождающий иллюстративный графический материал. В отдельных темах математическая составляющая достаточно сложная, поэтому важно иметь возможность делать выкладки и преобразования, причём в математической нотации, на персональном компьютере. Наглядность представления материала, возможность конструирования воображаемых моделей по их математическим описаниям - одно из необходимых требований для корректного понимания сути моделей и их описаний. Создание интерактивных, динамических графиков, поясняющих примеров-иллюстраций, обычно, предполагает сложные геометрические расчёты и аналитические преобразования. Подготовка соответствующих программных приложений не только требует специальных навыков, но и очень трудоёмка. Система *Mathematica* предоставляет решение этой проблемы, в частности, через использование программных модулей Wolfram Demonstrations Project.

2.1. О программных модулях, используемых в темах дисциплины.

В процессе преподавания в БГУ на факультете прикладной математики и информатики предмета «Компьютерная графика» используются интерактивные демонстрации (программные приложения-проекты) из коллекций [Wolfram Demonstrations Project, 2015] по следующим темам:

- Цвет в компьютерной графике. Аддитивная, субтрактивная цветовые системы, модель «цветовое пространство», модель «цветовой куб». Интуитивные цветовые модели и их геометрическая интерпретация. Стандартные цветовые системы и преобразования между ними.

- Математические основы машинной графики. Точка, вектор, расстояние на плоскости и в пространстве. Уравнения отрезка, луча в 2D и 3D: параметрические, с направляющим вектором. Нормаль. Расстояние до точки. Функции угла между векторами, лучами, прямыми.

- Преобразования координат. Однородные координаты. Геометрические преобразования в 2D и 3D. Матричное представление преобразований (сдвиг, отражение/симметрия, поворот, масштаб). Задачи поворота вокруг произвольной оси, относительно точки. Композиция 3D преобразований, их коммутативность. Конвейер геометрических преобразований. Проекция, алгоритмы, матрицы проективных преобразований.

- Основы обработки цифровых изображений. Линейные, нелинейные фильтры, примеры, морфологические операторы. Поиск границ на основе градиента, лапласиана.

- Построение реалистичных изображений. Модели освещения в компьютерной графике. Моделирование прозрачности. Построение теней. Текстура. Понятие, примеры воксельной графики.

Отметим несколько типичных первоисточников.

Все перечисленные ниже проекты вызываются (а также можно скачать исходные коды или CDF документы) на сайте [Wolfram Demonstrations Project, 2015], достаточно набрать название при удалении в приведенных названиях пробелов.

Одна из начальных тем в компьютерной графике – цвет и цветовые модели. Из коллекции используются интерактивные модули визуализации различных цветовых моделей, выполнения преобразований между ними, в частности приложения: Colors of the Visible Spectrum; Overlapping Light Colors; Colored Lights; Named Colors; Select, View, and Compare Named Colors; Analogous and Complementary Colors; Newton's Color Wheel; Color Cube; Color Triangles; Color Space; Cartesian Color Coordinate Spaces; RGB and CMYK Colors; RGB Explorer; Orthogonal Views of Named RGB Colors; HSV Colors; HSV Loci in the RGB Color Space; CIE Chromaticity Diagram.

Модули, которые применяются при изучении разделов «Математические основы машинной графики»: Understanding 2D Translation; Understanding 2D Shearing; Understanding 2D Rotation; Understanding 2D Reflection; Understanding 2D Rescaling; 3D graphics modules: Understanding 3D Rotation; Understanding 3D Scaling; Understanding 3D Reflection; Understanding 3D Shearing; Two Models of Projective Geometry; Orthographic Projection of Parallelepipeds; Stereographic Projection of Platonic Solids; Dissection of a Prismatoid into Prisms; Cutoff Parallelepipeds.

Во всех модулях визуализации в NB, CDF документах кроме управления ползунками на панелях (геометрическими параметрами, цветами, прозрачностью выводимых объектов) пользователь может менять масштаб объекта, поворачивать его во всех направлениях. Т.к. доступны исходные коды, студентам предоставляются доработанные приложения, с использованием русскоязычной терминологии, оформления графики, как в базовых учебниках и пособиях; комментарии в модулях переведены на русский язык.

2.2. Технические решения, состав электронных ресурсов на примере одной из тем дисциплины.

Интерактивные электронные ресурсы при преподавании дисциплины «Компьютерная графика» используются на всех этапах: в лекциях, практических занятиях, контролируемой самостоятельной работе, текущем контроле знаний, в итоговой диагностике результатов учебной деятельности, которая выполняется в формате компьютерного тестирования. Приведём и поясним компоненты электронного учебно-методического комплекса на примере изучения темы «Геометрические преобразования в 2D и 3D. Матричное представление, композиция 3D преобразований». На рисунках 1 - 8 представлены скриншоты, иллюстрирующие кадры программного модуля, предоставляемого студентам.

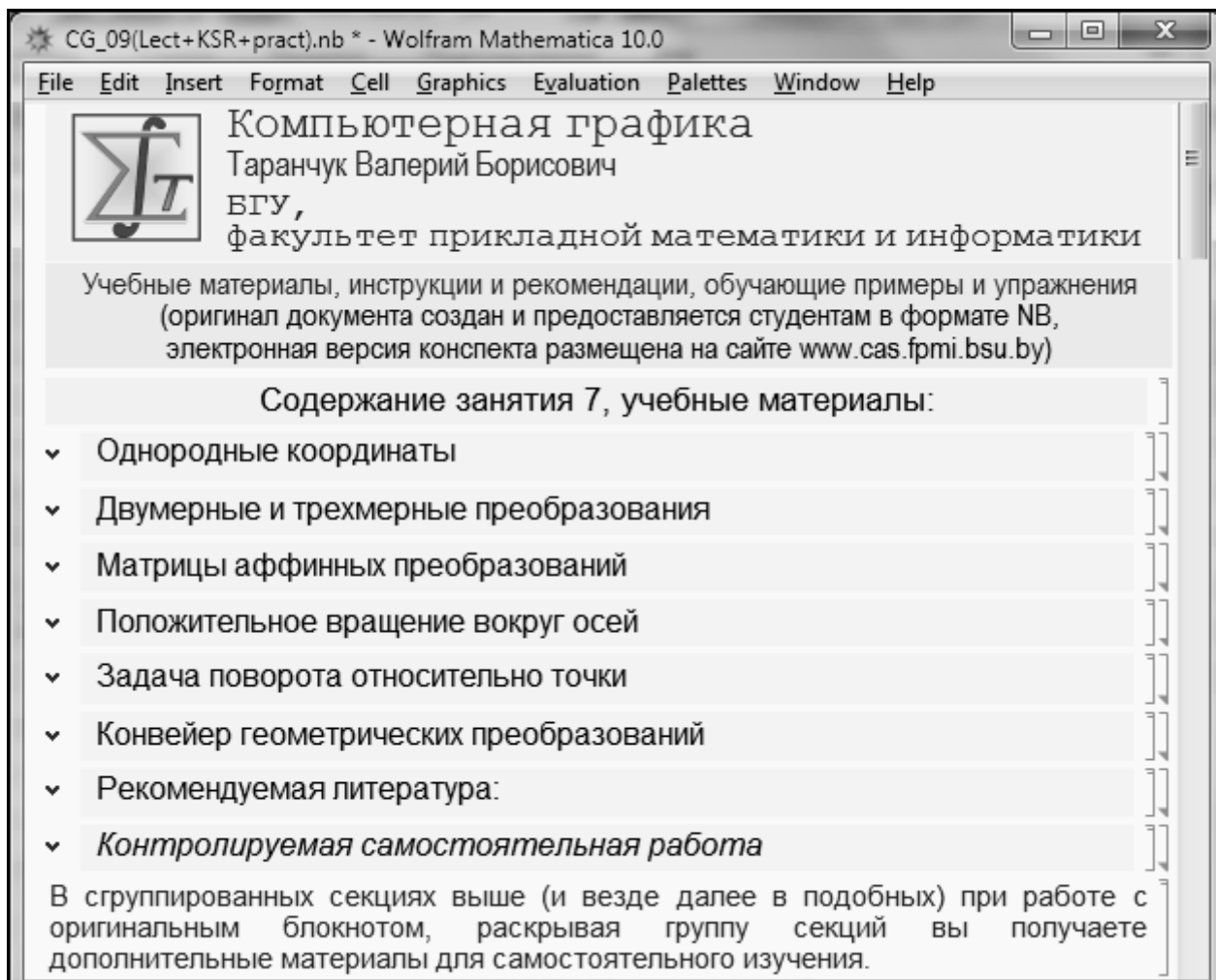


Рисунок 1 – Скриншот главного окна программного модуля

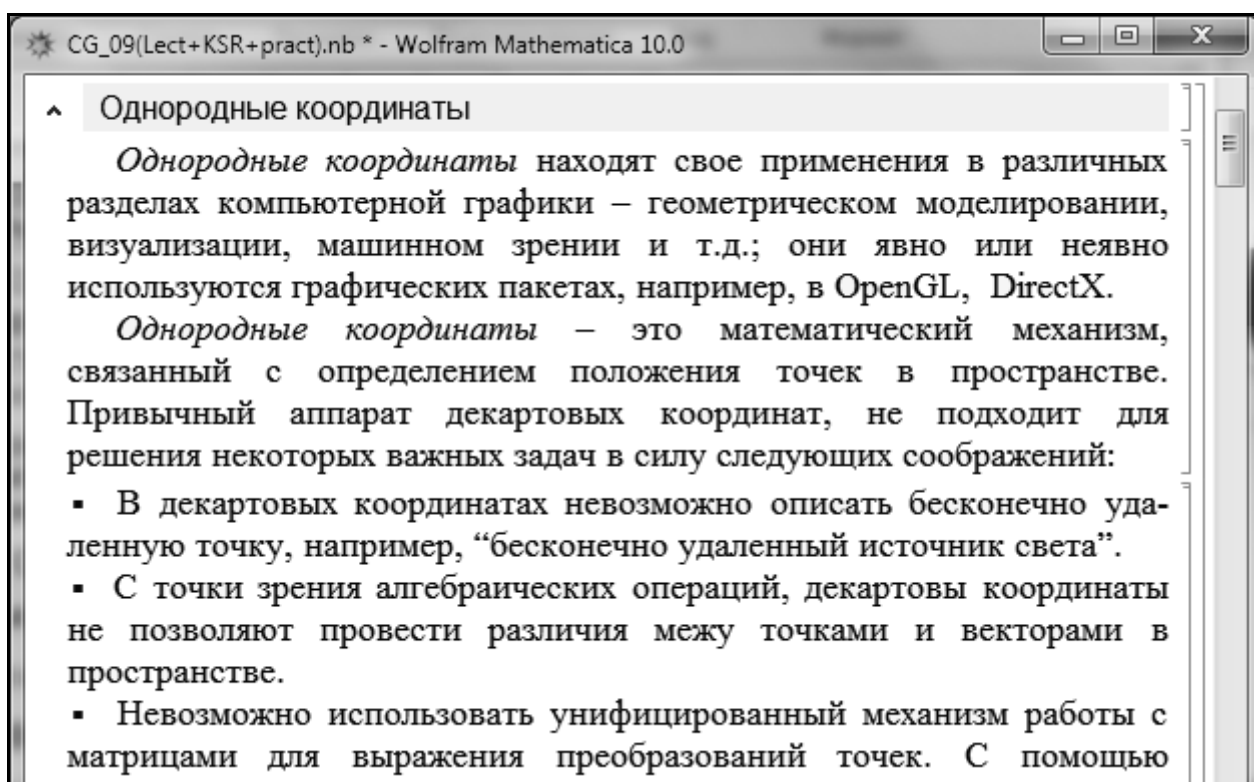


Рисунок 2 – Скриншот фрагмента блока «Однородные координаты»

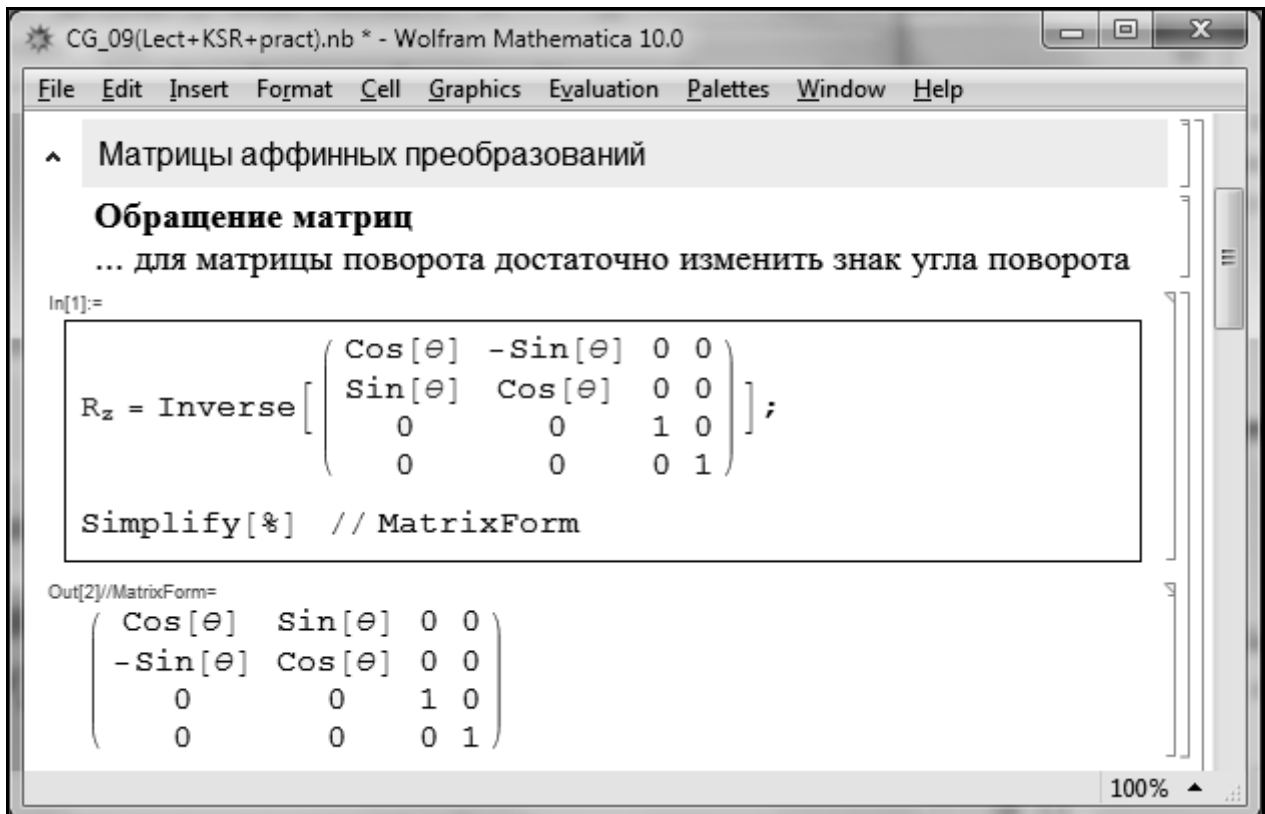


Рисунок 3 – Скриншот фрагмента блока «Матрицы аффинных преобразований»

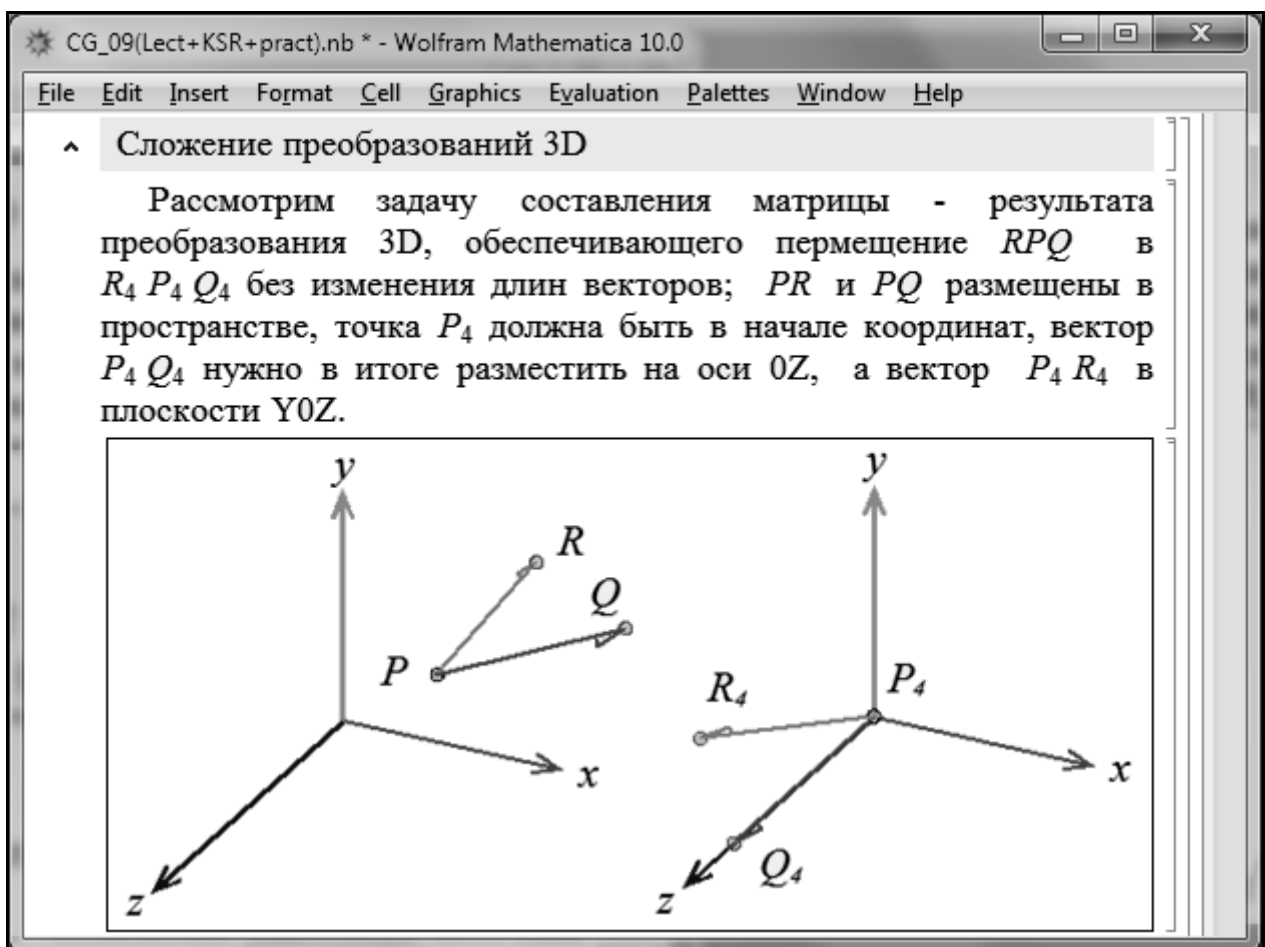


Рисунок 4 – Скриншот фрагмента блока «Сложение преобразований 3D»

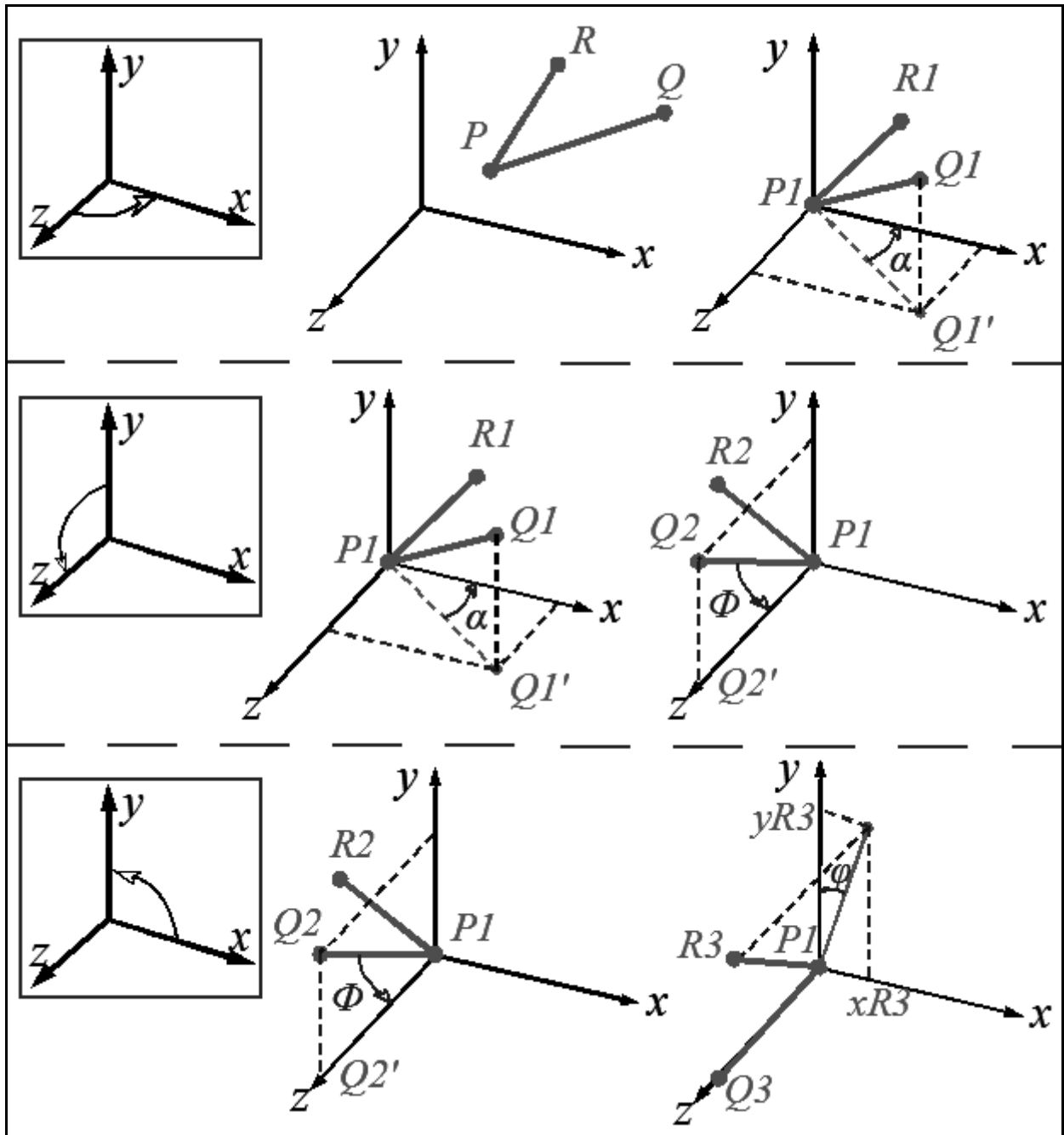


Рисунок 5 – Скриншот фрагментов иллюстраций шагов конвейера преобразований

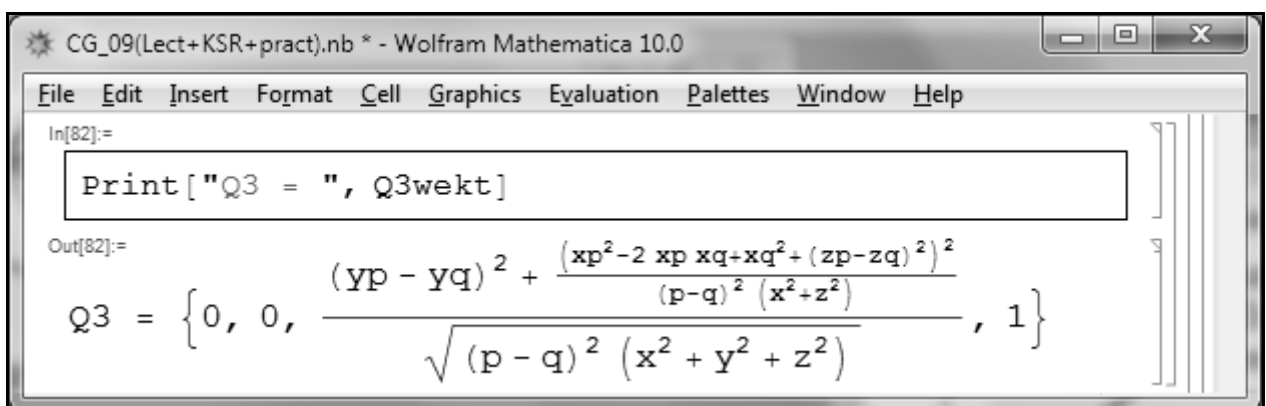


Рисунок 6 – Скриншот фрагмента с результатами проверки правильности преобразования

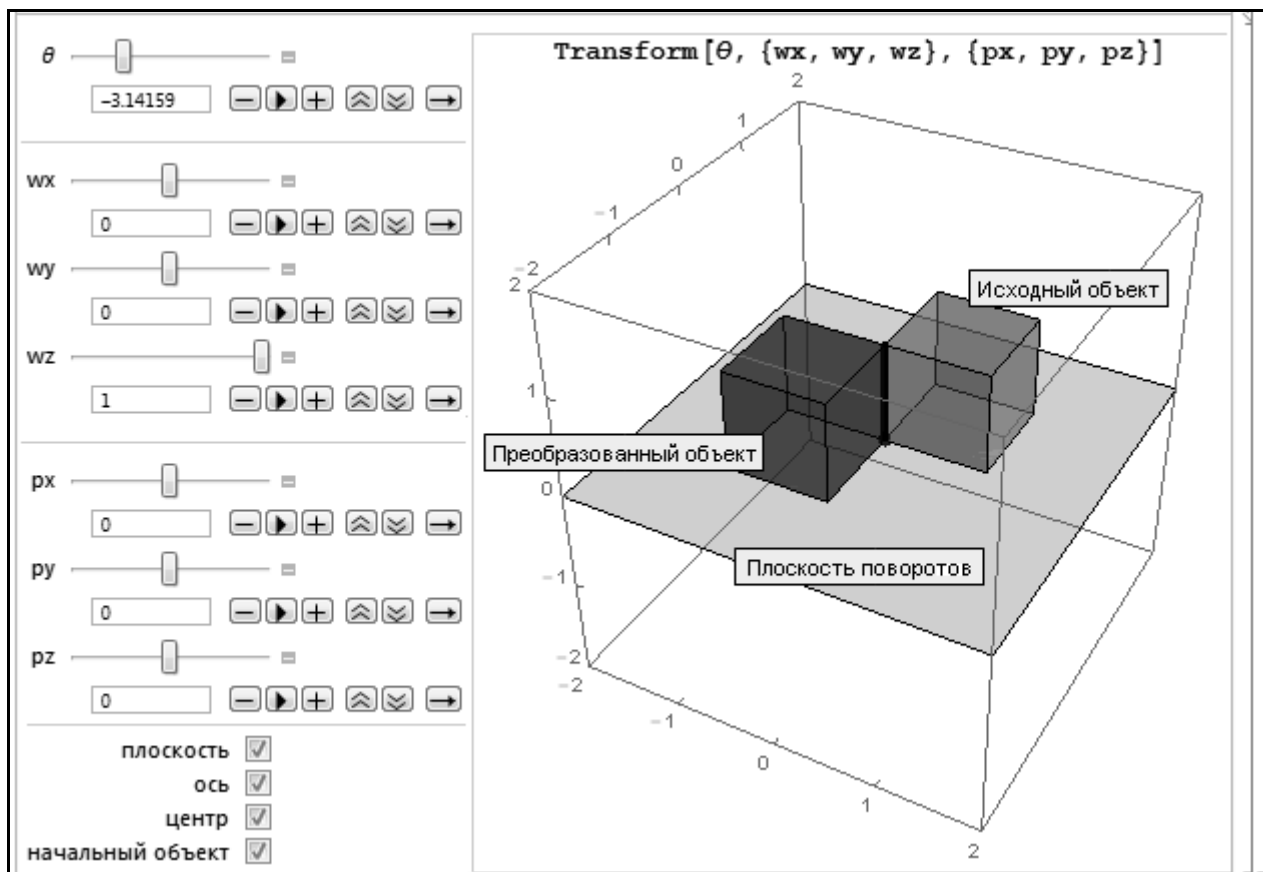


Рисунок 7 – Скриншот окна управления интерактивного модуля результатов преобразований

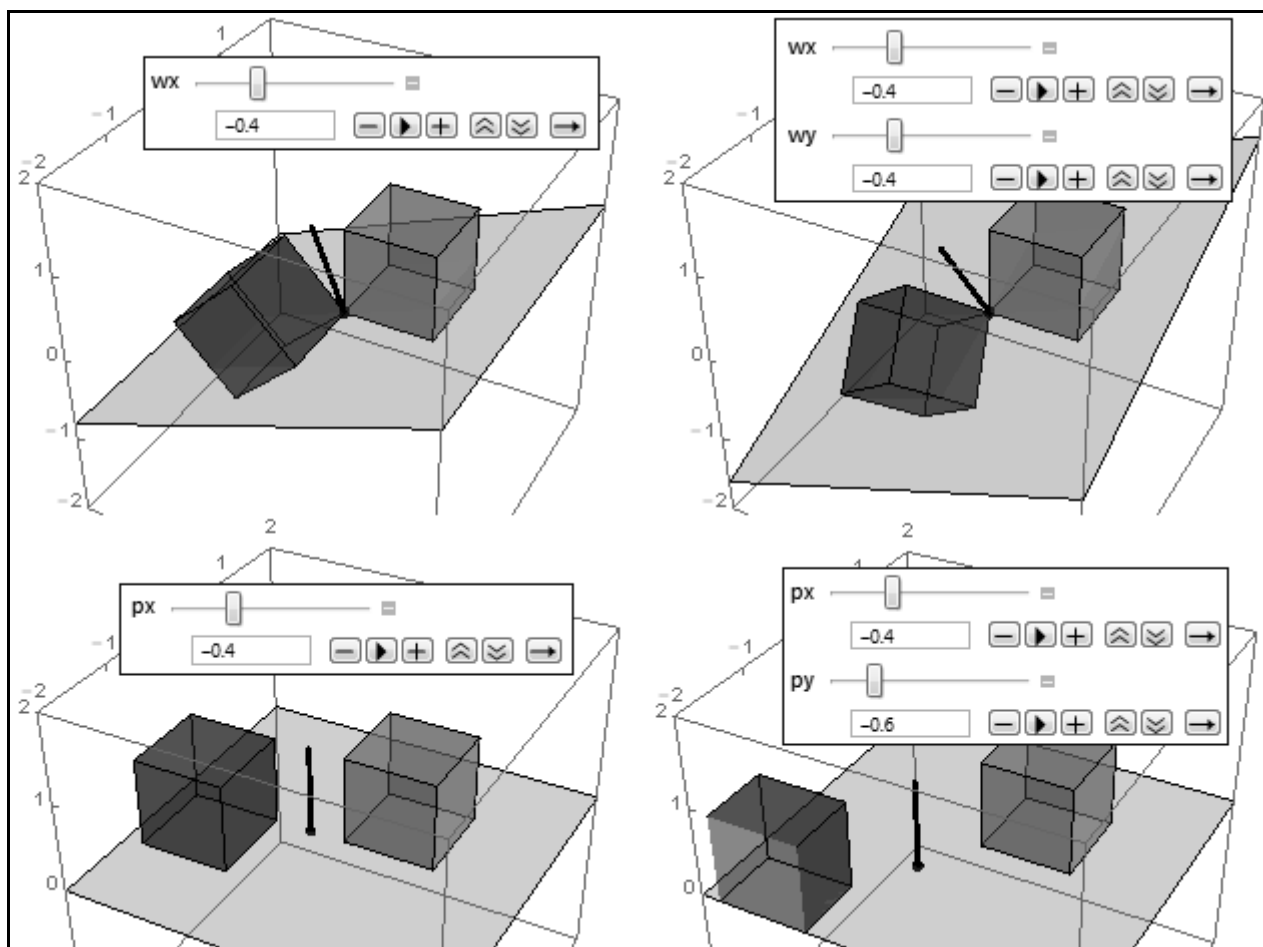


Рисунок 8 –Фрагменты скриншотов кадров отображения результатов преобразований поворотов и перемещений

На рисунке 1 показан скриншот главного окна программного модуля. С модулем можно работать в *Mathematica* или, используя CDF Player, есть изложение теории, ссылки, формулировки заданий для выполнения; все тексты, где есть формулы, записаны в математической нотации. Кроме секций с пояснениями есть секции для выполнения и получения результатов – можно выполнять операции с матрицами, строятся требуемые изображения. Например, в блоке «Матрицы аффинных преобразований» (фрагмент показан на рисунке 3) даны пояснения функций системы *Mathematica*: *MatrixForm* – вывод элементов одномерного или двумерного массива (списка) в матричном формате; *Inverse*, *Transpose* – обращение, транспонирование матрицы; *Simplify* – упрощение выражения. Эти функции представлены в виде упражнений, когда в заготовках можно менять значения, получать и просматривать результаты.

Скриншот начальных секций блока «Конвейер геометрических преобразований» с постановкой задачи «Сложение преобразований 3D» показан на рисунке 4. Рассматривается классическая задача получения итоговой матрицы преобразования, решение состоит в выполнении 4-х шагов: перенос и 3 поворота вокруг координатных осей (эскизы на рисунке 5) – эти шаги реализуются стандартными действиями применения соответствующих матриц. В модуле после каждого шага выводятся графики (исходный, результат), а также рассчитываются и выводятся координаты точек. Так на рисунке 6 контролируются координаты точки Q_3 , которая после этого шага преобразований должна оказаться на оси OZ – первые 2 координаты нулевые.

Кроме секций документа с пояснениями и иллюстрациями алгоритмов преобразований студентам для освоения предлагается программный модуль *Understanding3DRotation+.cdf*, который адаптирован (перевод, начальные ракурсы и масштаб просмотра, пояснения частей кода) по оригиналу *Understanding3DRotation-author.nb* из каталога *Wolfram Demonstrations Project*. На рисунках 7, 8 показаны скриншоты панели управления, несколько кадров с результатами работы модуля. В приложении можно перемещать и поворачивать получаемую сцену; каждое действие управления можно выполнять, задавая значение параметра в поле ввода или перемещением бегунка, также можно запускать просмотр с автоматическим изменением параметра, регулировать скорость и направления прокрутки видео.

В модуле даны комментарии к основным функциям и опциям кода, чтобы студенты могли вносить изменения, а также заимствовать приёмы написания программы, упражняться, используя другие графические примитивы и фигуры. Например, поясняя использованную в коде функцию формирования и вывода графики *Graphics3D*, предлагается вместо примитива *Cuboid* получить изображения с *Sphere* или *Cylinder*, *Cone*, *Ball*, *Parallelepiped*, *Prism*. Относительно применяемых в

модуле функций преобразования *RotationTransform*, *TranslationTransform* предложено освоить *AffineTransform*, *GeometricTransformation*. В части оформления, настройки вида объектов сцены в упражнениях включены пояснения правил задания толщины и типа линии (*Thickness*, *Dashed*, *Dotted*, *DotDashed*), цветов и прозрачности (*Colors*, *Opacity*), управления кадром вывода – *PlotRange*, *PlotRegion*. Вопросы программирования динамического вывода, использования инструментов интерактивности – поясняются функции и опции: *Manipulate*, *Dynamic*, *Initialization*, *PopupMenu*, *Checkbox*, *ControlType*, *Locator*, *SaveDefinitions*, *AutorunSequencing*.

Заключение

Описание компонент электронного документа поясняет, как применяемые средства обеспечивают процесс изучения дисциплины, когда теоретический материал увязывается с задачами усвоения и реализации базовых алгоритмов. Предлагаемая методика интегрирования интеллектуальных средств системы *Mathematica*, формата вычисляемых документов CDF, модулей коллекции демонстрационных интерактивных приложений расширяет границы создания и свободного распространения электронных интерактивных образовательных ресурсов.

Библиографический список

- [Дьяконов, 2009] Дьяконов, В.П. Энциклопедия компьютерной алгебры / В.П. Дьяконов. – М. : ДМК Пресс, 2009. - 1264 с.
- [Таранчук, 2013] Таранчук, В.Б. Основные функции систем компьютерной алгебры : пособие для студентов фак. прикладной математики и информатики / В.Б. Таранчук. – Минск : БГУ, 2013. – 59 с.
- [CDF, 2015] CDF. Документы оживают благодаря возможностям вычислений [Электронный ресурс] – Режим доступа : <http://www.wolfram.com/cdf>. – Дата доступа: 18.01.2015.
- [Wolfram Demonstrations Project, 2015] Wolfram Demonstrations Project [Электронный ресурс] – Режим доступа : <http://demonstrations.wolfram.com>. – Дата доступа: 18.01.2015.
- [Таранчук, 2014] Таранчук, В. Б. О создании интерактивных образовательных ресурсов с использованием технологий Wolfram / В. Б. Таранчук // Информатизация образования, 2014. № 1. С. 78–89.

WOLFRAM MATHEMATICA TOOLS AND TECHNOLOGIES OF DEVELOPMENT OF INTELLIGENT TUTORING SYSTEMS

Taranchuk VB

*Belarusian State University,
Minsk, Republic of Belarus*

taranchuk@bsu.by

The paper describes the new features and recommendations on the use of technology Wolfram Research for creation and maintenance of intelligent tutoring systems. Examples are given of the practice of training materials discipline "Computer Graphics".



OSTIS-2015

(Open Semantic Technologies for Intelligent Systems)

УДК 621.3

РАЗРАБОТКА КОМПЛЕКСНОЙ СЕМАНТИЧЕСКОЙ МЕТОДИКИ К КОМПОНЕНТНОМУ ПРОЕКТИРОВАНИЮ СЛОЖНЫХ ПРОЕКТНО- ПРОМЫШЛЕННЫХ СРЕД ОБУЧЕНИЯ (ППСО)

Афанасьев А.Н., Войт Н.Н.

Ульяновский государственный технический университет, г. Ульяновск, Россия

a.afanasev@ulstu.ru

n.voit@ulstu.ru

Разработаны программные методы повышения качества процесса обучения с помощью процедур интеллектуальной диагностики персональных характеристик обучаемого, синтеза индивидуальной траектории с использованием предложенных компьютерных семантических моделей предметной области, обучаемого, сценария, информационных потоков данных, реализованных в системе Moodle

Ключевые слова: интеллект; среда; обучение; дистанционное

Введение

Усложнение существующих проектных решений, разработка новых САПР требует наличие квалифицированных кадров на предприятии, которые во многом определяют успешность предприятия на рынке. Специалисты такой степени подготовленности обязаны постоянного повышения профессиональной компетенции. Возможность обучения без отрыва от производства является важным критерием в процессе обучения. Разработанный комплекс моделей, методов, технологий позволит реализовать проектно-промышленные среды обучения и повысить качество обучения.

1. Архитектура ППСО

Методологической и технологической основой разработки ППСО является образовательная среда стандарта IEEE 1484 LTSA, в которой взаимодействуют системы «окружение – концептуальное взаимодействие – модель взаимодействия» (рисунок 1).

Уровни образовательной среды.

Окружение. Система, представляющая сквозной процесс обучения в школах, колледжах, вузах и организациях повышения квалификации с приемственностью лучших методик обучения.

Концептуальное взаимодействие. Система, обеспечивающая интерфейсное взаимодействие программных компонентов (интеллектуальная система обучения, предметная область, опыт решений, профессиональная зрелость и компетентность) ориентирована на повышение качества процесса обучения с помощью применения конкретной образовательной программы.

Модель взаимодействия. Система, описывающая технологии реализации компонентов, их взаимодействия, исходный код, API-интерфейсы и протоколы, СУБД и структуру БД с помощью UML, ER-диаграмм.

Результатом взаимодействия обучаемого и среды обучения на уровне окружения являются информационные потоки: обучаемый-контент, обучаемый-(виртуальный) тьютор, обучаемый-обучаемый, обучаемый-университет и тьютор-университет (тьютором может быть виртуальный преподаватель).

ИНТЕЛЛЕКТУАЛЬНАЯ СРЕДА ОБУЧЕНИЯ (IEEE 1484 LTSA)

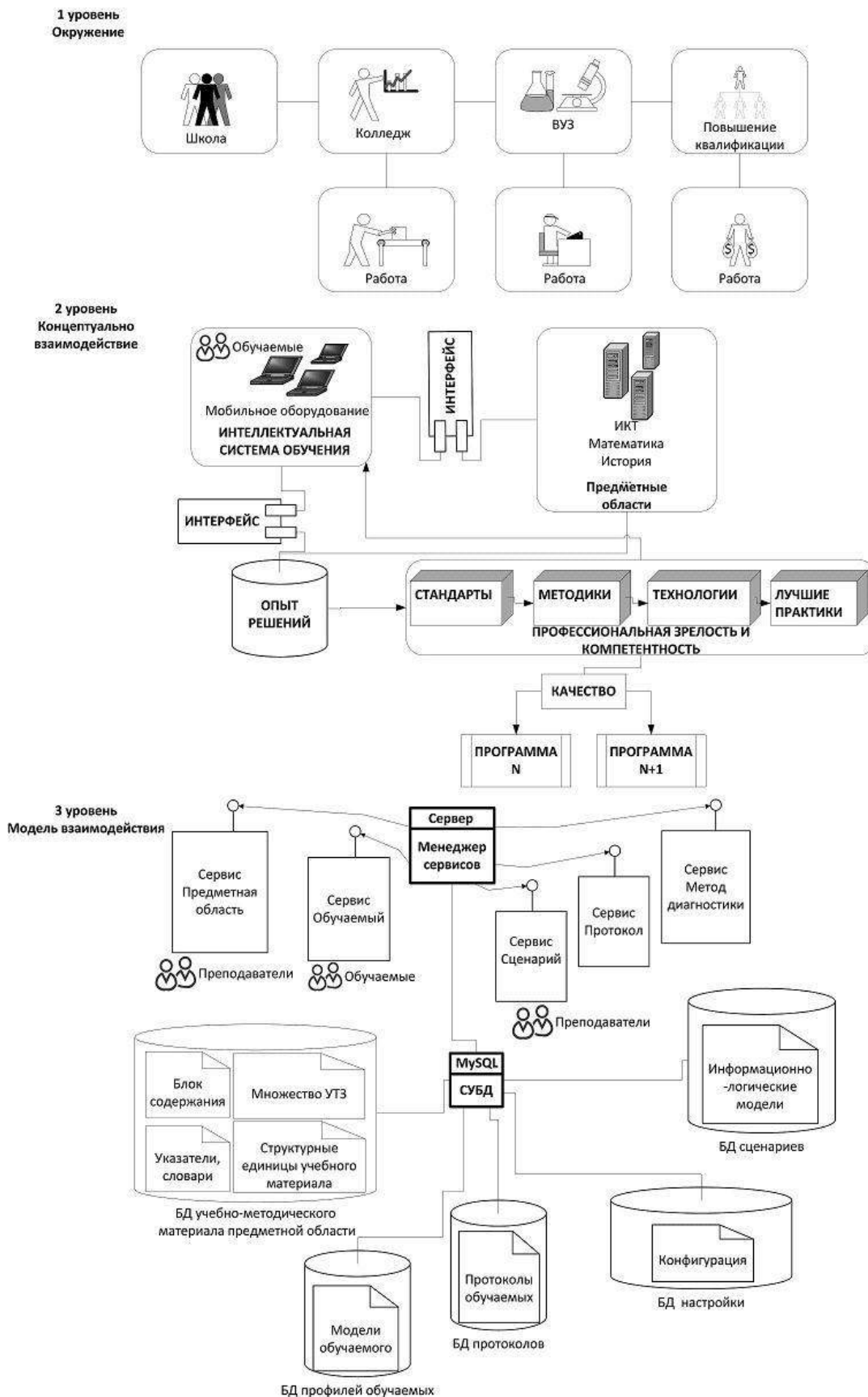


Рисунок 1 – Архитектура ИПСО

Таблица 1 – Семантическая характеристика разработанных моделей

Наименование моделей	Назначение	Математический аппарат	Отличительные свойства	Положительные эффекты
Предметная область	Представление объектов и процессов	Ассоциативное дерево онтологий с иерархическими, порядковыми и ассоциативными связями	Динамическое использование связей, интеграция по содержанию	Повышение качества содержания учебно-методического материала в моделировании изделий
Обучаемый	Представление индивидуальных профессиональных характеристик обучаемого	Система из нечетких лингвистических индивидуальных характеристик обучаемого в виде знаний, умений, владения навыками и компетентности	Нечеткий характер оценок теоретической и практической степени подготовки обучаемого в предметной области, учет предыстории обучения	Создание адаптивной индивидуальной траектории
Сценарий	Представление и упорядочивание процесса обучения	Система из орграфа, отображения вершин и правил выбора траектории	Наглядность и структурированность учебно-методического материала	Высокая скорость разработки и редактирования сценариев

Таблица 2 – Характеристика методов среды обучения

Наименование метода	Назначение	Математический аппарат	Отличительные свойства	Положительные эффекты
Диагностика знаний, умений, владения навыками и компетентности	Оценка уровня подготовленности к решению задач	Классификация индивидуальных профессиональных характеристик с помощью нечетких карт Кохонена	Уменьшение числа ошибок в оценке уровня подготовленности к решению задач	Повышение качества технологий обучения
Адаптивное планирование и управление траекторией обучения	Достижение требуемых индивидуальных профессиональных характеристик в процессе обучения	Система моделей «Предметная область», «Обучаемый», «Сценарий»	Гибкое управление сценарием в процессе обучения	Сокращение сроков процесса обучения

2. Разработка компьютерных семантических моделей ППСО

В основе математического обеспечения уровня концептуального взаимодействия положены компьютерные модели, представляющие предметную область, степень готовности обучаемого к решению задач, сценарий и информационный поток данных [Афанасьев и др., 2010].

Разработана модель предметной области в виде дерева онтологий, отличающаяся динамическим использованием иерархических, упорядоченных и ассоциативных связей, обеспечивающая адекватное представление процессов и объектов, повышающая качество содержания обучения [Афанасьев и др., 2009а].

Разработана новая модель обучаемого, в которой используются нечеткие лингвистические индивидуальные характеристики, соответствующие теоретическому и практическому уровням его подготовки в предметной области [Афанасьев и др., 2009б].

Разработана модель сценария траектории обучения, позволяющая представить процесс обучения, упорядочивая учебно-методический материал.

Семантическая характеристика разработанных моделей приведена в таблице 1.

3. Разработка компьютерных методов системы обучения

Разработан интеллектуальный метод диагностики знаний, умений, владения навыками и компетентности обучаемого на базе классификации с помощью нечетких карт Кохонена [Афанасьев и др., 2010, Афанасьев и др., 2009б].

Индивидуальные профессиональные характеристики модели обучаемого меняются событийно в контрольных точках K_i сценария. Оценки-баллы в виде входных векторов поступают на вход нечеткой карты, которая классифицирует полученные данные и формирует нечеткие характеристики степени подготовленности обучаемого.

Размерность входных векторов для классов знаний, умений, владения навыками определена числом вопросов в тестах, задач, временных затрат на каждую задачу соответственно, а для класса компетентности – трем (знания, умения и владение навыками).

Разработан метод адаптивного планирования и управления траекторией обучения, использующий комплекс моделей «Предметная область», «Обучаемый», «Сценарий» для достижения требуемых характеристик.

Управление траекторией обучения рассматривается как выбор вариантов сценария обучения и его реконструкция. В зависимости от степени подготовленности обучаемого принимается решение о выборе траектории обучения (сценарная или структурно синтезированная). Реконструкция сценария изменяет сценарий, дополняя его сконструированной траекторией из элементов модели предметной области, когда метод интеллектуальной диагностики показывает результат, не удовлетворяющий заданным целевым характеристикам обучаемого в точке K_i . Характеристика методов приведена в таблице 2.

Заключение

В настоящее время на базе разработанных компьютерных моделей и методов ведутся разработки программных модулей для открытой платформы Moodle.

Библиографический список

- [Афанасьев и др., 2010] Афанасьев А.Н., Войт Н.Н. Интеллектуальная обучающая система концептуальному проектированию автоматизированных систем // Известия Самарского научного центра Российской академии наук. – 2010. – Т. 12, № 4 (2). – С. 465 – 468.
- [Афанасьев и др., 2009а] Афанасьев А.Н., Войт Н.Н. Организация когнитивной автоматизированной обучающей системы (КАОС) промышленных пакетов САПР // Обозрение прикладной и промышленной математики. – 2009. – Т. 16. В. 5. – С. 804.
- [Афанасьев и др., 2009б] Афанасьев А.Н., Войт Н.Н. Разработка компонентной автоматизированной обучающей

системы САПР на основе гибридной нейронной сети // Автоматизация и современные технологии. – 2009. – № 3. – С. 14 – 18.

INTELLECTUAL TECHNOLOGY DEVELOPMENT OF COMPONENT DESIGN OF DESIGN AND INDUSTRIAL LEARNING ENVIRONMENTS

Afanas'ev A.N. *, Voit N.N. **

* *Ulyanovsk State Technical University,
Ulyanovsk, Russia*
a.afanasev@ulstu.ru

** *Ulyanovsk State Technical University,
Ulyanovsk, Russia*
n.voit@ulstu.ru

The models and methods are the mathematical basis of intellectual technology of the development of learning environments. They are used at large engineering and industrial enterprises and provide better quality of learning. Procedures of intelligent diagnostic of the personal characteristics of student, the synthesis of individual trajectory by using the proposed computer semantic domain models, learner, script, information data flow are described. The technology is implemented on the platform Moodle.



OSTIS-2015

(Open Semantic Technologies for Intelligent Systems)

УДК 004.891.2

ОНТОЛОГИЧЕСКИЙ ПОДХОД К АНАЛИЗУ КОМПЕТЕНЦИЙ ВЫПУСКНИКОВ ВУЗОВ

Шеркунов В.В.

*Ульяновский государственный технический университет,
г. Ульяновск, Россия*

v.sherkunov@gmail.com

В работе представлен краткий обзор образовательных стандартов и квалификационного справочника. Дано краткое описание разработанных онтологий, их формальное представление и практическая реализация. Также дано описание их дальнейшего применения в разрабатываемой интеллектуальной системе анализа компетенций и подбора выпускников на открытые вакансии предприятия.

Ключевые слова: квалификационный справочник, образовательный стандарт, онтология, компетенция.

Введение

При переходе в 2009 году на двухуровневую систему образования, возникла проблема не востребованности бакалавров со стороны промышленности, вследствие непонимания или неприятия производственными специалистами профессионального уровня бакалавров [Пушных, 2011]. В связи с этим, появилась проблема определения соответствия уровня квалификации выпускников, получивших степени бакалавров и магистров требованиям работодателей.

Целью данной работы является создание алгоритмов и онтологически-ориентированной интеллектуальной системы, позволяющей оценить рынок специалистов или группы выпускников и подобрать наиболее подходящих кандидатов на открытые вакансии, а также определить пробелы по соответствующим вакансиям и при необходимости сделать вывод, по повышению квалификации уже действующих сотрудников.

1. Образовательные стандарты и профессиональные требования

В настоящее время в задачах поиска экспертов, компетентных в определенных областях или задачах, используется онтологический подход. Примерами могут служить работы [Панкова и др., 2011], [Крюков и др., 2013], [Ranwez et al., 2012]. Так, например, в работе [Крюков и др., 2013] свидетельствами наличия компетенций в некоторой области являются документы, авторами которых являются претенденты. Анализ соответствия

документов указанной предметной области происходит на основе онтологии.

Но при этом, остаются актуальными вопросы, связанные с подбором специалистов на открытые вакансии, которые не имеют документальных свидетельств своей компетенции. Примером могут служить выпускники ВУЗов.

Основные квалификационные характеристики, которыми должен обладать выпускник, указаны в образовательных стандартах.

Образовательные стандарты, представляют собой «совокупность требований, обязательных при реализации основных образовательных программ», направленных на обучение бакалавров, магистров и специалистов в соответствии со сформулированной в компетентностном формате образовательной целью.

Квалификационные требования образовательного стандарта по направлениям подготовки сформулированы в терминах компетенций. Разделяют компетенций двух типов – общекультурные и профессиональные, - каждый из которых включает в себя несколько видов. В первом случае это социально-личностные, общенаучные, инструментальные, во втором – общепрофессиональные и определяемые видом деятельности выпускников. В свою очередь, каждый вид компетенции характеризуется наличием компонентов – знаний, способностей, умений, готовностей, навыков и мотиваций [Кузнецова и др., 2009].

Отражение и конкретизация образовательных стандартов происходит в учебных планах

специальностей, составляемых в ВУЗах. Где, помимо основного перечня дисциплин, указаны дополнительные дисциплины, соответствующие направлению подготовки.

В свою очередь, на крупных предприятиях для определения требований к занимаемой должности используется квалификационный справочник должностей руководителей, специалистов и других служащих. Данный справочник содержит основные квалификационные характеристики, утвержденные постановлением Минтруда России от 21.08.1998 №37. Квалификационные характеристики являются нормативными документами, предназначенными для обоснования рационального разделения и организации труда, правильного подбора, расстановки и использования кадров, обеспечения единства при определении должностных обязанностей работников и предъявляемых к ним квалификационных требований, а также принимаемых решений о соответствии занимаемым должностям при проведении аттестации руководителей и специалистов [Тихомирова и др., 2014]. Квалификационный справочник отражает основные требования к знаниям, навыкам и умениям специалиста, для занятия им конкретной должности.

При этом нет четкого соотношения между квалификационными требованиями, указанными в квалификационном справочнике и компетенциями, которыми обладает выпускник в соответствии с образовательными стандартами.

Для решения задачи сопоставления компетенций выпускника и требований к квалификации занимаемой должности, предлагается использование онтологического подхода, что в дальнейшем позволит эксперту связать компетенции с квалификационными требованиями, и даст возможность проведения анализа соответствия выпускника на претендующую должность.

Свидетельствами владения компетенциями будем считать изученные дисциплины, где уровень владения будет определяться полученной оценкой. Для формального представления выпускника или специалиста, строится его онтологическая модель, в которой указываются степень, направление подготовки (профиль) и изученные дисциплины с оценкой. Каждой дисциплине соответствуют определенные компетенции, указанные в онтологической модели учебного плана. Так же в модели указывается стаж работы, который может нести дополнительную информацию о квалификации выпускника.

Учебный план и квалификационный справочник аналогично представляются в виде онтологии. Онтологическая модель учебного плана включает в себя степень, направление подготовки, дисциплины, изучаемые по данному направлению, а также компетенции, которыми должен обладать выпускник после освоения дисциплины. Онтологическая модель квалификационного

справочника содержит множество должностей, и соответствующие им должностные обязанности, необходимые знания и требования к квалификации.

2. Формализация задачи

Онтологию учебного плана формально можно записать как кортеж следующего вида:

$$O^U = \langle D, Ds, St, C, R^U, R^{DU} \rangle, \quad (1)$$

где $D = \{\text{бакалавр, магистр}\}$ – конечное множество степеней, присваиваемых по окончанию ВУЗа;

$Ds = \{d_1, d_2, \dots, d_n\}$ – множество направлений подготовки специалистов, d_i – определенное направление подготовки;

$St = \{t_1, t_2, \dots, t_n\}$ – множество дисциплин, изучаемых по всем направлениям;

R^U – множество отношений типа «объект-объект», определяемых следующим образом:

$$R^U = \{R_{Ds}^U, R_{St}^U, R_C^U\},$$

где R_{Ds}^U – отношение, связывающее степень с направлением («имеет направление»);

R_{St}^U – отношение, связывающее выпускника с дисциплиной («изучена»);

R_C^U – отношение, связывающее дисциплину с компетенциями («соответствует»).

R^{DU} – отношение типа «объект-тип данных», указывающий текстовое описание компетенции («характеризует»).

Формально, онтологическую модель выпускника можно представить в виде кортежа:

$$O_i^S = \langle S_i, D^S, Ds^S, St^S, E, R^S, R^{DS}, R_O^S \rangle, \quad (2)$$

где S_i – конкретный выпускник или специалист;

$D^S \subseteq D$ – конечное множество степеней, которые получили заданные специалисты;

$Ds^S \subset Ds$ – множество направлений подготовки, по которым получены степени конкретными выпускниками или специалистами;

$St^S \subset St$ – множество дисциплин, изученных определенными выпускниками или специалистами;

$E = \{e_1, e_2, \dots, e_n\}$ – множество предприятий, на которых работают или работали выпускники или специалисты, e_i – предприятие, на котором работает или работал i -й выпускник;

R^S – множество отношений типа «объект-объект», определяемых следующим образом:

$$R^S = \{R_E^S, R_{St}^S, R_J^S\},$$

где R_E^S - отношение, указывающее образование специалиста («имеет образование»);

R_{St}^S - отношение, связывающее выпускника с дисциплиной («изучена дисциплина»);

R_J^S - отношение, связывающее выпускника с местом работы («имеет место работы»);

$R_O^S = \{\text{степень, направление, дисциплина, место работы}\}$ – конечное множество свойств, характеризующих образование, изученные дисциплины и место работы;

$R^{DS} = \{\text{оценка, стаж}\}$ – конечное множество отношений типа «объект-тип данных», указывающие конкретные значения оценки для дисциплины и стажа для места работы.

Модель онтологии, для квалификационного справочника, формально определяется следующим образом:

$$O^Q = \langle P, F, K, Q, R^Q, R^{DQ} \rangle, \quad (3)$$

где $P = \{p_1, p_2, \dots, p_n\}$ – множество должностей квалификационного справочника, p_i – конкретная должность справочника;

$F = \{f_1, f_2, \dots, f_n\}$ – множество должностных обязанностей, f_i – конкретная должностная обязанность, соответствующая определенной должности;

$K = \{k_1, k_2, \dots, k_n\}$ – множество знаний требуемых для занятия должности, k_i – определенное знание;

$Q = \{q_1, q_2, \dots, q_n\}$ – множество требований к квалификации к должностям, q_i – определенное требование;

R^Q – отношение типа «объект-объект», указывающих связь должности с должностными обязанностями, знаниями и требованиями к квалификации («имеет»);

R^{DQ} – отношений типа «объект-тип данных», связывающий должностные обязанности, знания и требования к квалификации с их текстовым описанием.

Для того чтобы осуществить подбор специалистов, эксперту по кадрам для должностных обязанностей нужно указать, какие компетенции необходимы для их выполнения:

$$\forall q_i RC^Q.$$

где R – некоторое бинарное отношение ассоциации,

связывающее должностные обязанности с множествами соответствующих им компетенций из учебного плана;

$C^Q \subset C$ – множество компетенций учебного плана, соответствующих заданным должностным обязанностям.

Графическое представление онтологической модели взаимодействия изображено на рисунке 1.

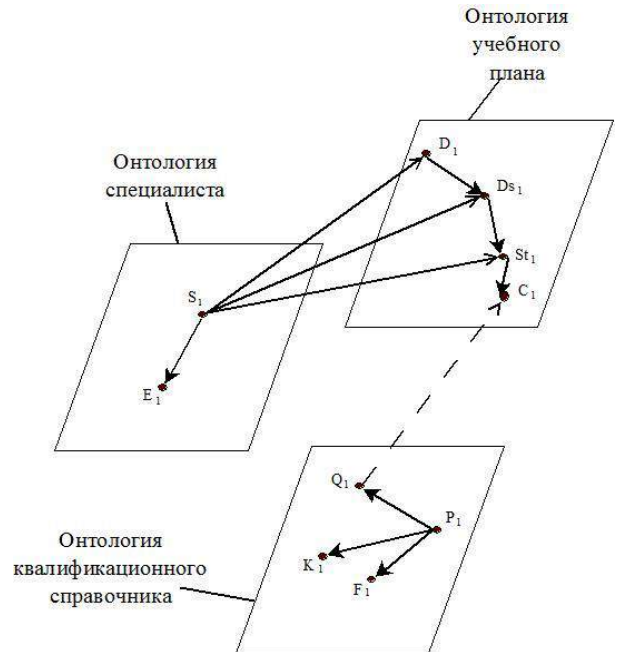


Рисунок 1. Модель взаимодействия онтологий.

Стрелками внутри каждой онтологии обозначены внутренние связи между объектами, определенными внутри каждой онтологии. Стрелки между онтологиями учебного плана и специалиста соответствуют отношениям из множества R_O^S , которые характеризуют степень, направление и изученную дисциплину. Пунктирной стрелкой обозначено бинарное отношение ассоциации, которое необходимо установить эксперту, чтобы связать должностную обязанность с компетенциями, которые необходимы для ее выполнения.

3. Реализация онтологий

Построение онтологий специалиста и учебного плана осуществлялось с помощью редактора онтологий Protégé 4.3 в формате OWL. Построение онтологии квалификационного справочника осуществляется в автоматическом режиме, посредством разработанного программного модуля.

При создании онтологии, было принято некоторое допущение в отношении свойств, связанных с особенностями языка OWL, не позволяющего моделировать атрибуты у предметных отношений. В работе [Трофимов, 2011] предлагается два варианта решения проблемы, один из которых заключается в декларировании отношений как классов, а не как свойств.

Суть данного подхода заключается в следующем. Отношение определяется как класс, а экземпляры этого класса будут выступать в роли экземпляров отношения. В свою очередь, аргументы этого отношения связываются с ним при помощи механизма свойств и их количество уже не ограничено.

В связи с этим, в онтологии специалиста были определены следующие классы (Рисунок 2):

- «Степень_Направление»;
- «Дисциплина_Оценка»;
- «МестоРаботы_Стаж».

Для которых определено множество свойств R_O^S .

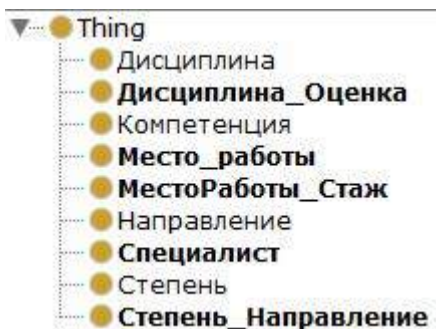


Рисунок 2. Представление классов онтологии выпускника в редакторе Protégé.

Фрагмент онтологии учебного плана в формате OWL приведен ниже:

```

<owl:Class rdf:about="#Дисциплина"/>
<owl:Class rdf:about="#Компетенция"/>
<owl:ObjectProperty rdf:about="#соответствует">
<rdfs:domain rdf:resource="#Дисциплина"/>
<rdfs:range rdf:resource="#Компетенция"/>
</owl:ObjectProperty>
<owl:NamedIndividual
rdf:about="#Дискретная_математика">
<rdf:type rdf:resource="#Дисциплина"/>
<соответствует
rdf:resource="#Прикладная_информатика_ОК-2"/>
</owl:NamedIndividual>
<owl:NamedIndividual
rdf:about="#Прикладная_информатика_ОК-2">
<rdf:type rdf:resource="#Компетенция"/>
<характеризует rdf:datatype="http://www.w3.org/2001
/XMLSchema#string">
способен логически верно, аргументировано и
ясно строить устную и письменную речь, владеть
навыками ведения дискуссии и полемики
</характеризует>
</owl:NamedIndividual>

```

Далее приведем фрагмент онтологии специалиста:

```

<owl:Class rdf:about="#Дисциплина_Оценка"/>
<owl:Class rdf:about="#Место_работы"/>
<owl:Class rdf:about="#МестоРаботы_Стаж"/>
<owl:Class rdf:about="#Специалист"/>
<owl:Class rdf:about="#Степень_Направление"/>
<owl:ObjectProperty
rdf:about="#имеет_образование">

```

```

<rdfs:domain rdf:resource="#Специалист"/>
<rdfs:range
rdf:resource="#Степень_Направление"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="#степень">
<rdfs:range rdf:resource="#Степень"/>
<rdfs:domain
rdf:resource="#Степень_Направление"/>
</owl:ObjectProperty>
<owl:DatatypeProperty rdf:about="#оценка">
<rdfs:domain
rdf:resource="#Дисциплина_Оценка"/>
<rdfs:range rdf:resource="http://www.w3.org/2001/
XMLSchema#integer"/>
</owl:DatatypeProperty>
<owl:NamedIndividual
rdf:about="#Бакалавр_Прикладная_информатика_1">
<rdf:type rdf:resource="#Степень_Направление"/>
<степень rdf:resource="#Бакалавр"/>
<направление
rdf:resource="#Прикладная_информатика"/>
</owl:NamedIndividual>
<owl:NamedIndividual rdf:about="#Дисциплина_1">
<rdf:type rdf:resource="#Дисциплина_Оценка"/>
<оценка rdf:datatype="http://www.w3.org/2001/
XMLSchema#integer">5</оценка>
<дисциплина
rdf:resource="#Дискретная_математика"/>
</owl:NamedIndividual>

```

На рисунке 3 изображено графическое представление онтологической модели выпускника или специалиста в виде графа. Вершинами графа являются индивиды соответствующих классов, характеризующие выпускника, стрелками обозначены отношения, описанные выше: R_E^S , R_{St}^S , R_J^S , R_O^S , R^{DS} .

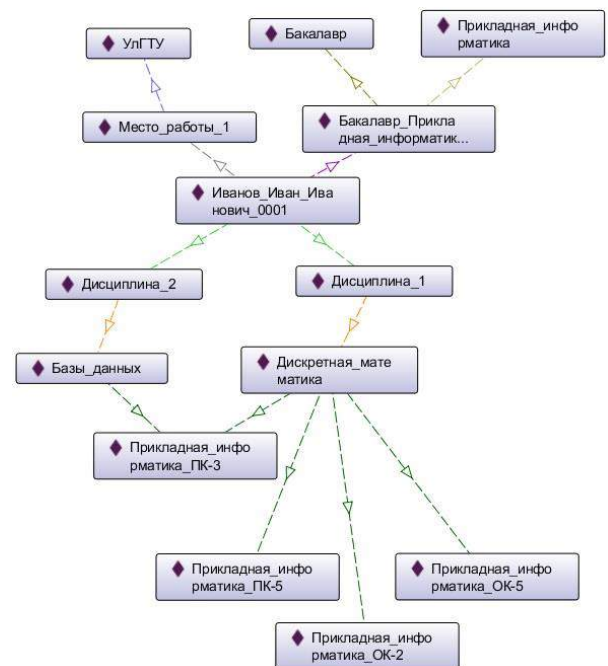


Рисунок 3. Онтологическое представление выпускника в редакторе Protégé.

Онтология квалификационного справочника формируется путем синтаксического анализа текстового документа данного справочника опираясь на его структуру. Структура описания квалификационных характеристик должностей разделяется на три группы:

- должностные обязанности;
- требования к навыкам и знаниям;
- требования к квалификации.

Для синтаксического анализа используется квалификационный справочник в формате PDF. Из которого извлекаются должности, путем перебора и фильтрации закладок документа, не принимая во внимания закладки общего значения. Далее, учитывая структуру справочника, для каждой должности извлекаются соответствующие разделы описания квалификационных характеристик. Для этого, выбирается текст между двумя должностями, расположенными в порядке следования в документе, и делится на группы, соответствующие формату справочника. Для выделения группы должностных обязанностей из полученного текста извлекается фрагмент между разделами «Должностные обязанности» и «Требования к навыкам и знаниям» (в справочнике, указателем на раздел служит словосочетание «Должен знать»), соответственно, для выделения требований к навыкам и знаниям извлекаем фрагмент между разделом «Требования к навыкам и знаниям» и «Требования к квалификации», а для требований к квалификации – фрагмент между разделом «Требования к квалификации» и окончанием текста. После чего, каждый фрагмент делится на предложения и записывается в файл OWL в виде утверждений соответствующих концептуальной схеме [Тихомирова и др., 2014]. Фрагмент полученных утверждений, записанных в файл, представлен ниже:

```
<Declaration>
<ClassIRI="#Должность"/>
</Declaration>
<Declaration>
<ClassIRI="#Должностные_обязанности"/>
</Declaration>
<Declaration>
<ObjectProperty IRI="#имеет"/>
</Declaration>
....
<ObjectPropertyAssertion>
<ObjectProperty IRI="#имеет"/>
<NamedIndividual
IRI="#Главный_конструктор_проекта"/>
<NamedIndividual
IRI="#Должностные_обязанности_главного_
конструктора_проекта"/>
</ObjectPropertyAssertion>
<DataProperty IRI="#включают"/>
<NamedIndividual
IRI="#Должностные_обязанности_главного_
конструктора_проекта"/>
<LiteraldatatypeIRI=
```

```
"http://www.w3.org/2001/XMLSchema#string">
Обеспечивает соблюдение требований и
нормативов по организации труда при
проектировании новых и реконструкции
действующих предприятий, разработке
технологических процессов и оборудования,
охраны окружающей среды</Literal>
</DataPropertyAssertion>
```

В настоящее время ведутся работы по разработке интеллектуальной системы подбора специалистов, основанной на разработанных онтологиях с применением компетентностного подхода. Где, подбор будет происходить с учетом связи «должностная обязанность-компетенция» и опираться на дисциплины, освоение которых дает необходимые компетенции, уровень владения которых определяется оценкой, полученной по данной дисциплине.

Для выявления наиболее подходящих кандидатов, возможно применение генетических алгоритмов и других методов интеллектуального анализа данных, позволяющих определить, кто из кандидатов обладает наиболее подходящим уровнем компетенций необходимых для занятия им должности.

Заключение.

В данной работе был дан краткий обзор основных нормативных документов, таких как образовательные стандарты и квалификационный справочник. Представлено формальное описание онтологий учебного плана, специалиста и квалификационного справочника, также представлена их практическая реализация. Дано представление дальнейшей перспективы применения данных онтологий, в разрабатываемой интеллектуальной системе подбора кандидатов на открытые вакансии.

Данная система, должна позволить специалисту кадровой службы, а также руководителям структурных подразделений производить анализ кандидатов, выявлять необходимые компетенции и получать наиболее эффективное распределение кандидатов. Что, в свою очередь, позволит устранить пробел между выпускниками, получившими образование по новой образовательной системе и работодателями, в понимании уровня квалификации данных выпускников.

Библиографический список

- [Ranwez et al., 2012] Ranwez Sylvie et al. User Centered and Ontology Based Information Retrieval System for Life Science/ Available from BMC Bioinformatics, 2012. <http://www.biomedcentral.com/1471-2105/13/S1/S4> (05/11/2014).
- [Крюков и др., 2013] Крюков, К.В. О понятии формальной компетентности научных сотрудников / К.В. Крюков, О.П. Кузнецов, В.С. Суховеров // Материалы III международной научно-технической конференции «Открытые семантический технологии проектирования интеллектуальных систем» (OSTIS-2013). Минск, 2013. С. 143-146

[Кузнецова и др., 2009] Кузнецова, Т. А. Согласование квалификационных требований, предъявляемых профессиональными и образовательными стандартами к выпускникам ВУЗов / Т. А. Кузнецова, Н. Н. Матушкин, С. И. Пахомов // Мониторинг образования, 2009, № 4, С. 3-9

[Панкова и др., 2011] Панкова, Л.А. Онтологические модели поиска экспертов в системах управления знаниями научных организаций / Л.А. Панкова, В.А. Пронина, К.В. Крюков // Проблемы управления. 2011. №6. С. 52-60

[Пушных , 2011] Пушных, В.А. Взгляд на проблему и переход на двухуровневую систему обучения в российских инженерных вузах с позиций теории «обучающейся» организации. / В.А. Пушных // Инженерное образование. 2011. №7. С. 16-21

[Тихомирова и др., 2014] Тихомирова, Д.Ю. Интеллектуальная система формирования проектных команд на основе компетенций/Д. Ю. Тихомирова, В. В. Шеркунов// КИИ-2014. Труды конференции. Т 2, С. 141-149

[Трофимов, 2011] Трофимов, И.В. Эволюция выразительных способностей языка OWL/И.В. Трофимов// Программные системы: теория и приложения. 2011. № 4(8). С. 85–94.

ONTOLOGICAL APPROACH TO THE ANALYSIS OF COMPETENCIES OF UNIVERSITY GRADUATES

Sherkunov V.V.

*Ulyanovsk State Technical University, Ulyanovsk,
Russia*

v.sherkunov@gmail.com

In the paper provides an overview of educational standards and qualification manual. A formal description of the developed ontologies and their practical realization are given. Also presented the prospects for their further use.

Introduction

Upon transition in 2009 to a two-level education system, there was a problem not of a demand of bachelors from the industry, owing to misunderstanding or rejection by production workers of professional level of bachelors. In this regard, there was the problem of determining compliance with the skill level of graduates with bachelor's degrees and master the requirements of employers.

Main Part

The main qualification characteristics, which the graduate has to possess, are specified in educational standards. Qualification requirements of the educational standard for the directions of preparation are formulated in terms of competencies. Reflection and specification of educational standards happens in curricula of the specialties made at universities. In turn, at the large enterprises for definition of requirements to a post the qualification manual of positions of heads, experts and other employees is used.

Thus, there is no accurate ratio between the qualification requirements specified in the qualification manual and competencies, which are possessed by the graduate according to educational standards.

To solve the problem of comparison of the graduate competencies and requirements to qualification of a

position, use of ontological approach is offered that further will allow the expert to connect competences with qualification requirements, and enable to carrying out the analysis of relevant graduate on claiming the post.

As the evidence of possession of competences, we will consider the studied disciplines where the level of proficiency will be defined by the received assessment. For formal representation of the graduate or the specialist, ontological model is designed in which are specified degree, the direction of preparation and the studied disciplines with an assessment. To each discipline, there correspond the certain competencies specified in ontological model of the curriculum. In addition, length of service, which can contain additional information on qualification of the graduate, is specified in model. The curriculum and the qualification manual also presented in the ontology form.

Creation of ontologies of the specialist and the curriculum implemented in the editor of ontologies Protégé 4.3 in the OWL format. Creation of ontology of the qualification manual is carried out in the automatic mode, by means of the developed program module.

Conclusion

In the paper the short review of the main normative documents, such as educational standards and the qualification manual are given. The ontologies formal description of the curriculum, the expert and the qualification manual are submitted, their practical realization is also presented. The representation of further prospects of application of these ontologies in the developed intelligent system of selecting candidates for open positions is given.



OSTIS-2015

(Open Semantic Technologies for Intelligent Systems)

УДК 004.853

НЕЙРОСЕТЕВАЯ МОДЕЛЬ ПРОЦЕССА ПРОФЕССИОНАЛЬНОГО ОБУЧЕНИЯ МОЛОДЫХ СПЕЦИАЛИСТОВ

Федяев О.И.

*Донецкий национальный технический университет,
г. Донецк, Республика Украина*

fedyaev@r5.dgtu.donetsk.ua

oleggedyayev@yahoo.com

Научная работа посвящена разработке нейросетевой модели процесса обучения студентов для агентной системы моделирования рынка труда. Эта модель имитирует процесс передачи профессиональных навыков и знаний в зависимости от личностных характеристик студентов. Система моделирования на основе искусственных агентов позволит анализировать процесс подготовки молодых специалистов и прогнозировать возможность их дальнейшего трудоустройства.

Ключевые слова: модель обучения; профессиональные знания; ментальность студента; нейронная сеть.

Введение

Рассматривается задача построения модели процесса профессионального обучения для агентной системы моделирования рынка труда, которая позволит анализировать процесс подготовки молодых специалистов, а также прогнозировать возможность их дальнейшего трудоустройства.

Сам процесс обучения, как объект исследования, является динамическим и характеризуется большой инерционностью. Последствия изменения одного из факторов можно узнать только по окончании обучения студентов. Поэтому актуальной как в экономическом, так и в социальном плане, является разработка моделей, позволяющих оптимизировать затраты на образование и прогнозировать результаты инновационных преобразований в подготовке кадров. Однако формально (математически) описать процесс обучения не представляется возможным. В этом случае целесообразно разрабатывать имитационные модели на основе нейронных сетей, которые могут обеспечить проведение необходимых исследований по этой проблеме [Федяев и др., 2013].

Целью данной работы является разработка нейросетевой модели, способной функционально описать зависимость получаемых студентом профессиональных знаний и умений от факторов, влияющих на полноту этих знаний. Для достижения этой цели необходимо решить следующие задачи:

- рассмотреть процесс обучения студентов университета как систему с распределённым интеллектом;
- определить внешние и внутренние факторы, влияющие на качество усваивания знаний и навыков;
- разработать методику определения ментальных и психофизиологических особенностей преподавателей и студентов;
- разработать структуру нейросетевой модели и алгоритм её обучения;
- исследовать трудоёмкость настройки модели и адекватность её процессу обучения.

1. Система профессионального обучения как объект моделирования

Высшее учебное заведение, как центр подготовки квалифицированных специалистов, представляет собой распределённую систему, субъектами которой являются преподаватели и студенты, взаимодействующие в предоставленной ВУЗом учебной среде, оснащённой соответствующим оборудованием. Качество подготовки в основном зависит от следующих факторов:

- профессионализма преподавателей;
- подготовленности абитуриентов и системы их отбора;
- среды обучения (материально-технической базы ВУЗа);

- стандартов образования (учебные планы и т. п.);
- системы повышения квалификации преподавателей;
- организационной структуры управления образованием и трудоустройством.

Система подготовки специалистов основывается на взаимодействии министерства образования, университетов и студентов (рисунок 1).

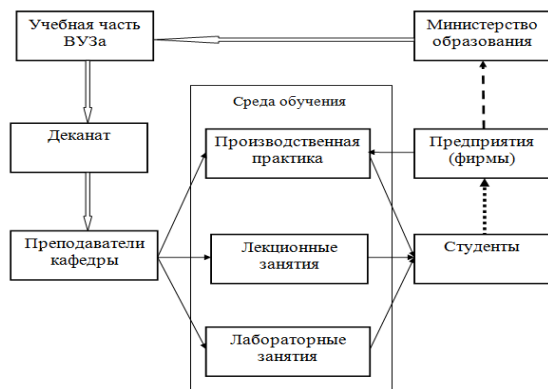


Рисунок 1 – Схема подготовки специалистов в ВУЗе для предприятий: (→) – передача знаний и умений; (⇌) – нормативные документы; (---▶) – заказ на подготовку специалистов; (••••▶) – подготовленные специалисты

Объектно-ориентированный анализ процесса подготовки молодых специалистов показал, что его субъекты взаимосвязаны, образуют распределённую, неоднородную и интеллектуальную систему. Поэтому разработка имитационной модели такой системы, которая будет использоваться для анализа и прогнозирования образовательных процессов, может быть успешно выполнена на основе методов агентно-ориентированного моделирования [Федяев и др. 2006], [Zhabska et al., 2011].

Основной функцией процесса обучения студентов как системы является передача профессиональных знаний и выработка умений у будущих специалистов решать определённые производственные задачи. Данная функция реализуется определённой совокупностью образовательных процессов: лекционные и практические занятия, производственная практика.

С позиции системного анализа [Федяев и др., 2010] процесс профессиональной подготовки специалистов реализуется сложной распределённой системой, которая состоит из автономных, взаимосвязанных, целенаправленно функционирующих элементов и находится во взаимодействии с внешней средой. Системный анализ предполагает рассмотрение функционирования системы во времени и пространстве с целью выделения основных характеристик и параметров как самой системы, так и составляющих её процессов.

1.1. Постановка задачи анализа подготовки студентов

В системе подготовки и трудоустройства молодых специалистов можно выделить следующие элементы: молодые специалисты, учебное заведение, министерство образования и фирмы (предприятия). Перечисленные элементы системы неоднородны по своей структуре, территориально распределены и обладают сложной функциональностью. Все вместе они образуют распределённую систему, в рамках которой можно решить следующие задачи:

- организация эффективного учебного процесса подготовки кадров по профессиям в соответствии с учебными планами (ВУЗ);
- определение качественных стандартов и учебных планов подготовки кадров (министерство науки и образования, учебное заведение);
- подбор необходимых кадров для решения производственных задач (фирмы);
- поиск специалистами работы на фирме (предприятии), соответствующей полученной квалификации и удовлетворяющей социальным запросам претендентов на работу.

Перечисленные задачи являются трудно формализуемыми и поэтому не могут быть решены традиционными математическими методами. Кроме того, участники рассматриваемого процесса территориально удалены друг от друга, неоднородны по структуре и их деятельность интеллектуальна по своей природе. Эти особенности обуславливают целесообразность применения теории интеллектуальных агентов к разработке имитационной модели для анализа процесса подготовки кадров (рисунок 2).

На макроуровне расположены искусственные агенты, моделирующие в рамках ограниченной рациональности функции министерства образования, учебного заведения и предприятий (фирм). Микроуровень содержит сообщества агентов, реализующих диалог студентов с работодателями и учебным заведением.

Задача 1. Настройка модели по данным наблюдений. Это обратная задача, связанная с нахождением параметров модели, т. е. с построением функции f по наблюдаемым данным M_c , M_n , C и P_c :

$$P_c = f(M_c, M_n, C), \quad (1)$$

где M_c – ментальность студента; M_n – ментальность преподавателя; C – среда обучения; P_c – профессионализм студента по одной изучаемой дисциплине.

Ментальность студента (M_c) определяется элементами, которые характеризуют его воспитательный аспект и приобретённый жизненный опыт:

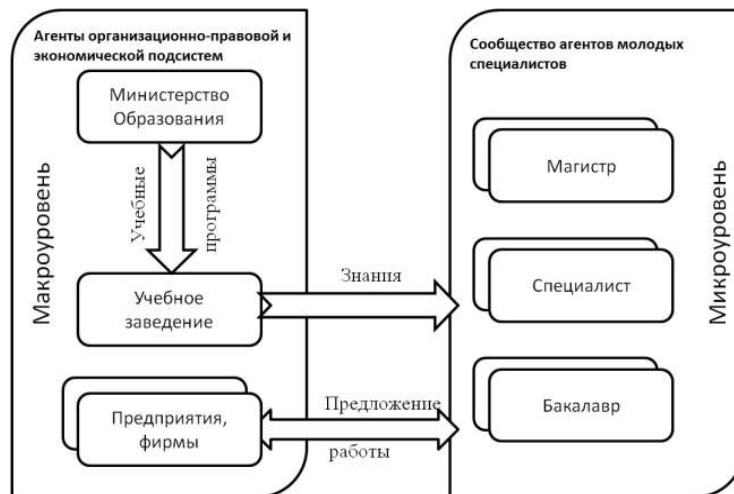


Рисунок 2 – Общая структура агентно-ориентированной модели процесса подготовки молодых специалистов

$$M_c = (m, i, p, s, \dots), \quad (2)$$

где m – ментальность; i – интеллект; p – психология; s – здоровье.

Ментальность преподавателя (M_n) в данном случае определяется факторами, от которых зависит качество передачи знаний от преподавателя к студенту:

$$M_n = (us, uz, h, v, a, \dots), \quad (3)$$

где us – учёная степень; uz – учёное звание; h – стаж; v – возраст; a – артистизм.

Среда обучения (C) характеризуется состоянием учебно-методического и технического обеспечения учебного процесса, а также уровнем организации обучения студентов.

Профессионализм студента по одной изучаемой дисциплине (P_c) определяется объёмом знаний (z_c) и умений (u_c), которые он получает в процессе изучения данной дисциплины:

$$\begin{aligned} z_c \subseteq Z_d \subseteq Z, \quad u_c \subseteq U_d \subseteq U, \\ P_c = z_c \cup u_c, \quad P_d = Z_d \cup U_d, \\ P_c \subseteq P_d, \end{aligned} \quad (4)$$

где Z_d – объём знаний, определяемый учебной программой дисциплины, которая читается на кафедре; Z – объём знаний по данному профессиональному направлению, определяемый современным состоянием науки и техники;

Задача 2. Формирование знаний и умений по ментальности участников образовательного процесса. Данная задача состоит в явном нахождении профессионализма студента (P_c), т. е. его знаний и умений, после изучения конкретной дисциплины по замеренным данным о ментальности студента (M_c) и преподавателя (M_n)

по построенной модели f :

$$P_c = f(M_c, M_n, C) \quad (5)$$

Эта задача относится к классу прогнозных задач. С её помощью можно исследовать влияние различных параметров (содержание учебной программы, контингента студентов и т. д.) на качество образования в конкретном университете.

2. Факторы влияния на качество усвоения студентами знаний

Для построения модели агента «Студент» необходимо учесть все факторы, влияющие на студента в той или иной степени, а также определить степень их влияния. Так как каждый отдельный студент является, прежде всего, личностью, то и анализировать необходимо его личностные характеристики.

Были проанализированы все факторы, влияющие на ментальный портрет студента. Были использованы популярные психологические методы их анализа [Ильин, 2004]. В результате для всестороннего анализа личности были выделены следующие типы факторов: мотивация студента к учёбе, интеллектуальные способности студента, психологические особенности студента, физические факторы, влияющие на обучение. Каждый из этих типов разбивается на несколько показателей, которые можно определить по результатам тестов, опросов и т. д. [Айзенк, 2003].

На наш взгляд, факторы, влияющие на усвоение студентом учебного материала, можно систематизировать так, как это показано на рисунке 3. Анализ этих факторов позволит изучить личность обучаемого с разных сторон, выявить наиболее важные ментальные особенности, влияющие на успешность обучения [Дейнека, 2009].



Рисунок 3 – Факторы, влияющие на усвоение материала студентом

Были разработаны методики определения ментальных и психофизиологических особенностей студента. Результаты по оценке каждого из вышеперечисленных параметров могут быть систематизированы и стандартизованы. Эти методики в совокупности образуют систему, которая определяет ментальный портрет студента. В таблице 1 представлена данная систематизация.

Таблица 1 - Систематизация характеристик студента

Характеристика ментальности	Способ определения	Оригинальная градация
Уровень интеллекта	Тест на IQ Айзенка	от 0 до 160 баллов
Тип темперамента	Тест Айзенка «Тип темперамента»	4 вида темпераментов
Социальный интеллект	Тест Гилфорда	от 0 до 57 баллов
Уровень мотивации	Тест Гречикова	5 видов мотивации
Уровень креативности	Тест Торренса	от 0 до 70 и более баллов
Специальные способности	Тест Айзенка по 3 видам спец. способностей	от 0 до 150 баллов за каждый
Умение работать в команде	Самостоятельное определение	от 0 до 100%
Жилищные условия	Самостоятельное определение	от 0 до 100%
Состояние здоровья	Самостоятельное определение	от 0 до 100%
Пол	Самостоятельное определение	женский или мужской

После прохождения всех опросов и тестов будет определен многопрофильный портрет студента, который можно будет использовать при разработке модели передачи знаний.

3. Нейросетевая модель зависимости остаточных знаний студентов от их ментальности

Процесс обучения студентов заключается в передаче знаний и навыков от преподавателей. Качество обучения фиксируется в экзаменационной ведомости. Разрабатываемая модель процесса обучения должна формировать на выходе остаточные знания студента по отдельной дисциплине, с которыми он выходит на рынок труда. По ним работодатели решают вопрос о трудоустройстве кандидатов на вакантные должности.

Остаточные знания зависят от ментальности студента и других факторов, которые описаны в предыдущем разделе. Эта связь трудно формализуема, т. е. математически описать её сложно. В таких случаях, как уже было указано выше, целесообразно использовать нейронную сеть [Круглов, 2001], которая позволит выявить существующую связь путем её обучения. Для обучения нейросети имеется в распоряжении следующая объективная информация:

- психологический портрет, характеризующий ментальность студента;
- учебная программа дисциплины;
- критерии оценки знаний;
- экзаменационная ведомость, отображающая успешность обучения студентов.

Прогноз остаточных знаний по одной конкретно взятой дисциплине для одного студента осуществляется в два этапа. На первом этапе прогнозируется экзаменационная оценка на основании ментальности обучаемого. На втором этапе, исходя из прогнозируемой оценки, формируется усреднённый набор остаточных знаний и умений, соответствующий данной оценке.

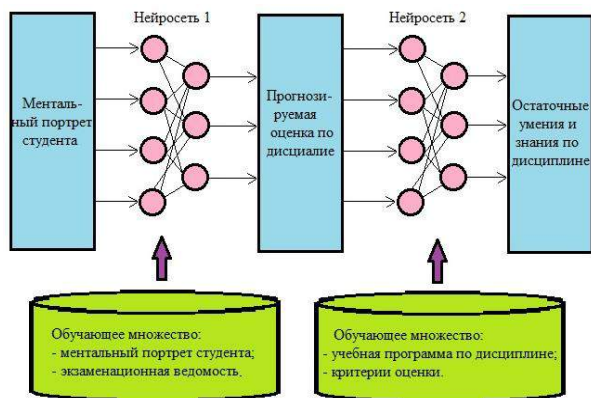


Рисунок 4 - Схема нейромодели профессионального обучения студентов на примере одной дисциплины

Каждый из этих этапов невозможно формализовать математически, поэтому будут использованы две нейросети. Первая нейронная сеть будет обучаться на основании ментальных портретов группы студентов и экзаменационной ведомости. Вторая нейросеть – на основании критериев оценки и учебной программы дисциплины, в которой содержится перечень знаний и умений. Схема описанной двухкаскадной модели представлена на рисунке 4.

Нейроалгоритм моделирования зависимости экзаменационной оценки от личностных характеристик студента реализуется первой нейросетью. Входными сигналами первой нейросети являются ментальные характеристики студентов, полученные в процессе их тестирования (таблица 1). Входные сигналы образуют вектор $X=(x_1, x_2, \dots, x_{10})$, компоненты которого описаны в таблице 2.

Таблица 2 - Входные параметры первой нейросети

Тип ментальной характеристики	Входной сигнал нейросети	Код
Мотивация	Тип мотивации	x_1
Интеллектуальные способности	Уровень IQ	x_2
	Уровень специальных способностей (в данном случае - вычислительных)	x_3
	Уровень социального интеллекта [0..1]	x_4
Психологические особенности	Тип темперамента [0..1]	x_5
	Умение работать в команде	x_7
Физические факторы	Жилищные условия	x_8
	Состояние здоровья	x_9
	Пол	x_{10}

На выходе нейросеть должна формировать сигналы, определяющие прогнозную экзаменационную оценку, соответствующую студенту с определенной ментальностью, которая подаётся на её вход. Оценка выставляется по пятибалльной шкале.

Нейросетевая модель формирования экзаменационной оценки строится на базе

многослойного персептрона с нелинейной функцией активации. Достаточно использовать 2-3 слоя, чтобы обеспечить реализацию любой нелинейной зависимости между выходом и входом.

Вторая нейросеть реализует нейроалгоритм моделирования зависимости остаточных знаний студента по дисциплине от полученной экзаменационной оценки. Входные сигналы второй нейросети представляют собой экзаменационную оценку, полученную с выхода первой нейросети.

Выходные сигналы нейросети образуют вектор, компоненты которого фиксируют наличие или отсутствие соответствующего остаточного знания или умения. Размер вектора определяется суммарным количеством знаний и умений, предусмотренных учебной программой дисциплины. Они обозначены вектором

$$Y = (y_1, y_2, \dots, y_n),$$

где n – количество знаний и умений; $y_i \in [0,1]$. Выходные сигналы показаны в таблице 3.

Таблица 3 - Выходные сигналы второй нейросети

№	Знания и умения	Код	
1	Знание 1 (Например, знание методов, алгоритмов)	y_1	Список знаний
2	Знание 2	y_2	
3	Знание 3	y_3	
...	
k	Знание k	y_k	
k+1	Умение 1 (Умение применять методы, алгоритмы)	y_{k+1}	Список умений
...	
n	Умение m	y_n	

Структура у обеих нейросетей относится к классу однородных многослойных персептронов с полными последовательными связями и с сигмоидальной функцией активации [Круглов, 2001]. Обучение нейросетей проводилось по стратегии «обучение с учителем» по алгоритму обратного распространения ошибки. Обучающее множество для второй нейросети составляет преподаватель-профессионал (эксперт) по своей дисциплине, используя утверждённые критерии оценки и учебную программу дисциплины, которая содержит перечень знаний и умений.

4. Обучение и анализ адекватности нейросетевых моделей

В качестве среды моделирования искусственных нейронных сетей использовался пакет Neural Network Toolbox, который входит в стандартную поставку MATLAB [Дьяконов, 2001]. Пакет Neural Network Toolbox обеспечивает всестороннюю поддержку типовых нейросетевых парадигм и имеет открытую модульную архитектуру.

Таблица 4 - Результаты тестирования студентов

Характеристика	Студенты					
	1	2	3	4	5	6
Уровень интеллекта	125	110	115	100	98	105
Тип темперамента	Хол	Флег	Мел	Хол	Санг	Санг
Социальный интеллект	45	50	38	42	23	33
Тип мотивации	Патр	Проф	Инст	Изб	Изб	Инст
Уровень креативности	64	30	62	24	78	49
Специальные способности	123	140	113	94	75	96
Умение работать в команде	6/8	5/8	6/8	3/8	2/8	7/8
Жилищные условия	60%	75%	90%	90%	50%	45%
Состояние здоровья	90%	80%	80%	75%	80%	75%
Пол	М	М	Ж	М	Ж	Ж
Полученная оценка	5	5	4	3	2	2

Пакет содержит функции командной строки и графический интерфейс пользователя для быстрого пошагового создания различных программных моделей нейросетей [Круглов, 2001].

При построении обучающего множества для первой нейросети были выбраны 6 студентов, прослушавших учебный курс «Интеллектуальные системы в экономике» и уже получивших экзаменационные оценки. Студенты для тестирования были выбраны таким образом, чтобы в обучающем множестве были представлены все экзаменационные оценки.

Эти студенты были протестированы согласно методике, изложенной в разделе 2. Результаты их анкетирования и тестирования показаны в таблице 4.

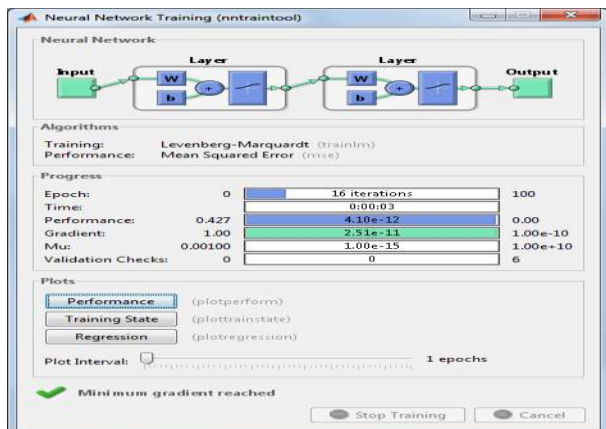


Рисунок 5 - Результаты обучения нейросети «Ментальность студента – прогноз - оценка»

Для обучающего множества были взяты данные первых пяти студентов. Результаты же студента под номером 6 будут использованы для проверки обученной нейросети. Как видно на рисунке 5, для обучения двухслойной нейросети хватило 16 эпох.

Обучающее множество для второй нейросети должен готовить преподаватель, который читает студентам учебную дисциплину. Из учебной программы (а это утверждённый нормативный документ) был взят список знаний и умений, которыми должен овладеть студент по данной дисциплине, и для него преподавателем сформирована таблица, показывающая, за какие знания и навыки ставится определённая оценка.

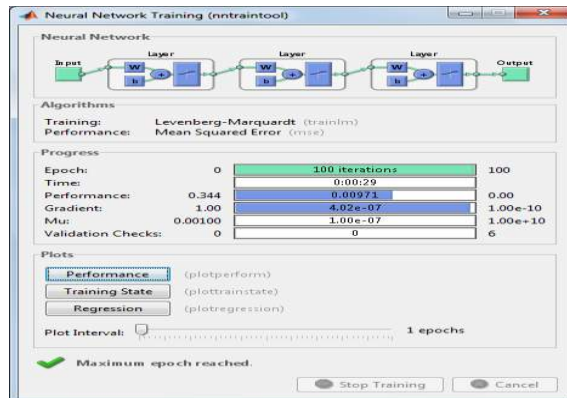


Рисунок 6 - Результаты обучения нейромодели «Оценка – прогноз - остаточные знания»

По аналогии с предыдущей нейромоделью была построена модель второй трёхслойной нейросети с количеством нейронов в слоях: 4-20-35. Входные сигналы нейросети – это вектор оценок, а выходные – вектор усреднённых знаний и навыков. Процесс обучения модели представлен на рисунке 6.

Совместная работа двух обученных нейронных сетей оценивалась на характеристиках ментальности студента с номером 6 (таблица 4), который не участвовал в обучении. Моделирование проводилось в соответствии с двухкаскадной схемой на рисунке 4. Анализ результата работы первого каскада показал, что значения компонент выходного вектора близки к коду (1,0,0,0). Эта кодировка соответствует экзаменационной оценке «неудовлетворительно», которую в действительности получил этот студент на экзамене (таблица 4).

Спрогнозированная оценка с выхода первой нейросети подавалась на вход второй нейросети, которая формировала результирующий вектор Y остаточных знаний и умений этого студента (рисунок 7).

Значения компонент вектора Y можно трактовать как степени уверенности в том, что у данного студента сохраняются в его памяти соответствующие знания и умения (конечно, относительно используемых обучающих множеств). Если сопоставить полученный результат с критериями оценки по учебной дисциплине «Интеллектуальные системы в экономике», то представленная на рисунке 7 совокупность спрогнозированных знаний и умений соответствует оценке «неудовлетворительно».

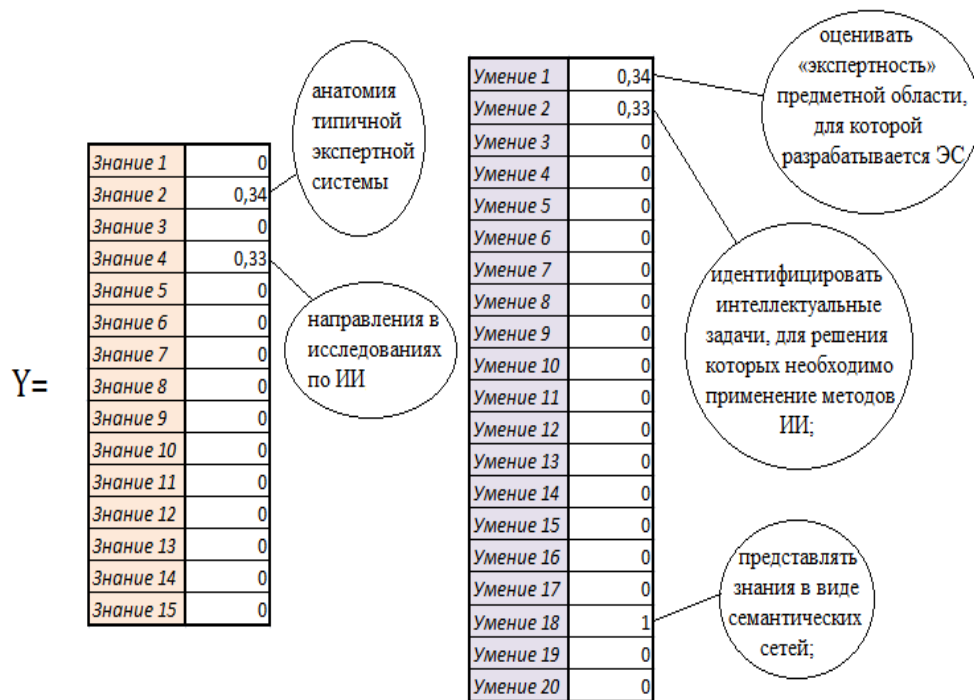


Рисунок 7 - Прогнозные остаточные знания и навыки для шестого студента по рассмотренной учебной дисциплине

Заключение

Предложен подход к нейросетевому моделированию трудно формализуемого процесса профессионального обучения молодых специалистов, основанный на имитации процесса передачи профессиональных навыков и знаний в зависимости от личностных характеристик студентов.

Установлены внешние и внутренние факторы, влияющие на успеваемость студентов и качество усваивания знаний и навыков. Особое внимание было уделено студенту как личности и его месту в процессе обучения. Исходя из этого, была разработана специальная методика, позволяющая анализировать психологические, эмоциональные, природные и физические способности студента. Методика была основана на классических психологических подходах, обладающих универсальностью, а также сравнительной легкостью для их реального применения.

После получения ментального портрета студента был разработан нейроалгоритм построения двухкаскадной нейромодели, имитирующей результат профессионального обучения путём выявления остаточных знаний и навыков студента, которые будут использованы на рынке труда.

Предварительные результаты исследования на программных моделях показали правильность предложенных идей по решению поставленной задачи.

На основе данной модели обучения будут разработаны искусственные программные агенты [Ивашкин, 2013], которые в комплексе будут

моделировать динамику процессов обучения группы студентов и их трудоустройство.

Библиографический список

- [Федяев, 2013] Федяев, О.И. Анализ и прогнозирование процесса трудоустройства молодых специалистов с помощью мультиагентной имитационной модели / Федяев О.И., Лукина Ю.Ю., Стрпалов А.С. // Труды конференции ИАИ-2013, КПИ, Киев, 2013. - С. 47-53.
- [Федяев, 2006] Федяев, О.И. Многоагентная модель процесса обучения студентов на кафедральном уровне // Наукові праці Донецького національного технічного університету. Серія "Проблеми моделювання та автоматизації проектування динамічних систем". Випуск 5 (116). - Донецьк: ДонНТУ, 2006. - С. 105-116.
- [Zhabska, 2011] Tetiana Zhabska, Oleg Fedyaev. The development of agent-based intellectual e-learning environment // Proceedings of the IADIS International conference Intelligent systems and agents 2011 Rome, Italy July 24-26, 2011/ Pages 143-147.
- [Федяев, 2010] Федяев, О.И. Проектирование виртуальной кафедры университета на основе многомодельного агентно-ориентированного подхода / Федяев О.И., Жабская Т.Е. // Искусственный интеллект. - 2010, №3. - С. 679-686.
- [Ильин, 2004] Психология творчества, креативности, одарённости / Ильин Е.П. ; - СПб.: Питер, 2004. -537 с.
- [Айзенк, 2003] Новые тесты IQ / Айзенк Г. ; - М.: Изд-во "ЭСКМО", 2003. - 189 с.
- [Дейнека, 2009] Современные тенденции в управлении персоналом. Учебное пособие / Дейнека А.В. [и др.]; - М.: Изд-во "Академия естествознания", 2009. - 294 с.
- [Круглов, 2001] Нечётка логика и искусственные нейронные сети / Круглов В.В. [и др.]; - М.: Физматлит, 2001. - 224 с.
- [Дьяконов, 2001] Математические пакеты расширения MATLAB. Специальный справочник / Дьяконов В. [и др.]; - СПб.: Питер, 2001. -268 с.
- [Ивашкин, 2013] Агентные технологии и мультиагентное моделирование: учебное пособие / Ивашкин Ю.А. - М.: МФТИ, 2013. -268 с.

NEURAL NETWORK MODEL OF THE PROCESS OF PROFESSIONAL EDUCATION OF YOUNG SPECIALISTS

Fedyayev O.I.

*Donetsk National Technical University, Donetsk,
Republic of Ukraine*

fedyaev@r5.dgtu.donetsk.ua

oleggedyayev@yahoo.com

The scientific work is devoted to the development of a neural network model of the learning process for students agent system modeling of the labor market. This model simulates the transfer of skills and knowledge, depending on the personal characteristics of students. The system will allow the simulation to analyze the process of training young professionals and to predict their future employment prospects.

Introduction

The problem of constructing a model of the professional training for the multi-agent system modeling of the labor market, which will analyze the process of preparing young professionals, as well as to predict their future employment prospects. The aim of this work is to develop a model based on a neural network capable of functionally describe the dependence of students receive professional knowledge and skills of the factors affecting the completeness of this knowledge.

Main Part

The main function of the learning process of students as the system is to transfer knowledge and develop professional skills of the future experts to solve specific production problems. The learning process is difficult to formalize, and therefore cannot be described by conventional mathematical methods. In addition, participants in the learning process geographically distant from each other, are heterogeneous in structure and activity of intelligent by nature. These features cause the feasibility of the theory of intelligent agents to the development of a simulation model for the analysis of the training process.

Have been developed methods for determining the mental and psycho-physiological characteristics of the student. These techniques combine to form a system that determines a mental portrait of a student.

The developed model of the learning process should form the output residual knowledge of the student on a separate discipline, with whom he comes into the labor market. Residual knowledge depend on the mentality of students and other factors. This relationship is difficult to formalize. In such cases it is advisable to use a neural network that will identify an existing connection through her training. Forecast residual knowledge on one particular discipline taken for one student is carried out in two stages. In the first phase is predicted on the basis of the examination score mentality of the student.

In the second stage, based on the projected estimates formed averaged residual set of knowledge and skills corresponding to this assessment. Each of these steps cannot be formalized mathematically, so the two will be used by the neural network. The first neural network will be trained on the basis of mental portrait of a group of students and the examination sheet. The second neural network - based on the evaluation criteria and curriculum discipline, which contains a list of knowledge and skills. As a medium of artificial neural networks modeling package was used Neural Network Toolbox, which is included in the standard package MATLAB.

Preliminary results of a study on software models showed the correctness of the proposed ideas for solving the problem. On the basis of this model will be developed artificial learning software agents, which together will simulate the dynamics of the processes of training a group of students and their employment.

Conclusion

An approach to the modeling of neural networks is difficult to formalize the process of vocational training of young professionals, which allows to simulate the transfer of skills and knowledge, depending on the personal characteristics of students. The authors have developed a technique that analyzes the psychological, emotional, environmental and physical abilities of the student. After receiving mental portrait student was developed neural algorithm for constructing a two-stage neural model to simulate the result of vocational training by identifying residual knowledge and skills of the student, which will be used in the labor market.



OSTIS-2015

(Open Semantic Technologies for Intelligent Systems)

УДК 004.02:519.816

METHODS OF INDISTINCT REGULATION IN MANAGEMENT PROBLEMS EDUCATIONAL PROCESS

A.H.Nishanov *, M.Yu.Doshchanova **, D.A.Mirzaev **

* *Urgench branch of the Tashkent university of information technology,
Urgench, Republic Uzbekistan*

nishanovaxram@mail.ru

malikadoshanova@mail.ru

** *The Tashkent university of information technology,
Tashkent, Republic Uzbekistan*

mdilshod@mail.ru

In management problems, educational process the big sets of alternatives appear at the decision of the subtasks connected with distribution. In work for them special methods are allocated: the method of generation of set of alternatives-distributions, and of an estimation of alternatives and a conclusion of the unique decision is developed the device of an indistinct regulator is adapted.

Keywords: decision-making, an indistinct regulator, generation of alternatives, fuzzification.

Introduction

In management problems educational process methods of an expert estimation of alternatives we will apply when capacity of set of alternatives and restrictions is not great, otherwise it is difficult to expert to execute their estimation. The big capacities of set of alternatives appear at the decision of problems of distribution.

Let's use in this case for an estimation of alternatives methods of unsharp regulation which found reflection in the device of creation of an unsharp regulator. The mechanism of an indistinct logic conclusion is put in a basis of indistinct regulators.

The given methods, as a rule, are applied at the decision of problems of automatic control.

In work it is offered to adapt them for the decision of problems of organizational management. It is connected by that distribution problems at management of educational process always are accompanied by the indistinct entrance data and the decision of problems characterized by considerable influence of the person. It confirms validity of a choice of the given device. The indistinct regulator allows estimating set of alternatives of the big capacity, at ranging which (distribution) the set of restrictions is used. The basic means of maintenance of the given possibility are solving rules, which are intended in this case for reflexion of

influence of the person on an estimation and a choice of alternatives. Thus, the choice of alternatives is carried out as a conclusion of the unique decision.

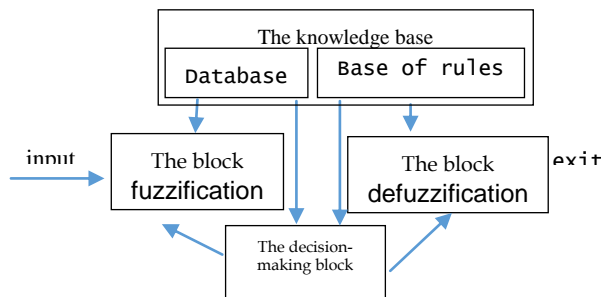
1. Introduction methods in an illegibility

Thus, for the purpose of the decision of questions of adaptation, we will consider structure of an indistinct regulator and we will describe methods, which are necessary for developing.

The indistinct regulator [1] consists of five functional blocks (drawing 1):

- The block fuzzification, reformative numerical entrance values in conformity degree to linguistic variables;
- The base of rules containing a set of indistinct rules of type "if";
- A database in which functions of an accessory of the indistinct sets used in indistinct rules are defined;
- The block of decision-making making operations of a conclusion on the basis of available rules;
- The block defuzzification, reformative results of a conclusion in numerical values.

Usually the base of rules and a database unite in the general block – the knowledge base.



Drawing 1 – Structure of an indistinct regulator

The procedure of an indistinct conclusion, which is carried out in an indistinct regulator, includes five following procedures:

- Formation of base of rules of system of an indistinct conclusion;
- Transformation entrance variable in values of functions of an accessory of elements of indistinct sets of entrance linguistic variables (fuzzification);
- Comparison among themselves values of functions of an accessory of various entrance variables for reception of weight of each rule (aggregation);
- Definition of target indistinct values in production (accumulation);
- Transformation of values of an accessory of target variables to target value (defuzzification).

For adaptation of the device of an indistinct regulator to problems of distribution of organizational management, it is necessary to consider methods and the algorithms underlying realization of first two procedures. These procedures are connected with preparation of the initial information and rules for an indistinct regulator, the others can be used without change.

Rules work with linguistic variables, which prepare in procedure fuzzification. Therefore, we will consider the methods realizing given procedure, and then – formation of base of rules of system of an indistinct conclusion.

Procedure fuzzification is realized on the basis of introduction methods in an illegibility and defined as follows.

Fuzzification is a procedure of transformation of the entrance functions of an accessory of elements of indistinct sets of the entrance linguistic variables described by a train given in values.

For performance of the given procedure it is necessary to generate elements of trains of linguistic variables [2].

The linguistic variable is described by a kind train $\langle \beta, T, U, G, M \rangle$, where β – a name of a linguistic variable; T – the set of its values (terms) representing the name of indistinct variables, area of interpretation of everyone is set U named universal set of a linguistic variable; G – syntactic procedure of formation of term-set T ; M – semantic procedure of formation of indistinct sets of elements of set T .

Let's consider the data on which basis linguistic variables are formed.

As us, the decision that methods of indistinct regulation we will use at the decision of problems of distribution the initial data of an indistinct regulator is formed on the basis of sets of alternatives-distributions is accepted.

At generation of sets of alternatives-distributions we carry out splitting of signs $X = \{x_l, K, x_k\}$ sets of objects $U = \{U^1, \dots, U^n\}$ and signs $Y = \{y_l, K, y_k\}$ sets of objects $V = \{V^1, \dots, V^m\}$. Sets X and Y have been broken into subsets: simple $X^1 \subseteq X$ and compound $X^2 \subseteq X$ signs. On the basis of subsets of signs X^1 and Y^1 classes of equivalence on sets of objects U and V are constructed and the set of alternatives-distributions $A = \{a_i \mid a_i = (u_i, v_i), u_i, v_i \text{ are described by set of signs } X^1 \text{ and } Y^1 \text{ accordingly}\}$.

The alternative is generated on conformity each other signs $x_r \in X^1$ and $y_s \in Y^1$, having accurate character. For an alternative estimation a_i it is necessary to analyse compound signs on set in rules of an indistinct regulator conformity, i.e. on maintenance of the restrictions set in a problem.

At formation of set of entrance and target linguistic variables we will use signs $x \in X$ and $y \in Y$, characterized by indistinct character and described by trains $\langle \beta, U, T, G, M \rangle$ and $\langle \beta, U, T, G, M \rangle$ accordingly.

Values of a linguistic variable are indistinct variable (terms) of term-set T .

2. Term-set formation

For formation of term-set T syntactic procedure G and semantic procedure of M are used.

According to works [1,3,4,5] procedure G has two definitions:

1. G – a syntactic rule which can be set in the form of the context-free grammar generating terms of set T by means of approximate splitting of universal set U on ranges [1].

2. G – the syntactic procedure describing process of formation from set T of new values comprehended for a given problem of a linguistic variable [3].

According to definition 1 term-set T of a linguistic variable β is formed by syntactic procedure G . By definition 2 syntactic procedure G forms new additional elements of set T of elements of in advance generated set T .

In our opinion, both definitions are necessary, definition 2 supplements definition 1, therefore we will enter the generalized definition 3.

3. Syntactic procedure G is a procedure of formation of term-set T on the basis of generation of primary terms of set T' and compound (additional) terms of set T'' , comprehended for the given problem.

Let's consider an essence of syntactic procedure G . Term-set T is a set of the indistinct variables defined on universal set U . According to an indistinct variable the kind train is called

$$\langle X, U, \tilde{X} \rangle,$$

where X - the name of an indistinct variable; U - universal set;

$$\tilde{X} = \bigvee_{u \in U} \mu_X(u)/u$$

- Indistinct set on U , describing restriction on possible numerical values of an indistinct variable X .

Definition of an indistinct variable is an appointment to it of the term - names of an indistinct variable and definition for it indistinct set \tilde{X} .

Thus, the essence of syntactic procedure G consists in the decision of the first of two problems – formation of elements X of trains of indistinct variables. The second problem dares by means of semantic procedure of M - formation of elements of trains of indistinct variables.

3. Formation of base of rules

The base of rules of system of an indistinct conclusion is intended for formal representation of solving rules of a problem of decision-making. In system of an indistinct conclusion rules indistinct productions are used.

The base of rules indistinct production represents final set of rules indistinct production, coordinated concerning linguistic variables used in them.

The coordination of rules concerning used linguistic variables means that as conditions and the conclusions of rules indistinct linguistic statements can be used only, thus in each of indistinct statements functions of an accessory of values of term-set for each of linguistic variables should be defined.

In systems of an indistinct conclusion, linguistic variables are used in indistinct statements of conditions indistinct production and are called as entrance linguistic variables. Variables, which are used in indistinct statements of the conclusions indistinct production, are called as target linguistic variables.

Thus, at the task or formation of base of rules indistinct production it is necessary to define set of rules indistinct production $P = \{R_1, R_2, \dots, R_n\}$, set of entrance linguistic variables $In = \{\beta_1, \dots, \beta_n\}$ and set of target linguistic variables $Out = \{\omega_1, \dots, \omega_n\}$. The base of rules indistinct production is considered set if sets P , In and Out are set.

Formal representation of solving rules in the form of indistinct production is based on a formalism indistinct production models.

For creation of the formalized record of a solving rule in the form of indistinct production, it is necessary to define the elements entering into its structure. Indistinct production is defined by the expression which is looking like $(i): Q; P; A \Rightarrow B; S, F, N$, where (i) - a name of indistinct production; Q - scope of application of indistinct production; P - a condition of applicability of a kernel of indistinct production; $A \Rightarrow B$ - production kernel; F - factor of definiteness of indistinct production or weight of indistinct production; N - production postcondition, describes actions and

procedures which are necessary for executing after realisation B .

The base of rules represents system indistinct production, consisting of several subsystems (blocks). Each subsystem is intended for the decision of separate subtasks. The accessory of production to a subsystem is reflected in scope of application of indistinct production.

The condition of applicability looks like the logic statement which validity defines necessity of activation of a kernel of production.

The logic statement has predicate idea and registers in the form of conjunction of the facts which defines a current situation on the entrance data. The entrance data are sets of signs $X = \{x_1, \dots, x_k\}$ and $Y = \{y_1, \dots, y_l\}$, describing distributed objects P and Q and having accurate character. The proof of the validity of statements is carried out on a method described in work[6].

The kernel of indistinct production looks like «if A , that B » in which A is a condition, and B – the conclusion. Condition A and conclusion B – some expressions of indistinct logic which are most often represented in the form of indistinct statements. As expressions A and B compound logic indistinct statements, i.e. the elementary indistinct statements connected by indistinct logic sheaves, such as indistinct negation NOT , indistinct conjunction AND and an indistinct disjunction can be used OR .

The requirement of a coordination of base indistinct production is connected with restriction on formation of elementary indistinct statements of a condition A and conclusions B . It consists that elementary indistinct statements of a condition A should be elements of set In – entrance linguistic variables, and elementary indistinct statements of conclusion B should be elements of set Out – target linguistic variables.

As linguistic variables, we will use signs $x \in X$ and $y \in Y$, characterized by indistinct character and being properties of distributed objects P and Q . And entrance linguistic variables can be both signs $x \in X$, and signs $y \in Y$, and signs $y \in Y$ as set Y describes objects Q can be target linguistic variables only. We will remind that objects Q are objects of appointment on which objects P are distributed.

Let's consider a subtask «Reduction of intensity of studying of disciplines of various cycles to one average value» problems «Formation of the curriculum of a speciality». In it on the basis of intensity of studying of disciplines of cycles (naturally – scientific disciplines – EN , the general professional disciplines - OPD , special disciplines – SD) average intensity of studying of disciplines of a cycle is calculated.

Set In of entrance linguistic variables make Int^{EN} «Intensity of studying of disciplines of cycle EN », Int^{OPD} «Intensity of studying of disciplines of cycle OPD » and Int^{SD} «Intensity of studying of disciplines of cycle SD », $In = \{Int^{EN}, Int^{OPD}, Int^{SD}\}$. A target linguistic variable is average intensity of studying of disciplines of cycle $IntAll \in Out$. The term-set of entrance linguistic

variables looks like $T = \{Low, Normal, High\}$, a term-set of a target linguistic variable – $T = \{Low, SlightlyLow, Normal, SlightlyHigh, High\}$.

Then examples of kernels indistinct production can have the following appearance:

1. If $(Int^{EN}=Low)$ and $(Int^{OPD}=Low)$ and $(Int^{SD}=Low)$ then $IntAll = Low$.
2. If $(Int^{EN}=Low)$ and $(Int^{OPD}=Low)$ and $(IntSD=Normal)$ then $IntAll=SlightlyLow$.
3. If $(Int^{EN}=Low)$ and $(Int^{OPD}=Low)$ and $(IntSD =High)$ then $IntAll=SlightlyLow$.
4. If $(Int^{EN}=Low)$ and $(Int^{OPD} =Normal)$ and $(IntSD =Low)$ then $IntAll=Low$.
5. If $(Int^{EN}=Low)$ and $(Int^{OPD} =Normal)$ and $(IntSD =High)$ then $IntAll=SlightlyHigh$.
6. If $(Int^{EN}=Low)$ and $(Int^{OPD} =Normal)$ and $(IntSD =Normal)$ then $IntAll=SlightlyLow$.
7. If $(Int^{EN}=Low)$ and $(Int^{OPD} =High)$ and $(IntSD =Low)$ then $IntAll=SlightlyLow$.
8. And so on.

Value of target linguistic variable $IntAll$ will be used further by a regulator, which is carrying out check of conformity of the list of disciplines of a semester to restrictions, imposed on a semester.

Factor of definiteness of indistinct production name weight of indistinct production. In the given work for all rules $F = I$. It means that all indistinct production of base of rules of an indistinct regulator have equal weight.

Production postcondition describes actions and procedures, which are necessary for executing after realization B . Performance N can occur right after realizations of a kernel of production.

Procedure of an indistinct conclusion includes operations of aggregation, activation, accumulation and unessential operation defuzzification.

Conclusion

In management problems, educational process the big sets of alternatives appear at the decision of the subtasks connected with distribution. In work for them special methods are allocated: the method of generation of set of alternatives-distributions, and of an estimation of alternatives and a conclusion of the unique decision is developed the device of an indistinct regulator is adapted.

The method of generation of set of alternatives-distributions is constructed on strategy of the directed search and consists in a data control on the basis of the analysis of signs of distributed sets of alternatives and allocation in them of subsets of simple and compound signs.

Methods of indistinct regulation, which can be divided into two blocks roughly, are applied to an estimation of set of alternatives-distributions in work: methods fuzzification and methods of an indistinct conclusion.

Methods fuzzification are adapted for the decision of problems of management by educational process, methods of an indistinct conclusion can be applied without changes.

As a rule, methods of indistinct regulation are used in automatic control problems. Adaptation of methods of indistinct regulation for the decision of problems of organizational management expands area of their application.

The bibliographic list

[Алтунин, 2000] Алтунин А.Е., Семухин М.В. Модели и алгоритмы принятия решений в нечетких условиях / Монография. - Тюмень: Изд-во Тюменского гос. университета, 2000. - 352 с.

[Найханова, 2004] Найханова Л.В., Дамбаева С.В. Методы формирования лингвистической переменной и управления диапазонами ее универсального множества. Теоретические и прикладные вопросы современных информационных технологий // Материалы всерос. конф. - Улан-Удэ: Изд-во ВСГТУ, 2004, - С.51-54.

[Гитман, 1998] Гитман М.Б. Введение в теорию нечетких множеств и интервальную математику. - Ч 1. Применение лингвистической переменной в системах принятия решений. // Пермь, 1998.

[Заде, 1976] Заде Л.А. Понятие лингвистической переменной и его применение к принятию приближенных решений. // М: Мир, 1976. - 165с.

[Кофман, 1982] Кофман А. Введение в теорию нечетких множеств. // М: Радио и связь, 1982. - 432с.

[Найханова, 2004] Найханова Л.В., Евдокимова И.С. Аппарат описания формальных моделей лингвистического транслятора. Теоретические и прикладные вопросы современных информационных технологий // Материалы всерос. конф. - Улан-Удэ: Изд-во ВСГТУ, 2004. - С.96-98.

МЕТОДЫ НЕЧЕТКОГО РЕГУЛИРОВАНИЯ В ВОПРОСАХ УПРАВЛЕНИЯ ОБРАЗОВАТЕЛЬНЫМ ПРОЦЕССОМ

Нишанов А.Х. *, Дошанова М.Ю. **, Мирзаев Д.А. * *

Ташкентский университет информационных технологий, г. Ташкент, Узбекистан

nishanovaxram@mail.ru

malikadoshanova@mail.ru

mdilshod@mail.ru

В задачах управления учебным процессом большие множества альтернатив появляются при решении подзадач, связанных с распределением. В работе для них выделены специальные методы: разработан метод генерации множества альтернатив-распределений, а для оценки альтернатив и вывода единственного решения адаптирован аппарат нечеткого регулятора.

Ключевые слова: принятие решений, нечёткое регулирование, генерирование альтернатив, подготовка задачи для решения методами нечеткой логики.



OSTIS-2015

(Open Semantic Technologies for Intelligent Systems)

УДК 004.02:519.816

INTELLECTUAL MEANS OF AUTOMATION OF MANAGEMENT OF TRAINING

M.Z.Babamukhamedova, M.Yu.Doshchanova, K.A.Djangazova*

** The Tashkent university of information technology,
Tashkent, Republic Uzbekistan*

malikadoshanova@mail.ru

Предложен гибридный подход реализации интеллектуального управления индивидуализированным процессом обучения на основе синергетической модели управления, применение которого в создании образовательных сред позволяет повысить эффективность функционирования среды, улучшить её адаптивные свойства, придать целенаправленный и активный характер. Особенностью предложенного подхода является создание средств автоматизации управления обучением с учётом интегративных тенденций в условиях компетентного обучения.

Keywords: computer aided control system for teaching, synergetic model of teaching control, individual teaching trajectory, intellectual control, integrated teaching.

Introduction

The present stage of evolution of the automated training systems is characterized by a steady tendency to formation of the scientifically educational environments making a basis of functioning of scientifically educational space in the conditions of formation of an information society. The methodology of construction of scientifically-educational environments is at a formation stage, is based on methodology of working out of computer training systems, systems of various forms of electronic training on the basis of a principle of the complex decision didactic, technological, information, etc. the problems directed on creation of conditions for maintenance of computer support of process of rendering of qualitative educational services.

One of important indicators of quality of educational services is degree of personification of support of process of training which is in turn defined by adaptability and controllability of system of training. The Scientifically-educational environment as the difficult organizational-technical system which functioning is directed on the individualized management of process of formation defined competences, consists of co-operating subsystems, management automation with which demands perfection of models and methods of its working out. Therefore and unresolved the problem of perfection of models and methods of automation of system engineering of management of training in scientifically-educational environments is actual.

Features of training as operated process, consist that along with formalized and semi structured the class of problems of the mixed type which use both analytical, and heuristic models of the account of preferences exists problems in the conditions of the incomplete information. Such problems are characterized by accident of external influences, aprioristic incompleteness of the information, uncertainty of the purposes. Therefore, for management of training use of artificial intelligence techniques is expedient.

Therefore, considering conceptual changes in methodology of system engineering of automated management on the basis of application intellectual a component for the decision of not formalized and badly structured problems, and also complication of didactic requirements of the individualized training, within the limits of the allocated problem, one of unresolved problems is the problem of synthesis of the hybrid scheme of intellectual transformations for automation of management by process of the individualized training.

1. Synergetic model of management of training

The analysis of a difficult complex of system of didactic requirements to the scientifically-educational environment taking into account dynamics of the basic tendencies of their development, and also the account of features of training as operated process, and tendencies of development of the theory of management of difficult systems, allows to draw a conclusion on expediency of application synergetic the approach in creation of a control system by training [1].

The Synergetic model of the management [2], which main principles organically correspond to features of process of training, allows to form operating influences on the basis of research of a tendency of self-development of the trainee. The model constructed within the limits of the given approach synergetic managements of training displays two-class structure «knowledge and abilities» with a vector of conditions (x, y) and a management vector (h, U) [3]:

$$\frac{dx}{dt} = fUy, \quad \frac{dy}{dt} = c(1-U)xy, \quad (1)$$

$$\frac{d}{dt}(Ux + (1-U)y) = \frac{h(t)}{1+r} + \frac{c-f}{1+r}(Ux + (1-U)y),$$

where $h(t)$ - speed of granting of the information, r - resistance factor to didactic process, f - forgetting factor, c - conclusion factor, U - a part of time which has been taken away on accumulation of knowledge and abilities, x, y - normalized volumes of the saved up knowledge and abilities.

Optimization of management by training is reached on the basis of the account of distribution of a vector of intelligence that provides bases of the individualized training. Realization of the offered model by means of intellectual management is carried out on the basis of the analysis of necessary intellectual transformations of the information on parametres of basic elements synergetic models.

2. The Structurally functional scheme of management

It is possible to present training process by set of the unified cyclic actions directed on management by following elements of the maintenance: an educational element (EE) - a subject matter - (SM) the competence (CMP) a set (system) competence ($SCMP$). Control over success is defined according to diagnosticity the set vector of purpose C [4]:

$$C = \{N_{HE}, U, A, M, F_M, F_S\} \quad (2)$$

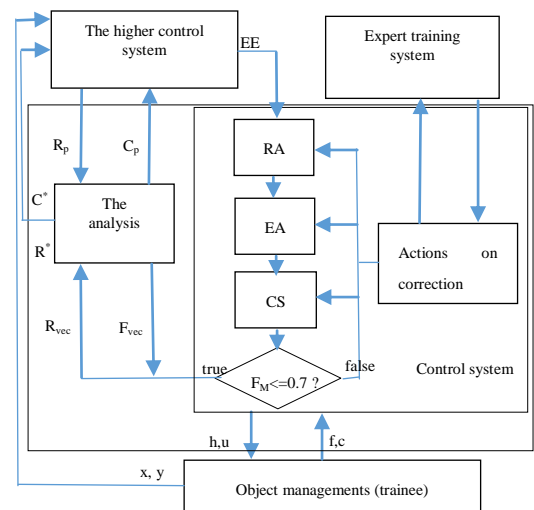
where N_{HE} - quantity of educational elements (EE); U - level of mastering $EE: U \in \{1, 2, 3, 4\}$; A - an exponent of abstraction $EE: A \in \{1, 2, 3, 4\}$; M - an exponent of sensibleness of mastering $EE: M \in \{0, 1, 2, 3\}$; F_M - factor of mastering $EE: 0 \leq F_M \leq 1$; F_S - factor of skill of mastering $EE: 0 \leq F_S \leq 1$.

All components of a vector of the purpose can be presented in a numerical kind. According to recommendations [4] purpose of training can be formulated as follows: to study the set educational elements (EE) at level of mastering of activity U with factor of mastering F_M , degree of abstraction A , factor of skill F_S at level of sensibleness M . Such formulation of the purpose is diagnostic set.

On the basis of the offered way of formalization of training it is possible to define two basic structures of object of management. In the first - «one teacher - some trainees» the teacher carries out functions of measurement of results of training of each trainee, compares their set C , the decision on necessary

operating influence makes and realizes it. In such system automation of managerial processes promotes elimination of an information overload of the teacher. Functions of such automated control system $EE - ACS$ EE (fig. 1) the following:

1. Reception of the entrance information from the higher control system - identifier EE , a vector diagnostic set purpose C , time of studying EE ;
2. Reception of the entrance information from the trainee - values of parametres of intelligence: f - forgetting factor and c - conclusion factor;
3. Maintenance of performance of training actions under the chosen scenario which obligatory stages rough actions (RA), executive activity (EA), a control stage (CS), correcting actions concern;
4. Performance of auxiliary calculations of success of achievement of the purpose - formation of vector F_{vec} and transfer of these parametres to model of the trainee;
5. Transfer to the higher control system of the information on a vector of conditions (x - relative volume of the saved up knowledge; y - relative volume of the saved up abilities).

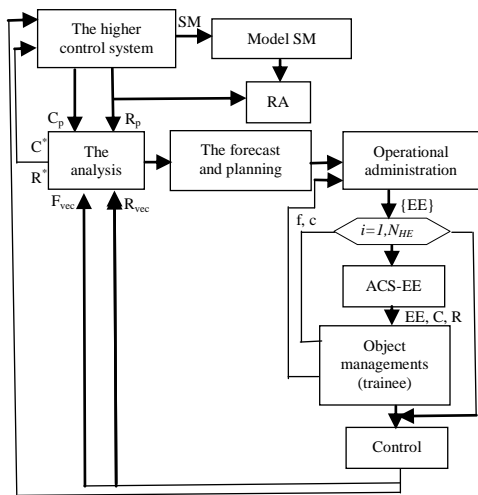


Drawing 1- Scheme ACS-EE

It is possible to carry their weak formalization, absence of means of exact measurement, heuristic character of interrelations to features of entrance, target and operated parametres between results of training and actions on time distribution. Therefore, most an effective remedy of management of such scheme is neuro-indistinct management [5].

Automation of management by process of training EE on the basis of intellectualization use is directed on reception of individual distribution of time, the account of features of a vector of intelligence of the trainee, hence - on improvement of quality of operating decisions without an overload of the teacher.

The management block this process is a component of all other levels of training. The greatest efficiency can be reached in the presence of interrelation with the expert training system, which conclusion recommendations about correction are.



Drawing 2 - Scheme ACS-SM

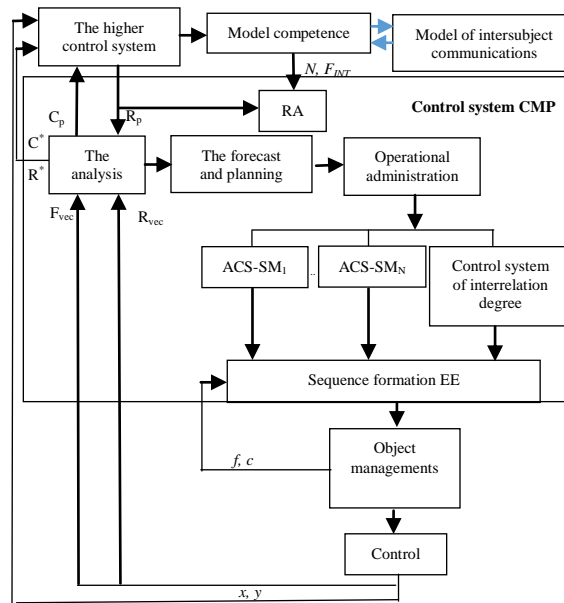
The block of management of training to a subject matter – ACS SM (fig. 2) differs from previous that contains procedures of definition of admissible sequences of training *EE*, a choice next *EE* on the basis of the account of logic interrelations, temporary restrictions. The basic functions of the block the following:

1. Reception of the entrance information from the higher control system - logic structure *SM*: list *EE* with a vector diagnostic set purpose *C*, time of studying *SM*;
2. Sequence Definition (generally - several) *EE* which studying makes an overall aim of studying *SM*;
3. Forecasting of achievement of the purpose for set time the concrete trainee on the basis of its characteristics of development of a material;
4. Planning of sequence *EE* on the basis of the account of restriction of resources (for example, time);
5. Support of process of an operational administration, as cyclic process of a consecutive call ACS - *EE*;
6. Control of success of achievement of the purpose - formations of vector F_{vec} , real time of training R_{vec} and transfer of these parametres to the analysis block;
7. Transfer to the higher control system of the information on a vector of conditions (x - relative volume of the saved up knowledge; y - relative volume of the saved up abilities);
8. Divergence Definition between planned and actual indicators:

$$C^* = (C_p - F_{vek}) / C_p; \quad R^* = (R_p - R_{vec}) / R_p. \quad (3)$$

Except features of managerial process *EE*, it is necessary to carry procedures of the forecast and planning which are expedient for carrying out on the basis of application of means of intellectual management to features of management *SM*. A basis for decision-making on formation of recommended sequence *EE* is the information received from experts-teachers. This information is indistinct, characterizes degree of logic interrelation between *EE*. Therefore, procedures of the forecast and planning demand application intellectual a component of converters of the information.

The second structure of object of management - «some teachers - one pupil». Here as the training purpose the maintenance component, which should be generated at the given grade, level should be used. Management of formation process competences - ACS-CMP (fig. 3) is in relation to previous (ACS-SM) - the higher system. The Structurally functional scheme of this block contains calls subordinated ACS-SM those disciplines which studying leads to formation of the certain competence. Feature of the given scheme is the interrelation between system of intersubject communications and model of the competence, which realizes the corresponding intellectual converter on the basis of neyro-indistinct clustering.



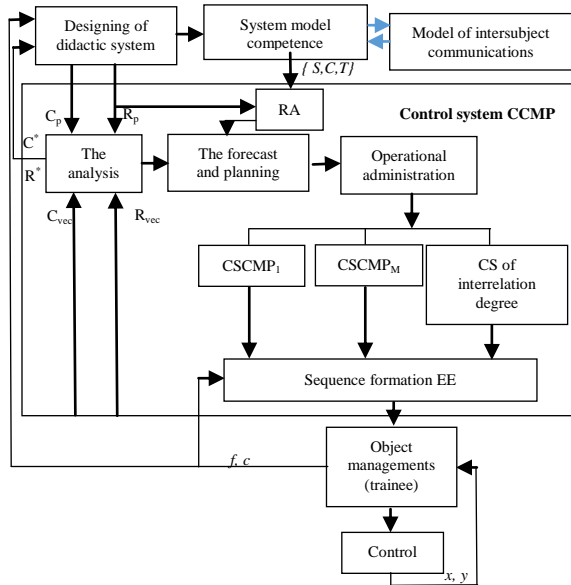
Drawing 3 - Scheme ACS-CMP

The basic functions, which are carried out ACS-CMP, the following is:

1. Reception from the senior control system of the identifier of the formed competence;
2. Definition on the basis of model competence identifiers of corresponding subject matters, the purposes, training terms;
3. Definition of the most expedient intersubject communications for formation of the certain competence, transfer to the count of training of value of factor of integration F_{INT} ;
4. Forecasting of achievement of the competence in a definite time;
5. Planning of vectors of the purposes and training time on each subject matter, their correction on the basis of use of intersubject communications and redistribution;
6. Formation of individual sequence *EE* on the basis of the account of a vector of intelligence of model of the pupil;
7. Call of subordinated systems ACS-SM for each of certain subject matters;
8. Realization of regulation by degree of interrelations during formation of an individual trajectory of training;
9. Control over degree of achievement of the

certain competence, definition of actual indicators of time, achievement of the purpose, a condition vector (x, y).

Management of process of formation of system competence - ACS-CCMP (fig. 4) is carried out on the basis of sequence of calls of subordinated control systems ACS-CMP with use of the intellectual converter which for each quantum of time forms the best, from the point of view of the account of parameters of a vector of intelligence, influence on competence formation.



Drawing 4 - Scheme ACS-CCMP

Thus, control system ACS-CCMP carries out transformations of the entrance information on the purpose, training time to the information for choice systems ACS-CMP, that is works as the switch. ACS-CCMP carries out following basic functions which define its structure:

1. Definition of requirements to projected didactic system for system formation competence according to the qualifying characteristic;
2. Formation of the maintenance for realization of process of formation of a set competence $\{S\}$;
3. Definition of system of intersubject communications, which according to experts are the most expedient at formation corresponding competence;
4. Call of subordinated systems ACS-CCMP and a control system of interrelation degree;
5. Formation of individual sequence EE ;
6. Optimization of the individualized distribution of time for studying of the interconnected subject matters, forming the competence;
7. Transfer to the higher system (in environment which forms requirements to system competence) result of a divergence with diagnostic an object in view of training and allocated time of training.

Thus, management realization by all complete process of training as operated, it is carried out on the basis of the enclosed structure of calls of the subordinated control systems realizing training according to hierarchy of objects of training. The

detailed description of each of systems features of their entrance and target data allow to draw a conclusion concerning expediency of automation of all process on the basis of realization of intellectual transformations. One of the basic elements providing an individualization of training, the control system of degree of integration of the maintenance for which functioning working out of structurally parametrical model of system of intersubject communications is necessary is.

The entrance data for construction of the count of training are structures of subject matters, which in the given statement of problems are formed on the basis of the unified model of the monosubject subject matter having hierarchical structure which parameters are defined by degree of intrasubject interrelations.

The model allows on the basis of the received opinions of experts on expediency of interrelations between EE in the form of binary indistinct relations to display them in the form of the indistinct count. The analysis of features of such count has allowed to allocate four basic types of the tops, which composition by certain rules indistinct product allows to form educational blocks automatically. The educational block represents auxiliary logic formation in structure SM . Parametrical filling of structure of the educational block (EB) is carried out on the basis of application max-min compositions of indistinct binary relations.

The received model makes a basis for the further formation of model of system of intersubject communications.

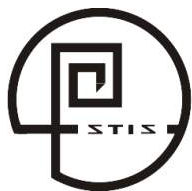
Conclusion

The hybrid approach of realization of intellectual converters of the structurally functional scheme of automation of a control system is offered by the individualized training constructed on a basis synergetic of model.

Feature of the offered approach is creation of means of automation of management by training with the account integrative tendencies in conditions competent training. Efficiency of use of the offered approach is investigated.

The bibliographic list

- [Князева, 2007] Князева Е.Н., Курдюмов С.П. Основания синергетики. Человек, конструирующий себя и свое будущее. – М.: КомКнига, 2007. – 232 с.
- [Колесников, 2006] Колесников А.А. Синергетические методы управления сложными системами: Теория системного синтеза. – М.: КомКнига, 2006. – 240 с.
- [Мазурок, 2010] Мазурок Т.Л. Синергетическая модель индивидуализированного управления обучением // Математические машины и системы, 2010, № 3, С.124-134.
- [Беспалько, 2002] Беспалько В.П. Образование и обучение с участием компьютеров (педагогика третьего тысячелетия). – М.: МПСИ, 2002. – 325 с.
- [Мазурок, 2010] Мазурок Т.Л. Нейро-нечёткая реализация синергетического управления индивидуализированным обучением // Искусственный интеллект. – 2010. - №4. – С. 596-605.



УДК 658.512.22:004.9

КОНЦЕПТУАЛЬНОЕ ЭКСПЕРИМЕНТИРОВАНИЕ В ПРОЕКТИРОВАНИИ КОНФИГУРИРУЕМЫХ ШАБЛОНОВ АВИАЦИОННЫХ ДЕТАЛЕЙ

Соснин П.И., Чоракаев О.Э.

*Ульяновский государственный технический университет,
г. Ульяновск, Российская Федерация*

sosnin@ulstu.ru

olegchorakaev@yandex.ru

В статье представляются средства структурного проектирования конфигурируемых шаблонов авиационных деталей. Специфику подхода к проектированию такого вида оснастки определяет использование средств концептуального экспериментирования, в основу которого положено отображение состояний жизненного цикла шаблона на семантическую память вопросно-ответного типа.

Ключевые слова: концептуальное экспериментирование, проектирование, семантическая память, шаблон .

Введение

В авиастроительном производстве для изготовления, контроля и сборки деталей фюзеляжа, крыльев и элерона, включая детали их обшивки, широко используется шаблонное оснащение, в состав которого входят десятки тысяч шаблонов разной степени сложности и назначения. Этот факт обусловлен такими особенностями деталей названного класса, как сложность их геометрических форм, малая жесткость, большие габариты, высокие требования точности изготовления и точности увязки. Кроме того, для увязки деталей, входящих в каждое плоское сечение конструкции самолета, необходима система жестких носителей, фиксирующих контуры внутренних деталей, входящих в состав данного сечения. Так, например, большие по величине детали приходится увязывать на листовых металлических контрольно-контурных шаблонах.

В общем случае, шаблон, не только является носителем геометрии и формы детали, но также включает конструктивные и технологические базы, контуры и оси внутренних деталей, попавших в данное сечение, различные конструктивные и технологические отверстия. Кроме того на шаблоны наносится различная информация: название шаблона, шифр и номер чертежа изделия, марка и толщина материала, указания о линиях сгиба и малки борта, контуры отверстий облегчения, маркировка отверстий и другая информация.

В проектировании металлических листовых шаблонов общего случая приходится творчески решать задачи их конструктивного конфигурирования, учитывающие создание технологических условий [Босинзон, 2010] для применения шаблонов в изготовлении, контроле и сборке авиадеталей. С шаблонами общего случая, в статье связан класс «конфигурируемых шаблонов», примеры которых приведены на рисунке 1.

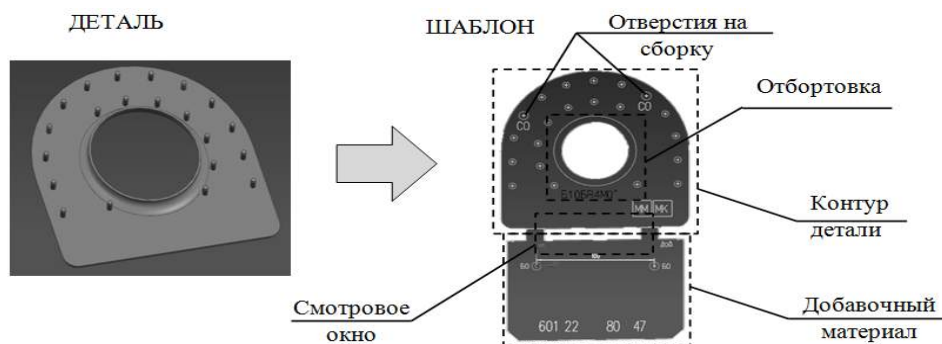


Рисунок 1 - Образец шаблона и детали

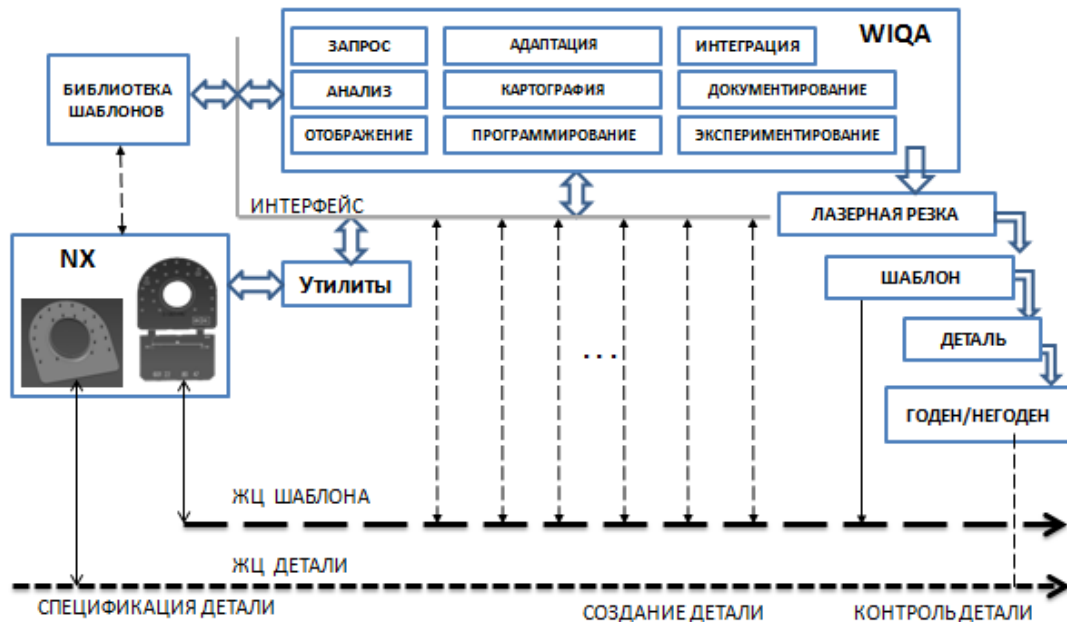


Рисунок 2 - Жизненный цикл конфигурируемого шаблона

Образец шаблона представлен вместе с деталями для демонстрации того «дополнительного», что пришлось включить в шаблон его разработчиком. Следует отметить, что геометрия конфигурируемого шаблона существенно сложнее геометрии детали, для работы с которой он создан.

Ниже для структурного проектирования конфигурируемых шаблонов предлагается ряд средств, предоставляющих разработчикам таких шаблонов проводить эксперименты с их семантическими моделями, в том числе с моделями, имитирующими изготовление шаблонов с использованием числового программного управления лазерной резки [Colombo, 2013]. Для проведения экспериментов предлагается использовать вопросно-ответную моделирующую среду WIQA, средства которой были адаптированы к инструментальной поддержке жизненного цикла шаблонов [Соснин, 2014].

Обобщённая схема структурного проектирования шаблонов

Выбор и реализация той части структуры шаблона, которая дополняет включенную в шаблон форму (и размеры) определенного сечения детали, выводит на ряд проектных задач, допускающих альтернативные решения. Даже в тех случаях, когда для проектирования шаблонов применяется библиотека их «лучших образцов», которые подтвердили свою «рациональность» в производстве уже освоенных типов самолетов, в переходе к производству нового самолета шаблонную оснастку придется создавать заново, разумеется, используя освоенные шаблоны, модифицируя их, и разрабатывая новые шаблоны. То есть творческий характер проектирования шаблонов и их взаимосвязки останется [Сосонкин,

2008]. Именно такое положение дел и подсказывает целесообразность включения экспериментов (не с шаблонами, а с их моделями) в жизненный цикл проектирования (особенно сложных, конфигурируемых) шаблонов.

Предлагаемые средства структурного проектирования шаблонов и схема их комплексирования (в контексте жизненного цикла шаблона) приведены на рисунке 2.

Проектирование шаблона начинается после того, как спецификации детали нашли свое геометрическое представление в NX-системе (ранее «Unigraphics»). Информация о детали, включая геометрию той плоскости детали, для которой разрабатывается шаблон, с помощью специальной утилиты передается в инструментально-моделирующую среду WIQA, во-первых, для отображения детали на семантическую память этой системы [Соснин, 2014], а, во-вторых, для анализа этой информации и формирования запроса к библиотеке шаблонов.

На текущий момент времени библиотека шаблонов находится за рамками среды WIQA и содержит образцы шаблонов, ответственность за выбор которых и загрузку в NX несёт проектировщик. После этого, геометрия подходящего образца шаблона также загружается в семантическую память среды WIQA. Подчеркнём, что на тех интервалах жизненного цикла детали и жизненного цикла исходной модели, которые уже рассмотрены, действия по проектированию шаблона должны осуществляться в соответствии со стандартом СТП 687.07.0873-2004. Отличие только в том, что все эти действия и последующие (в процессе жизненного цикла шаблона) выполняются под управлением (псевдо-кодовой) программы, в среде WIQA, причём отображения детали и

исходной версии шаблона выполняют роль входных данных для осуществления действий непосредственно в моделирующей среде.

Инструментально-технологическая среда структурного проектирования шаблонов

Прежде чем продолжить представление предлагаемых в статье средств структурного проектирования, раскроем причины выбора инструментария WIQA и его специфику с позиций разработки конфигурируемых шаблонов. Система WIQA разработана для инструментальной поддержки концептуального экспериментирования [Соснин, 2008] проектировщиков в процессах решения проектных (но не только) задач. Для проектировщика любой осуществляемый им концептуальный эксперимент – это мысленный эксперимент, содержание и процесс которого оперативно отображаются на семантическую память, а результаты отображения используются по ходу экспериментирования с определенными полезными целями. Средства и специфика концептуального проектирования, ориентированного на его отображение на семантическую память инструментария WIQA, детально раскрыты в публикациях [Соснин, 2014] и [Sosnin, 2014].

Особенности семантической памяти инструментария WIQA раскрывают структура и содержание ячейки памяти, представленной на рисунке 3 и ориентированной на загрузку вопросно-ответных единиц (вопроса или ответа или их композиции).

Спецификации ячейки включают богатую атрибутику, семантика которой может быть

доопределена проектировщиком с помощью прикрепления к любой из ячеек дополнительных атрибутов файлов и ссылок, если в этом появляется необходимость. Но особо важным для QA-памяти является то, что над ней определен расширяемый (при необходимости) псевдо-кодовый язык программирования L^{WIQA} . Программы на этом языке могут исполняться с помощью пооператорного (пошагового) интерпретатора, интерпретатора с компиляцией выбранных групп операторов и компилятора, причем, проектировщик в таком исполнении выполняет роль «интеллектуального процессора» [Соснин, 2014], взаимодействующего с компьютерным процессором.

Применение программных средств инструментария WIQA к задаче проектирования конфигурируемых шаблонов потребовало расширить язык L^{WIQA} за счет включения в него грамматику специального раздела данных (для спецификации шаблонов разных типов) и операторов (управления процессом проектирования, и имитации лазерной резки в формах, подобных ее осуществлению по программам числового управления). Более того, в работе с псевдо-кодовыми аналогами команд числового управления лазерной резкой поддерживается две интерпретации, одна из которых используется для имитации резки в поле специализированного графического редактора инструментария WIQA. А вторая интерпретация позволяет перейти от псевдо-кодовой программы, имитирующей определенную резку, к ее версии в рабочих G-кодах с инструкциями для подготовки к резке и оперативному взаимодействию с лазерным оборудованием, что демонстрирует пример, приведенный на рисунке 4.

Проектировщик D

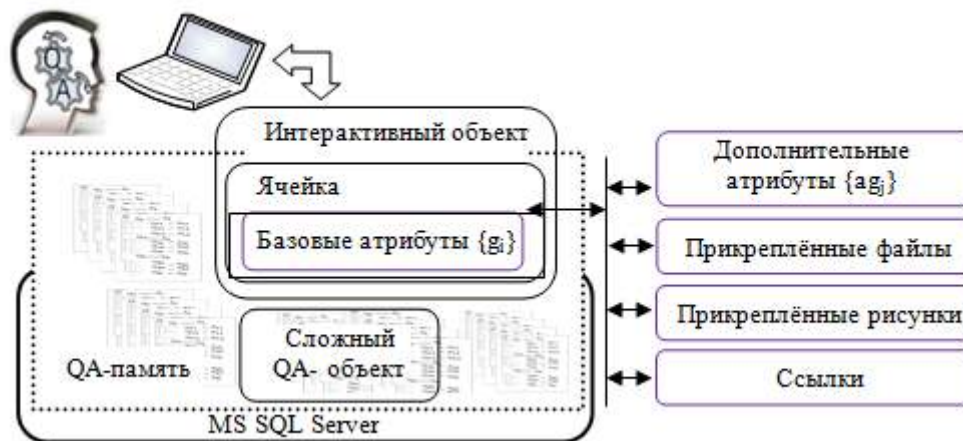
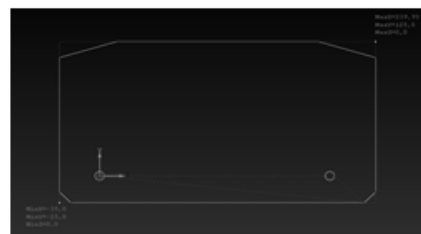


Рисунок 3 - Структура ячейки вопросно-ответной памяти

НАЧАЛО ()
 Установить мощность (СКВОЗНОЙ РЕЗ)
 ПОЛИЛИНИЯ по массиву точек
 ОКРУЖНОСТЬ С ЦЕНТРОМ () и диаметром 8
 ОКРУЖНОСТЬ С ЦЕНТРОМ () и диаметром 8
 Установить мощность (РИСКИ)
 ЛИНИЯ()
 НАНЕСТИ НОМЕР ДЕТАЛИ
 Установить мощность (ИНФОРМАЦИЯ)
 ОБОЗНАЧЕНИЕ БО
 ОБОЗНАЧЕНИЕ КРОМКИ
 КОНЕЦ()



```
N1 G17 G40 G21 G71 G90 G94 G49 G61 F2000
N2 S1000
N3 G0 X229.93 Y-25 F2000
N4 M3
N5 G1 X239.93 Y-15 F1130
N6 G1 X239.93 Y110
N7 G1 X189.93 Y125
N9 G0 X15 Y125 F2000
N11 G1 X-35 Y110 F1130
N12 G1 X-35 Y-15
N13 G1 X-25 Y-25
N14 G1 X229.93 Y-25
N16 G0 X204 Y0 F2000
N18 G3 X204 Y0 I-4 J0 F1130
N20 G0 X4 Y0 F2000
N22 G3 X4 Y0 I-4 J0 F1130
N24 G1 X0 Y0 F2000
N25 M30
```

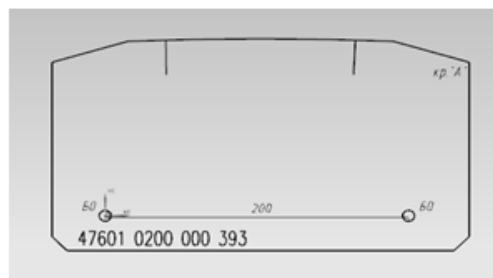


Рисунок 4 - Пример псевдо-кодовой программы с ее аналогом в G-кодах

На уровне геометрии шаблонов и примитивов лазерной обработки расширение языка включает следующий набор правил (в символике расширенной БНФ-нотации):

КОНТУР = {ПРИМИТИВ}.

ПРИМИТИВ = ОТРЕЗОК | ДУГА | СПЛАЙН.

ТЕКСТОВЫЙ БЛОК = {СИМВОЛ}, ТОЧКА ЧИСЛО

ОТРЕЗОК = 'L(' ТОЧКА ',' ТОЧКА ')'

ОТРЕЗОК = ['G0 X', ЧИСЛО 'Y', ЧИСЛО 'J', 'G1 X', ЧИСЛО 'Y', ЧИСЛО.

ДУГА = 'A1 (' ТОЧКА ',' ТОЧКА ',' ТОЧКА ',' ЧИСЛО ') | ОКРУЖНОСТЬ.

ДУГА = 'G3 X', ЧИСЛО 'Y', ЧИСЛО 'I' ЧИСЛО 'J' ЧИСЛО | 'G4 X' ЧИСЛО 'Y' ЧИСЛО 'I' ЧИСЛО 'J' ЧИСЛО.

ОКРУЖНОСТЬ = 'C1(ТОЧКА ',' ЧИСЛО ')'

СПЛАЙН = 'SP(' ТОЧКА ',' {ТОЧКА ','} ТОЧКА ')'

СПЛАЙН = {'G3 X', ЧИСЛО 'Y', ЧИСЛО 'I', ЧИСЛО 'J' ЧИСЛО | 'G4 X', ЧИСЛО 'Y', ЧИСЛО 'I', ЧИСЛО 'J', ЧИСЛО }.

ТОЧКА = (' ЧИСЛО ',' ЧИСЛО ')'

ЧИСЛО = {ЦИФРА}.

ЦИФРА = '0' | '1' | '2' | '3' | '4' | '5' | '6' | '7' | '8' | '9'.

СИМВОЛ = 'a' | 'b' | ... | 'Я'.

У правил две интерпретации, одна из которых является «декларативной» и используется для построения геометрических форм и символьных спецификаций «чертежа» шаблона, а вторая используется как средство псевдо-кодового программирования лазерной обработки заготовки, подготовленной для шаблона. В этом плане отображение шаблона на семантическую память используется как исходный код псевдо-кодовой программы декларативного типа. Именно такая интерпретация фигурно-символьного представления шаблона привела авторов к решению ввести в

процесс проектирования шаблона его картографической модели [Чоракаев, 2014].

Картографическое представление шаблонов

Картографическое представление шаблона подразумевает, что документ определяющий шаблон составляется путем добавления набора слоев, каждый из которых объединяет данные, собранные по определенному признаку. Порядок отображения слоев зависит от его положения, слои находящиеся внизу списка отображаются первыми, остальные слои – сверху.

На рисунке 5 для одного из шаблонов, приведенного выше, его картография разбивается на слои, исходя из очередности его изготовления на станке лазерного раскроя:

- слой 1 содержит заготовку с координатной плоскостью (**Z**);
- слой 2 содержит элементы внешнего сквозного контура (Gr_i^{HDO});
- слой 3 содержит элементы внутреннего сквозного контура (Gr_i^{HDI});
- слой 4 содержит элементы, отображаемые рисками (Gr_i^{PD});
- слой 5 содержит текстовые блоки (I_{text}).

Такое разбиение на слои является типовым для любых конфигурируемых шаблонов, но в процессах экспериментирования полезно выделять слой дополнительных элементов, связанных с условиями создания шаблонов, например, их контроля, или их применения для позиционирования детали и шаблона, а также для различных видов крепления, например, такие элементы как пластины и упоры.

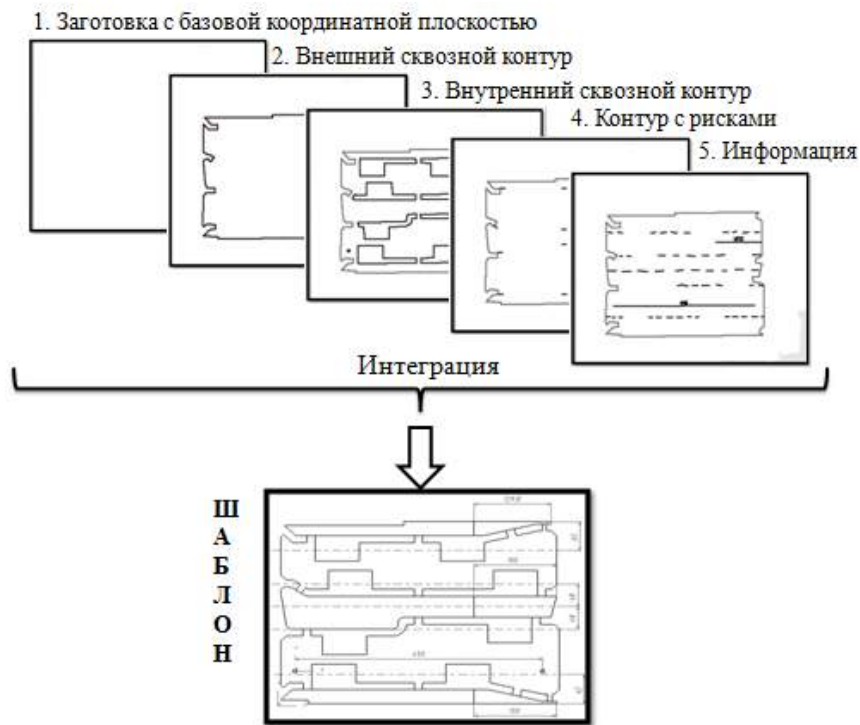


Рисунок 5 - Картографическое представление шаблона

Для объединения элементов в картографический слой используется функция селекции S^k , а для совмещения (проецирования) картографически-ориентированных слоев вводится операция «», по аналогии с конкатенацией в программировании. Тогда шаблон представляется следующим образом:

$$sh = [S^k(Z)][S^k(G_{det})][S^k(G_{add})][S^k(I_{text})][S^k(G_{oe})]$$

Картографическое описание шаблона **SH** в конечном итоге должна быть отображена на выходных документах, основными из которых является *программа обработки* – отображает слой для сквозной вырезки и нанесения рисок, различные только мощностью лазерного луча и *эскиз для расчетно-технологической карты* – графическое представление шаблона (рис.6)

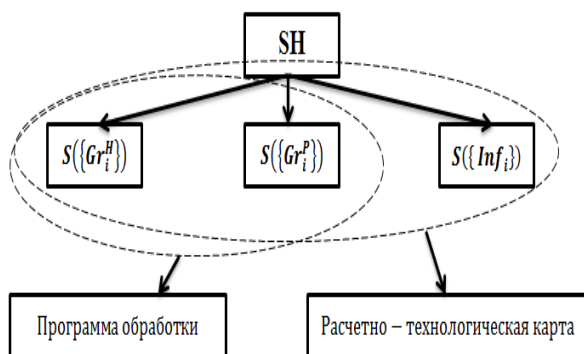


Рисунок 6 - Преобразование картографической модели

В отличие от детали при проектировании шаблонов приходится учитывать условия, в которых шаблон будет использоваться для изготовления детали, ее контроля и при сборке с другими

детальями, а также условия изготовления самого шаблона. Для учета описанных факторов требуется инструмент для проверки решений на модели, а не в металле, где каждое отклонение приводит к потере материальных ресурсов. Картографически-ориентированный подход позволяет уменьшить объем экспериментирования, применяя представленное выше выделение картографических слоев.

При разработке шаблона, представленного на рисунке 5, приходится начинать с выделения контролируемых контуров детали и размещения мысленных копий детали в пространстве (рисунок 7). Эти контролируемые контуры проецируются на плоскость листа заготовки шаблона с учетом размеров стандартных заготовок детали, для ее последующей обрезки, что предполагает размещения на шаблоне упоров. Для совмещения шаблона с контролируемой или изготавливаемой деталью применяются струбцины, под которые в теле шаблона необходимо предусмотреть окна. На следующих этапах проектирования, конструктивные элементы объединяются в одно тело с помощью добавочного материала, который служит для крепления рабочих контуров между собой, для придания жесткости конструкции, размещения информации и базовых отверстий для контроля самого шаблона.

Так что при проектировании шаблона приходится работать со слоями в порядке, который отличается от последовательности слоев на рисунке 4, рекомендуемой для их изготовления с помощью лазерной резки.

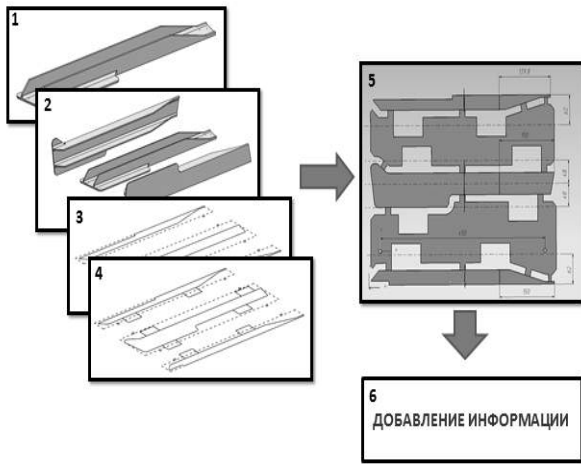


Рисунок 7 - Применение картографии в процессе проектирования шаблона

Концептуальное экспериментирование

Как уже отмечалось выше, отображение на семантическую память открывает возможности для использования концептуальных экспериментов в решении проектных задач при проектировании конфигурируемых шаблонов. Более того, в предыдущем пункте было отмечено, что проектные решения целесообразно связать с теми функциями, которые шаблон должен выполнять в жизненном цикле соответствующей детали (изготовление детали, использование шаблона в контрольных и сборочных операциях). Ряд проектных решений должен быть связан с изготовлением шаблона и его

контроля на соответствие производственно-технологическим требованиям. Но, следует отметить, что принципиальными из этих решений будут те, которые связаны с формированием (геометрической) структуры шаблона. Так что к числу принципиальных задач концептуального экспериментирования следует отнести задачи связанные с включением в шаблон фрагментов его геометрической структуры.

Для работ с плоскими геометрическими структурами в инструментарий WIQA включен специализированный графически редактор, потенциал которого достаточен для переноса в его рабочее поле плоских геометрических структур из NX (Unigraphics), то есть плоских изображений деталей и шаблонов, причём с учётом размерностей и даже в версии G-кодов. Другими словами, отрезки линий и кривые в среде графического редактора WIQA являются программируемыми. Более того, структуры в поле графического редактора WIQA можно представить псевдо-кодowymi программами, исполняя которые можно построить рисунок запрограммированной структуры повторно. То есть, структуру можно отобразить на псевдо-кодovou программу, а исполнив псевдо-кодovou программу (в операторах геометрического программирования системы WIQA) можно построить закодированную в программе геометрическую структуру. Именно такое взаимное отображение лежит в основе концептуального экспериментирования с фрагментами шаблонов. Одна из задач выбора по результатам экспериментов приведена на рисунке 8.

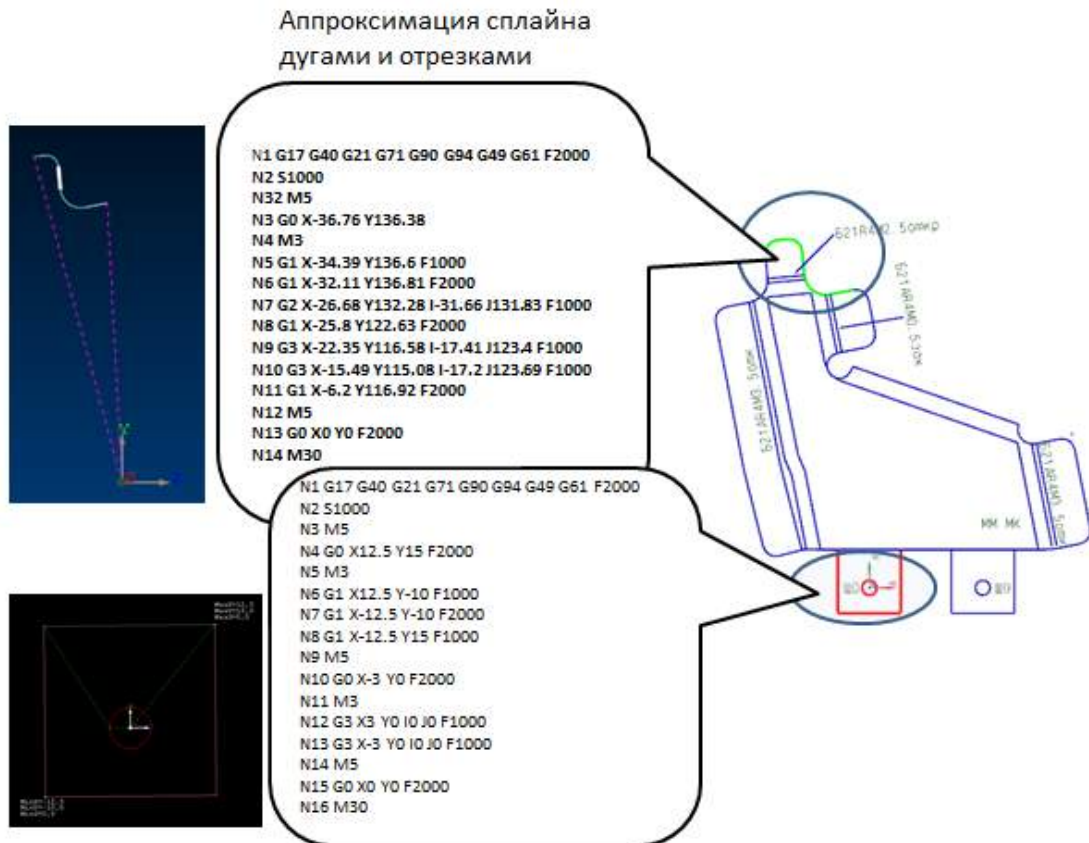


Рисунок 7 - Две версии аппроксимация сплайн

На рисунке специально приведены программы лазерной резки для выделенных фрагментов для того, чтобы подчеркнуть тот факт, что (с помощью предлагаемой системы средств структурного проектирования шаблонов) псевдо-код фрагмента геометрической структуры может быть автоматически переведён на язык G-кодов. Как выглядит псевдо-код лазерной резки было показано выше на рисунке 4.

Формирование документов

Схема на рисунке 6 отражает тот факт, что по ходу проектирования шаблона требуется формировать определенную нормативную документацию, в состав которой входит и расчетно-технологическая карта (РТК), отображающую как геометрию спроектированного шаблона, вариант

применения (эскиз приложения шаблона к детали), так и сопутствующую информацию, вносимую в определенные ячейки табличного документа.

Документирование является одной из базовых функциональностей инструментария WIQA, для осуществления которой любые также документные формы отображаются на семантическую память, причем единицы отображения доступны для их использования в псевдо-кодовом программировании. Так что любой результат экспериментирования, если он оказался рабочим и встраивается в проект шаблона, можно оперативно переносить в соответствующие документные формы. Схема документирования РТК, осуществляемого в среде WIQA, приведена на рисунке 6.

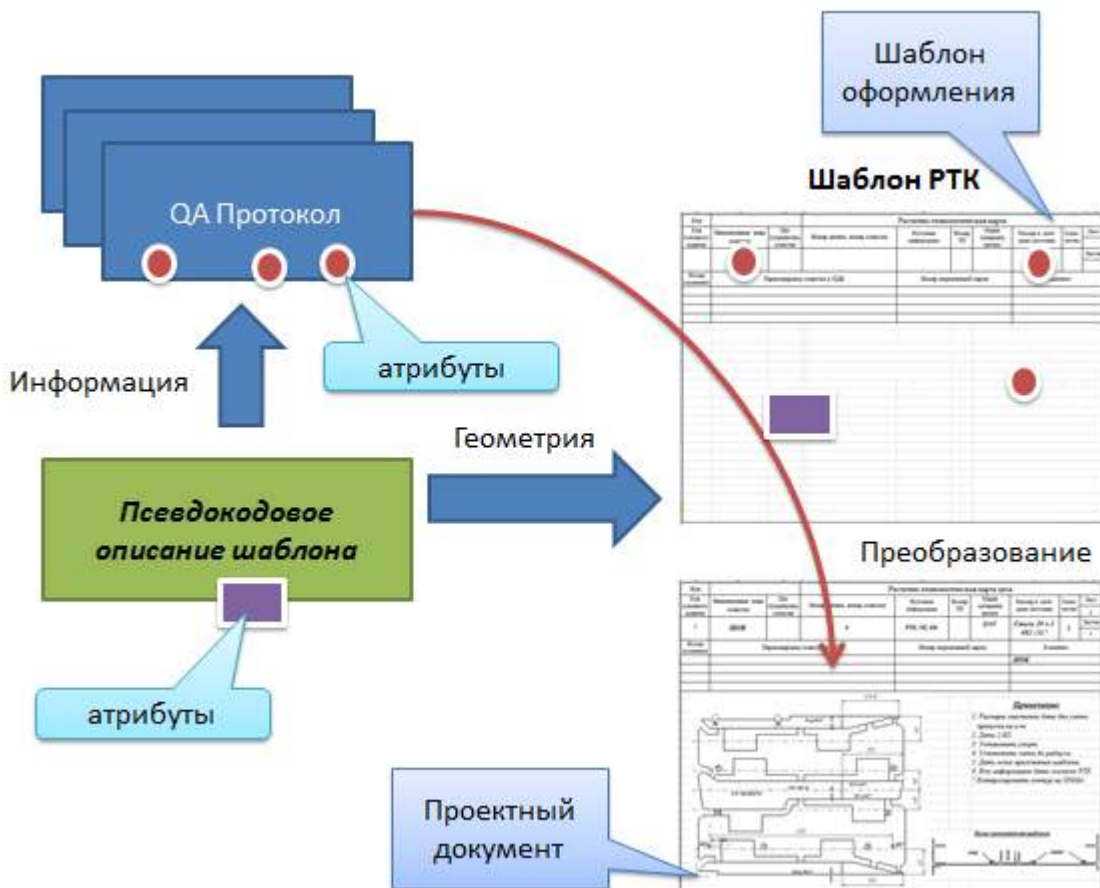


Рисунок 3 - Формирование РТК путем связывания шаблона документа и данных

Схема демонстрирует, что оперативное документирование в среде WIQA переносится в QA-память, причем для каждого типового документа формируются вопросно-ответное представление его структуры и шаблон визуального представления документа по образцу его твердой копии.

Заключение

В статье представлен подход к структурному проектированию шаблонов авиационных деталей, в процессе которого для проектировщиков открыты

возможности концептуального экспериментирования с геометрическими фрагментами шаблонов, включение которых в их конфигурацию предназначено для выполнения производственно-технологических функций изготовления и контроля деталей, а также их сборки в более сложные узлы. Такие возможности обеспечиваются за счёт использования в проектировании инструментально-моделирующей среды WIQA, в функционал которой введены средства отображения геометрии детали и шаблона

на семантическую память в виде, допускающем интерпретацию геометрических фрагментов шаблонов как декларативных псевдо-кодовых программ. Открыты две версии исполнения программ, одна из которых осуществляется в формах имитационного моделирования, а вторая преобразуется в программу числового управления лазерной резки запрограммированного геометрического фрагмента.

Включение представленных в статье средств в процессы проектирования конфигурируемых шаблонов способствует повышению степени автоматизации при решении творческих проектных задач, предотвращению и обнаружению ошибок проектирования, а также повторному использованию полезных проектных решений, в том числе и построенных шаблонов. Для решения задач повторного использования важным направлением является создание эффективной в применениях библиотеки шаблонов, вопросы о которой исключены из содержания статьи.

Список литературы

- [Босинзон, 2010] Босинзон М. А. Современные системы ЧПУ и их эксплуатация. М.: Академия, 2010. 192 с.
- [Соснин, 2014] Соснин П.И. Онтологическая Поддержка Концептуального Экспериментирования в Вопросно-Ответных Моделирующих Средах / Труды Конгресса по интеллектуальным системам и информационным технологиям», Научное издание в 4-х томах. – М.: Физматлит, 2014. – Т.1. с. – 488-495 с.
- [Соснин, 2008] Соснин, П. И. Концептуальное моделирование компьютеризованных систем/ П. И. Соснин. – Ульяновск: УлГТУ, 2008. – 198 с.
- [Сосонкин, 2008] Сосонкин В. Л., Мартинов Г. М. Программирование систем числового программного управления. М.: Логос. Университетская книга, 2008. 344 с.
- [Чоракаев, 2014] Чоракаев О.Э., Соснин П.И. Подход к формальному описанию структуры шаблонов авиационных деталей. Материалы Российской конференции «Информатика, Моделирование, Автоматизация, Программирование» – Ульяновск: Изд-во УлГТУ, 2014, с. –211-215.
- [Colombo, 2013] Colombo R.M., Guerra G., Marcellini F., Herty M. A hyperbolic model for the laser cutting process. Applied Mathematical Modelling, 37 (2013), 14-15, 7810-7821
- [Sosnin, 2013] Sosnin P., Pseudo-Code Simulation of Designer Activity in Conceptual Designing Of Software Intensive Systems Proceedings - 27th European Conference on Modelling and Simulation, ECMS 2013, 2013. С. 85-92.
- [Sosnin, 2014] Sosnin P., Lapshov Y., Svyatov K., Programmable Managing of Workflows in Development of Software-Intensive Systems, A. Moonis et al. (Eds.): IEA/AIE 2014, Part I, LNAI 8481, pp. 138–141, 2014.

CONCEPTUAL EXPERIMENTING IN DESIGNING THE CONFIGURED PATTERNS OF AVIATION PARTS

P.I. Sosnin*, O.E. Chorakaev*

**Ulyanovsk State Technical University,*

Ulyanovsk, Russia

sosnin@ulstu.ru

olegchorakaev@yandex.ru

Key words: configured pattern, conceptual experimenting, designing, semantic memory.

Introduction

The paper presents a set of means for structural designing the configured patterns of aviation parts. Specificity of the offered approach to designing this kind of an equipment is defined with the use of means of conceptual experimenting. Such type of experimenting suggests a reflection of the life cycle of any pattern to a semantic memory of the question-answer type. The structure and attributes of the semantic memory cells are focused on coding of models of questions and answers from the viewpoint of their use in solving the creative tasks. In such modeling, any conceptual experiment is conducted as automated mental experiment the structure and content of which are operatively used in designing of patterns.

Main Part

Configured patterns as a specialized equipment for manufacturing and checking of aviation details have a set of features the main of which are the following: complicated geometrical forms, small rigidity, greater scales, strict requirements to accuracy of manufacturing.

Moreover, in the creation of each pattern, it should repeat not only the geometrical forms of the corresponding detail, but needed utility forms should be included to the pattern also. Such additional components of the patterns are oriented on conditions of their use, for example, in assembly operations with other details. Necessity to take into account the diverse points of view on a pattern leads to creative tasks, which had better to solve preliminary (before manufacturing a pattern) on models, by experimenting with their decisions.

The decisions that are offered in the paper are realized and checked up in the toolkit WIQA (Working In Questions and Answers) the environment of which supports conceptual modeling and conceptual programming. For designing the configured patterns, cartographical modeling of the structure and specifications of patterns, and also pseudo-code programming of their laser processing are included in WIQA functions.

Conclusion

Inclusion in the life cycle of designing the patterns the opportunities of experimenting with their conceptual models leads to increasing the automation degree of designing and quality of its results, reducing the technological time of pattern production. The use of experimenting also promotes the prevention of mistakes and detection of mistakes, and it facilitates the reuse of numerical programs for controlled tools.



OSTIS-2015

(Open Semantic Technologies for Intelligent Systems)

УДК 658.512.22:004.9

ОНТОЛОГИЯ ПРОЕКТИРОВАНИЯ ШАБЛОНОВ АВИАЦИОННЫХ ДЕТАЛЕЙ

Гришин М.В., Ларин С.Н., Соснин П.И.

*Ульяновский государственный технический университет,
г. Ульяновск, Российская Федерация*

likani7@mail.ru

larinmars@rambler.ru

sosnin@ulstu.ru

В статье обосновывается необходимость разработки и применения классификатора технологической оснастки основанного на онтологическом подходе, в качестве информационной основы для создания комплекса программ автоматизированного проектирования шаблонной и объемной технологической оснастки в авиационной промышленности.

Ключевые слова: классификатор, проектирование, онтология, шаблоны.

Введение

Плазovo - шаблонный метод вот уже более пятидесяти лет остается основным методом подготовки производства авиационной техники. Сложность формы конструктивных элементов планера самолета не позволяет задавать геометрические свойства сопрягаемых деталей и увязывать их (согласовывать форму и размеры) с помощью традиционных машиностроительных чертежей [Самсонов и др., 2000].

Развитие вычислительной техники, математического и программного обеспечения, создание средств автоматизации ввода и вывода графической информации позволили, к сегодняшнему дню, полностью отказаться от конструктивных и технологических плазов, а также от шаблонов основной группы (шаблоны контрольно-контурные (ШКК) и отпечатки контрольные (ОК)). Первоисточником для производства рабочих шаблонов теперь служит математическая модель детали, созданная в памяти компьютера. Изготовление шаблонов, технологической оснастки, эталонов поверхностей, а также деталей осуществляется на станках с ЧПУ, контроль шаблонов – по размерам снятым с ЭВМ, а контроль оснастки и, в случае сложных контуров, механообрабатываемых деталей – по шаблону, например, формовочная оснастка контролируется по шаблону КШКС (контрольный шаблон контура сечения). В ходе проектирования электронной модели шаблона возникают ситуации, когда один и тот же шаблон, на одну и ту же деталь может иметь

несколько вариантов исполнения. На рисунке 1 представлены возможные вариации исполнения шаблона обрезки и кондуктора (ШОК) на угловой профиль с отверстиями под заклепки.

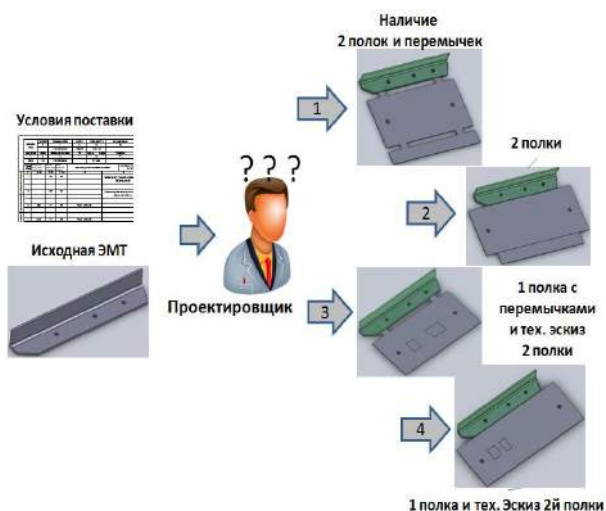


Рисунок 1 – Возможные вариации исполнения ШОК

Следует отметить, что все вариации исполнения шаблона на представленную деталь верны, однако не все они удачны с точки зрения эксплуатации, металлоёмкости и трудоёмкости исполнения. К примеру когда одна полка детали имеет простой контур и не имеет отверстий, то целесообразно с целью экономии металла и машинного времени станка с ЧПУ на котором изготавливается шаблон, дать ее в виде технологического эскиза на добавке.

Основываясь на изложенных выше положениях, не нашедших своё отражения ни в стандартах, ни в технологических инструкциях, авторами предлагается разработать более подробный технологический классификатор шаблонной оснастки на основе имеющегося классификатора шаблонной оснастки в СТП 687.07.0873-2004. [СТП, 2004]

1. Разработка и назначение классификатора технологической оснастки

Классификатор технологической оснастки представляет собой систематизированный свод наименований классификационных группировок объектов классификации - изделий плоской и объемной технологической оснастки (рисунок 2).

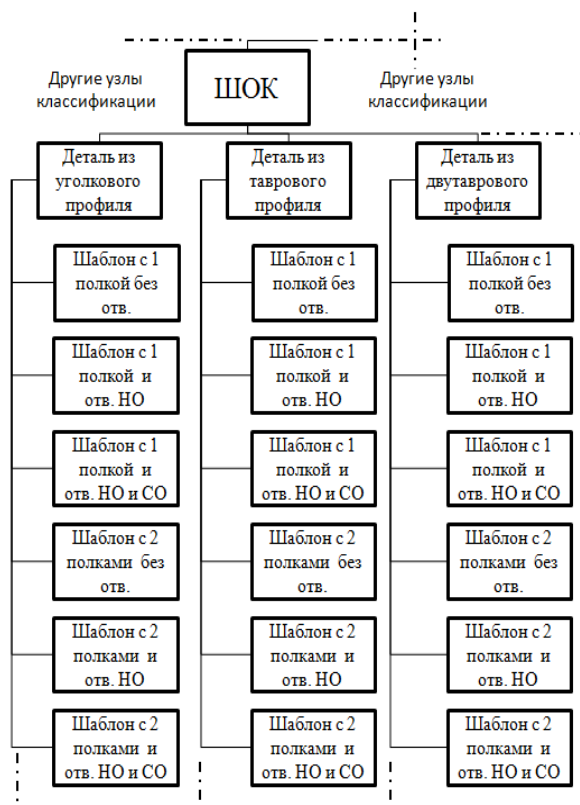


Рисунок 2 – Детализация видов ШОК

На рисунке представлен фрагмент для одного из классов шаблонов, а именно шаблонов обрезки и кондуктора. В Классификатор включены классификационные характеристики изделий плоской и объемной оснастки, комплектов технологической оснастки, групп-комплектов, на которые разработана и разрабатывается конструкторско - технологическая документация по единой системе технологической документации (ЕСТД), а также общетехнические документы (нормы, правила, требования, методы и т.д.) на оснастку, входящую в Классификатор. Основанием для разработки Классификатора явилось отсутствие в СТП 687.07.0873-2004 [СТП, 2004], СТП 687.06.0802-79 [СТП, 1979] и ОСТ 1.51451-73 [ОСТ,

1973] более глубокой классификации шаблонов, наличие не используемой в настоящее время номенклатуры типов шаблонной оснастки и отсутствие в последних требований к проектированию технологической оснастки согласно ISO 9001, облегчающих в дальнейшем автоматизацию проектно - конструкторских работ с использованием передовых информационных технологий.

Классификатор разработан в качестве информационной основы для разработки комплекса программ автоматизированного проектирования шаблонной и объемной технологической оснастки в авиационной промышленности. Его использование обеспечивает:

- создание единого информационного языка для автоматизированных систем проектирования и тематического поиска изделий и конструкторско - технологических документов, ориентированного на повторное использование результатов;
- определение объектов, процессов и направлений унификации и стандартизации; использование различными предприятиями и организациями электронной конструкторско - технологической документации, разработанной другими предприятиями, без ее переоформления, при проектировании, производстве, эксплуатации и ремонте;
- внедрение средств вычислительной техники в сфере проектирования и управления проектированием технологической оснастки (ТО);
- повышения уровня автоматизации проектно - конструкторских работ;
- применение кодов оснастки по Классификатору технологической оснастки как опережающей информации при решении задач технологической подготовки производства (ТПП).

2. Реализация Классификатора шаблонной оснастки в среде WIQA.

Проектирование шаблонной оснастки находит своё отражение в определенной системе понятий, которые полезны в решении различных задач, включая задачу построения интерактивного классификатора шаблонов. В рассматриваемом случае для материализации системы понятий было решено использовать инструментально-моделирующую среду WIQA (Work In Questions and Answers), одна из компонентов которой «Онтология» предназначена для построения прикладных онтологий [Соснин, 2014].

Центральное место в онтологии проектирования шаблонов, как и в любой другой онтологии, порождаемой в среде WIQA занимает «Словарь», в структуре которого выделены разделы для представления основных видов шаблонов. Статьи разделов содержат не только определения шаблонов, но также ссылки на модели шаблонов и ключи для поиска по оперативным запросам (рисунок 3).

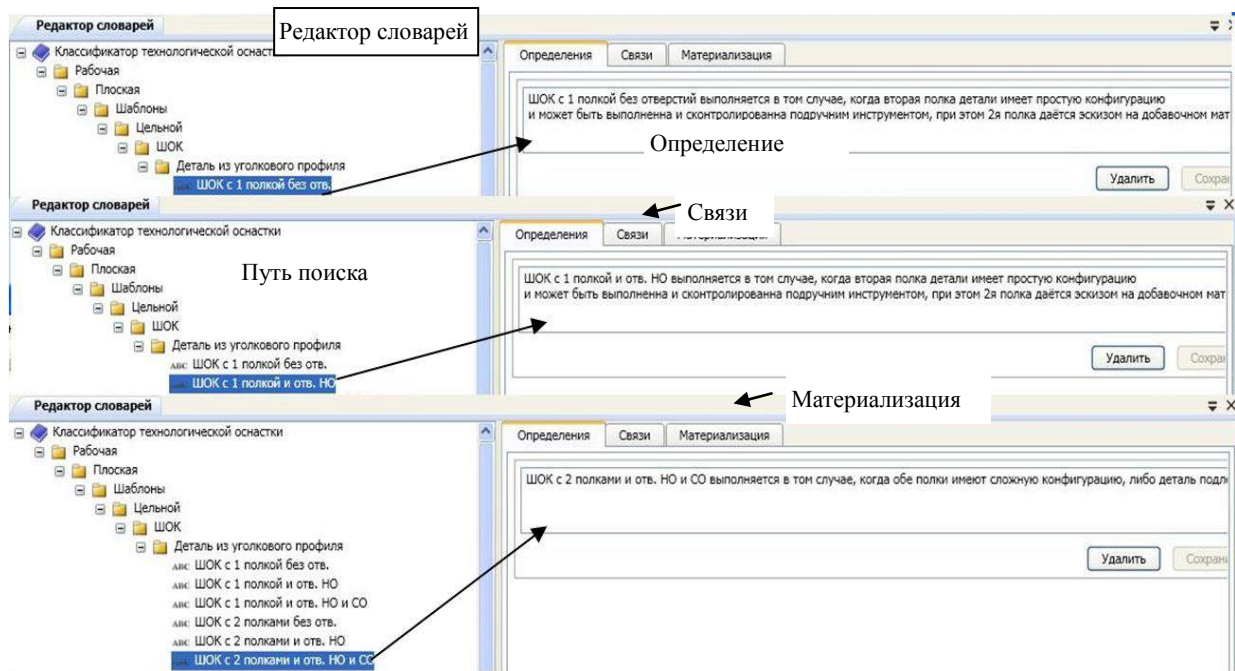


Рисунок 3 –Реализация проектной онтологии в WIQA

Среда WIQA ориентирована на концептуальное моделирование в процессах решения проектных задач [Соснин, 2008], в основу которого положено псевдо-кодовое порогаммирование активности проектировщиков [Sosnin, 2013], что переносится и на программный доступ к составляющим онтологии, включая доступ по ссылкам к концептуальным моделям шаблонов.

Формирование классификации шаблонов предшествовало её погружению в структуры Словаря проектной онтологии, в которых сохранены особенности построения классификации для её расширения (множество шаблонов является открытым).

Классификатор построен по смешанному методу, основанному на дедуктивном логическом делении классифицируемого множества. Этим методом достигается конкретизация признаков классификации оснастки на каждой последующей ступени классификации. В словаре выделяются классификационные группировки (группы понятий) составленные по определенным конструкционным признакам шаблонов и слова, являющиеся формальным описанием шаблонов.

Разработка Классификатора основана на следующих логических правилах:

- деление множества оснастки на классификационные группировки произведено на каждой ступени классификации по одному и тому же признаку или их сочетанием;
- на каждой ступени классификации исчерпывается объем делимого множества;
- деление множества произведено последовательно, однако имеются пропуски очередной (-ых) ступеней классификации в тех

случаях, когда оснастка не имеет признака классификации удовлетворяющего данной ступени.

Для того, чтобы при классификации исчерпывался объем делимого множества, в классификаторе предусмотрена классификационная группировка под наименованием «Конструкционные Элементы шаблонов». Эти группировки, как правило использованы для описания состава шаблона и также имеют свои признаки.

Каждое понятие имеет свое определение. Текстовое определение водится в первую очередь для обозначения общих признаков шаблона (или их группы), а также для поиска в Словаре онтологии. Для визуального представления шаблонов в Словаре используется материализация, выполненная в виде простейших эскизов оснастки. Это позволяет в первую очередь задать будущий облик проектируемой в последствии электронной модели шаблона, а также использовать эскиз в качестве уточняющей информации в запросах на оснастку.

Благодаря инструментарию WIQA имеется возможность полного описания необходимой оснастки в онтологии словаря. Имея в наличии классифицированный шаблон и его составляющие элементы создаются связи для формализации шаблона по признакам "часть-целое", т.к. в него входят определяющие его элементы, "наследование" т.к., в большинстве своём случаев у шаблона имеется родитель или предок, и заполняется необходимые атрибуты (рис.4).

Отметим, что отношениям можно не только приписать имена, выводящие на их семантику, но и прокомментировать. Такая возможность особо полезна для прагматических отношений, например, инструментального типа (связывающего шаблон с

инструментами для его обработки или процесс механической обработки со средствами его реализации).

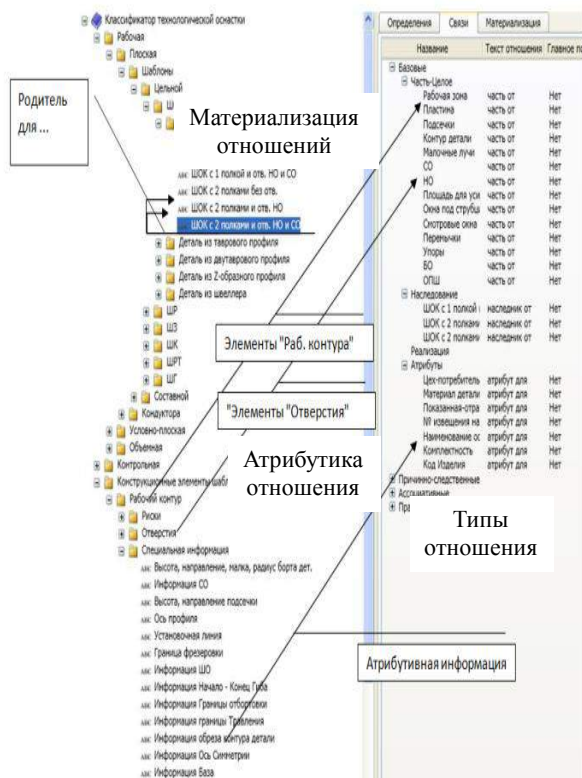


Рисунок 4 – Представление связей в онтологии

Заключение

В статье представлен подход к проектированию классификатора шаблонной технологической оснастки с помощью инструментария WIQA. Использование средств WIQA обеспечивает моделирование не только имеющихся шаблонов, но и создание новых шаблонов, с возможностью спецификации и аккумуляции их моделей в онтологическом словаре.

Модельное представление шаблонов полезно для их повторного использования в задачах, когда модели приходится адаптировать к изменившимся условиям производства, например, при создании новых образцов авиационной техники. Интерактивная версия классификатора в виде проектной онтологии с программным доступом к её составляющим способствует повышению степени автоматизации проектно - конструкторской деятельности в разработке шаблонной оснастки.

Библиографический список

- [Самсонов и др., 2000] Самсонов О.В. Тарасов Ю. Е., Бесплазовое производство авиационной техники: проблемы и перспективы// САПР и Графика, №9, 2000. – С. 33-38.
- [Соснин, 2008] Соснин П.И. Концептуальное моделирование компьютеризованных систем. – Ульяновск : УЛГТУ, 2008. – 198 с
- [Соснин, 2014] Соснин П.И. Персональная онтология профессионального опыта, Материалы 4-ой Международной конференции «Открытые семантические технологии проектирования интеллектуальных систем» (OSTIS-2014). – Минск:БГУИР, 2014. – С. 147-154.

[СТП, 2004] СТП 687.07.0873-2004. Система качества. Технологическая подготовка производства. Изготовление и применение плазово-шаблонной оснастки. – 168 с.

[СТП, 1979] СТП 687.06.0302-1979 – «Формблоки. Типовые конструкции и рекомендации по изготовлению» – 52 с.

[ОСТ, 1973] ОСТ 1.51451-73. Шаблоны плазовые. Номенклатура. – 12 с.

[ОСТ, 1978] ТИ 687.25303.00002-1978 Изготовление плазовых шаблонов для универсального стенда групповой отработки и контроля. – 40 с.

[Sosnin, 2013] Sosnin P., Pseudo-Code Simulation of Designer Activity in Conceptual Designing of Software Intensive Systems, Proceedings of the 27th European Conference on Modelling and Simulation, ECMS 2013, 2013. – С. 85-92.

ONTOLOGY OF DESIGNING FOR TEMPLATES OF AIRCRAFT PARTS

Grishin M.V., Larin S.N., Sosnin P.I.

Ulyanovsk State Technical University,

Ulyanovsk, Russian Federation

likani7@mail.ru

larinmars@rambler.ru

sosnin@ulstu.ru

The necessity of the development and application tooling classifier based on the ontological approach as an information basis for the creation of complex computer-aided design templates and volume production tooling in the aviation industry.

Main Part

Lofting a generic method for more than fifty years, remains the main method of pre-production aircraft. The complexity of the shape of the structural elements of the airframe does not allow to define the geometric properties of mating parts and link them (according the shape and dimensions) using traditional engineering drawings. The main disadvantage of this method is the high labor intensity and duration of the cycle of technological preparation of production as for the start of production of the aircraft to be produced tens of thousands of different in their characteristics patterns.

Conclusion

Use of funds WIQA offers opportunities not only for conceptual modeling templates available, but also to create new, with the possibility of presenting the material and accumulation in the ontological dictionary. Later presented materialization can be used as advance information for the template designer in order to avoid shortcomings in the design and planning for a given type of pattern, as well as contributing to a significant increase in the level of automation of design - design activity.



OSTIS-2015

(Open Semantic Technologies for Intelligent Systems)

УДК 681.31

СЕМАНТИЧЕСКИЙ АНАЛИЗ ДИАГРАММАТИЧЕСКИХ МОДЕЛЕЙ В ПРОЕКТИРОВАНИИ СЛОЖНЫХ АВТОМАТИЗИРОВАННЫХ СИСТЕМ

Афанасьев А.Н. *, Гайнуллин Р.Ф. **, Афанасьева Т.В. *

* *Ульяновский государственный технический университет,
г. Ульяновск, Россия*

a.afanasev@ulstu.ru
tv.afanasjeva@gmail.com

** *Общество с ограниченной ответственностью «Эквид»
г. Ульяновск, Россия*

r.gainullin@gmail.com

Предлагаются автоматнo-ориентированные методы и средства семантического анализа и контроля диаграмматических моделей при проектировании сложных автоматизированных систем. В качестве основного графического базиса выбран язык UML

Ключевые слова: автоматизированная система; диаграмматика; семантический анализ и контроль

Введение

В повышении успешности разработки сложных автоматизированных систем (САС) ключевую роль играет разработка качественных диаграмматических моделей, особенно на этапах концептуального проектирования [Афанасьев, 2012а]. Использование диаграмматики значительно повышает эффективность процесса проектирования и качество создаваемых систем за счет унификации языка взаимодействия участников процесса создания САС, строгого документирования проектно-архитектурных, функциональных решений и формального контроля корректности диаграммных нотаций.

Наиболее распространенным в промышленной практике диаграмматическим инструментом, используемым на всех этапах создания САС, является язык UML. Однако в современной теории и практике применения UML-диаграмматики в проектировании САС наблюдается слабое развитие методов и средств анализа и контроля корректности проектируемых диаграмм. Отсутствуют средства контроля корректности семантической согласованности диаграммных нотаций в процессе коллективного проектирования. Данные факты открывают дополнительный источник трудно диагностируемых и «дорогих» ошибок в создании

АС, их анализ и контроль является **актуальной научно-технической задачей**.

Многоуровневые RVM-грамматики

Целью работы является расширение класса диагностируемых ошибок в процессе проектирования САС за счет развития и реализации методов и средств анализа и контроля диаграммных нотаций, что позволяет сократить ошибки и время создания САС.

Базовый математический аппарат семантического анализа и контроля основан на применении семейства RV-грамматик [Афанасьев 2012а, Шаров 2005, 2008, 2011].

При коллективном проектировании диаграмматические модели имеют сложную иерархическую структуру (комплексные диаграммы), количество термов в них увеличивается многократно. Классическая графическая RV-грамматика становится громоздкой, ее разработка усложняется, не обеспечивается контроль взаимосвязанных вершин и диаграмм комплексных моделей. Для устранения этих недостатков предлагается аппарат многоуровневых грамматик.

Рассмотрим многоуровневую систему RV-грамматик, представленную в виде кортежа из четырех элементов:

$$RVM = \langle n, \Sigma^n, RV^n, r_0 \rangle,$$

где n – индекс грамматики; Σ^n – алфавит n -й грамматики; RV^n – множество продукций n -й грамматики; r_0 – аксиома грамматики верхнего уровня.

Грамматика RV^i содержит в качестве одного из состояний грамматику RV^j . Грамматика RV^j также может быть составной.

RV^n – грамматикой языка $L(G)$ является упорядоченная пятерка непустых множеств

$$G = (V, \Sigma, \tilde{\Sigma}, R, r_0),$$

где $V = \{v_l, l = \overline{1, L}\}$ – алфавит операций над внутренней памятью; $\Sigma = \{a_t, t = \overline{1, T}\}$ – терминальный алфавит графического языка (множество примитивов графического языка); $\tilde{\Sigma} = \{\tilde{a}_t, t = \overline{1, \tilde{T}}\}$ – квазитерминальный алфавит, являющийся расширением терминального алфавита. Алфавит включает:

- квазитермы графических объектов, не являющихся продолжателями анализа,
- квазитермы графических объектов, имеющих более одного входа,
- квазитермы связей – меток с определенными для них семантическими различиями,
- квазитерм для завершения анализа;

$R = \{r_i, i = \overline{1, I}\}$ – схема грамматики G (множество имен комплексов продукций, причем каждый комплекс r_i состоит из подмножества P_{ij} продукций $r_i = \{P_{ij}, j = \overline{1, J}\}$);

$r_0 \in R$ аксиома RV -грамматики (имя начального комплекса продукций), $r_k \in R$ – заключительный комплекс продукций.

Продукция $P_{ij} \in r_i$ имеет вид:

$$\tilde{a}_t \xrightarrow{W_\gamma(\gamma_1, \dots, \gamma_n)} r_m,$$

где $W_\gamma(\gamma_1, \dots, \gamma_n)$ – n -арное отношение, определяющее вид операции над внутренней памятью в зависимости от $\gamma \in \{0, 1, 2, 3\}$; $r_m \in R$ – имя комплекса продукции – преемника.

В структуру внутренней памяти входят стеки для обработки графических объектов, имеющих более одного выхода, и эластичные ленты для обработки графических объектов, имеющих более одного входа.

Система RVM получает с входной ленты символы терминального алфавита и передает их на соответствующий уровень. Элементы, которые переводят грамматику на другой уровень, назовем «сабтермами». Тогда описание RV^n -грамматики примет вид:

$$G = (V, \Sigma, \tilde{\Sigma}, \bar{\Sigma}, R, r_0),$$

где $\bar{\Sigma}$ – множество сабтермов, т.е. элементов грамматики, переводящих автомат на следующий более низкий уровень.

Продукция, содержащая сабтерм, имеет вид:

$$\bar{a}_t \xrightarrow[r_0^{n+1}]{W_\gamma(\gamma_1, \dots, \gamma_n)} r_m^n,$$

где r_0^{n+1} – комплекс-преемник – начальный комплекс грамматики следующего уровня, r_m^n – комплекс-преемник, к которому производится переход при достижении r_k^{n+1} .

Семантический анализ и контроль

При коллективном проектировании важным является контроль онтологической согласованности комплекса проектируемых диаграмм. Ошибки такого типа являются семантическими. В этом плане для анализа семантической корректности предложено использовать многоуровневую грамматику. Верхним уровнем многоуровневой грамматики является грамматика диаграмм вариантов использования, так как разработка АС, согласно методологии RUP, начинается именно с этой диаграммы. В процессе анализа накапливается семантическая информация о предметной области в виде графа связей между семантическими понятиями (текстовой информацией), нагруженными к блокам и связям диаграмм. Каждая новая диаграмма анализируется на условия непротиворечивого расширения понятий предметной области.

При построении первых диаграмм проверяется только семантическая непротиворечивость внутри диаграммы: возможность использования понятий в семантической паре. При добавлении новых диаграмм проверяется согласованность диаграммы обособлено и на согласованность комплексной модели проектируемой АС.

Для проверки комплексной модели необходимо построить граф семантических связей между элементами АС. Для решения данной задачи выбран адаптированный метод лексико-синтаксических шаблонов.

Предложенный метод позволяет диагностировать следующие семантические ошибки: Большая синонимия, Антонимия объектов,

Конверсивность отношений. Один из классов ошибок не контролируется предложенным методом – несовместимость объектов.

Разработанные методы анализа и контроля позволяют диагностировать ошибки, разделенные контекстом, а также семантические ошибки, которые не определяются в большинстве современных редакторов.

В процессе исследования были выявлены следующие классы ошибок, встречающиеся в диаграммах UML (таблица 1).

Таблица 1 – Классы ошибок UML-диаграмм

№	Тип ошибки	Д И	Д А	Д П	Д К	Д Р
1	Отсутствие связи	+	+	+	+	+
2	Ошибка передачи управления					
3	Ошибка кратности входов		+			+
4	Ошибка кратности выходов		+			
5	Недопустимая связь	+	+	+	+	+
6	Ошибка связи	+	+			+
7	Ошибка уровня доступа				+	
8	Ошибка передачи сообщения		+	+		
9	Ошибка делегирования управления				+	
10	Количественная ошибка элементов диаграммы		+			+
11	Исключающие связи неверного типа				+	
12	Вызов, направленный в линию жизни			+		
13	Несвязанная связь	+	+	+	+	+
14	Нарушение кратности зависимостей	+	+			
15	Взаимоисключающие связи	+				
16	Множественная связь	+				
17	Бесконечный цикл	+				
18	Кольцевые связи	+				
19	Синхронный вызов до получения ответа			+		
20	Ошибка удаленного контекста		+			

В таблице использованы следующие сокращения: ДИ – диаграмма использования, ДА – диаграмма активности, ДП – диаграмма последовательности, ДК – диаграмма классов, ДР – диаграмма разветвления.

Исследование современных систем создания диаграмматических нотаций UML показало, что они позволяют обнаруживать первые 16 из

перечисленных типов ошибок. Авторский аппарат RVM-грамматик позволяет обнаруживать 4 дополнительных типа ошибок: множественная связь, кольцевые связи, синхронный вызов до получения ответа, ошибка удаленного контекста.

Ошибкой удаленного контекста называется ошибка, возникающая в парных элементах, например, условное ветвление и слияние условных ветвей, и характеризующаяся возможным присутствием неограниченного количества блоков и связей между ними. Примером такой ошибки может служить условное ветвление на диаграмме активности, которое предполагает исполнение только одной из возможных ветвей, и парное ему слияние параллельных ветвей, которое передает управление дальше только при достижении всех входных связей. Такое сочетание элементов никогда не позволит достигнуть конца диаграммы.

Разработаны программные средства анализа и контроля [Афанасьев, 2012б, Брагин, 2013]: анализатор диаграмматических моделей потоков бизнес-процессов вопросно-ответной системы моделирования САС, синтаксически-ориентированный анализатор UML-диаграмм для MS Visio, сетевая система анализа и контроля диаграмматики моделей, предлагающая полный набор функциональности для анализа и контроля синтаксических и семантических ошибок.

Инструментарий последней системы позволяет выполнять следующие основные функции для пользователя:

- создание диаграмм моделей бизнес-процессов с использованием различных графических языков;
- добавление новых нотаций в графические языки;
- анализ построенных диаграмм, с использованием аппарата RVM-грамматик, по предварительно загруженным в систему описаниям языка и правилам анализа;
- добавление новых алгоритмов анализа с помощью плагинов;
- добавление синтаксических и семантических правил графических языков, для использования в RVM-анализаторе;
- создание взаимосвязи между построенными проектировщиками диаграммами;
- одновременный доступ нескольких проектировщиков к базе данных построенных диаграмм.

Заключение

Предложен метод анализа и контроля семантических ошибок диаграмматических моделей в составе комплексной диаграммы, созданной в процессе коллективного проектирования, на основе автоматных графических RVM-грамматик. RVM-грамматики позволяют обнаруживать 4 дополнительных типа ошибок, что составляет 20 % от общего числа ошибок. Разработаны

инструментальные средства семантического анализа и контроля диаграмматических моделей.

Библиографический список

[Афанасьев, 2012а] Афанасьев А.Н. Методология и инструментальный анализ и контроля потоков работ в автоматизированном проектировании сложных компьютеризованных систем // Труды конгресса по интеллектуальным системам и информационным технологиям «IS&IT'12». В 4 т. Т.1. – М.: Физматлит, 2012. – С. 391-399.

[Шаров, 2005] Шаров О.Г., Афанасьев А.Н. Синтаксически-ориентированная реализация графических языков на основе автоматных графических грамматик // Программирование. – 2005. - № 6. – С. 56-66.

[Шаров, 2008] Шаров О.Г., Афанасьев А. Н. Нейтрализация синтаксических ошибок в графических языках // Программирование. – 2008. – №1. – С. 61-66.

[Шаров, 2011] Шаров О.Г., Афанасьев, А. Н. Методы и средства трансляции графических диаграмм // Программирование. – 2011. – №3. – С. 65-76.

[Афанасьев, 2012б] Афанасьев А.Н., Гайнуллин Р.Ф., Шаров О.Г. Программная система анализа диаграммных языков // Программные продукты и системы. – 2012. № 3. – С. 138-141.

[Брагин, 2013] Брагин Д. Г., Гайнуллин Р. Ф. Анализатор диаграммных языков для диаграммного редактора Microsoft Visio // Информационные системы. – 2013. – № 6. – С. 18-21.

SEMANTIC ANALYSIS DIAGRAMMATICAL MODELS IN THE DESIGN OF COMPLEX AUTOMATED SYSTEMS

Afanasjev A.N.* , Gainullin R.F. ** ,

Afanasjeva TV.*

*Ulyanovsk State Technical University, Ulyanovsk,
Russia

a.afanasev@ulstu.ru

tv.afanasjeva@gmail.com

** Ltd «Ecvid», Ulyanovsk, Russia

r.gainullin@gmail.com

Offered automaton-oriented methods and means of semantic analysis and control of diagrammatical models in the design of complex automated systems. As a basic graphical basis chosen the UML

Introduction

In a pinch, the successful development of complex automated systems (CAS) plays a key role to develop quality diagrammatical models, especially at the stage of conceptual design. Using diagrammatic significantly increases the efficiency of the design process and the quality of the systems by unifying language of interaction between participants of the process of creating the CAS, rigorous documentation of architectural design, functional solutions and formal control of correctness of diagram notations. The most common in industrial practice diagrammatic tool used at all stages of CAS, is the language UML. However, in the modern theory and practice of using UML-diagram in designing CAS observed weak development of methods and tools for analysis and control of the correctness of the designed diagrams.

There are no means of controlling the correctness of semantic coherence diagram notation in the process of collective design. These findings open up an additional source of difficult to diagnose and "expensive" errors in the creation of the CAS, analysis and control is an important scientific and technical problem.

Main Part

The aim is to expand the class of diagnosed errors in the design process of CAS through the development and implementation of methods and tools for analysis and control of diagram notations, reducing errors and time the CAS.

When collective designing diagrammatical models have a complex hierarchical structure (complex diagrams), the number of terms they increased many times. Classical graphic RV-grammar becomes cumbersome, its development is complicated by the non-controlling interconnected nodes and diagrams of complex models. To address these shortcomings is provided an apparatus multilevel grammars.

When designing the collective control is an important ontological consistency complex designed diagrams. Errors of this type are semantic. In this respect, to analyze the semantic correctness is proposed to use a multi-level grammar. The upper level of a multi-level grammar is a grammar use case diagrams as well as the development of the CAS, according to the methodology RUP, starts with this chart. In analyzing the semantic information is stored on the domain as a graph of relations between semantic concepts (textual information), loaded to the blocks and relationship diagrams. Each new diagram is analyzed for conditions consistent expansion domain concepts .

The proposed method makes it possible to diagnose the following semantic errors: Large synonyms, antonyms objects of conversion relations. One class of errors is not controlled by the proposed method - Incompatible objects.

The methods developed for the analysis and control can diagnose errors separated context and semantic errors, which are not defined in most modern editors.

Software tools for analysis and control analyzer diagrammatic flow models of business processes Question answering modeling CAS, syntax-oriented analyzer UML-diagrams for MS Visio, network system analysis and control diagrammatical models offered full set of functionality for analysis and control of syntactic and semantic errors.

Conclusion

Proposed a method for analysis and control of semantic errors diagrammatic's models in complex diagrams created in the process of collective design, based on the automatic graphics RVM-grammars. RVM-grammar allow to detect 4 additional types of errors, which is 20% of the total number of errors. Developed tools of semantic analysis and control of diagrammatic's models.



OSTIS-2015

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822:514

ПРИМЕНЕНИЕ НАВИГАЦИОННОЙ СТРУКТУРЫ ЭЛЕКТРОННОГО АРХИВА ПРОЕКТНОЙ ОРГАНИЗАЦИИ В ЗАДАЧАХ ИНТЕЛЛЕКТУАЛЬНОГО АНАЛИЗА ТЕХНИЧЕСКИХ ДОКУМЕНТОВ

Субхангулов Р.А., Филиппов А.А.

*Ульяновский государственный технический университет,
г. Ульяновск, Россия*

subkhangulov-ruslan@yandex.ru

al.filippov@ulstu.ru

В работе рассматривается применение интеллектуальных методов и алгоритмов анализа текстовых документов с целью построения навигационной структуры электронного архива (ЭА) проектной организации. Представление содержимого ЭА в виде иерархии кластеров, содержащих технические документы, близкие по тематике в контексте применяемых стандартов проектируемых систем, позволяет сократить пространство поиска и тем самым ускорить процедуры нахождения требуемых документов по их содержанию.

Ключевые слова: электронный архив, стандарты проектирования, онтология, навигационная структура.

Введение

Современная крупная проектная организация обладает значительным по объему ЭА конструкторской и технической документации, большая часть которой представлена в текстовом неструктурированном виде. Фактически, такой ЭА текстовой документации содержит в себе опыт и знания большого количества высококвалифицированных специалистов, которые на протяжении многих лет занимались разработкой и проектированием сложных систем. При увеличении объема ЭА затрудняется анализ документов по заранее заданным реквизитам, а от лиц, занимающихся проектированием сложных технических систем, требуются навыки в области семантической обработки большого объема технической документации, а также глубоких знаний предметной области. В результате, довольно часто важный опыт предыдущих разработок, зафиксированный в ЭА, остается невостребованным и, как следствие, увеличивается время выполнения цикла опытно-конструкторских работ.

Решение указанной проблемы может основываться на применении интеллектуальных методов и алгоритмов анализа текстовых документов с целью построения навигационной структуры ЭА технической документации. Представление содержимого такого архива в виде иерархии кластеров, содержащих технические

документы, близкие по тематике в контексте применяемых стандартов проектируемых систем, позволяет сократить пространство поиска и тем самым ускорить процедуры нахождения требуемых документов по их содержанию.

Учет специфики проектных знаний приводит к необходимости формирования онтологии проектной организации особой структуры, включающей в себя особенности процесса проектирования в форме системы понятий предметной области, отношений между ними и функций интерпретации. Таким образом, ЭА должен обладать свойствами интеллектуальной системы. Актуальной является задача разработки моделей, методов и алгоритмов построения навигационной структуры ЭА технической документации на основе предметно-ориентированной кластеризации слабоформализованных информационных ресурсов.

1. Формальная модель онтологии электронного архива

1.1. Основные требования к онтологии с точки зрения процесса проектирования

Одно из определений процесса проектирования дано Норенковым И.П. в работе [Норенков, 2002]: «Проектирование технического объекта – это создание, преобразование и представление в принятой форме образа этого еще не

существующего объекта». В каждом конкретном случае такая принятая форма образа проектируемого объекта определяется на основе ряда ограничений и правил, в том числе, с использованием стандартов оформления конструкторской, программной, эксплуатационной документации и другими нормативными документами. В основе любого процесса проектирования сложной системы лежит понятие жизненного цикла (ЖЦ), который отражает различные состояния проектируемой системы, начиная с момента возникновения необходимости в данной системе и заканчивая моментом ее снятия с эксплуатации [Норенков, 2002].

При построении модели предметной области в виде прикладной онтологии для решения задач анализа технической документации необходимо сформулировать основные требования к онтологии. Такие требования должны опираться на особенности предметной области и, кроме того, на особенности тех информационных ресурсов, которые подвергаются анализу. Цель применения онтологии заключается в привлечении дополнительных знаний об окружающей среде проектируемых средств при анализе документации для сокращения времени поиска необходимых документов. Фактически, для отдельно взятого информационного ресурса из ЭА технической документации онтология задает новую систему координат, в которой кластер документов может рассматриваться как группа связанных между собой по смыслу технических документов.

На основании вышесказанного, сформулируем основные требования к прикладной онтологии:

- Онтология ЭА проектной организации должна включать в себя описания применяемых в организации моделей ЖЦ проектируемых систем.
- Структура онтологии должна основываться на использовании множества возможных серий стандартов, для каждой из которых определяется свой набор понятий и отношений между ними.
- Множество понятий онтологии должно включать в себя те понятия, которые соответствуют уже реализованным проектам, результаты которых в документальном виде зафиксированы в ЭА.

1.2. Структурно-аналитическая модель прикладной онтологии

Формально онтология ЭА состоит из двух прикладных онтологий и записывается как кортеж вида:

$$O = \langle O^D, O^{LC}, R_A \rangle, \quad (1)$$

где O^D – компонент онтологии предметной области; O^{LC} – онтология ЖЦ проектируемых систем; R_A – отношение однонаправленной ассоциации между компонентами онтологии. Рассмотрим более подробно компоненты онтологии ЭА (1).

Онтологию предметной области запишем в виде кортежа:

$$O^D = \langle C, W, R^D, F^D \rangle,$$

где C – множество понятий ЭА, которое образует основу понятийного аппарата проектирования сложной системы; $W = W^S \cup W^P$ – множество терминов предметной области; W^S – множество терминов на уровне стандартов; W^P – множество терминов на уровне проектов; R^D – множество отношений:

$$R^D = \{R_G^D, R_C^D, R_A^D\},$$

где R_G^D – антисимметричное, транзитивное, неререфлексивное бинарное отношение обобщения («subclass_of»); R_C^D – бинарное транзитивное отношение композиции («part_of»); R_A^D – бинарное отношение однонаправленной ассоциации.

Множество понятий C записывается следующим образом:

$$C = (C^{S_1} \cup C^{S_2} \cup \dots \cup C^{S_k}) \cup C^P,$$

где $C^{S_i}, i = \overline{1, k}$ – множество понятий предметной области, рассматриваемых в рамках i -й группы стандартов, используемых в проектной организации (например, ГОСТ 34, ГОСТ 19 и т.д.); C^P – множество понятий предметной области, извлекаемых из технической документации по реализованным проектам.

Множество интерпретирующих функций представлено в виде:

$$F^D = \{F_{WC}^D, F_{C^P C^S}^D\},$$

где $F_{WC}^D : \{W\} \rightarrow \{C\}$ – функция, сопоставляющая набору терминов подмножество понятий предметной области, задаваемая алгоритмически; $F_{C^P C^S}^D : \{C^P\} \rightarrow \{C^S\}$ – функция интерпретации подмножества понятий на проектном уровне онтологии, позволяющая осуществить переход на уровень понятий, определенных в стандартах.

Онтология ЖЦ как компонента кортежа (1) записывается следующим образом:

$$O^{LC} = \langle M^{LC}, St^{LC}, R^{LC} \rangle,$$

где M^{LC} – множество моделей ЖЦ проектируемых систем; St^{LC} – множество стадий (этапов) ЖЦ. Отношение R^{LC} имеет вид «часть-целое (part_of)» и позволяет декомпозировать стадии ЖЦ проектируемой системы в онтологии на этапы и т. д.

1.3. Представление технического документа в контексте онтологии предметной области

Технический документ (ТД) в контексте интеллектуального ЭА будем рассматривать как информационный ресурс. Любой ТД можно рассматривать как контейнер слабоструктурированной информации: с одной стороны мы имеем дело с естественно-языковым текстом, а с другой стороны ТД имеет четко выраженную структуру, определяемую различными нормативными документами.

Если рассматривать ТД с позиций «данные-метаданные», то отдельно взятый ТД можно считать носителем данных, а онтологию ЭА – метаданными. Отношения ТД с вершинами онтологии рассматриваются применительно к двум уровням: уровень стандартов проектирования и уровень проектов.

Рассмотрим виды отношений между ТД и вершинами онтологии более детально:

- отношение «connect_to» позволяет зафиксировать в архиве принадлежность ТД множеству понятий предметной области, определенных в рамках стандартов проектирования (определяется тематика содержания документа);

- отношение «contain» между разделом ТД и термом онтологии на проектном уровне и на уровне стандартов определяет набор тех терминов, которые содержатся в разделе ТД.

Структура ТД определяется набором его разделов и подразделов, порядком их следования и вложенностью. Для отношения «contain» вес определяется как f_i^j – частота встречаемости i -го термина в j -м разделе документа. Для формирования оценок терминов будем использовать следующие правила [Наместников, 2009]:

- высокочастотные термины не являются узкими или специфическими (однако, дают большое число совпадений при сравнении терминов запроса и документа);

- низкочастотные термины вносят небольшой вклад в поиск нужных документов (поскольку редкие термины дают небольшое число совпадений);

- наилучшими конденсационными терминами являются термины не слишком редкие и не слишком частые.

Частоту появления терминов в одном ТД будем сравнивать с частотой появления тех же терминов во всем объеме документов. Если частоты терминов в анализируемом документе значительно превосходят частоту терминов по всему объему документов, делается предположение, что соответствующие термины являются ценными. Математически такая зависимость выражается следующим образом [Барсегян, 2009]:

$$f_i = tfidf_i = tf_i \cdot \log\left(\frac{N}{df(w_i)}\right),$$

где $tfidf_i$ – относительная важность термина w_i в документе; tf_i – нормализованная частота встречаемости термина w_i ; N – количество всех документов; $df(w_i)$ – количество документов, содержащих термин w_i .

2. Формирование навигационной структуры электронного архива

Под терминологическим окружением понятия предметной области будем понимать множество терминов (слов) из ТД по реализованным проектам в ЭА, которые наиболее близки с данным понятием в семантическом смысле.

Определение семантического расстояния между понятием и терминами в ТД будем определять, основываясь на идее анализа семантических отношений, представленной в работе [Serrano-Guegogo и др., 2005], и заключающейся в использовании «дистанции» между словами.

В документе отношение между понятием, определенном экспертом и, одновременно являющимся термином документа, и «обычным» термином, расположенных в одном предложении, должно отличаться от отношения между понятием и термином из двух разных абзацев. Кроме того, если идея повторяется в нескольких абзацах, то она может считаться более важной, чем, если бы она была зафиксирована в одном абзаце.

После определения семантических расстояний между исследуемым понятием, для которого строится терминологическое окружение, и терминами документа необходимо определить подмножество терминов, которое наиболее тесным образом в семантическом смысле связано с понятием. При определении терминологического окружения будем использовать гипотезу λ -компактности [Загоруйко, 1999] Таким образом, используя гипотезу λ -компактности, определяется подмножество терминов, которое включается в терминологическое окружение рассматриваемого понятия.

Каждое терминологическое окружение W_k понятия $C_k^{P(S)}$ представим следующим образом:

$$\{(w_{1k}, f_{1k}), (w_{2k}, f_{2k}), \dots, (w_{ik}, f_{ik}), \dots, (w_{lk}, f_{lk})\},$$

где w_{ik} – i -й терм k -о понятия онтологии; l_k – общее количество термов, ассоциированных с k -м понятием; f_{ik} – нормализованный семантический вес i -о термина в терминологическом окружении k -о понятия (нормализованное семантическое

расстояние между термином и понятием в рамках одного терминологического окружения).

В основе онтологического индексирования ТД лежит следующая функция:

$$F_{ov} : ch_j^d \rightarrow oV_j^d,$$

где ch_j^d – j -ый раздел ТД d ; oV_j^d – онтологическое представление j -го раздела Тда d .

Под степенью выраженности понятия онтологии интеллектуального ЭА будем понимать степень совпадения терминологического окружения понятия с набором терминов некоторого фрагмента ТД при условии, что в терминологическое окружение включены термины, наиболее близкие в семантическом отношении с понятием.

Вычисление степеней выраженности понятий онтологии для каждого раздела ТД производится с применением аппарата нечетких соответствий [Берштейн и др., 2005]. Образом множества \tilde{W}^d (множество терминов ТД d) при соответствии \tilde{I} будем называть нечеткое множество $\tilde{I}(\tilde{W}^d)$ в $C^{P(S)}$, определяемое выражением:

$$\tilde{I}(\tilde{W}^d) = \{ \langle \mu_{\Gamma(W^d)}(c^{P(S)}), c^{P(S)} \rangle \mid c^{P(S)} \in C^{P(S)} \},$$

$$\text{где } \mu_{\Gamma(W^d)}(c) = \bigvee_{w^d \in W^d} (\mu_{W^d}(w^d) \& \mu_O < w, c^{P(S)} >).$$

Фактически, применив функцию интерпретации онтологии $F_{WC}^D : \{W\} \rightarrow \{C\}$ на уровне проектов и стандартов, получаем первоначальное онтологическое представление каждого раздела документа в виде:

$$o\hat{V}_j^d = \langle ch_j, \{\hat{C}_j^P \cup \hat{C}_j^S\} \rangle, \hat{C}_j^P \subseteq C^P, \hat{C}_j^S \subseteq C^S \Big|_{S_k^{tc}},$$

где \hat{C}_j^P , \hat{C}_j^S – первоначальные (ориентировочные) наборы понятий уровней проектов и стандартов соответственно, которые требуют уточнения.

Для генетической оптимизации ТД ЭА используем метод сравнения терминологического окружения каждого понятия в онтологии предметной области уровня проектов с анализируемым текстом [Наместников, 2012]. Минимальным фрагментом анализируемого текста является отдельное предложение, а максимальным – ТД в целом, так как в различных частях (фрагментах) документа делается акцент на разных понятиях предметной области.

Проведенные эксперименты с выделенными фрагментами ТД на основе генетической оптимизации показали, что в среднем около 30% понятий в сумме дают 70% от общей степени выраженности всех понятий фрагмента ТД. Учитывая данный факт, первоначальные наборы понятий \hat{C}_j^P и \hat{C}_j^S j -го раздела документа

расширяются наиболее значимыми понятиями каждого фрагмента.

Заключительным шагом в формировании онтологического представления ТД является применение интерпретирующей функции $F_{C^P C^S}^D : \{C^P\} \rightarrow \{C^S\}$, которая позволяет уточнить набор понятий уровня стандартов, опираясь на найденное подмножество понятий в ТД уровня проектов онтологии.

Реализуя вышеуказанные процедуры, получаем окончательные онтологические представления для каждого j -го раздела документа в следующем виде:

$$oV_j^d = \langle ch_j, \{C_j^P \cup C_j^S\} \rangle, C_j^P \subseteq C^P, C_j^S \subseteq C^S \Big|_{S_k^{tc}}.$$

Для нахождения формальной меры расстояния между документами представим каждое онтологическое представление документа в качестве дерева (иерархии) понятий предметной области. Такая иерархия определяется путем нахождения минимального дерева, включающего все понятия из онтологического представления [Загоруйко, 1999].

Редакционное расстояние между иерархиями определяется на основе вычисления стоимости редакционной операции, которая находится отдельно для каждого типа семантического отношения. Итоговое редакционное расстояние между иерархиями вычисляется по следующей формуле:

$$\tau_{ov}^* = \max_i \left(\sum_{s=1}^m \varphi_{S_i}(R_G^D)_s + \sum_{l=1}^n \varphi_{S_i}(R_C^D)_l \right),$$

где $\varphi_{S_i}(R_G^D)$ – стоимость редакционной операции отношения обобщения; $\varphi_{S_i}(R_C^D)$ – стоимость редакционной операции отношения «часть-целое»; i – номер группы стандартов; s – номер добавляемого отношения обобщения; l – номер добавляемого отношения «часть-целое»; S_i – принадлежность значения редакционной операции к i -й группе стандартов.

Коэффициент нормализации T_{ov} рассчитываем исходя из всех семантических отношений обобщенной иерархии. Мера расстояния между онтологическим представлениями ТД определяется с помощью следующего выражения:

$$\|oV^{d_1} - oV^{d_2}\| = \frac{\tau_{ov}^*}{T_{ov}}.$$

Для выполнения процесса формирования навигационной структуры в виде вложенного набора кластеров технических документов необходимо решить задачу настройки весов семантических отношений между понятиями онтологии на уровне стандартов, при которых качество кластеризации, определяемое выражением:

$$F^* = \frac{\max(\bar{K}_+ + \bar{K}_-, \hat{K}_+ + \hat{K}_-)}{N} \rightarrow \min \quad (2)$$

было бы наилучшим. В выражении (2) \hat{K}_- и \bar{K}_- – множества отсутствующих документов соответственно в первом и во втором кластерах; \hat{K}_+ и \bar{K}_+ – множества лишних документов соответственно в первом и во втором кластерах, N – количество документов.

3. Формальный критерий эффективности применения навигационной структуры

Будем предполагать, что искомая навигационная структура ЭА ТД образуется в процессе иерархической кластеризации документов. Особо следует отметить тот факт, что сформированная навигационная структура ЭА сохраняет свою инвариантность в пределах определенной стадии (этапа) ЖЦ проектируемой системы.

На каждом шаге происходит формирование двух новых кластеров. На нулевом шаге кластеризации электронный архив представляется в виде полного множества документов D_0 (рисунок 1). На первом шаге иерархической кластеризации получаем два множества ТД, для которых справедливо выражение: $D_1 \cup D_2 = D_0$. На втором шаге аналогичным образом формируются множества ТД $D_{11}, D_{12}, D_{21}, D_{22}$, причем $(D_{11} \cup D_{12} = D_1) \cup (D_{21} \cup D_{22} = D_2) = D$. Указанный процесс продолжается до тех пор, пока не сработает условие остановки иерархической кластеризации ТД и не будет построена искомая навигационная структура архива.

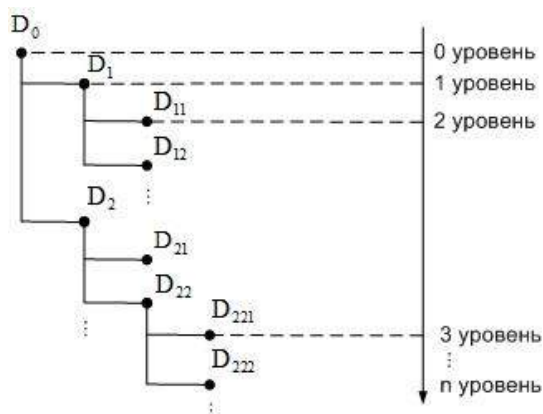


Рисунок 1 – Пример навигационной структуры ЭА

Количество документов в кластере на n -м уровне будет определяться как величина, равная $\frac{N}{2^n}$, где n – номер уровня навигационной структуры; N – количество документов в ЭА.

Применяя сформированную навигационную структуру ЭА ТД при поиске необходимого

документа у проектировщика возникает преимущество, состоящее в сокращении времени нахождения документа близкого по содержанию с некоторым эталонным документом.

Пусть Δt – время, затрачиваемое на анализ одного ТД. Суммарное время анализа содержимого кластера на уровне n будет определяться величиной

$$k_n \left(\Delta t \cdot \frac{N}{2^n} \right)$$

где $k_n \in [0, 1]$ – коэффициент, который определяет долю ТД, просматриваемых проектировщиком на n -м уровне. Действительно, находясь на нулевом уровне навигационной структуры, проектировщик никогда не будет анализировать все документы архива. Следовательно, k_n будет принимать наименьшее значение. При увеличении n у проектировщика появляется возможность просматривать большую долю документов в кластере. Следовательно, коэффициент k_n будет увеличиваться.

Средний процент сокращения времени поиска документов в ЭА, основываясь на навигационной структуре, определяется по следующей формуле:

$$\Delta k = \frac{k_0 \Delta t N - k_n \left(\Delta t \cdot \frac{N}{2^n} \right)}{k_0 \Delta t N} \cdot 100\% = \left(1 - \frac{k_n}{k_0} \cdot \frac{1}{2^n} \right) \cdot 100\%.$$

Пусть $k_0 = 0.1$; $k_n = 0.5$; $n = 6$, тогда получим

следующее значение сокращения времени поиска документов:

$$\Delta k = \left(1 - \frac{0.5}{0.1} \cdot \frac{1}{8} \right) \cdot 100\% = 37.5\%.$$

Предполагая, что в среднем около 20% времени научно-исследовательской и опытно-конструкторской работы (НИОКР) расходуется на поиск документов в ЭА проектных организаций, можно определить, на сколько сократится время выполнения НИОКР в среднем при использовании ЭА ТД с навигационной структурой, рассматриваемой выше:

$$\Delta = \Delta k \cdot 0.2 = 7.5\%.$$

Заключение

Представление содержимого ЭА в виде иерархии кластеров, содержащих технические документы, близкие по тематике в контексте применяемых

стандартов проектируемых систем, позволяет сократить пространство поиска и тем самым ускорить процедуры нахождения требуемых документов по их содержанию.

Библиографический список

[Serrano-Guerrero и др., 2005] Serrano-Guerrero J Physical and Semantic Relations to Build Ontologies for Representing Documents / Serrano-Guerrero J., Olivas J. A., De la Mata J., Garcés P. // Fuzzy logic, Soft Computing and Computational Intelligence (Eleventh International Fuzzy Systems Association World Congress IFSA), 2005, №. 1. P. 503-508.

[Барсегян, 2009] Барсегян А.А. Анализ данных и процессов: учеб. пособие. // А. А. Барсегян // Санкт-Петербург: БХВ-Петербург, 2009.

[Берштейн и др., 2005] Берштейн Л. С. Нечеткие графы и гиперграфы. / Л. С. Берштейн Л.С., А. В. Боженок // М.: Научный мир, 2005.

[Загоруйко, 1999] Загоруйко Н. Г. Прикладные методы анализа данных и знаний /Н. Г. Загоруйко // Новосибирск: ИМ СО РАН, 1999.

[Наместников, 2009] Наместников А.М. Интеллектуальные проектные репозитории. // А. М. Наместников // Ульяновск: УлГТУ, 2009.

[Наместников, 2012] Наместников А.М. Концептуальное индексирование проектных документов на основе генетической оптимизации. // А. М. Наместников // Автоматизация процессов управления. - 2012. - №27, С. 62 – 66.

[Норенков, 2002] Норенков И. П. Основы автоматизированного проектирования: учеб. для вузов. – 4-е изд., перераб. и доп. /И. П. Норенков // М.: Изд-во МГТУ им. Н. Э. Баумана, 2009.

APPLICATION OF NAVIGATION STRUCTURE OF DIGITAL ARCHIVE OF PROJECT ORGANIZATION IN TASKS OF THE INTELLECTUAL ANALYSIS OF CAD DOCUMENTS

Subkhangulov R.A., Filippov A.A.*

*Ulyanovsk State Technical University,
Ulyanovsk, Russia
al.filippov@ulstu.ru

This article is about application of intellectual methods and algorithms of analysis of text documents for the purpose of creation of navigation structure of digital archive (DA) of project organization. Representation of contents of DA in the form of hierarchy of the clusters containing technical documents, similar by content in the context of design standards allows reducing search space and accelerating procedures of finding of the required documents based on their contents.

Introduction

The modern project organization possesses the large DA of CAD documentation which most part is presented in the text unstructured form. The important experience of the previous development which is kept in archive remains unclaimed and, as a result, is increased the runtime of a cycle of development.

The solution of this problem can be based on application of intellectual methods and algorithms of

analysis of text documents for the purpose of creation of navigation structure of DA of technical documents.

The specifics of project knowledge led to the necessity of formation the ontology of the project organization of the special structure including features of design process.

Main Part

The main requirements to application-oriented ontology:

- The ontology of DA of the project organization shall include descriptions of the life cycle (LC) models of the designed systems applied in the organization.
- The structure of ontology shall be based on use of a set of design standards, for each of which is defined the set of concepts and the relations.
- The set of concepts of ontology shall include those concepts which correspond to realize projects which are kept in a documents look in DA.

The navigation structure of DA is formed in the course of a hierarchical clustering of documents. Especially it should be noted that the created navigation structure of DA keeps the invariance within a certain stage of LC of the designed system.

On each step there is a formation of two new clusters. On a zero step of a clustering the DA is presented in the form of a full set of documents D_0 (figure 1). On the first step of a hierarchical clustering we receive two sets of documents: $D_1 \cup D_2 = D_0$. On the second step sets of documents are similarly formed $D_{11}, D_{12}, D_{21}, D_{22}$, and $(D_{11} \cup D_{12} = D_1) \cup (D_{21} \cup D_{22} = D_2) = D$. The specified process proceeds until performed a condition of a stop of a hierarchical clustering of documents and the required navigation structure of archive won't be constructed.

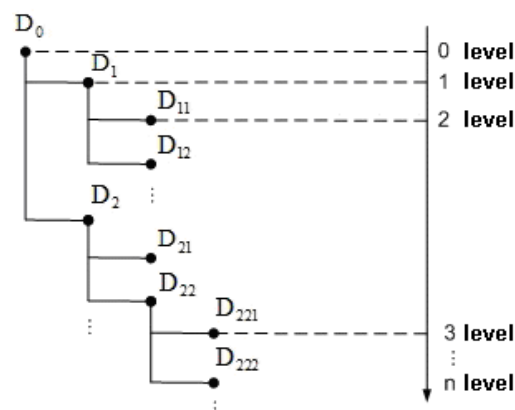


Figure 1 – Example of navigation structure of DA

Conclusion

Representation of contents of DA in the form of hierarchy of the clusters containing technical documentation, similar by content in the context of design standards allows reducing search space and accelerating procedures of finding of the required documents based on their contents.



OSTIS-2015

(Open Semantic Technologies for Intelligent Systems)

УДК 004.892

ОЦЕНКА ТЕРМИНОЛОГИЧНОСТИ ЛЕКСИЧЕСКИХ ЕДИНИЦ НА ОСНОВЕ ОНТОЛОГИИ ПРЕДМЕТНОЙ ОБЛАСТИ

Андреев И.А., Башаев В.А., Клейн В.В., Мошкин В.С., Ярушкина Н.Г.

*Ульяновский государственный технический университет,
г. Ульяновск, Российская Федерация*

ares-ilya@yandex.ru

perevod73@yandex.ru

vikklein93@gmail.com

postforvadim@yandex.ru

jng@ulstu.ru

В данной статье описана семантическая метрика извлечения списка терминов из текстов конкретной проблемной области, основанная на анализе ее онтологии. Представлена формальная модель используемой OWL-онтологии, также рассмотрена реализация моделей и алгоритмов оценки степени терминологичности слов или сочетаний слов текстовых массивов в программной системе извлечения терминологии из текста.

Ключевые слова: извлечение терминов, семантическая метрика, онтология.

Введение

Принцип работы существующих алгоритмов извлечения терминологии (term extraction) в лексикографии и терминоведении основан на статистических и лингвистических методах. В основе статистических методов лежит вычисление степени терминологичности на основании числовых закономерностей, присущих термину или нетермину. В основе лингвистических методов лежит отбор по определенным лексико-грамматическим шаблонам и другим лингвистическим признакам термина [Андреев и др., 2013].

Главным недостатком использования статистических и лингвистических методов в процессе извлечения терминологии из текста является отсутствие возможности выделения из получившегося множества терминов только тех, которые относятся к рассматриваемой проблемной области [Yarushkina, 2001].

На множестве информационных единиц в некоторых случаях полезно задавать отношение, характеризующее ситуационную близость информационных единиц, т. е. силу ассоциативной связи между информационными единицами. Его можно было бы назвать отношением релевантности для информационных единиц [Namestnikov et al., 2002].

При анализе больших массивов документации необходимо учитывать специфику ее предметной области, чтобы получить в качестве результата список терминов, характерных для конкретной предметной области. Именно для решения подобных задач используются семантические алгоритмы, базирующиеся на определенных семантических метриках.

В настоящее время одной из наиболее универсальных методик представления экспертных знаний с точки зрения полноты семантического описания информационной единицы предметной области является онтологический подход. Именно поэтому одним из важнейших направлений решения задачи извлечения терминологии из большого массива технической документации является разработка и использование семантических метрик на основе онтологических моделей [Ярушкина и др., 2007a].

1. Формальная модель онтологии предметной области "эксплуатация токарно-фрезерного станка с ЧПУ"

Сущность онтологического подхода заключается в том, что предметная область представляется в виде организованной совокупности понятий, их свойств и связей.

Наиболее удобным форматом представления онтологии с точки зрения машинной обработки и

наглядности описания особенностей предметной области является язык OWL.

Выделим обязательные требования к OWL-онтологии, используемой в рамках решения задачи извлечения терминологии:

- Онтология должна наиболее полно отражать особенности объектов предметной области.
- Онтология не должна быть избыточной.
- Онтология должна быть наглядной.

Онтологический подход хранения знаний предполагает представление их в следующем виде:

$$O = \langle T, R, F \rangle \quad (1)$$

Исходя из модели (1), онтология «Эксплуатация токарно-фрезерного станка с ЧПУ» имеет следующие составляющие:

1. T – термины прикладной области, которую описывает онтология. Например, объекты «Резцедержатель», «Станина», «Поплавковое реле».

2. R – отношения между терминами предметной области, при этом $R \subset \{R_{inc}, R_{add}, R_{term}, R_{lem}, R_{NC}\}$:

- R_{inc} – множество встроенных отношений объектов таких, как «sameAs» и «SubClassOf»;
- R_{add} – множество отношений, позволяющих расширять набор объектов описываемой предметной области за счет сочетания лемм связанных объектов;
- R_{term} – отношение «является Термином», имеющее логический тип значения. Это свойство является вспомогательным и определяется экспертом исходя из критерия, насколько данный объект онтологии является характерным конкретно для этой предметной области;
- R_{lem} – отношение «имеет Лемму», имеющее строковое значение, полученное путем леммирования (приведения к начальной форме) наименования объекта с помощью программы Mystem компании Яндекс по соответствующим морфологическим признакам термина;
- R_{NC} – множество отношений объектов, а также свойств типа данных, наиболее полно описывающих особенности взаимодействия объектов рассматриваемой предметной области.

3. F – множество функций интерпретации (аксиоматизации), заданных на терминах и/или отношениях онтологии [Добров и др., 2003]. Примеры таких функций в разработанной онтологии представлены выражениями (2) и (3):

$$F_{СОЖ} : X_{ТипСверления} \rightarrow Y_{ТипПодачи} \quad (2)$$

где $F_{СОЖ}$ – отношение «является Типом Подачи СОЖ»,

$X_{ТипСверления}$ – множество объектов класса «Тип Сверления»,

$Y_{ТипПодачи}$ – множество объектов класса «Тип Подачи СОЖ».

$$F_{InEng} : X_{Контекст} \rightarrow Y_{Eng} \quad (3)$$

где F_{InEng} – отношение «имеет Английский Эквивалент»,

$X_{Контекст}$ – множество объектов класса «Контекст»,

Y_{Eng} – множество объектов класса «Английский Аналог».

2. Семантическая метрика оценки терминологичности слов/сочетаний слов

Использование семантической метрики «термин/нетермин» на множестве слов конкретного текста с использованием заранее разработанной OWL-онтологии в процессе извлечения терминологии предполагает определение для каждого поступающего слова или сочетания слов степени близости к терминам рассматриваемой области.

Степень близости входных слов/сочетаний слов к терминам проблемной области (k_{Ont}) может иметь значение от 0 до 1: чем ближе полученное значение к 1, тем с большей долей вероятности данное одно-/многословие является термином [Афанасьева и др., 2011].

В ходе решения поставленной задачи было разработано два критерия выделения терминов из предметной области посредством использования онтологии:

- тезаурусный критерий,
- критерий вложенных связей.

Результаты проведенных экспериментов должны показать, какой из данных семантических критериев является наиболее продуктивным и оптимальным применимо к имеющейся модели онтологии.

2.1. Тезаурусный критерий

Тезаурус представляет собой контролируемый словарь терминов на естественном языке, явно указывающий отношение между терминами и предназначенный для информационного поиска. Любая онтология является усложненной версией тезауруса [Кураленок и др., 2002].

Тезаурусный подход к извлечению терминологии предполагает непосредственный поиск вхождений лемм поступающих на вход слов и их сочетаний среди терминов, определенных в онтологии. Для этого в разработанной онтологии для каждого класса определено свойство «имеет Лемму», которое имеет строковое значение, полученное путем леммирования (приведения к начальной форме) имени объекта с помощью программы Mystem компании Яндекс по соответствующим морфологическим признакам термина.

Алгоритм определения степени близости слов/сочетания слов терминам проблемной области согласно тезаурусному критерию предполагает:

- Оценку степени близости поступающего на вход алгоритма слова/сочетания слов каждому объекту онтологии без учета онтологического критерия оценки;
- Определение опорного объекта онтологии, наиболее близко ассоциирующегося с входным одно-/многословием.

Опорный объект онтологии, используемый в дальнейшем анализе, имеет степень близости по отношению к входному слову/сочетанию слов, рассчитанную по следующей формуле:

$$k_i = \max_{i=1}^m \left(\frac{n_i}{p_i} \right) \quad (4)$$

где m – количество всех объектов онтологии;
 n_i – число слов из леммы входного многословия, найденных в лемме объекта онтологии;
 p_i – общее число слов в лемме объекта онтологии.

Общая схема оценки степени близости слов/сочетания слов терминам проблемной области согласно тезаурусному критерию приведена на рисунке 1.

Объект онтологии предметной области (лемма)



Входное слово/сочетание слов (лемма)

Рисунок 1 – Поиск опорного объекта онтологии

При этом порядок следования слов многословия в опорном объекте должен сохраняться.

Если несколько разных объектов онтологии имеют одинаковое значение коэффициента k_i , то опорным будет считаться тот объект, которому соответствует максимальное n_i . Если таких объектов несколько, то они все будут считаться опорными и анализ по онтологическому критерию будет проведен для каждого из этих объектов.

Структура онтологии предполагает наличие у каждого из ее объектов свойства (DatatypeProperty) «является Термином», имеющее логический тип значения. Это свойство является вспомогательным и определяется экспертом исходя из критерия, насколько данный объект онтологии является характерным конкретно для этой предметной области.

Степень близости слова/сочетания слов терминам рассматриваемой предметной области в

соответствии с тезаурусным критерием оценивается по следующей формуле:

$$k_{Ont} = \frac{k_i}{c + 1} \quad (5)$$

где k_i – результат первого этапа анализа; c – число отношений, связывающих опорный объект онтологии с ближайшим объектом, имеющим истинное значение свойства «является Термином».

В случае, если сам опорный объект имеет истинное значение данного свойства, то $c = 0$.

Таким образом, процесс оценки степени близости одно-/многословия к терминам проблемной области по метрике «термин/нетермин» в его онтологической составляющей представляет собой движение по графу, в узлах которого находятся объекты соответствующих классов онтологий.

2.2. Критерий вложенных связей

Помимо оценки степени терминологичности отдельно взятого слова/сочетания слов, разработанная метрика позволяет извлечь термины из текста посредством их сопоставления с имеющимися объектами и сочетаниями лемм соответствующих объектов с помощью отношений R_{add} , определенных в онтологии.

Таким образом, при сопоставлении входных сочетаний и объектов предметной области, связанных между собой однонаправленными отношениями R_{add} , термином рассматриваемой предметной области будет считаться многословие, лемма которого полностью совпадает с объединением лемм соответствующих объектов онтологии.

Определяющими для использования этого метода являются отношения R_{add} , связь объектов посредством которых позволяет формировать словосочетания естественным образом. Пример формирования многословий с помощью свойства «имеет Отношение»:

1. Найденная цепочка объектов: «Вращение» + «имеет Отношение» + «Двигатель» + «имеет Отношение» + «Переменный ток»;
2. Объединение лемм объектов онтологии: «вращение двигатель переменный ток»;
3. Термин, извлекаемый из обрабатываемого текста: «вращение двигателя переменного тока».

При этом извлеченные термины, входящие в свою очередь в термины, состоящие из большего количества слов, не рассматриваются в качестве терминов с целью избегания избыточности.

2.3. Метрики оценки результатов

Рассмотрим метрики оценки, применимые к задаче классификации. Сгруппируем ответы нашего гипотетического анализатора следующим образом:

- Истинно-положительные (**true positives, tp**) – те категории, которые мы ожидали увидеть и получили на выходе;
- Ложно-положительные (**false positives, fp**) – категории, которых быть на выходе не должно, и анализатор их ошибочно вернул на выходе;
- Ложно-отрицательные (**false negatives, fn**) – категории, которые мы ожидали увидеть, но анализатор их не определил;
- Истинно-отрицательные (**true negatives, tn**) – категории, которых быть на выходе не должно, и на выходе анализатора они тоже совершенно правильно отсутствуют.

В этом случае мера точности (P , *precision*) определяется так:

$$P = \frac{tp}{tp + fp} \quad (6)$$

Мера точности характеризует, сколько полученных от классификатора положительных ответов являются правильными.

Мера точности, однако, не дает представление о том, все ли правильные ответы вернул классификатор. Для этого существует так называемая мера полноты (R , *recall*):

$$R = \frac{tp}{tp + fn} \quad (7)$$

Мера полноты характеризует способность классификатора «угадывать» как можно большее число положительных ответов из ожидаемых [Ярушкина и др., 2014].

Помимо этого, удобно для характеристики классификатора, использующего разработанную семантическую метрику, использовать унифицированную метрику F_1 :

$$F_1 = 2 \times \frac{P \cdot R}{P + R} \quad (8)$$

Фактически это просто среднее гармоническое величин P и R . Именно F_1 используется, чтобы сформулировать пороговое качество разработанного семантического анализатора [Ярушкина и др., 2007b].

3. Семантическая метрика оценки терминологичности слов/сочетаний слов

В рамках решаемой поставленной задачи были проведены следующие действия:

1. Экспертом в области эксплуатации токарно-фрезерного станка с числовым программным управлением построена OWL-онтология соответствующей проблемной области;
2. Была разработана онтологически-ориентированная система извлечения терминологии, применяющая описанные выше метрики для решения задачи определения

терминологичности одно/многословий, извлекаемых из больших объемов технических текстов.

3.1. Онтологически ориентированная система извлечения терминологии

Разработанная OWL-онтология имеет иерархическую организацию и включает в себя 261 экземпляр классов и порядка 746 отношений объектов классов. На данный момент онтология имеет 4 уровня иерархии, что позволяет максимально конкретизировать термины предметной области, используемой при решении поставленной задачи.

Пример объявления класса «Концевая фреза» разработанной OWL-онтологии:

```
<owl:Class rdf:ID="Концевая_фреза ">
  <rdfs:label
rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Концевая фреза</rdfs:label>
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Фреза"/>
  </rdfs:subClassOf>
</owl:Class>
```

Для реализации описанных алгоритмов была разработана программная система «Онтологически-ориентированная система извлечения терминологии».

Алгоритм работы разработанной системы извлечения терминологии предполагает следующую последовательность действий:

1. Обработка текста модулем морфологического анализа;
2. Подсчет лингвистических и статистических характеристик полученного текста, содержащего морфологическую разметку, основным модулем системы;
3. Подсчет семантических характеристик слов и словосочетаний обрабатываемого текста, базирующийся на представленных методиках с использованием разработанной OWL-онтологии.

3.2. Результаты вычислительных экспериментов извлечения терминов из текстов соответствующей предметной области

Приводимые результаты экспериментов имеют целью изучение эффективности разработанных показателей. Были рассмотрены результаты работы двух показателей: тезаурусного и критерия внутренних связей; четырех категорий словоупотреблений: одиночных слов, двух-, трех-, четырехсловных словосочетаний.

Для проведения эксперимента использовался текст объемом около 62000 слов из руководства по эксплуатации токарно-фрезерного станка с ЧПУ.

К особенностям текстов данной предметной области можно отнести высокую насыщенность терминами, влияние научного стиля на лексико-семантические, морфологические, синтаксические параметры и формализованность содержания, опирающегося на логико-понятийную схему предметной области.

Для оценки эффективности подсчета показателей рассмотрены меры *Precision* (6), *Recall* (7) и F_1 -мера (8) для каждого показателя в каждой категории словоупотреблений.

Результаты экспериментов по извлечению терминов посредством применения тезаурусного критерия представлены в таблице 1, критерия вложенных связей – в таблице 2.

Так как в случае применения тезаурусного критерия оценивается терминологичность каждого слова/сочетания, поступающего на вход алгоритма, то для формального отделения терминов от нетерминов в результате его выполнения, необходимо ввести пороговое значение $k_{От} = 0,5$.

Таблица 1 – Результаты применения тезаурусного критерия

Кол-во слов	Термины	$k_{От} > 0,5$	Верно	<i>P</i>	<i>R</i>	F_1 -мера
1	294	120	88	0,73	0,29	0,42
2	631	305	133	0,43	0,21	0,28
3	361	379	214	0,56	0,59	0,57
4	107	196	120	0,61	1,12	0,79

Таблица 2 – Результаты применения критерия вложенных связей

Кол-во слов	Термины	Кандидаты	Верно	<i>P</i>	<i>R</i>	F_1 -мера
1	294	168	154	0,91	0,52	0,66
2	631	431	372	0,86	0,58	0,69
3	361	370	327	0,88	0,9	0,89
4	107	159	129	0,81	1,2	0,97

Анализ результатов выполнения разработанных методик необходимо рассматривать с учетом различий вариантов словоупотреблений:

3.2.1. Одиночные слова

Исходя из полученных выше результатов, следует отметить, что наилучшие показатели извлечения однословных терминов были получены при применении второго критерия. Почти все извлеченные алгоритмом одиночные слова являются терминами, в то время как всего было извлечено немногим более половины всех однословных терминов рассматриваемой проблемной области. *Recall* у тезаурусного показателя для однословных терминов хоть и ненамного ниже, но значение *Recall* позволяет судить о более низкой эффективности этого показателя. Таким образом, показатель вложенных связей оказался наиболее эффективным при вычислении однословных терминов, о чем свидетельствует и наивысшее значение F_1 -меры среди показателей.

3.2.2. Двухсловные словосочетания

Исходя из результатов анализа, можно сделать вывод, что тезаурусный признак значительно уступает по полноте и точности второму критерию, имеющему лучшие значения *Precision*, *Recall* и F_1 -меры среди всех результатов. При достаточно высокой точности критерий вложенных связей извлекает более половины двухсловных терминов предметной области. Таким образом, для извлечения двухсловных терминов наиболее эффективным также является второй критерий.

3.2.3. Трехсловные словосочетания

Удовлетворительным можно считать результат работы тезаурусного признака по извлечению трехсловных терминов: извлекается более половины трехсловных терминов предметной области при среднем *Precision*. Результаты работы второго критерия можно назвать лучшими, о чем позволяют судить достаточно высокие значения *Precision* и *Recall*.

3.2.4. Четырехсловные словосочетания

Тезаурусный признак и критерий вложенных связей оказались сопоставимыми по эффективности. Значение *Recall*, превышающее 1 для обоих показателей, свидетельствует об извлечении ими терминов, ранее не выделенных в ходе экспертного анализа. Несмотря на схожие результаты, тезаурусный признак проигрывает второму признаку за счет более низкого значения *Precision*. Таким образом, признак вложенных связей оказался наиболее эффективным для извлечения и четырехсловных терминов.

Полученные результаты экспериментов по извлечению терминов из инструкции по эксплуатации токарно-фрезерного станка с ЧПУ с использованием разработанной онтологии соответствующей предметной области позволяют сделать вывод о высокой эффективности использования критерия вложенных связей для решения поставленной задачи, особенно в случаях анализа трех- и четырехсловий.

Заключение

Таким образом, предложенная в данной работе семантическая метрика «термин/нетермин» на основе онтологии проблемной области позволяет выделить из массива поступающих одно-/многословий только те термины и сочетания, которые относятся к данной предметной области, устанавливая для каждого из входных сочетаний слов численное значение степени их близости к терминам рассматриваемой предметной области.

Данная метрика может быть использована как в качестве самостоятельной, так и в сочетании с лингвистическими и статистическими метриками, используемыми в процессе извлечения терминологии с целью обеспечения всестороннего анализа поступающих данных.

Библиографический список

[Андреев и др., 2013] Андреев И.А., Башаев В.А., Клейн В.В. Разработка программного средства для извлечения терминологии из текста на основании морфологических признаков, определяемых программой Mystem // Интегрированные модели и мягкие вычисления в искусственном интеллекте. – М.: Физматлит, 2013. – С. 1227–1236.

[Yarushkina, 2001] Yarushkina N. Soft computing and complex system analysis // International Journal of General Systems. – 2001. – Vol. 30, № 1. – pp. 71–88.

[Namestnikov et al., 2002] Namestnikov A., Yarushkina N. Efficiency of Genetic algorithms for automated design problems // Известия Российской академии наук. Теория и системы управления. – 2002. – № 2. – С. 127–133.

[Ярушкина и др., 2007а] Ярушкина Н.Г., Вельмисов А.П., Стецко А.А. Средства data mining для нечетких реляционных серверов данных // Информационные технологии. – 2007. – № 6. – С. 20–29.

[Добров и др., 2003] Добров Б.В., Лукашевич Н.В., Сыромятников С.В. Формирование базы терминологических словосочетаний по текстам предметной области // Тр. 5-й Всеросс. науч. конф. «Электронные библиотеки: перспективные методы и технологии, электронные коллекции» (RCDL-2003). – СПб., 2003. – С. 201–210.

[Афанасьева и др., 2011] Афанасьева Т.В., Ярушкина Н.Г. Нечеткий динамический процесс с нечеткими тенденциями в анализе временных рядов // Вестник Ростовского государственного университета путей сообщения. – 2011. – Т. 3. – С. 7–16.

[Кураленок и др., 2002] Кураленок И.Е., Некрестьянов И.С. Оценка систем текстового поиска // Программирование. – 2002. – 28(4). – С. 226–242.

[Ярушкина и др., 2014] Ярушкина Н.Г., Мошкин В.С. Применение онтологического подхода к анализу состояния локальной вычислительной сети // Радиотехника. – 2014. – № 7. – С. 120–124.

[Ярушкина и др., 2007б] Ярушкина Н.Г., Афанасьева Т.В. Нечеткие временные ряды как инструмент для оценки и измерения динамики процессов // Датчики и системы. – 2007. – № 12. – С. 46–50.

ALGORITHMS FOR EVALUATION OF WORD COMBINATIONS OR WORDS MEMBERSHIP DEGREE TO TERM LIST BASED ON SUBJECT AREA ONTOLOGY

I.A. Andreev., V.A. Bashaev, V.V. Klein, V.S. Moshkin, N.G. Yarushkina.

Ulyanovsk State Technical University, Russia

ares-ilya@yandex.ru

perevod73@yandex.ru

vikklein93@gmail.com

postforvadim@yandex.ru

jng@ulstu.ru

This article describes a semantic metric retrieve a list of terms from the texts for this specific problem, based on an analysis of its ontology. A formal model used OWL-ontologies, as well as models and algorithms for assessing membership degree of word or combinations of words to term list.

In addition, the metrics of performance evaluation of semantic algorithms and the implementation of formal models of representation of the domain

knowledge in the form of ontological and developed algorithms in software system terminology extraction from text are considered.

Introduction

Operation principles of the existing term extraction algorithms in lexicography and terminology are based on statistical and linguistic methods.

When analysing big documentation corpora, the domain specific nature is essential to be taken into consideration to obtain a list of terms specific for this domain. Semantic algorithms based on the determined semantic metrics are used to solve such tasks.

One of the most universal methods to present expert knowledge in the context of semantic description comprehensiveness is the ontological approach. That is why one of the most important ways to solve the problem of terminology extraction from a big corpus of technical documentation is development and use of semantic metrics based on ontological models.

Main part

The operation algorithm of the term extraction system developed supposes the following execution sequence:

1. Processing a text via the morphological analysis module
2. Calculation of linguistic and statistical characteristics of the text processed and containing morphological tagging via the main system module
3. Calculation of semantic characteristics of words and word combinations of the processed text based on the presented methods using the OWL ontology developed

We used a text of 62,000 words that is a part of an operators manual for a CNC turning milling machine to carry out an experiment. To estimate effectiveness of the indicators calculation, we considered such criteria as *Precision*, *Recall* and F_1 for each indicator in each linguistic usage category. The results of the experiment on term extraction from the operators manual for a CNC turning milling machine using the developed ontology of the relevant domain enable us to conclude that nested bond criterion is high efficient to solve the set task, especially in case of tri- and quadri-words.

Conclusion

Thus, the «term/non-term» semantic metric proposed in this paper based on a domain ontology enables us to extract only the terms and term combinations relevant to the domain from an incoming corpus of single- or multi-words establishing a numerical value of relevance to the domain terms for each of the input ones.



OSTIS-2015

(Open Semantic Technologies for Intelligent Systems)

УДК 004.891

МЕТОДИКИ ПОСТРОЕНИЯ НЕЧЕТКИХ ОНТОЛОГИЙ СЛОЖНЫХ ПРЕДМЕТНЫХ ОБЛАСТЕЙ

Мошкин В.С., Ярушкина Н.Г.

*Ульяновский государственный технический университет,
г. Ульяновск, Российская Федерация*

PostForVadim@yandex.ru

jng@ulstu.ru

В данной работе представлена методология построения нечетких онтологий с точки зрения организационного процесса в условиях сложной предметной области и наличия нескольких экспертов, описывающих её в терминах OWL. Помимо этого, рассмотрены особенности построения нечеткой онтологии с использованием методологии Fuzzy OWL на примере онтологии состояния локальной вычислительной сети с последующей интеграцией связанных знаний, представленных в форме системы продукционных правил.

Ключевые слова: онтология, нечеткая онтология, логический вывод, fuzzy owl.

Введение

В настоящий момент для разработки систем, основанных на знаниях, актуальной является задача объединения различных репрезентативных подходов с целью обеспечения наиболее полного представления знаний в рассматриваемой области.

Типичными моделями представления знаний в экспертных системах поддержки принятия решений являются:

- продукционная модель;
- модель, основанная на использовании фреймов;
- модель семантической сети;
- логическая модель;
- онтологическая модель.

Одной из возникающих проблем при решении задачи логического вывода знаний, необходимых в процессе анализа показателей состояния реально существующей сложной технической системы, является проблема полноты и адекватности описания предметной области.

Онтология, будучи формальным отображением реальности, требует четкого представления информации, а реальность не всегда может быть четко выражена. Такое несоответствие порождает сложности концептуального характера. Среди таких сложностей – реализация процедур адекватного нечеткого логического вывода, реализация адекватных блоков интеграции онтологической

информации и прочее. В связи с тем, что объединение различных форматов представления знаний на онтологической основе является наиболее удачным с точки зрения автоматизированной обработки и восприятия человеком методом формирования базы знаний экспертных систем, одной из наиболее актуальных задач в сфере интеллектуализации таких систем является решение проблемы учета нечеткости в человеческих рассуждениях [Ярушкина и др., 2014].

Переход от обычных онтологий к нечетким выглядит вполне естественным, поскольку понятия и отношения естественного языка, представляющие собой исходный материал для построения онтологии неоднозначны, неточны и не имеют жестких границ. Поэтому адекватным средством формализации онтологий могут служить модели на базе лингвистических переменных, нечеткие множества, нечеткие отношения, нечеткие графы и нечеткие деревья, нечеткие ограничения, нечеткие реляционные и алгебраические системы [НГС, 2007].

При интеграции онтологии и систем продукционных правил, пользователь не получает возможности выбора варианта действия в связи с ограниченностью инструмента логического вывода на основе четких экспертных знаний. В случае же введения в онтологию отдельных наборов нечетких отношений и объектов, в результате вывода пользователь получает более широкое множество вариантов решения поставленной задачи, каждый из которых обладает определенной мерой

уверенности, выраженной соответствующим значением функции принадлежности.

1. Построение нечетких онтологий сложной предметной области

Онтология представляет собой систему, состоящую из множества понятий, их определений и аксиом, необходимых для ограничения интерпретации и использования понятий. Чем сложнее формализуемая проблемная область, тем больше внимания необходимо уделять унификации форматов представляемых данных.

Помимо этого, при построении онтологии нужно учесть субъективную оценку понятий предметной области отдельным экспертом, исходя из его компетентности в данной конкретной подобласти рассматриваемой области. Это также возможно реализовать с помощью задания значений соответствующих функций принадлежности, т.е. посредством внесения нечеткости в разрабатываемую онтологию.

На рисунке 1 в общем виде представлена методика построения нечетких онтологий сложной предметной области.

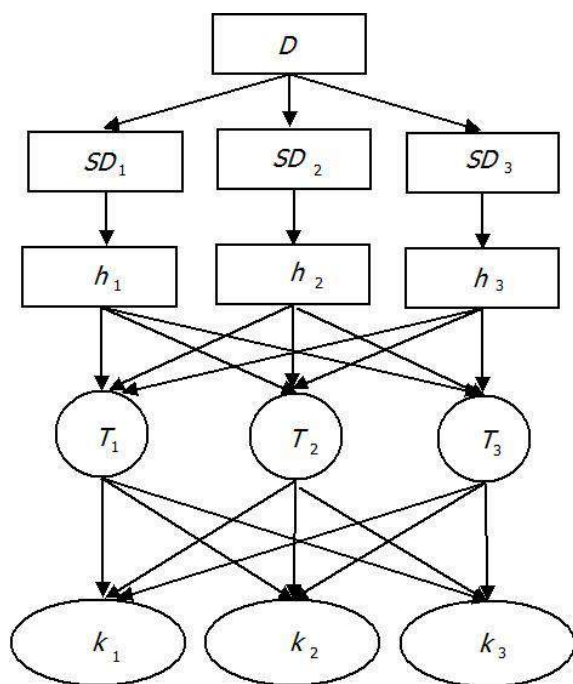


Рисунок 1 – Построение нечеткой онтологии

Здесь на первом (верхнем) уровне расположена комплексная проблемная область D , которая предполагает слияние источников информации (экспертов, коллекций текстов) из разных областей. Проблемная область (ПрО) разбивается на соответствующие подобласти SD (второй уровень), причем каждая подобласть характеризуется своими источниками информации (третий уровень). Чаще всего такими источниками информации являются эксперты каждой конкретной подобласти, которые при этом имеют определенное представление о

категориях и понятиях смежных подобластей в рамках единой ПрО D .

Экспертами строится иерархия понятий ПрО, где на четвертом уровне находятся базовые понятия или категории онтологии T_i , где $i = [1 \dots n]$, а на пятом (нижнем) уровне – лингвистические переменные, относящиеся к именам категорий k_j , где $j = [1 \dots m]$. При этом нечеткость отношения между категорией онтологии и конкретной лингвистической переменной выражается в присвоении этому отношению значения функции принадлежности $0 \leq \mu(t, k) \leq 1$, где

$t \subseteq T_i, k \subseteq K_j$. При этом совокупность категорий онтологии, лингвистических переменных, а также значений соответствующих функций принадлежности образуют нечеткие множества категорий рассматриваемой онтологии [Тарасов и др., 2012].

При разработке онтологической системы эксперты производят выбор и соотнесение имен категорий с ключевыми словами. Поскольку каждый эксперт есть специалист в определенной подобласти ПрО, а уровень его компетентности в других подобластях ниже, предварительно определяется вес эксперта по разделам (подобластям) ПрО, выражаемый числом из интервала $[0, 1]$. Далее эксперты определяют степень связи между категорией T_i и ключевым словом k_j , которая также задается числом из интервала $[0, 1]$. Сближение значений весов может характеризовать «степень разделения» онтологии разными специалистами.

Рассмотрим особенности данной модели на примере построения нечеткой онтологии тестирования состояния ЛВС при искусственном повышении трафика. Формализуемая предметная область (D) была разделена на подобласти, основными объектами рассмотрения которых были:

- SD_1 : сетевое оборудование ЛВС;
- SD_2 : ПО администрирования ЛВС;
- SD_3 : ПО и аппаратное обеспечение (АО) тестирования сети.
- SD_4 : основные характеристики работоспособности сети.

Так как построение онтологии производилось коллективом экспертов на базе конкретной ЛВС, третий уровень модели включает лишь один источник информации h и каждое понятие ПрО, выделенное группой экспертов в виде класса или объекта онтологии, имеет 1 в качестве значения веса выделившего его эксперта.

Для некоторых свойств онтологии введены лингвистические переменные: например, для отношения «имеетКоличествоКолизий» – «большое», «среднее», «малое». Значения свойств вводятся как нечеткие утверждения с некоторым значением, соответствующим функции принадлежности $\mu(t, k)$. Значения данных свойств

рассматриваются как предикаты p , зависящие от фактического значения показателей характеристик работоспособности ЛВС:

< «Сеть», «имеетКоличествоКоллизий», «большое», $p_{\text{большое}}$ (Значение показателя) >

Потребность в онтологии подобного вида не подлежит сомнению в связи с тем, что она позволит решать задачи оптимизации процессов в общем виде. Блок нечеткого логического анализа, использующий помимо таким образом построенной нечеткой онтологии еще и набор SWRL-правил с заложенными лингвистическими переменными в качестве параметров правил, позволит эффективно извлекать полезную информацию из малого набора утверждений [Кучеренко и др., 2005].

2. Методика построения нечетких онтологии Fuzzy OWL

Формальная семантика OWL описывает, как получить логические следствия, имея такую онтологию, т.е. получить факты, которые не представлены в онтологии буквально, но следуют из ее семантики.

OWL (англ. Web Ontology Language) — язык описания онтологий, позволяющий представлять классы и отношения между ними. В основе языка — представление действительности в модели данных «объект — свойство». OWL является переформулировкой дискрипционной логики (ДЛ) с использованием синтаксиса XML.

2.1. Расширение OWL с помощью Fuzzy OWL

В настоящее время существует наиболее универсальный и имеющий достаточную поддержку со стороны разработчиков интеллектуальных экспертных систем подход к построению нечетких онтологий - методология Fuzzy OWL.

Синтаксис Fuzzy OWL предполагает три формата используемого алфавита: для представления нечетких классов (fuzzy concepts), нечетких отношений (fuzzy roles) и нечетких объектов классов онтологии (fuzzy individuals).

В отличие от OWL DL, реализующей разновидность дискрипционной логики $SHOIN(D)$, концепция Fuzzy OWL охватывает все положения логики $SROIQ(D)$:

- S – это набор базовых классов, имен свойств и имен объектов, является подмножеством DL в связке с транзитивными свойствами;
- R – возможность определения составных аксиом вложенности ролей в наборе аксиом перечисленных видов : $R \circ S \subseteq R, R \circ S \subseteq S$, где $R \circ S$ - композиция ролей;
- O – наличие номиналов (так называемых классов единичных объектов);
- I – наличие обратных свойств;

- Q – качественные ограничения кардинальности ролей;
- (D) – типы данных, свойства, которые связывают объекты с типами данных [Bobillo et al., 2009].

2.2. Свойства-аннотации OWL

Основополагающим элементом методологии построения нечетких онтологий с точки зрения структуры OWL, является наличие свойства-аннотации fuzzyLabel, которое хранит параметры и значения функции принадлежности, соотносимые с конкретным классом, объектом, отношением внутри заданного нечеткого множества данных [Bobillo et al., 2011].

Свойства-аннотации OWL могут использоваться для добавления информации для классов, отдельных индивидов и свойств объектов/типов данных. OWL позволяет аннотации с различной информацией и метаданными классов, свойств, индивидов и онтологий. Эти части информации могут принимать форму аудита или редакционной информации.

Разновидность языка OWL - OWL DL, используемая в экспертных системах для дальнейшего получения знаний из уже существующих в онтологии, накладывает ряд ограничений на использование свойства аннотации — двумя из наиболее важных ограничений являются:

- наполнителем для свойства аннотации должны быть символьные данные, ссылки URI или индивидуумы;
- в свойствах аксиом нельзя использовать свойства аннотации, например они не могут использоваться в иерархии свойств, поэтому они не могут иметь подсвойства, или быть подсвойством другого свойства. Также для них не должны устанавливаться домен и диапазон [Lee et al., 2005].

2.3. Особенности методологии Fuzzy OWL

Методология Fuzzy OWL с помощью необходимых параметров нечеткости в аннотациях к элементам онтологии предполагает возможность делать нечеткими:

- классы;
- объекты классов онтологий;
- свойства (свойства типов данных и свойства объектов);
- аксиомы онтологий.

Алгоритм задания нечеткого отношения, согласно методике Fuzzy OWL, выглядит следующим образом:

1. Создается новое свойство-аннотация fuzzyLabel, в котором будут задаваться параметры нечеткости каждого элемента онтологии;
2. Лингвистические переменные определяются в форме новых созданных типов данных (Datatype).

3. Для каждого созданного типа данных задаются верхнее и нижнее границы принимаемых значений, выбирается тип функции принадлежности и для нее задаются соответствующие параметры [Straccia, 2005]. Пример определения лингвистической переменной «БольшоеЧислоКоллизий» и соответствующей функции принадлежности:

```
<AnnotationAssertion>
  <AnnotationProperty IRI="#fuzzyLabel"/>
  <IRI>#БольшоеЧислоКоллизий</IRI>
  <Literal datatypeIRI="&rdf;PlainLiteral">
    <fuzzyOwl2 fuzzyType="datatype">
      <Datatype type="rightshoulder"; a="15.0"
b="30.0"/>
    </fuzzyOwl2>
  </Literal>
</AnnotationAssertion>
```

Методология Fuzzy OWL предполагает аналитическое определение нескольких возможных типов функций принадлежности:

- Треугольные (triangular);
- Трапециевидные (trapezoidal);
- Линейные (linear);
- Лево- и правосторонние (left-shoulder and right-shoulder).

3. Логический вывод на базе нечеткой онтологии

3.1. Особенности представления знаний в SWRL

Формой представления знаний, позволяющей делать непосредственный вывод из содержащихся в онтологии данных и дополнять её новыми, является форма набора SWRL-правил.

Semantic Web Rule Language (SWRL) – это технология, которая основана на объединении OWL и RuleML, т.е. объединены онтологии (OWL DL) и правила.

Представление рассматриваемой области знаний в виде причинно-следственных связей объясняется тем, что рекомендации и выводы в области применения чаще всего имеют форму сложных условных высказываний [Воронина и др., 2010].

В общем виде SWRL-правила, обеспечивающие процесс логического вывода рекомендации, определены выражением 1.

$$a_1 \wedge a_2 \wedge \dots \wedge a_n \rightarrow b, \quad (1)$$

где
 a_1, \dots, a_n – предикаты (атомы) антецедента (условия),

b – консеквент (следствие) правила, состоящий из одного атома.

Выражение в SWRL, т.е. SWRL-правило, состоит из головы и тела: голова – это консеквент продукционного правила и может состоять только из одного атома, а тело – это антецедент, который может состоять из нескольких атомов.

Процесс вывода новых фактов из содержащихся в онтологии осуществляется посредством использования ризонера, поддерживающего соответствующий вид логики.

Принципы работы механизма ризонинга на основе SWRL-правил имеют ряд преимуществ перед остальными подобными технологиями:

- правила SWRL не содержат конкретных объектов, а только ссылаются на них, что дает возможность применять одно и то же правило к ряду групп объектов;
- правила SWRL могут быть добавлены к OWL-описанию, т.е. включены в онтологию;
- написание и «чтение» правил удобнее, если для этого существует специальный язык [Соловьев и др., 2006].

В общем виде SWRL-правила (на примере проблемной области оценки состояния ЛБС) с учетом возможности обработки нечетких значений отношений, представленных в онтологии, выглядят следующим образом:

*Ситуация(?x) \wedge
 включаетПоказатель(?x,Сеть) \wedge
 имеетКолличествоКоллизий (Сеть, большое) \wedge
 включаетПоказатель(?x,Сеть) \wedge
 имеетФоновуюНагрузку(Сеть, невысокая) \rightarrow
 предполагает(?x, Проблема_архитектуры)*

В правила, помимо нечетких отношений и объектов, могут включаться и конкретные численные значения соответствующих показателей, а также наименования, четкие отношения и другие характеристики, заданные в онтологии.

Значениям лингвистических переменных, заданным в теле SWRL-правила, соответствуют некоторые степени принадлежности, определенные в OWL-онтологии, и в дальнейшем формирующие некую величину степени уверенности в процессе вывода сыгравшего правила.

3.2. Алгоритм логического вывода на базе нечеткой онтологии

Логический вывод рекомендаций с использованием онтологии, содержащей нечеткие отношения, классы, а также объекты классов, имеет существенные отличия по сравнению с извлечением знаний из четкой онтологии (см. рисунок 2).

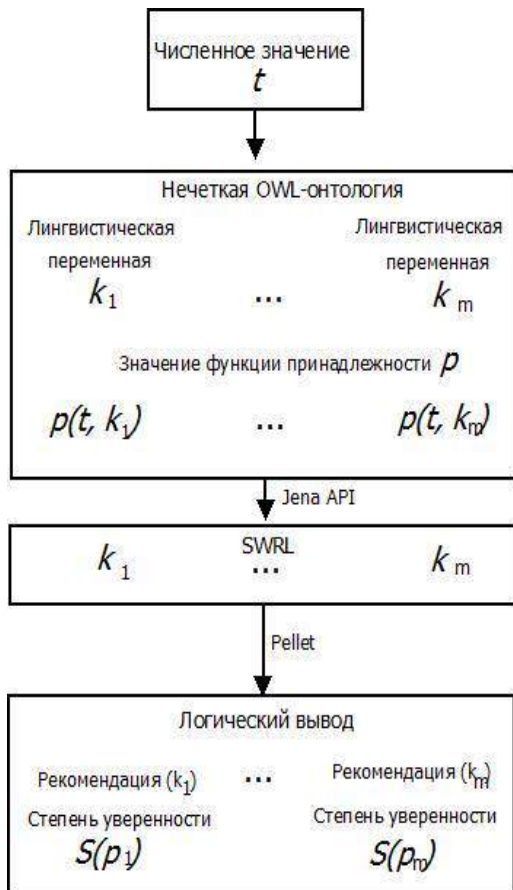


Рисунок 2 – Логический вывод на основе нечеткой онтологии

Основная последовательность действий в процессе логического вывода рекомендации на основе взаимодействия нечеткой OWL-онтологии и системы продукционных SWRL-правил включает в себя следующие этапы:

4. Проверка входных данных на наличие в онтологии соответствующих классов, отношений, объектов;
5. Определение лингвистических переменных и соответствующих значений функции принадлежности по численному значению входного параметра, определенного в онтологии;
6. Запрос на выполнение набора правил анализа ситуации и вывода рекомендации;
7. Запрос на получение связанных данных (обращение к родителям объектов);
8. Запрос на выполнение правил с учетом полученных связанных данных;
9. Вывод результата анализа (одна или несколько рекомендаций) в виде объекта разработанной OWL-онтологии с учетом показателя степени уверенности S .

При этом показатель степени уверенности является функцией от поступающих значений функций принадлежности, относящихся к заложенным в соответствующем правиле значениям предметной области (см. выражение 2):

$$S_{Rule} = \prod_{i=1}^l \mu(t_i, k_j) \quad (2)$$

где t – термин проблемной области, присутствующий в сработавшем SWRL-правиле;
 k – лингвистическая переменная, $j = [1..m]$;
 $\mu(t, k)$ – значение соответствующей функции принадлежности.

В целях повышения наглядности полученного результата, необходимо провести нормализацию показателей степеней уверенности всех сработавших правил:

$$S_{n\text{ NORM}} = \frac{S_n}{\max(S_n)} \quad (3)$$

где $S_{n\text{ Norm}}$ – нормализованное значение степени уверенности рекомендации каждого n -ого правила.

Таким образом, наиболее вероятное решение по исправлению той или иной ситуации, моделируемое значениями входных параметров, будет иметь значение степени уверенности равное 1.

В конечном же итоге, пользователь экспертной системы получит на выходе набор рекомендаций на естественном языке, каждой из которых будет поставлено в соответствие некоторое значение $S = [0..1]$. Таким образом, благодаря включению нечеткости в описание предметной области, значительно снижаются риски потери возможных результатов работы блока логического вывода, увеличивается гибкость процесса вывода, а пользователю предоставляется гораздо широкий выбор вариантов, которые при этом упорядочены по мере их релевантности с точки зрения совокупности экспертных знаний, заложенных в онтологии и наборе правил.

Заключение

Таким образом, предложенная в данной работе методология построения нечетких онтологий с точки зрения организационного процесса в условиях сложной предметной области и наличия нескольких экспертов, описывающих её в терминах OWL, позволяет:

- a. существенно сократить временные издержки построения онтологии,
- b. избежать противоречий в интерпретации и формализации того или иного элемента предметной области;
- c. снизить субъективную составляющую оценки понятий Про отдельным экспертом, исходя из его компетентности в данной конкретной подобласти рассматриваемой области.

Все это возможно реализовать с помощью задания степени принадлежности всем определяемым объектам и отношениям, т.е. посредством внесения нечеткости в разрабатываемую онтологию.

Рассмотренная в статье методология построения нечетких онтологий Fuzzy OWL позволяет значительно расширить возможности использования онтологий в процессах получения

новых знаний из уже определенных в ней, еще больше приблизить формализованное представление знаний о ПрО к естественному для человека варианту их представления и восприятия.

Описанный в работе модифицированный алгоритм интеграции онтологической и продукционной форм хранения и представления знаний, благодаря введению нечеткости в элементы онтологии, снижает риски потери возможных результатов работы блока логического вывода основанной на данной базе знаний экспертной системе, а также увеличивает гибкость процесса вывода рекомендации, предоставляя пользователю более широкий выбор вариантов, упорядоченных по степени их релевантности.

Библиографический список

[Воронина и др., 2010] Воронина И.Е., Пигалкова Е.А. Интеграция знаний продукционного характера в правовую онтологическую модель с помощью SWRL-правил // Вестник ВГУ, серия «Системный анализ и информационные технологии». – 2010. – № 2. – С. 139–144.

[Кучеренко и др., 2005] Кучеренко Е.И., Павлов Д.А. Некоторые аспекты анализа развития нечетких онтологий. // Искусственный интеллект. Донецк. 2005. – С.162-169.

[НГС, 2007] Нечеткие гибридные системы/ И.З.Батыршин, А.О.Недосекин, А.А.Стецко, В.Б.Тарасов, А.В.Язенин, Н.Г.Ярушкина. – М.: Физматлит, 2007.

[Соловьев и др., 2006] Соловьев В.Д., Добров Б.В., Иванов В.В., Лукашевич Н.В. Онтологии и тезаурусы. Учебное пособие.- Казань, Москва.-2006.

[Тарасов и др., 2012] Тарасов В.Б., Калущая А.П., Святкина М.Н. Гранулярные, нечеткие и лингвистические онтологии для Обеспечения взаимопонимания между когнитивными агентами // Открытые семантические технологии проектирования интеллектуальных систем (OSTIS-2012): материалы II междунар. науч.-техн. конф. – Минск: БГУИР, 2012. – С. 267-278.

[Ярушкина и др., 2014] Ярушкина Н.Г., Мошкин В.С. Применение онтологического подхода к анализу состояния локальной вычислительной сети // Радиотехника. – 2014. – № 7. – С. 120–124.

[Bobillo et al., 2009] Bobillo F., Straccia U. Fuzzy description logics with general t-norms and datatypes // Fuzzy Sets and Systems 160 (23) - 2009 – p. 3382–3402.

[Bobillo et al., 2011] Bobillo F., Straccia U. Fuzzy ontology representation using OWL 2 // International Journal of Approximate Reasoning. Vol. 52 - 2011. - p. 1073–1094

[Lee et al., 2005] Lee C.S., Jian Z.-W., Huang L.-K., A fuzzy ontology and its application to news summarization // IEEE Transactions on Systems, Man and Cybernetics, PartB Vol. 5.- 2005. - p. 859–880.

[Straccia, 2005] Straccia U. Towards a Fuzzy Description Logic for the Semantic Web // 2nd European Semantic Web Conference, 2005. – P. 167-181.

METHODS OF CONSTRUCTION OF FUZZY ONTOLOGIES OF COMPLEX SUBJECT AREAS

Moshkin V.S., Yarushkina N.G.

Ulyanovsk State Technical University, Russian Federation

PostForVadim@yandex.ru

jng@ulstu.ru

This paper presents a methodology for constructing fuzzy ontologies from the perspective of the

organizational process in a complex domain and the presence of several experts, describing it in terms of OWL. In addition, the features of the construction of fuzzy ontology using the methodology of Fuzzy OWL ontology for example, the state of the local area network with subsequent integration-related knowledge presented in the form of production rules.

Introduction

Ontology, as a formal representation of reality, requires a clear presentation of information, but the reality may not always be clearly expressed. This discrepancy raises conceptual complexity.

The transition from conventional to fuzzy ontologies looks quite natural, since the concepts and relations of natural language, which are the raw material for building ontology ambiguous, imprecise and do not have rigid boundaries.

Main Part

Fuzzy ontology consisting of five levels reduce the subjective component evaluation data domain concepts of individual experts on the basis of its competence in a given subregion this area.

There are the most versatile approach to the construction of fuzzy ontologies - Methodology of Fuzzy OWL. Fuzzy OWL syntax involves three formats used alphabet to represent fuzzy concepts, fuzzy roles and fuzzy individuals. The concept of Fuzzy OWL covers all the provisions of logic SROIQ (D).

Thanks to the inclusion of fuzziness in the description of the subject area, significantly reduces the risk of loss of possible outcomes of the unit inference increases the flexibility of the process output, and the user is given a much wide range of options that are ordered at the same time as their relevance in terms of combined expertise, embedded in ontology and a set of rules.

Conclusion

Thus, proposed in this paper a methodology for constructing fuzzy ontologies from the perspective of the organizational process in a complex domain reduce the subjective component evaluation data domain concepts of individual experts on the basis of its competence in this area.

Considered in the paper the methodology of constructing fuzzy Fuzzy OWL ontologies can significantly extend the possibilities of using ontologies in the process of acquiring new knowledge from already defined in it.

Modified algorithm of integration ontological forms a production storage and presentation of knowledge, thanks to the introduction of fuzziness in the ontology elements, reduces the risks of loss of the possible outcomes of the unit inference based on this knowledge base of the expert system.



OSTIS-2015

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822:514

ОНТОЛОГИЧЕСКИЙ ПОДХОД К ФОРМИРОВАНИЮ ПРОЕКТНЫХ ЗАПРОСОВ ИНТЕЛЛЕКТУАЛЬНОГО АГЕНТА

Наместников А.М., Субхангулов Р.А.

*Ульяновский государственный технический университет,
г. Ульяновск, Россия*

nam@ulstu.ru,

subkhangulov-ruslan@yandex.ru

В статье рассматривается применение интеллектуальных агентов в процессе формирования информационных запросов к электронному архиву технических документов. Интеллектуальный агент содержит онтологическое описание профиля проектировщика, которое выражается в виде фрагмента онтологии предметной области. Задачами интеллектуального агента являются: пополнение информации о пользовательских потребностях и нахождение текстовых документов по запросу пользователя электронного архива.

Ключевые слова: интеллектуальная система; онтология; интеллектуальный агент; информационный поиск.

ВВЕДЕНИЕ

Основными задачами электронного архива является обеспечение коллективной работы проектно-конструкторских отделов над проектом, добавление, хранение и поиск технических документов (ТД). Поиск часто осуществляется по заранее определенным реквизитам документов и по ключевым словам. Однако данные модели поиска не имеют представления об информационных потребностях пользователя и, тем самым, всегда присутствует вероятность того, что документы, которые были отобраны, не позволяют сократить информационную неопределенность проектировщика. Современные системы информационной поддержки используют механизмы интеллектуального поиска. Интеллектуальный поиск – это ключевая тенденция в современном информационном поиске, которая предполагает способность поисковой системы к самоорганизации, осуществление независимого общения с пользователем, эффективный поиск текстовых документов, реагирующий на изменения информационной потребности пользователя. В основе интеллектуального поиска есть возможность использовать интеллектуальных агентов, функционирование которых основано на предметно-ориентированных знаниях. Эти знания могут быть представлены в виде онтологии предметной области [Добров Б.В. и др., 2006, Гаврилова Т.А. и др., 2000]. Интеллектуальные агенты изучают историю пользовательских запросов, выполняют поиск

документов, обмениваются метаинформацией между собой.

В данной статье представлена модель формирования поисковых запросов, основанная на использовании интеллектуальных агентов, которые учитывают предпочтения проектировщика в процессе поиска ТД. Фактически, речь идет о формировании индивидуального профиля проектировщика, который активно взаимодействует с электронным архивом ТД проектной организации. Такой профиль может применяться в задачах онтологически-ориентированного информационного поиска текстовых документов, что позволит максимально удовлетворить информационную потребность пользователя.

1. Структура интеллектуального агента

Под интеллектуальным агентом [Рассел и др.] понимается сущность, которая через систему датчиков получает информацию о среде и воздействующая посредством исполнительных механизмов на эту среду. Под интеллектуальностью следует понимать наличие у агента модели пользовательских потребностей и механизма их удовлетворения. Таким образом, интеллектуальный агент должен обладать следующими свойствами:

- Автономность - агент должен выполнять большую часть своей работы автономно, не взаимодействуя с человеком или другими агентами.

- Коммуникабельность – агент должен уметь общаться с пользователем, получая от него задания и предоставляя результаты.
- Адаптируемость и адаптивность поведения – в ходе общения с пользователем агент должен уметь настраиваться (или, хотя бы быть настраиваемым) под привычки и методы работы конкретного пользователя.
- Восприимчивость – агент, находясь в окружающей его информационной среде, должен воспринимать некоторым образом изменения окружающей среды и реагировать на изменения.
- Проактивность - агент не только должен формально выполнять поставленную задачу поиска, но и должен собирать при этом полезную для пользователя информацию, относящуюся к запросу пользователя.

Задачами интеллектуального агента в данной работе являются: пополнение знаний о предметной области; анализ пользовательских потребностей; формирование ранжированного списка документа на основе анализа потребностей пользователя в поисковой системе.

Для данных задач будем применять многослойный интеллектуальный агент с иерархической базой знаний, который имеет следующую архитектуру (рисунок 1).

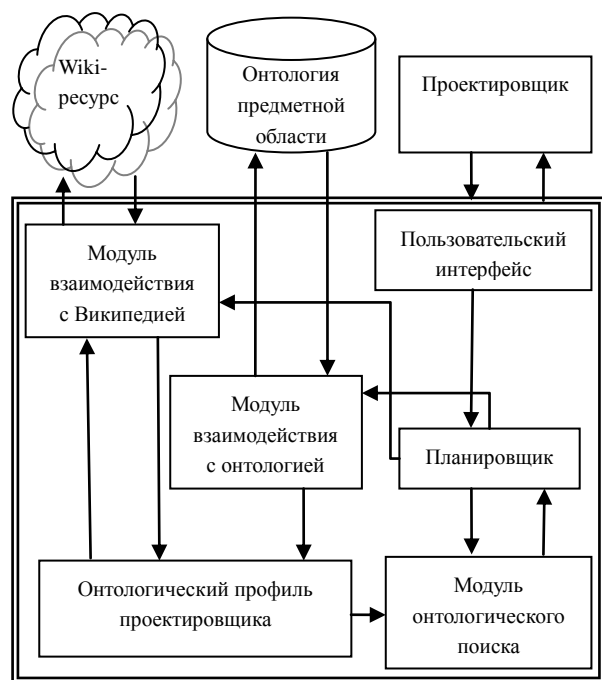


Рисунок 1. Архитектура интеллектуального агента

Интеллектуальный агент состоит из следующих слоев:

- Пользовательский интерфейс;
- Планировщик;
- Слой пользовательского профиля;
- Слой онтологического поиска;
- Слой взаимодействия с Википедией;
- Слой взаимодействия с онтологией проекта.

Пользовательский интерфейс осуществляет взаимодействие пользователя с интеллектуальным агентом. Пользователь вводит поисковые запросы, выраженные в виде набора терминов, получает список релевантных документов и оценивает документы на степень удовлетворенности своих потребностей.

Планировщик осуществляет следующие функции: преобразует пользовательский запрос к онтологическому виду, выраженный в виде множества концептов предметной области; передает преобразованный запрос модулю онтологического поиска; получает список релевантных документов и выводит в модуле пользовательского интерфейса; пополняет пользовательский профиль знаниями о предметной области.

Слой пользовательского профиля содержит знания о предметной области. Данные знания выражены в виде концептов онтологии проектов и концептов извлеченных из Википедии. В дальнейшем, пользовательский профиль используется в процессе онтологического поиска.

Слой онтологического поиска получает от планировщика онтологический запрос, выполняет поиск релевантных документов и возвращает набор ранжированных документов планировщику.

В модуле взаимодействия с Википедией реализованы функции извлечения концептов. Далее подробно рассмотрено построение профиля проектировщика на основе онтологии предметной области.

2. Онтология предметной области

Онтология предметной области представим в виде кортежа:

$$O = \langle C, W, R^D, F \rangle,$$

где C - это множество понятий предметной области, которая образует основу понятийного аппарата процесса проектирования; $W = W^S \cup W^P$ - множество терминов предметной области (W^S - множество терминов на уровне стандартов, W^P - множество терминов на уровне проектов); R^D - множество отношений:

$$R^D = \{R_G^D, R_C^D, R_A^D\},$$

где R_G^D - антисимметричное, транзитивное, нереклексивное бинарное отношение обобщения; R_C^D - бинарное транзитивное отношение композиции («часть-целое»); R_A^D - бинарное отношение однонаправленной ассоциации.

Множество понятий C записывается следующим образом:

$$C = (C^{S_1} \cup C^{S_2} \cup \dots \cup C^{S_k}) \cup (C^{W_1} \cup C^{W_2} \cup \dots \cup C^{W_k}) \cup C^P,$$

где $C^{S_i}, i = \overline{1, k}$ - множество понятий предметной области, рассматриваемых в рамках i -ой серий стандартов, используемых в проектной организации (например, ГОСТ 34.602-89, ГОСТ 19.201-78 и т.д.); C^{W_i} - множество понятий предметной области, извлекаемые из wiki-ресурса; C^P - множество понятий предметной области, извлекаемых из ТД по реализованным проектам.

Множество интерпретирующих функций представлено в виде:

$$F^D = \{F_{WC}^D, F_{CW}^D, F_{C^P C^S}^D, F_{C^S C^W}^D\},$$

где $F_{WC}^D : \{W\} \rightarrow \{C^P\}$ - функция, сопоставляющая набору терминов подмножество понятий предметной области, задаваемая алгоритмически; $F_{CW}^D : \{C^P\} \rightarrow \{W\}$ - функция интерпретации концептов, сопоставляющая каждому концепту набор терминов из словаря; $F_{C^P C^S}^D : \{C^P\} \rightarrow \{C^S\}$ - функция интерпретации подмножества понятий на проектном уровне онтологии, позволяющая осуществить переход на уровень понятий (концептов), определенных в стандартах; $F_{C^S C^W}^D : \{C^S\} \leftrightarrow \{C^W\}$ - функция интерпретации подмножества понятий на уровне стандартов, позволяющая осуществить переход на уровень понятий (концептов), определенных в Википедии и проектных документах.

3. Формирование онтологического профиля проектировщика

Рассмотрим процесс формирования онтологического профиля проектировщика на основе информации, извлекаемой из Википедии. Википедия — свободная общедоступная мультязычная универсальная интернет-энциклопедия, реализованная на принципах Wiki. Концепты в данной библиотеке представлены в виде HTML-страницы, для связи между страницами используются гиперссылки, тем самым гиперссылки между страницами символизируют семантическую связь между понятиями. Опираясь на систему гиперссылок, существует возможность в автоматическом режиме переходить от одной страницы к другой, извлекая знания о понятиях предметной области.

Рассмотрим данный алгоритм поэтапно:

1. На первоначальном этапе на вход модуля взаимодействия с Википедии поступают понятия из технического задания на проектирование.

2. Далее выполняется извлечение понятий из Википедии. В основе этого процесса лежит модифицированный алгоритм волновой

трассировки. Данный процесс состоит из ряда последовательных этапов:

2.1. На вход системы поступают множество понятий полученных на этапе (1).

2.2. Выполняется поиск страниц в Википедии, соответствующих полученным концептам.

2.3. Выполняется анализ страницы, полученный на этапе (2.2), результатом является нахождение тех концептов, для которых одновременно выполняются следующие условия:

- существует страница, которая описывает концепт;
- анализируемая страница содержит гиперссылки на страницу найденного концепта;
- страница концепта должна содержать обратную гиперссылку на анализируемую страницу;

2.4. Обнаруженные концепты добавляются в черновик профиля проектировщика.

2.5. Проверяется условие существования маршрута между всеми первичными концептами, которые получены на этапе (1).

2.6. Если условие (2.5) выполняется, то это означает окончания модифицированного алгоритма волновой трассировки, если не выполняется, то пункты (2.2)-(2.5) выполняются снова для концептов, полученных на этапе (2.3).

Таким образом, на выходе второго этапа получим множество понятий, между которыми существуют не идентифицированные семантические отношения. Однако может оказаться так, что это множество содержит понятия, которые выходят за рамки исследуемой предметной области.

3. Полученные на втором этапе концепты приводятся в нормальную форму (с помощью алгоритма стемминга выделяются словарные основы концептов).

4. Удаление тех концептов, которые отсутствуют в словаре проектной организации, сформированного из терминов ТД электронного архива.

5. Идентификация типов отношений свеем к использованию логических правил, выявляющие отношения между понятиями. Экспертом подготавливается корпус текстов предметной области, которые содержат множество предложений. Каждое такое предложение представляет собой некоторую ситуацию, в котором оказались концепты, и которые свойственны тому или иному отношению и имеют предикатное представление. В основе отношений лежат трехместные предикаты [Найханова, 2008]. Для отношения «обобщение» используется предикат $PHier(a, x, y)$, которое описывает отношение $pod \leftrightarrow вид$. Отношению «целое-часть» соответствует предикат $Pwp(a, x, y)$. Определим отношение «ассоциация» между концептами, если между ними не существуют отношения «целое-часть» и «обобщение». Для распознавания вида отношения между концептами в текстах предметной области, необходимо дополнительная информация об отношениях в виде термов-спутников, которые составляют устойчивые

словосочетания с глаголом семантического отношения. Для отношения «обобщение» термами-спутниками являются: «к видам», «родом», «имя», «значения» и другие. Термы-спутники: «целое», «часть», «состав», «элементом» соответствуют отношению «целое-часть». В работе [Маркарова, 1996] подробно рассмотрено различные виды конструкции с предикатами, выражающие отношения «Целое-Часть». Используя знания об отношениях, построим логические правила по их идентификации. Формально логическое правило выглядит следующим образом:

$$R = (S = \{P, \{C^w\}, F^C, \{G\}\}),$$

где S – ситуация, которая описывается в предложении; $P = \{p\}$ – подмножество предложений из корпусов текста предметной области; C^w – концепты, извлеченные из wiki-ресурса; F^C – множество конструкций предикатов и терм-спутников; G – подмножество, описывающее результат исполнения логических правил.

4. Формальное представление концептуального индекса электронного архива

Пусть $C = \{c_i, i \in I, I = \{1, 2, 3, \dots, n\}\}$ – конечное множество понятий предметной области, зафиксированных в онтологии; $D = \{\tilde{d}_j, j \in J = \{1, 2, 3, \dots, m\}\}$ – семейство нечетких подмножеств в C [Берштейн Л.С. и др., 2005]. Пара $\tilde{CI} = (C, D)$ называется нечетким неориентированным гиперграфом, если $\tilde{d}_j \neq \emptyset$, $j \in J$ и $\bigcup_{j \in J} \tilde{d}_j = C$. При этом $c_1, c_2, \dots, c_n \in C$ являются вершинами гиперграфа, а множество D , состоящее из $\tilde{d}_1, \tilde{d}_2, \dots, \tilde{d}_m$ – множество нечетких ребер гиперграфа.

Принимая во внимание, что отдельно взятый ТД имеет онтологическое представление как результат концептуального индексирования, множество $D = \{\tilde{d}_j\}$ будем понимать как множество ТД в концептуальном индексе, а \tilde{d}_j – отдельно взятое онтологическое представление j -го ТД. Получаем, что нечеткий неориентированный граф $\tilde{CI} = (C, D)$ – концептуальный индекс электронного архива.

Два понятия концептуального индекса c_α и c_β (вершины гиперграфа) называются нечетко смежными, если существует документ (нечеткое ребро гиперграфа), которое включает оба понятия, причем величина

$$\mu(c_\alpha, c_\beta) = \bigvee_{d_i \in D} \mu_j(c_\alpha, c_\beta), \text{ где}$$

$$\mu_j(c_\alpha, c_\beta) = \mu_{d_j}(c_\alpha) \& \mu_{d_j}(c_\beta)$$

называется степенью смежности понятий c_α и c_β . Величина $1 - \mu(c_\alpha, c_\beta)$ представляет собой расстояние между понятиями c_α и c_β , основываясь фактически на содержании архива.

Данный показатель может найти применение при уточнении запроса пользователя к архиву ТД в том случае, когда известно доминирующее понятие в запросе, но результат оставляет желать лучшего. Для уточнения запроса привлекается терминологическое окружение того понятия онтологии, которое находится на меньшем расстоянии от исходного.

Два документа \tilde{d}_γ и \tilde{d}_δ называются нечетко смежными, если $\tilde{d}_\gamma \cap \tilde{d}_\delta \neq \emptyset$, причем величина

$$\mu(\tilde{d}_\gamma, \tilde{d}_\delta) = \bigvee_{c \in (\tilde{d}_\gamma \cap \tilde{d}_\delta)} \mu_{\tilde{d}_\gamma \cap \tilde{d}_\delta}(c)$$

называется степенью смежности документов \tilde{d}_γ и \tilde{d}_δ . Величина $1 - \mu(\tilde{d}_\gamma, \tilde{d}_\delta)$ описывает расстояние между документами в архиве, основываясь на содержании документов и онтологии предметной области. Данный показатель может применяться в задаче нечеткой кластеризации содержимого электронного архива, т. е. там, где важную роль в целевой функции играет расстояние между центром кластера (в качестве которого может выступать гипотетический документ) и анализируемыми документами.

5. Онтологический поиск документов

Поисковый запрос, сформулированный в терминах на естественном языке $Q = \{t_i\}$, преобразуется планировщиком к концептуальному виду, но прежде выполняется ряд преобразований:

- удаление стоп-слов,
- стемминг (выделение основы слова, получение термов).

С полученным запросом $Q = \{t_i\}$ возможны два варианта обработки:

- 1) $Q = C$ – поисковый запрос совпадает с названием концепта онтологии предметной области.
- 2) $Q = W$ – поисковый запрос совпадает с терминами онтологии предметной области.

В первом случае расширение поискового запроса достигается с использованием функции интерпретации концептов, т.е. запрос дополняется терминами, формирующими терминологическое окружение концепта:

$$Q = Q \cup F_{cw}^D(c_i).$$

Во втором случае применяется функция, сопоставляющая набор терминов подмножеству понятий предметной области. Таким образом, получим множество концептов, в терминологическом окружении которых присутствуют термины запроса. К полученному множеству концептов применим функции интерпретации концептов, дополняя запрос терминами, семантически связанных с концептами:

$$Q = Q \cup F_{CW}^D(F_{WC}^D(w_i)).$$

Таким образом, расширение запроса сводится к определению прямой и обратной функции интерпретации. Полученный после обработки расширенный запрос преобразуется к концептуальному виду с помощью следующего выражения:

$$\mu_{ij} = 1 - \frac{l}{l_k} \sum_{s=1}^{l_k} |f_s^k - f_s|,$$

где f_s, f_s^k — частоты встречаемости s -го термина в запросе и в описании k -го понятия онтологии проекта, соответственно; l_k — мощность текстового входа понятия c_k . В том случае, если термин s отсутствует в запросе, f_s принимается равным нулю; μ_{ij} — величина, характеризующая степень выраженности концепта [Филиппов и др., 2013] в запросе, где $0 \leq \mu_{ij} \leq 1$.

В традиционных моделях информационного поиска нахождение степени релевантности документа обозначается величиной, которая называется мерой сходства запроса к документу. В данной работе информационный запрос и ТД имеют концептуальные представления, т.е. рассматриваются в виде нечеткого множества. Для вычисления меры сходства между запросом и документом воспользуемся термином «степень включения», которое соответствует операций нечеткого включения множества [Берштейн и др., 2005]. Мера включения запроса и ТД вычисляется по следующей формуле:

$$\overline{\gamma(I_q, I_d)} = \&(\mu_{I_q}(c) \rightarrow \mu_{I_d}(c)),$$

где $\overline{I_q, I_d}$ — концептуальные индексы запроса и ТД соответственно, c — концепты онтологии, где $0 \leq \overline{\gamma(I_q, I_d)} \leq 1$.

6. Вычислительные эксперименты.

В ходе научно-исследовательской работы были разработаны следующие интеллектуальные подсистемы: подсистема автоматизированного формирования онтологической сети с использованием Wiki-ресурсов и подсистема онтологического поиска ТД, основанная на

использование информационных потребностях пользователя. С данными подсистемами были проведены вычислительные эксперименты, которые показали следующие результаты. В подсистему автоматизированного формирования онтологической сети был загружен концепт «СУБД». Данная подсистема сформировала концептуальную сеть (рисунок 2).

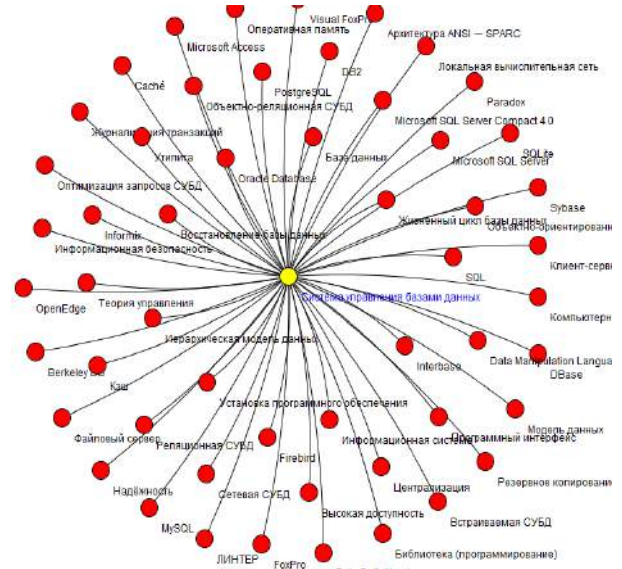


Рисунок 2. Сформированная концептуальная сеть

Были проведены вычислительные эксперименты с подсистемой онтологического поиска. Для оценки качества информационного поиска ТД использовались величина F -мера. Данная величина учитывает в себе два параметра: точность и полнота. F -мера представляет собой среднее гармоническое взвешенное:

$$F = \frac{1}{\alpha \frac{1}{P} + (1-\alpha) \frac{1}{R}} = \frac{(\beta^2 + 1)PR}{\beta^2 P + R}$$

где $\beta^2 = \frac{1-\alpha}{\alpha}, \alpha \in [0,1],$ т.е. $\beta^2 \in [0, \infty]$.

По умолчанию сбалансированная F -мера присваивает точности и полноте одинаковые веса, т.е. $\alpha = 1/2$, или $\beta = 1$. Если $\beta < 1$ предпочтение отдает точности поиска, при $\beta > 1$ полноте поиска. При $\beta = 1$ формула принимает вид:

$$F_\beta = \frac{2PR}{P + R}$$

Полученные результаты сравнивались с результатами следующих систем информационного поиска: Яндекс Персональный поиск (ЯПП), Архивариус 3000 (A300), AOL Desktop Search (AOL), Copernic Desktop Search (CDS). Сравнительные результаты эксперимента представлены на рисунке 3.

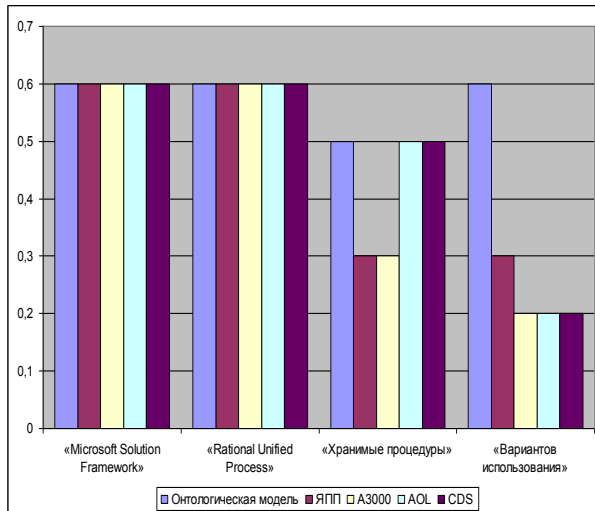


Рисунок 3. Результаты эксперимента

Как видно из рисунка, лучшее качество поиска онтологическая модель показывает в том случае, если в запросе присутствует высокая неопределенность в определении термина к конкретной тематике.

ЗАКЛЮЧЕНИЕ

В данной работе предлагается использовать интеллектуальные агенты в процессе поиска ТД. Интеллектуальный агент пополняет пользовательский профиль, обрабатывает поисковые запросы и выводит список ранжированных документов. В процессе научного исследования были разработаны подсистемы онтологического поиска и подсистема автоматизированного формирования онтология, с которыми были проведены вычислительные эксперименты.

Данная работа выполнена при частичной финансовой поддержки РФФИ, проект №14-01-31086.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- [Берштейн и др., 2005.] Берштейн Л.С., Боженюк А.В. Нечеткие графы и гиперграфы. М.: Научный мир, 2005 – 256 с.
- [Гаврилова и др., 2000] Гаврилова Т.А., Хорошевский В.Ф., Базы знаний интеллектуальных систем. –СПб. : Питер, 2000. – 384 с.
- [Добров и др., 2006] Добров Б.В., Лукашевич Н.В., Лингвистическая онтология по естественным наукам и технологиям: основные принципы разработки и текущее состояние // Десятая национальная конференция по искусственному интеллекту с международным участием (Обнинск, 25-28 сентября 2006 г.) – М.: Физматлит, 2006.
- [Маркарова, 1996] Маркарова Т.С., Конструкции с предикатами, выражающими отношения "Часть-целое" в современном русском языке. : автореф. дис. на соиск. учен. степ. канд.филол. наук (10.02.01);МГУ – М., 1996 - 23 с.
- [Найханова, 2008] Найханова Л.В., Технология создания методов автоматического построения онтологий с применением генетического и автоматного программирования. – Улан-Удэ.:БНЦ СО РАН, 2008. – 237 с.
- [Рассел и др., 2006] Рассел С., Норвиг П., Искусственный интеллект. Современный подход. – М.:Вильямс, 2006. – 1408 с.
- [Филиппов др., 2013] Филиппов А.А., Наместников А.М., Субхангулов Р.А. Применение нечетких моделей в задачах

кластеризации и информационного поиска текстовых проектных документов // Интегрированные модели и мягкие вычисления в искусственном интеллекте. Сборник научных трудов VII-й Международной научно-практической конференции (Коломна, 22-22 мая 2013 г.) В 3-х томах. Т.3. – М.:Физматлит, 2013. С. 1278-1289.

ONTOLOGICAL APPROACH TO THE FORMATION OF PROJECT REQUESTS OF INTELLIGENT AGENT

Namestnikov A.M. *, Subkhangulov R.A. *

* *Ulyanovsk State Technical University, Ulyanovsk, Russia*

nam@ulstu.ru

subkhangulov-ruslan@yandex.ru

INTRODUCTION

In this paper is considered the use of intelligent agents in the formation of information requests to the electronic archive of technical documents. Intelligent agent contains an ontological description of the profile of the designer, which is expressed as a fragment of domain ontology. Tasks intellectual agent are addition information on user needs and finding text documents requested by the user of an electronic archive.

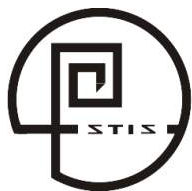
MAIN PART

In this article we propose to use intelligent agents in process of search for technical documents. Intelligent agent in process of search using user profiles and ontological model of search. The user profile contains a fragment of domain ontology. Thus, the search engine has information about preferences of user. This information helps to improve the quality of the search of documents

In the process of scientific research have been developed subsystem of ontological search and subsystem of automated formation of ontology.

CONCLUSION

Computational experiments were performed with the subsystem ontological search. Experiments showed that the ontological model search shows better results than the traditional model



OSTIS-2015

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822:514

ОПЫТ СОЗДАНИЯ СРЕДСТВ СЕМАНТИЧЕСКОГО МОДЕЛИРОВАНИЯ И ПРОЕКТИРОВАНИЯ НА МАССОВОЙ ПРОГРАММНОЙ ПЛАТФОРМЕ

Смирнов С.В.

*Институт проблем управления сложными системами Российской академии наук,
г. Самара, Россия*

smirnov@iccs.ru

В статье рассказывается об опыте разработки методов и компьютерных средств, поддерживающих семантическое моделирование и проектирование на платформе MS Excel. Описываются наиболее характерные технологии и компоненты созданных инструментов. Рассматривается открытая архитектура инструментальной системы и унифицированная архитектура конструируемых с ее помощью систем, основанных на знаниях.

Ключевые слова: семантическая технология; объектно-ориентированный стиль представления знаний; открытая архитектура; массовая платформа.

Введение

Точка зрения, что методы и средства построения программных продуктов (равно как сами такие продукты) опираются на *представление и обработку знаний*, получает все большее распространение. Экспликация «знаниевой» роли привычных программно-технических понятий позволило найти новые, не использованные ранее методы программирования, которые оказались чрезвычайно полезными при решении сложных задач в различных прикладных сферах приложения компьютеров. Эпистемологическое значение программно-технических единиц обнаруживает себя в различных формах. Это и знание последовательности шагов, необходимых для решения задачи, в обычном *императивном* программировании (которое ныне, как правило, *объектно-ориентированное*, т.е. прямо опирается на объектно-фреймовое представление знаний [Dilger, 1989]). Это и различные техники *декларативного* (логического, базирующегося на ограничениях и т.п.) программирования, в основе которых лежит отделение описания задачи от интерпретации этого описания. Словом, сегодня можно с достаточной уверенностью констатировать, что (по крайней мере) идея «семантизации» программной инженерии *«овладевает массами»*.

Семантические технологии проникают в арсеналы привычных практик и инструментов создания информационных систем (см., например, [Копайгородский, 2014], [Смирнов, 2013], вообще, тематику OSTIS-конференции). Поэтому для

удовлетворения «повышенного спроса» таких технологий в растущем круге пользователей вполне мотивирована разработка средств семантического моделирования и проектирования на массовых программных платформах, к которым, несомненно, принадлежит MS Excel.

Достоинства MS Excel общеизвестны. Это *широчайшее распространение*, простой, но достаточно мощный *встроенный язык программирования* - Visual Basic for Application (VBA) с развитой интегрированной средой разработки приложений, включающей средства визуального проектирования пользовательских интерфейсов и инструменты автоматизации программирования, - *единство* среды разработки пользовательских приложений и хранения данных, возможность использования стандартных для MS интерфейсов *межзадачных коммуникаций* для общения с другими приложениями и т.д.

Эти и другие резоны определили, что развиваемый в течение ряда лет в Институте проблем управления сложными системами РАН инструментарий, реализующий семантическое моделирование и проектирование – общецелевая система объектно-ориентированного моделирования *gB* (от англ. «*good Base*», являющегося в свою очередь акронимом от «*graceful object-oriented data Base*») – с некоторого времени обновляется как версия Excel-приложения. И уже в этой реинкарнации *gB*-система использована при разработке ряда экспериментальных и промышленных программных комплексов (см., например, [Виттих, 2009]).

1. Задачи семантического моделирования и проектирования

Задачи семантических технологических «переделов» в инструментальных системах в целом хорошо известны.

Согласно широко разделяемой точки зрения разработка прикладных интеллектуальных систем и приложений начинается с *онтологического анализа* для моделирования смысла предметной области (ПрО) [Смирнов, 2001]. Непосредственная компьютерная поддержка этой задачи инженерии знаний осуществляется либо путем предоставления эксперту ПрО инструментов «приобретения» [Гаврилова, 2008], или *автоформализации*, знаний (т.е. соответствующих языков, интегрированных сред и т.п.), либо когнитивным *онтологическим анализом данных* о ПрО [Загоруйко, 2013].

Так или иначе, пользователь инструментальной системы в результате онтологического анализа ПрО фиксирует ее понимание в оригинальной онтологии. Кроме такой *«целевой»* онтологии пользователь-разработчик неизбежно привлечет в конструкцию создаваемой прикладной системы (и, следовательно, должен получить возможность прежде построить) концептуальные описания (онтологии) *других* предметных областей (ПрО), сформировавшиеся независимо от актуальной ПрО и имеющие в контексте последней *методо-ориентированный*, инструментальный характер.

Формирование онтологий ПрО в объектно-фреймовом стиле и их визуализация в форме *семантических сетей* позволит разработчику увидеть и опубликовать (тем самым документируя) своё понимание на легко воспринимаемом и интерпретируемом языке, получить *шаблоны* для дальнейшего конструирования прикладной системы и др.

Онтология как база знаний регламентирует в последующем описание конкретных ситуаций в актуальной ПрО. Для реализации этой задачи пользователю инструментальной системы должны быть предоставлены средства генерации *объектно-ориентированных баз данных* для хранения онтологических *денотативных объектных моделей* ПрО как систем взаимосвязанных информационных моделей объектов, удовлетворяющих зафиксированным в онтологии ПрО аксиомам.

При этом любая онтология ПрО в силу необходимой технологической реакции на тезис Куайна об *онтологической относительности* должны быть в инструментальной системе денотативной моделью некоторой семантически замкнутой онтологии - *метаонтологии*, - которая будет денотатом самой себя [Смирнов, 2000].

Наконец, для моделирования и проектирования приложений следует предусмотреть возможность разработки специальных *трансформационных денотативных объектных моделей* – денотатов «технологической» ПрО, которая описывается

специальной онтологией *решателей задач* [Смирнов, 2008].

Таким образом, онтологические модели очерченной семантической технологии разработки интеллектуальных систем и приложений структурно *гомогенны*. Вместе с тем, комплектность программной реализации онтологических моделей будет отличаться в части их процедурных составляющих – методов и демонов:

- онтологии наряду с декларативной составляющей в виде информационных моделей объектов ПрО, описываемой метаонтологией, в общем случае включают программную реализацию свойств-методов и свойств-демонов;
- денотаты любой онтологии за исключением метаонтологии и онтологии решателей задач не имеют процедурных составляющих;
- каждая денотативная объектная модель решателя задач - приложения - включает демоны открытия и закрытия модели (в программно-техническом смысле), которые играют роль главной программы приложения.

Далее можно разделить онтологии на *служебные*, к которым следует отнести метаонтологию и онтологию решателей задач, и *пользовательские* – онтологии целевых и методо-ориентированных ПрО.

Приложения также можно разделить на *общезначимые*, призванные автоматизировать работу с онтологическими моделями (построение, редактирование, обозрение и т.п.), и *целевые*.

Это родовидовое разделение онтологических моделей иллюстрирует рисунок 1.

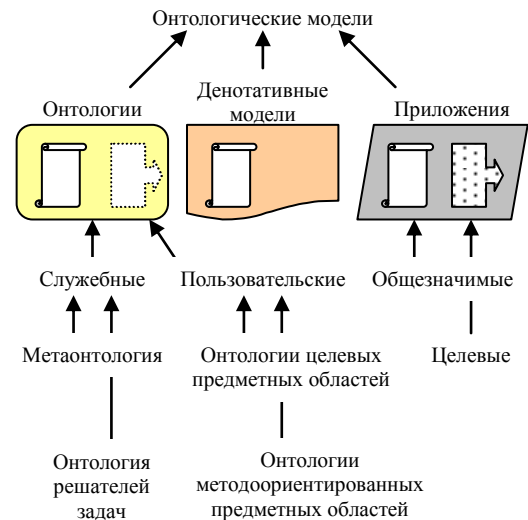


Рисунок 1 – Родовидовой состав онтологических моделей

2. Архитектура инструментальной системы и продуцируемых ею систем

На рисунке 2 представлена архитектура *gB*-системы и, вообще говоря, любой продуцируемой ею системы, основанной на знаниях, т.к. сама *gB* система разрабатывалась методом бутстрепинга и,

таким образом, является примером систем, создаваемых с ее помощью.

Базой создания и функционирования систем выступают метаонтология и отология решателей задач: на основе первой строятся все онтологии, а на основе второй – все приложения.

Общезначимые приложения могут импортировать и использовать все создаваемые интеллектуальные системы, а специализированные целевые приложения, вообще говоря, оригинальны для каждой такой системы.

Методо-ориентированные онтологии могут интегрироваться различными продуцируемыми системами, но соответствующие этим онтологиям денотативные объектные модели будут, скорее всего, оригинальны в каждой из этих систем.

Онтологии и денотативные объектные модели ПрО очевидно образуют самобытное ядро каждой создаваемой интеллектуальной системы.

Таким образом, архитектура gB-системы и продуцируемых ею систем представляет собой *открытую мультимодельную среду*, в которой обеспечена *однородность* всех без исключения

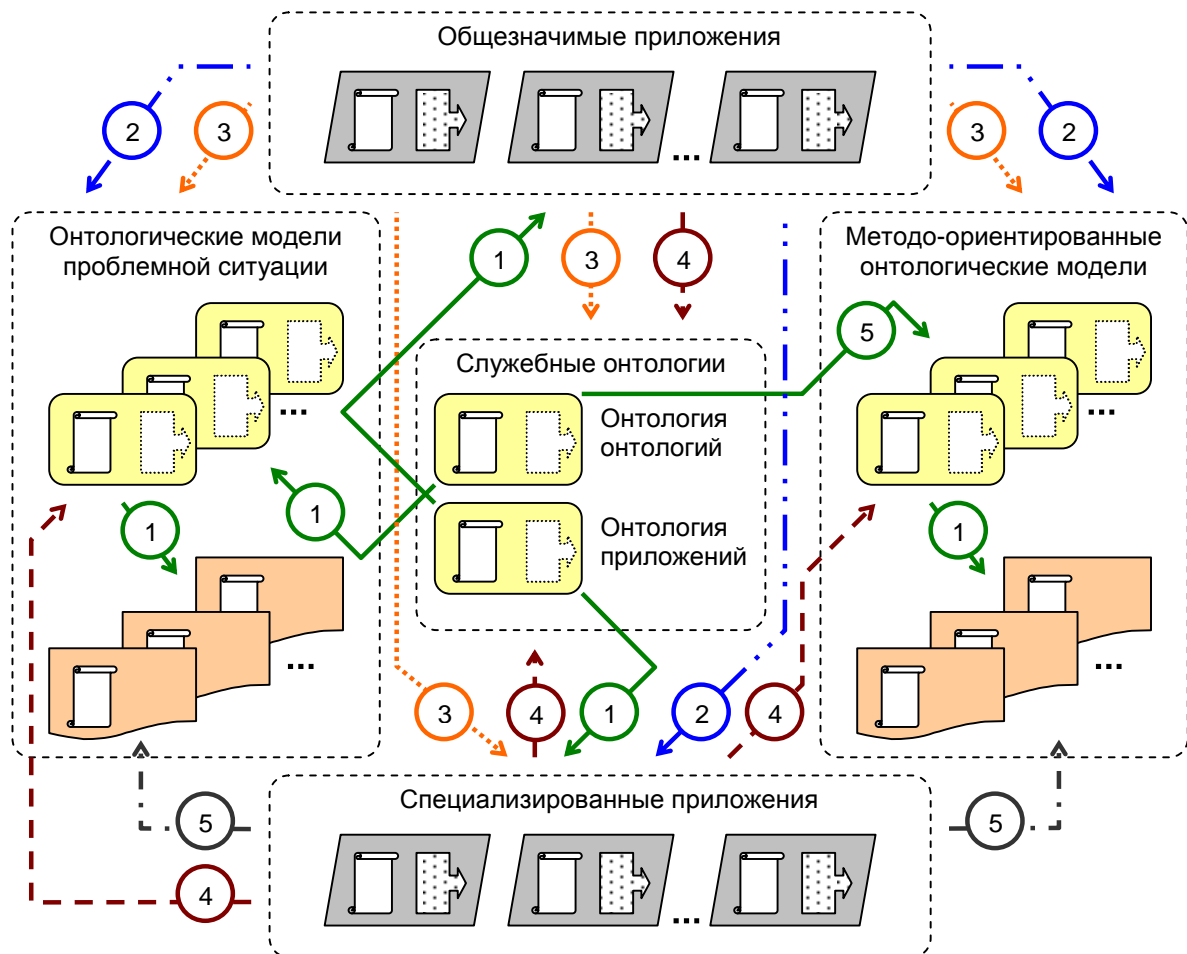
программно-технических единиц, создаваемых в процессе разработки и эксплуатации программных комплексов.

3. Реализация

Каждая программно-техническая единица gB-системы представляет собой стандартную Excel-книгу. Каждая родовидовая разновидность онтологических моделей имеет свой Excel-шаблон.

Ядро gB-системы образует объектно-ориентированная (с возможностью указания *активных* свойств объектов) СУБД на таблично-реляционном остоле Excel. Программный интерфейс СУБД является составной частью онтологии решателей задач и представляет собой совокупность методов объектов класса «Денотативная модель».

Книга-онтология или книга-приложение получает доступ к процедурным составляющим другой книги-онтологии посредством VBA-ссылок, которые генерируются автоматически при работе специальных общезначимых приложений. Аналогично генерируются каркасы свойств-методов и демонов.



Типы связей между моделями: 1 – служит основой построения; 2 - реализует создание и редактирование; 3 – осуществляет визуальный анализ; 4 - использует как базы знаний, активируя и их процедурные составляющие; 5 - использует как базы данных.

Рисунок 2 – Архитектура инструментальной системы и продуцируемых ею систем

4. Общезначимые приложения системы

К наиболее характерным общезначимым приложениям *gB*-системы системы сегодня можно отнести Обзоратель объектных моделей, Редактор онтологий, работающий как в режиме экспертного конструирования, так и онтологического анализа данных, а также Редакторы денотативных моделей и приложений.

Для иллюстрации потенциала *gB*-системы на рисунке 3 приведен скриншот Обзорателя моделей, демонстрирующего фрагмент онтологии решателей задач.

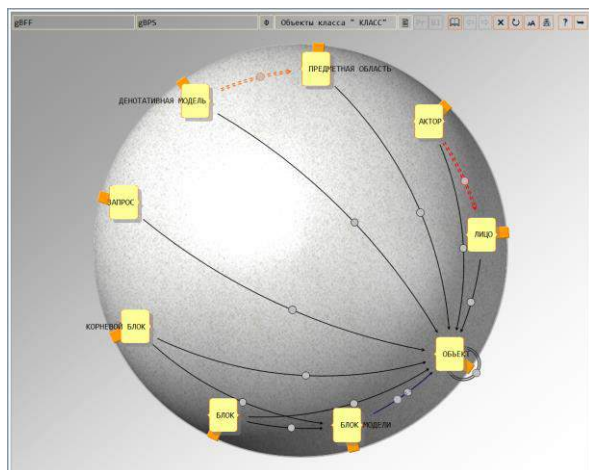


Рисунок 3 – Скриншот Обзорателя моделей

Формируемое отображение моделей удовлетворяет положениям концепции «фокус-контекст», когда с максимальной подробностью предоставляется фокус обзора, а для остальной части данных дается упрощенное замещающее изображение.

Обзоратель применяет как центрально-радиальный метод отображения *фокус-объекта* и непосредственно связанных с ним объектов, и кругового метода для обозрения произвольно отбираемой группы объектов (см. рисунок 3). И в том и другом случаях важнейшую когнитивную нагрузку несет составной образ невидимой части обозреваемой модели, формируемой 3D-фигурой «Инфоглобус», которая организует пространство обозрения модели в целом, и коннекторами – псевдосвязями присутствующих на экране объектов с невидимыми объектами модели.

Смена фокуса обозрения модели производится по указанию пользователя путем анимированного перемещения объектов по видимой поверхности Инфоглобуса с «восходом» актуализируемых объектов из-за линии «горизонта» и «заходом» неактуальных объектов за линию «горизонта» Инфоглобуса.

Заключение

В работе проанализирована проблематика создания «бюджетных» средства семантического

моделирования и проектирования на платформе MS Excel:

- выявлены типичные задачи семантического моделирования и проектирования при представлении знаний в наиболее усвоенном пользователями объектно-фреймовом стиле;
- дана классификация необходимых для реализации системы онтологических моделей;
- показана возможность реализации системы на основе открытой многомодельной гомогенной архитектуры.

Библиографический список

- [Виттих, 2009] Виттих, В.А. Онтологический подход к построению информационно-логических моделей в процессах управления социальными системами / В.А. Виттих, П.В. Ситников, С.В. Смирнов // Вестник компьютерных и информационных технологий. – 2009. - №5. - С. 45-53.
- [Гаврилова, 2008] Гаврилова, Т.А. Интеллектуальные технологии в менеджменте: инструменты и системы / Т.А. Гаврилова, Д.И. Муромцев. – СПб.: Изд-во «Высшая школа менеджмента»; Изд. дом СПбГУ, 2008. – 488 с.
- [Загоруйко, 2013] Загоруйко, Н.Г. Когнитивный анализ данных / Н.Г. Загоруйко. – Новосибирск: Академическое изд-во «Гео», 2013. – 186 с.
- [Копайгородский, 2014] Копайгородский, А.Н. Применение онтологий в семантических информационных системах // Онтология проектирования. – 2014. - №4. - С. 78-89.
- [Смирнов, 2000] Смирнов, С.В. Онтологическая относительность и технология компьютерного моделирования сложных систем / С.В. Смирнов // Известия Самарского научного центра РАН. 2000. Т. 2. № 1. С. 66-71.
- [Смирнов, 2001] Смирнов, С.В. Онтологический анализ предметных областей моделирования / С.В. Смирнов // Известия Самарского научного центра РАН. - 2001. - Т. 3. - №1. - С. 62-70.
- [Смирнов, 2008] Смирнов, С.В. Прагматика онтологий: объектно-ориентированная модель знаний о предметной области / С.В. Смирнов // 11-я Нац. конф. по искусственному интеллекту с международным участием КИИ-2008 (28 сентября-03 октября 2008 г., Дубна, Россия): Труды конф. Т. 3. – М.: ЛЕНАНД, 2008. - С. 208-216.
- [Смирнов, 2013] Смирнов, С.В. Пакеты программ как формальные онтологии: построение и использование // Открытые семантические технологии проектирования интеллектуальных систем OSTIS-2013: Материалы III международной науч.-тех. конф. (21-23 февраля 2013 г. Минск, Белоруссия) – Мн.: БГУИР, 2013. – С. 191-194.
- [Dilger, 1989] Dilger, W. Object-oriented Knowledge Representation – an Overview / W. Dilger // J. New Generation Computation Systems. - 1989. - Vol. 2. No 4. – P. 339-363.

THE EXPERIENCE OF SEMANTIC MODELING AND DESIGNING TOOLS ON WIDELY USED PLATFORM

Smirnov S.V.

*Institute for the Control of Complex System
of the Russian Academy of Sciences,
Samara, Russia
smirnov@iccs.ru*

The paper tells about the experience of creation methods and computer tools supporting semantic modeling and designing on MS Excel. The most distinctive components of these tools are described. The instrumental system open architecture and designing alienable knowledge-based systems unified architecture is considered.



OSTIS-2015

(Open Semantic Technologies for Intelligent Systems)

УДК 004.522

РЕАЛИЗАЦИЯ УДАЛЕННОГО УПРАВЛЕНИЯ ПАРАМЕТРИЗОВАННОЙ ТРЕХМЕРНОЙ МОДЕЛЬЮ САМОЛЕТА С ПОМОЩЬЮ КЛИЕНТ-СЕРВЕРНОГО ПРИЛОЖЕНИЯ

Боргест Н.М., Власов С.А., Коровин М.Д.

*Самарский государственный аэрокосмический университет имени академика С.П. Королева
(национальный исследовательский университет), г. Самара, Россия*

borgest@yandex.ru

maks.korovin@gmail.com

savlasoglavv@mail.ru

В работе рассмотрены основные принципы построения и создание приложения для удалённого доступа к системе автоматизированного проектирования сложных технических объектов. Целью подобных приложений является снижение временных затрат на доступ к автоматизируемым процедурам, упрощение ввода необходимой проектной информации и реализация возможностей использования удалённых программных и информационных ресурсов. В работе в качестве предметной области рассматривается «самолётостроение» и разрабатываемый для неё интеллектуальный помощник проектанта.

Ключевые слова: приложение, клиент, клиент-сервер, проектант, интерфейс.

ВВЕДЕНИЕ

Современные темпы развития пользовательских приложений приводят к усложнению программных интерфейсов и повышению требований к их функциональным возможностям. Наиболее ярко эта тенденция проявляется в «тяжелых» CAD-системах, на освоение которых требуются значительные временные затраты. Интерфейс программы играет немаловажную роль в работе с приложением. Повышение семантической прозрачности интерфейса значительно упрощает освоение программы, снижает время, необходимое на обучение персонала и повышает качество работы.

1. Клиент-серверное управление CAD системами

Современные CAD-системы обладают богатым функционалом и в состоянии обеспечивать автоматизированное решение широкого спектра задач. Однако, стоит заметить, что стоимость подобных систем крайне высока. В особенности это актуально для малых предприятий, обладающих относительно скромным бюджетом. Зачастую на предприятие покупается небольшое число лицензий на промышленные CAD системы, которые потом

находятся в коллективном пользовании. Это создает определенные трудности при эксплуатации. При этом стоит отметить, что часто выполняемые однотипные процедуры могут быть формализованы стандартными средствами системы, после чего возможна реализация отдельного специализированного интерфейса, в том числе, доступного удаленно. Реализация удаленного доступа к CAD системе позволяет значительно снизить требования к техническим характеристикам клиентской машины, так как все ресурсоемкие вычисления выполняются на стороне сервера.

В качестве примера применения технологии удаленного доступа рассмотрено удаленное управление параметризованной моделью тяжелого транспортного самолета, созданной в CAD CATIA, при помощи мобильного устройства на базе ОС Android. Удаленно варьируя проектные параметры самолета, пользователь получает возможность осуществлять перестроение трехмерной геометрической модели и связанной с ней CFD модели для расчета аэродинамических характеристик самолета без необходимости непосредственного взаимодействия с CAD и CFD системами.

2. Применяемый инструментарий

Управление процессом предварительного проектирования самолёта в системе САПА происходит с планшета на базе ОС Android.

В качестве платформы для разработки клиентской части приложения был выбран Corona – быстрый и простой инструмент разработки Iphone, Ipad и Android приложений. Corona SDK – это первый продукт в семействе продукции Corona от Anscа, предназначенный для разработки высокопроизводительных мультимедийных и графически оборудованных приложений. Anscа – это компания-разработчик Corona, а данный инструмент разработки позволяет разработчикам создавать быстрые и мощные кросс-платформенные приложения, у которых есть доступ к API, к которым у других платформ доступа нет, вроде API для управления камерой, GPS и акселерометром.

Преимущества Corona SDK:

- возможность разработки приложений специально под нужную систему;
- автоматическая интеграция OpenGL-ES;
- Corona использует язык программирования Lua [Fernandez, 2012].

Выбор операционной системы Android обусловлен открытостью системы и широкой номенклатурой доступных на рынке устройств. Приложение Corona Simulator позволяет сгенерировать исполняемые файлы системы Android (.apk файлы), который можно скопировать на любой планшет или телефон на базе ОС Android и там его установить.

Данные передаются на сервер с помощью post-запроса по протоколу http. В запросе передаётся json, в нем пары ключ – значение вида id:value, где id – идентификатор поля, описанный в файле settings.json, value – его числовое или логическое значение.

В settings.json находятся настройки, связанные с адресом сервера и полями для данных. Чтобы добавить еще поле, нужно отредактировать массив fields, чтобы изменить адрес сервера, нужно поменять значение у url, чтобы изменения вступили в силу необходимо скомпилировать проект.

Экран клиентского приложения на начальном этапе проработки состоит из массива названий полей и полей ввода значения (рисунок 1).

Приложение работает с любыми планшетами и телефонами с операционной системой Android начиная с версии 4.0.

Серверная часть приложения получает запрос от клиента, обрабатывает его, связывается с базой данных и отправляет обратно ответ клиенту.

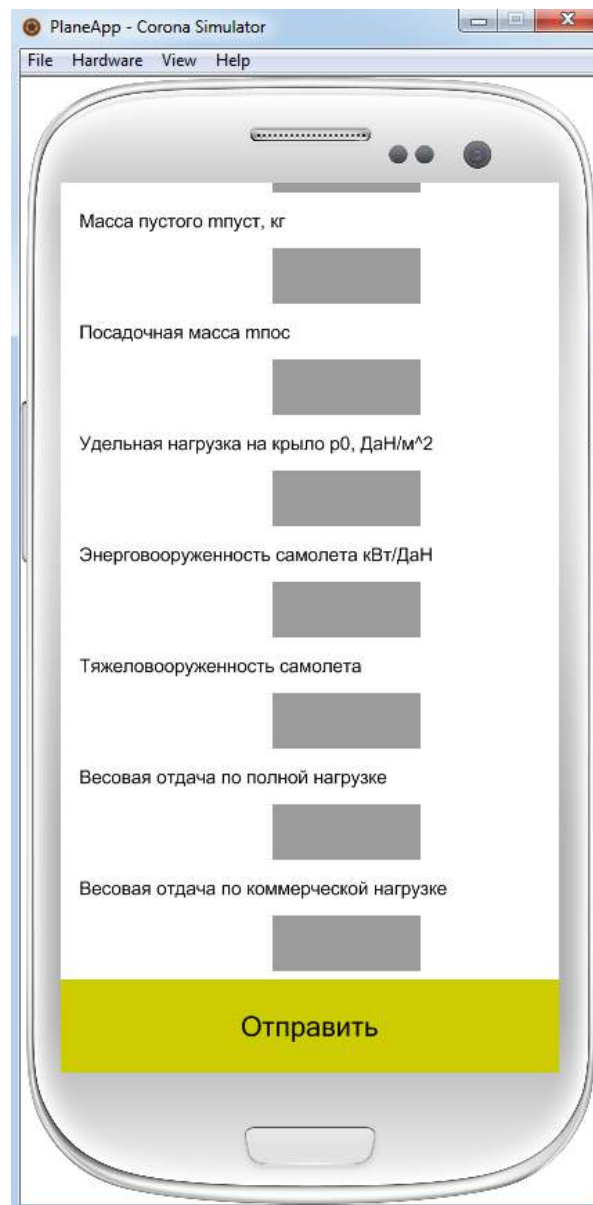


Рисунок 1 – Экран клиентского приложения в Corona SDK

В качестве базы данных была использована база проектных параметров тяжелого транспортного самолёта, интегрированная в программно-аппаратный комплекс автоматизации предварительного проектирования самолета «Робот-проектант» [Боргест, 2012]. База данных выполнена в системе MS Excel и связана с параметризованной моделью. Сохранение базы данных запускает регенерацию модели по актуализированным данным, включая выбор и расчет проектных параметров, не определённых пользователем на этапе ввода данных.

Пример схемы клиент-серверного приложения представлен на рисунке 2.

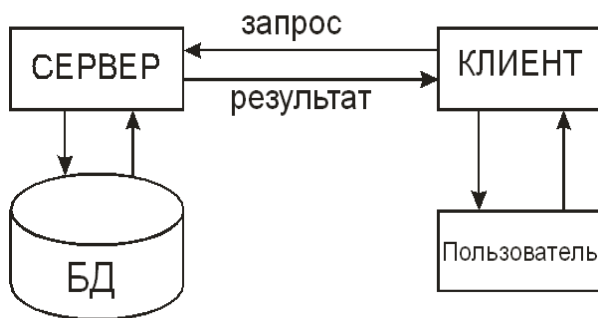


Рисунок 2 – Схема клиент-сервера

Для написания серверной части приложения был выбран язык С# — объектно-ориентированный язык программирования, разработанный в 1998—2001 годах группой инженеров под руководством Андерса Хейлсберга в компании Microsoft как язык разработки приложений для платформы Microsoft .NET Framework и впоследствии был стандартизирован как ECMA-334 и ISO/IEC 23270. С# относится к семье языков с С-подобным синтаксисом, из них его синтаксис наиболее близок к С++ и Java. Язык имеет статическую типизацию, поддерживает полиморфизм, перегрузку операторов (в том числе операторов явного и неявного приведения типа), делегаты, атрибуты, события, свойства, обобщённые типы и методы, итераторы, анонимные функции с поддержкой замыканий, LINQ, исключения, комментарии в формате XML [Рихтер, 2013]. Этот язык программирования, благодаря тому, что он разработан в компании Microsoft, хорошо подходит для приложений, взаимодействующих с Microsoft Excel, что немаловажно для разрабатываемой системы, так как база данных проектных параметров хранится в виде таблиц excel.

3. Применение удалённого управления САПР

В большинстве автоматизированных информационных систем применяют СУБД, поддерживающие реляционные модели данных.

Среди общих требований к СУБД можно выделить:

- обеспечение целостности данных (их полноты и достоверности);
- защита данных от несанкционированного доступа и от искажений из-за сбоев аппаратуры; удобство пользовательского интерфейса;
- в большинстве случаев важна возможность распределенной обработки в сетях ЭВМ.

Первые два требования обеспечиваются ограничением прав доступа, запрещением одновременного использования одних и тех же обрабатываемых данных (при возможности их модификации), введением контрольных точек (checkpoints) для защиты от сбоев и т.п.

Банк данных в САПР является важной обслуживающей подсистемой, он выполняет

функции информационного обеспечения и имеет ряд особенностей. В нем хранятся как редко изменяемые данные (архивы, справочные данные, типовые проектные решения), так и сведения о текущем состоянии различных версий выполняемых проектов. Как правило, банк данных работает в многопользовательском режиме, с его помощью осуществляется информационный интерфейс (взаимодействие) различных подсистем САПР.

Этот подход был реализован в системе автоматизации предварительного проектирования самолета «Робот-проектант» [Боргест, 2012]. С одной стороны, проектирование сложных технических систем – сфера деятельности, которая всегда требовала творческих решений, с другой – она достаточно хорошо формализована и алгоритмизирована.

Результатом работы робота-проектанта является модель изделия. Она состоит из 2 взаимосвязанных частей – матрицы проекта с логикой расчета (в текущем варианте выполненной на основе таблиц MS Excel) и параметризированной трехмерной модели, которая автоматически меняет собственную конфигурацию в зависимости от данных, содержащихся в матрице проекта.

Предполагается, что робот-проектант может работать в автоматическом режиме, или же в режиме интеллектуального помощника проектанта-человека, при этом степень участия человека в процессе расчета не является постоянной величиной и зависит от желания конкретного пользователя. Иными словами, для каждого оператора предварительно или динамически в процессе работы создается сценарий общения, включающий в себя степень автоматизации расчета, выбор предпочтительных устройств ввода-вывода данных, необходимость выполнения тех или иных этапов расчета. Таким образом, вид интерфейса робота-проектанта зависит от предпочтений конкретного пользователя.

Удалённый доступ к интерфейсу системы реализуется с целью упрощения взаимодействия пользователя с системой проектирования, повышения доступности благодаря возможности доступа с мобильных устройств (планшетных компьютеров и телефонов). На рисунке 3 изображена схема удаленного управления САПР [Кои, 2010].



Рисунок 3 – удалённое управление САПР

Выбор операционной системы Android обусловлен открытостью кода системы, поддержкой интегрированных онлайн-приложений, в том числе сервисов Google, широкой номенклатурой устройств на базе операционной системы, высоким быстродействием таких устройств как телефоны и планшеты.

ЗАКЛЮЧЕНИЕ

Расширение доступа к роботу проектанту, позволит управлять расчетом на большом расстоянии, вводить данные из любой точки, отслеживать этапы проектирования самолета, просмотр готовой 3D модели, а также возможность просмотра модели в демо-режиме.

Работа выполняется при финансовой поддержки Министерства образования и науки Российской Федерации.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

[Боргест и др., 2012] Боргест Н.М., Чернов Р.В., Шустова Д.В. Разработка интерфейса интеллектуального помощника проектанта / Н.М. Боргест, Р.В. Чернов, Д.В. Шустова // материалы международной научно-технической конференции OSTIS-2012. БГУИР. 2012

[Боргест, 2012] Боргест Н.М., Робот-проектант: фантазия и реальность. // Научный журнал «Онтология проектирования» №4(6), 2012. — с. 73-94.

[Орфали, 2000] Орфали Р., Харки Д. Java и CORBA в приложениях клиент-сервер. Двухуровневый лингвистический процессор ответных текстов на естественном языке / Роберт Орфали, Дэн Хаски // Лори, 2000. – с.734.

[Коржов, 1997] Коржов В.А. Многоуровневые системы клиент-сервер/ В.А.Коржов // Москва, 1997. – С. 169.

[Кou, 2010] Kou X.Y., Xue S.K.. Knowledge-guided inference for voice-enabled CAD / X.Y. Kou, S.K.. Xue // Tan Computer-Aided Design Volume. – 2010 - P. 545-557.

[Рихтер, 2013] Рихтер Джеффри,. Программирование на платформе Microsoft.NET Framework 4.5 на языке C#. 4-е издание / Д. Рихтер// Питер. - 2013.

[Fernandez, 2012] Fernandez Michelle M. Corona SDK Mobile Game Development: Beginner's Guide / M.M. Fernandez// Packt Publishing – 2012.

IMPLEMENTATION OF REMOTE CONTROL OVER A PARAMETERIZED THREE-DIMENSIONAL MODEL OF THE AIRPLANE BY A CLIENT-SERVER APPLICATION

Borgest N.M., Vlasov S.A., Korovin M.D.

*Samara State aerospace University
(national research university),
Samara, Russian Federation*

borgest@yandex.ru

maks.korovin@gmail.com

savlasoglavv@mail.ru

The paper discusses the basic principles of creation of applications for remote access to computer-aided design systems for complex technical objects. The purpose of these applications is to reduce the time required for the

access to automated procedures to simplify input of the required project information and to implement opportunities to use remote software and information resources. The domain in which the work is carried out is "aircraft design". The results are implemented into the intelligent assistant designer.

INTRODUCTION

The current pace of development of custom applications lead to more complicated software interfaces and increasing requirements for their functionality. Most clearly, this trend is manifested in the "heavy" CAD-systems for the development of which requires significant time costs. The program interface plays an important role in working with the application. Improving semantic transparency interface greatly simplifies the development of the program, reduces the time needed for staff training and improves the quality of work.

MAIN PART

As an example of remote access technology is considered remote control parameterized model of heavy transport aircraft, created in CAD CATIA, with the help of mobile devices based on OS Android. By remotely varying the design parameters of the aircraft, the user can perform three-dimensional geometric model changeover and associated CFD model to calculate the aerodynamic characteristics of the aircraft without the need for direct interaction with CAD and CFD systems.

The database of design parameters of heavy transport aircraft, integrated into the software and hardware automation preliminary design of the aircraft "robot designer" has been used to test the approach. The database system is made in MS Excel and is associated with a parameterized model. Saving database triggers regeneration model for updates to data, including the selection and calculation of design parameters are not specified by the user at the stage of data entry.

CONCLUSION

Expanding access to the robot designers, enables you to control the calculation at a great distance, to enter data from anywhere, track stages of the design of the aircraft, viewing the finished 3D model as well as the ability to view the model in demo mode.

The work is carried out with the financial support of the Ministry of Education and Science of the Russian Federation.



OSTIS-2015

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822:514

ПОДХОД К ПАРАМЕТРИЗАЦИИ ТРЕХМЕРНЫХ МОДЕЛЕЙ ДЛЯ РЕШЕНИЯ ЗАДАЧ АВТОМАТИЗАЦИИ ПРЕДВАРИТЕЛЬНОГО ПРОЕКТИРОВАНИЯ САМОЛЕТА

Боргест Н.М., Коровин М.Д., Спирина М.О.

*Самарский Государственный Аэрокосмический Университет имени академика С.П.Королева
(национальный исследовательский университет), г. Самара, Россия*

borgest@yandex.ru

maks.korovim@gmail.com

maryspirina@gmail.com

В статье рассматривается подход к автоматизации процесса предварительного проектирования самолета на основе параметризованных САД-моделей, связанных с внешними базами данных проектных параметров и процедур. Обосновывается необходимость разделения моделей и баз данных для реализации модульной архитектуры программной составляющей разрабатываемых автоматизированных систем на примере робота-проектанта самолета.

Ключевые слова: проектирование, автоматизация, трехмерная модель, параметризация.

Введение

Предварительное проектирование самолета – ответственный этап в процессе формирования облика будущего изделия, на котором определяется компоновка и основные проектные параметры, используемые в качестве исходных данных на последующих этапах проектирования. Различие в точности методов расчета, применяемых на разных этапах проектирования – от статистических формул на начальных этапах расчета к высокоточным методам математического моделирования на основе САЕ и CFD расчета, обусловлено тем, что исходных данных, доступных проектанту на начальном этапе, обычно недостаточно для построения трехмерной САД модели, необходимой для точного моделирования.

Задача формирования облика будущего самолета на ранних этапах проектирования может быть решена путем создания параметризованных моделей, связанных с базами данных проектных параметров и процедур, созданных на основе накопленных знаний в предметной области [Lazzara и др., 2009].

1. Виды параметризации

В современных САД/САМ-системах среднего и высокого уровня имеются средства параметризации компьютерных моделей объектов проектирования.

Идея параметризации компьютерной модели детали, заключается в следующем: так как объект конструирования постоянно меняется (возникают новые идеи, выявляются ошибки, меняются условия и факторы, приводящие к изменению конструкции), необходимо описать конструкцию объекта набором размерных характеристик, определяющих геометрию объекта, условия взаимосвязей и взаимоотношений отдельных элементов между собой. При этом набор параметров, описывающих способ построения геометрии объекта, не совпадает с набором параметров, определяющих функционирование и изготовление объекта конструирования. Параметризация конструкции заключается в создании алгоритма, который должен обеспечивать синтез конструкции из отдельных геометрических элементов (метод твердотельного моделирования), возможность модификации конструкции в определенном диапазоне, описывать законы функционирования конструкции, отражать определенную технологию ее изготовления. Подобные задачи, сложны даже для относительно простых устройств, например, таких, как технологическая оснастка (приспособления, штампы, пресс-формы). Однако, уровень развития современных САД систем и прогресс в области вычислительной техники позволяют формализовывать всё более сложные инженерные процессы, обеспечивая инструментарий для создания всё более сложных параметризованных моделей. Ниже приведены некоторые наиболее

распространенные типы параметризации САД моделей, применяемых в проектной деятельности.

1.1. Табличная параметризация

При использовании табличной параметризации создание нового экземпляра детали производится путём выбора из таблицы типоразмеров. Возможности этого типа параметризации ограничены, поскольку задание произвольных новых значений параметров и геометрических отношений обычно невозможно. Табличная параметризация находит широкое применение во всех параметрических САПР, так как позволяет существенно упростить и ускорить создание библиотек стандартных и типовых деталей, а также их применение в процессе конструкторского проектирования [Малюх, 2010].

1.2. Иерархическая параметризация

Параметризация на основе истории построений. В ходе построения модели вся последовательность построения отображается в отдельном окне в виде «дерева построения». В нем перечислены все существующие в модели вспомогательные элементы, эскизы и выполненные операции в порядке их создания.

Помимо «дерева построения» модели, система запоминает не только порядок её формирования, но и иерархию её элементов. Параметризация на основе истории построений присутствует во всех САПР, использующих трёхмерное твердотельное параметрическое моделирование. Такой тип параметрического моделирования обычно используется совместно с вариационной и/или геометрической параметризацией [Норенков, 2002].

1.3. Вариационная параметризация

Вариационная или размерная параметризация основана на построении эскизов с наложением на объекты эскиза различных параметрических связей и наложении пользователем ограничений в виде системы уравнений, определяющих зависимости между параметрами. Вариационная параметризация позволяет легко изменять форму эскиза и величину параметров операций, что позволяет удобно модифицировать трёхмерную модель [Малюх, 2010].

1.4. Геометрическая параметризация

Параметрическое моделирование, при котором геометрия каждого параметрического объекта рассчитывается в зависимости от положения родительских объектов, его параметров и переменных.

Параметрическая модель, в случае геометрической параметризации, состоит из элементов построения и элементов изображения. Элементы построения задают параметрические связи. К элементам изображения относятся линии изображения, а также элементы оформления.

Геометрическая параметризация является наиболее сложной в осуществлении, однако её функциональные возможности делают её предпочтительной при создании сложных параметризованных моделей [Голованов, 2002].

2. Робот-проектант

Внедрение автоматических систем — это сложный и дорогой процесс, иногда требующий значительных затрат времени, однако автоматизация часто возникающих типовых задач способна значительно повысить временную и экономическую эффективность работы проектанта.

Проектирование самолёта представляет собой сложный многодисциплинарный процесс, цель которого состоит в поиске рационального сочетания проектных параметров, которое наилучшим образом удовлетворяло бы поставленным задачам и выбранным критериям.

Робот-проектант — это компьютер с периферийными устройствами, инструментарии, включающие в себя речевые программы, языки описания, СУБД, САД системы, редакторы онтологий, и база знаний, как совокупность тезауруса, баз данных, правил и процедур, со сценариями проектирования [Боргест, 2012].

Результатом работы робота-проектанта является модель изделия. Она состоит из 2 взаимосвязанных частей — матрицы проекта с логикой расчета (в текущем варианте выполненной на основе таблиц MS Excel) и параметризованной трехмерной модели, которая автоматически меняет собственную конфигурацию в зависимости от данных, содержащихся в матрице проекта [Боргест и др., 2012].

Задача аналитического проектирования самолета требует построения математической модели изделия, которая характеризуется:

- совокупностью внешних параметров (требования технического задания на проект самолета: масса и другие параметры полезной нагрузки; потребная дальность полета; крейсерская скорость; класс аэродрома базирования и, соответственно, длина и ширина взлетно-посадочной полосы, её характеристики; категория и параметры системы автоматической посадки на аэродроме базирования и т. д.);
- совокупностью внутренних параметров (параметры самолета, определяемые в процессе решения задачи: прежде всего, взлетная масса, удельная нагрузка на крыло, стартовая тяговооруженность, аэродинамические характеристики, размеры самолета и его агрегатов, состав и параметры систем и т. д.);
- целевой функцией (критерием или критериями эффективности), позволяющей выбрать среди альтернативных проектов лучший, обеспечивающий экстремальное значение целевой функции [La Rocca и др., 2009].

Робот-проектант имеет модульную схему – отдельные подзадачи (например, определение удлинения крыла или его массы) решаются отдельными подпрограммами, выполнение которых инициируется роботом по заданной логике [Borgest и др., 2015].

Опыт разработки автоматизированной системы управления расчетом сложных объектов показал, что практическая реализация системы сводится к решению следующих задач:

- разработка базы данных для хранения необходимых параметров в процессе расчета;
- создание базы данных проектных решений, используемых для преобразования записей базы данных параметров в команды для построения CAD моделей;
- разработка графического интерфейса пользователя;
- разработка параметризованных CAD моделей;
- организация взаимодействия между клиентской и серверной частями;
- формирование исходных данных на входном языке базового программного средства;
- управление базовым программным средством;
- графическое отображение параметров и структуры модели.

Разделение геометрической модели и баз данных позволяет вносить изменения в алгоритм расчета и способы построения моделей, не затрагивая систему в целом, что позволяет реализовать модульный принцип построения системы. Модульная структура вычислительной подсистемы позволяет реализовать возможность выбора среди нескольких методов расчета, а также при необходимости комбинировать их.

Необходимость реализации отдельной базы данных проектных решений обусловлена особенностями процесса проектирования самолета, в частности, различиями в эвристических решениях, применяемых при проектировании самолётов различного назначения. Так, например, близкие по относительным параметрам носовые части фюзеляжей тяжелого и легкого самолетов, могут в значительной мере отличаться по конструкции и применяемым проектным решениям, как показано на рисунке 1.

Совокупность используемых в решении конкретной задачи расчетных модулей формирует сценарий расчета. Он зависит от типа решаемой задачи, степени вовлеченности человека в процесс работы робота-проектанта, а также от выбранного метода расчета. Стоит отметить, что в рамках выбранного проектного решения модель может претерпевать значительные изменения при корректировке соответствующих параметров. Например, после выбора и утверждения конфигурации центральной части фюзеляжа тяжелого транспортного самолета, модель может

изменяться в широком диапазоне проектных параметров, таких как шаг расположения продольных и поперечных элементов каркаса как показано на рисунке 2.

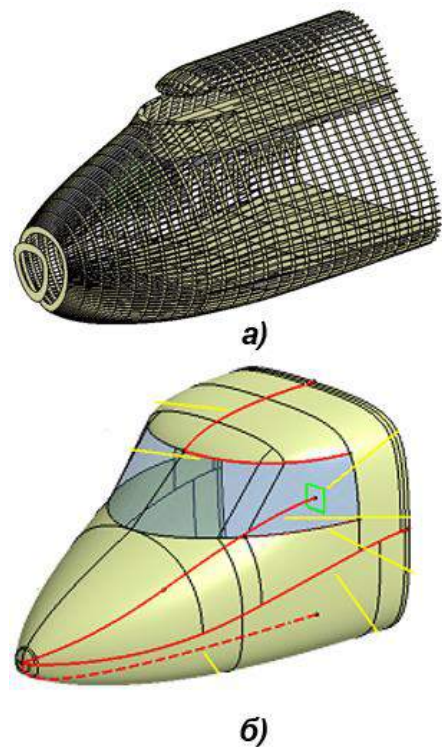


Рисунок 1 – Модель носовой части тяжелого (а) и легкого (б) самолета

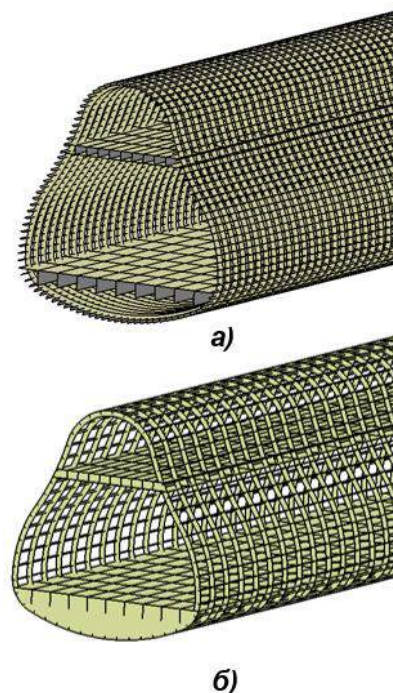


Рисунок 2 – Модели центральной части фюзеляжа тяжелого самолета с разным шагом расположения элементов каркаса

Заключение

В статье описаны современные подходы к параметризации трехмерных моделей. Выделен метод геометрической параметризации как

обладающий наибольшими функциональными возможностями. Описана организация базы данных трехмерной модели самолета в рамках работа-проектанта, программно-аппаратного комплекса автоматизации процесса предварительного проектирования самолета. Отмечена необходимость создания отдельных баз данных проектных параметров и баз проектных решений для реализации модульной схемы работа-проектанта.

Работа выполняется при финансовой поддержке Министерства образования и науки РФ.

Библиографический список

[Anderson и др., 2012] G. R. Anderson, M. J. Aftosmis, M. Nemec // Seventh International Conference on Computational Fluid Dynamics (ICCFD7) Big Island, Hawaii, July 9-13, 2012

[Borgest и др., 2015] Borgest N.M., Korovin M.D., Gromov A.A., Gromov A.I.A. The Concept of Automation in Conventional Systems Creation Applied to the Preliminary Aircraft Design // Soft Computing in Computer and Information Science, Springer International Publishing Switzerland, 2015, DOI 10.1007/978-3-319-15147-2_13

[La Rocca и др., 2009] La Rocca, G., Van Tooren, Knowledge-based engineering approach to support aircraft multidisciplinary design and optimization. J. Aircr. 46(6), 1875–1885 (2009)

[Lazzara и др., 2009] Lazzara, D.S., Haimes, R., Willcox, K.: Multifidelity geometry and analysis in aircraft conceptual design. In: 19th AIAA Computational Fluid Dynamics 22–25 June 2009, San Antonio, 207 Texas (2009)

[Боргест, 2012] Боргест Н.М., Робот-проектант: фантазия и реальность. // Научный журнал «Онтология проектирования» №4(6), 2012. — с. 73-94.

[Боргест и др., 2012] Боргест Н.М., Чернов Р.В., Шустова Д.В. Разработка интерфейса интеллектуального помощника проектанта / Н.М. Боргест, Р.В. Чернов, Д.В. Шустова // материалы международной научно-технической конференции OSTIS-2012. БГУИР. 2012

[Голованов, 2002] Голованов Н.Н., Геометрическое моделирование. — М.: Физматлит, 2002.- 472с.

[Малюх, 2010] Малюх В.Н., Введение в современные САПР: Курс лекций. — М.: ДМК Пресс, 2010.- 162с..

[Норенков, 2002] Норенков И.П., Основы автоматизированного проектирования. — М.: Изд-во МГТУ им. Н.Э.Баумана, 2002. . - 336с.

APPROACH TO THE PARAMETRIZATION OF THREE-DIMENTIONAL MODELS FOR SOLVING THE TASK OF THE AIRCRAFT PRELIMINARY DESIGN AUTOMATION

Borgest N.M., Korovin M.D., Spirina M.O.

*Samara State Aerospace University,
Samara, Russia*

borgest@yandex.ru

maks.korovim@gmail.com

maryspirina@gmail.com

In article approach to automation of process of preliminary design of the airplane on the basis of the parameterized CAD-models connected with external databases of design parameters and procedures is considered. Need of division of models and databases for realization of modular architecture of a program component of the robot designer of the plane locates.

Introduction

Preliminary design of the airplane – one of the most responsible stages of process of formation of shape of future product on which configuration and the key design parameters which are used as basic data at the latest design stages are defined. However, until now there is a gap in the accuracy of the methods of calculation applied at different design stages – from rough statistical formulas at the initial stages of calculation to high-precision methods of mathematical modeling on the basis of CAE and CFD calculation. Emergence of this gap is caused by that the basic data available to the designer at the initial stage of calculation, is usually not enough for construction three-dimensional CAD the model necessary for exact modeling.

The problem of formation of shape of future airplane at early design stages can be solved by creation of the parametrized models connected with databases of the design parameters and procedures created on the basis of the accumulated knowledge in subject domain.

Main Part

Division of geometrical model and databases allows making changes to algorithm of calculation and ways of creation of models without affecting system in general that allows to realize the modular principle of creation of system. The modular structure of a computing subsystem allows to realize possibility of a choice among several methods of calculation, and if necessary to combine them.

Need of realization of a separate database of design decisions is caused by features of process of design of the airplane, in particular, distinctions in the heuristic decisions applied at design of planes of different function. Therefore, for example, relatives in relative parameters nasal parts of fuselages of heavy and light airplanes can differ considerably on a design and the applied design decisions. Set of the settlement modules used in the solution of a specific objective forms the scenario of calculation. It depends on type of the solved task, degree of an involvement of the person into process of operation of the robot designer, and on the chosen calculation method.

It should be noted that within the chosen design decision the model could undergo considerable changes at adjustment of the corresponding parameters.

Conclusion

In article modern approaches to parametrization of three-dimensional models are described. The method of geometrical parametrization as possessing the greatest functionality is allocated. The organization of a database of three-dimensional model of the aircraft within the robot designer, hardware and software system automate the process of preliminary design of the aircraft is described. Need of creation of separate databases of design parameters and bases of design decisions for implementation of the modular scheme of the robot designer is marked out.



OSTIS-2015

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822

МЕТОДЫ ПРЕДСТАВЛЕНИЯ ЗНАНИЙ ДЛЯ РЕШЕНИЯ ЗАДАЧ МОДЕЛИРОВАНИЯ

Коршиков Д.Н., Лахин О.И., Носкова А.И., Юрыгина Ю.С.

*ООО «НПК «Разумные решения»,
г. Самара, Россия*

korshikov@smartsolutions-123.ru, lakhin@smartsolutions-123.ru,

noskova@smartsolutions-123.ru, yurygina@smartsolutions-123.ru

В работе приводятся методы представления знаний в виде предметных онтологий с использованием базиса языка OWL и его расширением до мета-онтологии Аристотеля, дается краткий обзор инструмента создания онтологий, а также описываются функции конструктора онтологий и решаемые задачи.

Ключевые слова: онтология; мета-онтология Аристотеля; семантическая сеть; RDF; OWL; мультиагентные системы; моделирование.

Введение

Требования к интеллектуальным системам управления ресурсами из года в год становятся все жестче, что обусловлено усложнением процессов современного рынка, неопределенностью и динамикой изменения спроса и предложения, возрастающей конкуренцией.

В условиях высокой неопределенности резко возросла ценность знаний специалистов в различных областях деятельности, а, следовательно, возникла потребность в их формализации, хранении и повторном использовании. Основной проблемой является то, что не все знания можно легко описать и запрограммировать, т.к. они обычно являются разнородными, многосвязными, часто как декларативными, так и процедурными, связанными не только иерархическими, но и сетевыми структурами. [Магюшин, 2014].

Поскольку сокращение трудоемкости разработки и сопровождения интеллектуальных программных систем является важной проблемой, то использование онтологий для управления знаниями является целесообразным [Шалфеева, 2011].

В статье рассматриваются методы представления знаний в виде онтологий, описываются особенности выбранного подхода, приводятся примеры решаемых задач.

1. Почему онтологии?

1.1. От классической системы к интеллектуальной

На самом начальном этапе создания системы всегда возникают вопросы: как именно она будет строиться, какой методологией руководствоваться, какие подходы использовать.

Классический подход, с одной стороны, позволяет использовать стандартные методологии, архитектуры и наборы готовых компонентов. Вся информация жестко структурирована и хранится в базах данных. Результат работы такой системы предсказуем и стабилен. С другой стороны, использование жестко закодированных данных и логики делает такие системы сложно расширяемыми и негибкими.

В свою очередь, интеллектуальные системы, в частности мультиагентные системы, использующие онтологии в качестве баз знаний, легко поддаются модификации при наличии такой необходимости. Онтологии также позволяют сильно расширить круг задач, решаемых интеллектуальной системой, в различных актуальных направлениях.

1.2. Спектр решаемых задач

Актуальность использования онтологий в качестве баз знаний легко оценить. Перечень задач, решаемых интеллектуальными системами на основе онтологий, достаточно велик. Рассмотрим основные из них:

- создание глобальной базы знаний путем интеграции различных онтологий, за счет широкого использования полисинонимии и ретранслируемости;
- решение задач, связанных с информационным поиском;
- решение задач, связанных с распознаванием образов;
- решение задач дедуктивного и индуктивного выводов и др.

Дополнительные преимущества онтологии предоставляют в вычислительном (сокращается время вычислений) и экономическом (сокращаются затраты на разработку) плане по сравнению с уже существующими решениями, основанными на классических подходах.

2. Методы и средства представления знаний

Онтология – это средство формализации знаний о предметной области в форме, упрощающей взаимодействие между пользователем и системой.

Термин «онтология» начал использоваться применительно к информационным технологиям сравнительно недавно, в начале 90х годов прошлого века. Но несмотря на это, уже существует достаточно большое количество различных форм и методов описания знаний в виде онтологий со своими достоинствами и недостатками. Также были выделены различные базисы, на которых специалисты могли бы основываться, описывая свои предметные области, так как одной из основных особенностей онтологического подхода является единство терминологии.

Различные формы представления знаний используются в системах, решающих разнообразные задачи, но ни одна из существующих форм не способна полностью покрыть потребности, возникающие при разработке интеллектуальных систем. В связи с этим возникла необходимость доработать существующие методы и определить свой базис для описания и хранения онтологий.

Для создания универсального метода было принято решение взять за основу общепринятые стандарты представления знаний. Согласно этим стандартам, онтологии можно классифицировать по целям создания на онтологии представления, онтологии верхнего уровня, онтологии предметных областей и прикладные онтологии. В настоящей работе более подробно рассматриваются первые два вида онтологий. Пользователь работает с онтологией верхнего уровня, которая автоматически программными средствами преобразуется в онтологию представления.

2.1. Онтологический базис

Для решения поставленных задач онтология представления описывается с помощью языка OWL. Это дает возможность использовать готовые инструменты, работающие с данными описанными

с помощью OWL, а также упрощает интеграцию со сторонними системами. Формат представления данных - RDF (т.е. технически модель данных, с которой работает система, представляет RDF граф).

RDF - это модель для описания объектов. Идея RDF состоит в том, чтобы одним простым способом можно было описать различные утверждения в таком структурированном виде, чтобы его могли обрабатывать компьютерные программы.

RDF – это формат представления информации, организованной в виде графов или сетей. RDF определяет управляемые графы связей, представленные тройками (триплетами), имеющими общий вид «субъект – предикат – объект». Множество RDF-утверждений образует ориентированный граф, в котором вершинами являются субъекты и объекты, а рёбра помечены предикатами.

OWL является расширением RDF-понятий и представляет собой вычислительный логико-ориентированный язык описания онтологий для семантической паутины. В основе языка – представление действительности в модели данных «объект – свойство». Структура OWL-онтологии включает: классы, таксономические отношения между классами, свойства (бинарные отношения), экземпляры классов и отношений.

Для расширения возможностей языка OWL при создании онтологий верхнего уровня в качестве онтологического базиса вводится мета-онтология Аристотеля, основными концептами которой являются «объект», «свойство», «процесс», «отношение» и «атрибут». Согласно этой модели:

- существуют объекты, которые обладают свойствами и характеризуются состояниями;
- свойства выражают способность объектов вступать в процессы взаимодействия на основе законов мира;
- отношения между объектами могут отражать структурные, функциональные, временные или любые другие виды связей;
- чтобы выполнить действие над объектом, необходимо соблюдение определенных условий, которые задаются свойствами и отношениями;
- действия (процессы) изменяют состояния объектов, их свойства и отношения;
- свойства, отношения и действия характеризуются значениями атрибутов;
- атрибуты объекта/отношения являются качественной или количественной характеристикой понятия;
- правила являются обобщенными понятиями для формализованных условий вида «если-то» (предикатов) и высказываний (утверждений, аксиом, фактов) [Скобелев, 2012].

Преимуществом использования такого онтологического базиса является возможность описывать процессы и действия, а значит, помимо физических объектов создавать и процедурные

онтологии, что актуально при решении задач моделирования различных бизнес-процессов.

2.2. Конструктор онтологий

Описываемый базис до сих пор не использовался ни в одном из существующих редакторов онтологий в качестве основного, поэтому параллельно с разработкой и описанием методов создается новый инструмент для работы с онтологиями – Конструктор Онтологий, который учитывает все указанные выше особенности. Его основными функциональными возможностями являются [Вакурина, 2014]:

- создание и редактирование онтологий;
- отображение онтологий в различных видах;
- хранение и использование онтологий.

2.2.1. Создание и редактирование онтологий

Пользователь может оценить преимущества выбранного подхода уже на этапе создания онтологии. Концепция OWL позволяет не только создавать онтологию с нуля, но и описывать новые знания, основываясь на уже существующих онтологиях, таким образом постоянно наращивая объемы информации, не дублируя ее. Реализация многопользовательского режима еще более упрощает эту работу. С этой точки зрения, задача создания единой глобальной базы знаний, о которой говорилось ранее, уже не кажется неразрешимой.

Следующим преимуществом работы в создаваемом Конструкторе является то, что пользователь может при необходимости совершать манипуляции с онтологиями, используя сравнительно простой язык взаимодействия DSL (Domain Specific Language). С его помощью можно вводить различные условия, правила, аксиомы и т.д.

Знания в онтологии могут создаваться не только пользователем, но и системой на основании рассуждений. За это отвечает модуль логического вывода. На данном этапе происходит тесное переплетение онтологического подхода и мультиагентных технологий, так как модуль логического вывода для решения задач классификации использует агентную модель.

Ниже рассматриваются основные принципы работы агентной модели для решения задач логического вывода:

- каждой сущности (экземпляру) и каждому классу онтологии назначается программный агент;
- задача агента - обладать всеми знаниями об этой сущности: каким классам принадлежит, в каких процессах участвует и т.д.;
- знания разделяются на факты и логические выводы.

Для получения знаний агент применяет свои аксиомы, а также запрашивает знания у других агентов. Набор аксиом агента расширяем, что позволяет наделять агента новыми знаниями.

Описание поведения агента:

- каждый агент хранит факты о себе;

- каждый агент делает логические выводы о себе и хранит их;
- каждый агент уведомляет других агентов о своих изменениях, на основе которых они делают новые выводы о себе;
- каждый агент по запросу предоставляет информацию о себе для других агентов;
- сообщения с фактами и уведомления об изменениях содержит информацию, о добавляемых и удаляемых утверждениях;
- при удалении факта, все логические выводы, которые основываются на нем, становятся недействительными.

Преимущества такого подхода:

- при появлении новых фактов будут проверены только те аксиомы, которые напрямую или опосредовано, связаны с этим фактом («локальные возмущения» в сцене);
- знания можно хранить децентрализованно, что делает систему легко расширяемой;
- архитектура система позволяет распараллелить решение задачи.

Однако есть и недостаток: при расширении онтологии и увеличении количества аксиом переговоры разрастаются. Но так как архитектура системы позволяет распараллелить решение задачи, то данная проблема решается расширением аппаратных мощностей.

Как упоминалось ранее, очевидной особенностью предлагаемого подхода является возможность описания процедурной онтологии. Принцип ее создания схож с принципом создания онтологии физического мира, но вместо классов объектов вводятся классы процессов и действий. В рамках описания процесса необходимо определить:

- набор участников процесса и условия их участия;
- тело процесса, которое описывает изменения, вносимые в сцену данным процессом при моделировании.

При описании участника процесса указывается его принадлежность к описанному ранее классу объекта, а также описываются некоторые правила, которые связывают его с другими участниками процесса.

Для описания изменений, вносимых в сцену данным процессом, используется язык DSL. Ниже приведены примеры команд языка DSL, которые могут использоваться (вначале имя команды, в скобках тип передаваемых параметров):

- создать объект;
- удалить объект;
- задать атрибут;
- создать отношение;
- удалить отношение;
- запустить процесс;
- остановить процесс.

Процессы могут состоять из других процессов, в этом случае в теле процесса указывается набор процессов, из которых он состоит.

Действия и процессы описываются практически одинаково, за исключением того, что в действии из участников должен быть выделен субъект, который может выполнить это действие.

На основании описанных в онтологии классов процессов и действий можно строить цепочки – бизнес-процессы, которые в свою очередь могут использоваться для описания алгоритмов действий, описывающих некоторые регламенты и правила.

2.2.2. Отображение онтологий

Для того, чтобы наглядно отображать созданные онтологии в различных разрезах, пользователю предоставляется возможность просмотра созданных онтологий в нескольких вариантах: табличная форма, словарь понятий и карточка концепта, семантическая сеть, алгоритм действий (в случае бизнес-процесса). Каждое из представлений обладает своими особенностями и может иметь разные варианты использования в зависимости от потребностей пользователя.

2.2.3. Использование онтологий для моделирования и решения прикладных задач

В рамках решения конкретных прикладных задач вначале создается онтология, описывающая статическую часть предметной области: классы объектов, атрибуты, отношения. Далее описывается динамическая часть: классы процессов, действий, бизнес-процессы, события. На основании онтологии создается набор моделей, которые описывают непосредственно объекты решаемой задачи, на данном этапе определяются значения атрибутов объектов, отношения между ними. После того как все необходимые модели готовы, на основании них создается сцена, которая используется для решения прикладных задач. Агенты, отслеживают изменения в сцене и выполняют заданное им поведение. Ниже приведены основные задачи агентов в рамках моделирования:

- решают задачи классификации того или иного объекта в сцене на основании описания классов и правил принадлежности к ним;
- находят участников процессов, запускают процессы на выполнение, в случае если все участники найдены, и условия взаимного участия удовлетворены;
- останавливают процессы, если прекращаются выполняться условия и ограничения, накладываемые на участников процесса.

В процессе работы со сценой средствами DSL языка можно управлять сценой, добавляя новые объекты, отношения между ними и выполнять другие команды. Для управления DSL-командами используется специализированный инструмент «командер», который предоставляет удобный пользовательский интерфейс для ввода команд. Контекстная подсказка помогает ориентироваться пользователю и оперативно вводить необходимые

изменения, меняя при необходимости не только сцену, но и непосредственно саму онтологию. Таким образом, знания и логика работы системы может меняться на лету, что является большим преимуществом рассматриваемого подхода, и может использоваться как непосредственно во время эксплуатации системы, так и при предварительной подготовке знаний системы, используемых при решении прикладных задач.

Заключение

В данной статье рассмотрены основные методы и принципы построения онтологий, рассмотрены методы представления знаний для решения задач моделирования. Выполнен краткий обзор функциональности Конструктора онтологий, который является средой разработки онтологий и предоставляет средства моделирования. Рассмотренные методы могут использоваться для описания любой предметной области и решения различных прикладных задач.

Библиографический список

[Матюшин, 2014] Матюшин, М.М. Методы и средства построения онтологий для визуализации связанных информационных объектов произвольной природы в сложных информационно-аналитических системах / М.М. Матюшин [и др.] // Информационно-управляющие системы. – 2014. - № 2(69). – С.9-17.

[Шалфеева, 2011] Шалфеева, Е.А. Возможности использования онтологий при разработке и сопровождении программных систем / Е.А. Шалфеева // Владивосток : ИАПУ ДВО РАН, 2011. – с.16. – 80 экз.

[Скобелев, 2012] Скобелев, П.О. Онтологии деятельности для ситуационного управления предприятием в реальном времени / П.О. Скобелев // Онтология проектирования. – 2012 - №1(3). - С.26-48.

[Вакурина, 2014] Вакурина, Т.Г. Онтология российского сегмента Международной космической станции и ее практическое использование в интеллектуальных аэрокосмических приложениях / В.В. Котеля, О.И. Лакхин, М.М. Матюшин, П.О. Скобелев // Материалы IV Международной научно-технической конференции «Открытые семантические технологии проектирования интеллектуальных систем» Open Semantic Technologies for Intelligent Systems (OSTIS-2014). - Минск: БГУИР, 20 – 22 февраля 2014 г. – С. 221-226.

METHODS OF KNOWLEDGE REPRESENTATION TECHNIQUES FOR USE IN MODELING

Korshikov D, Lakhin O, Noskova A,
Yurygina Yu

Software Engineering Company Smart Solutions,
Samara, Russian Federation
korshikov@smartsolutions-123.ru,
lakhin@smartsolutions-123.ru,
noskova@smartsolutions-123.ru,
yurygina@smartsolutions-123.ru

Knowledge representation techniques as a top-level ontology using OWL language and meta-ontology of Aristotle are listed, a brief overview on developing tools for creating ontologies are made, and some problems solved by using the proposed methods are discussed in this article.



OSTIS-2015

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822

ОНТОЛОГИЧЕСКОЕ МОДЕЛИРОВАНИЕ ТЕХНИЧЕСКОГО ОБСЛУЖИВАНИЯ

Федотова А.В.*, Давыденко И.Т.**

* *Московский государственный технический университет им. Н.Э.Баумана,
г. Москва, Россия*

afedotova.bmstu@gmail.com

** *Белорусский государственный университет информатики и радиоэлектроники,
г. Минск, Республика Беларусь*

ir.davydenko@gmail.com

В контексте интеллектуализации систем управления жизненным циклом сложной техники рассмотрена проблема управления знаниями о жизненном цикле. Главное внимание уделено структурированию и моделированию знаний о техническом обслуживании (на примере ТО самолета). Изложен онтологический подход к управлению знаниями, относящимися к техническому обслуживанию. Построена иерархическая структура онтологий ТО, разработаны онтологии нижнего уровня. На основе модифицированной многокритериальной классификации онтологий по Т.А.Гавриловой предложен вариант формализации технического задания на разработку онтологий.

Ключевые слова: интеллектуальные технологии, управление жизненным циклом, управление знаниями, база знаний, онтология, онтологическое моделирование, техническое обслуживание.

Введение

Основной стратегией производства XXI-го века является управление жизненным циклом (ЖЦ) сложной технической системы (СТС), который охватывает стадии проектирования, производства и эксплуатации. Модели ЖЦ выступают как метамодели для предприятий, которые объединяют их данные, процессы, организационные структуры и персонал в русле формирования компьютерно-интегрированных организаций [Stark, 2011].

Важнейшей стадией ЖЦ СТС является стадия эксплуатации, на основе которой реализуется, поддерживается и восстанавливается качество сложной технической системы, например, самолета. Эксплуатация включает этапы транспортировки и хранения СТС, ее ввода в действие и использования по назначению, технического обслуживания и ремонта. Развитие системного подхода к проблеме технического обслуживания предполагает изучение и моделирование этапа технического обслуживания (ТО) с позиции целостного ЖЦ.

Современными тенденциями построения систем управления ЖЦ являются их интеграция и интеллектуализация. Одним из предпочтительных вариантов разработки интеллектуальных систем

управления ЖЦ является использование массовой семантической технологии компонентного (модульного) проектирования [Голенков и др., 2011], [OSTIS, 2011]. Компонентное проектирование предполагает создание и постоянное расширение библиотек многократно используемых компонентов. Реализация единой технологии компонентного проектирования интеллектуальных систем требует:

- обеспечения полной совместимости (интегрируемости) компонентов на основе их единого представления;
- разработки онтологий компонентного проектирования;
- определения и уточнения типологии таких компонентов (предметные онтологии, онтологии верхнего уровня, многократно используемые фрагменты баз знаний, интерфейсные компоненты и т.д.);
- создания языка спецификации многократно используемых компонентов;
- построения средств компьютерной поддержки синтеза интегрированных систем из имеющихся компонентов.

Словосочетание «семантическая технология» подразумевает, что разработка интеллектуальной системы опирается на ее онтолого-семантические или логико-семантические модели, т.е. формальные описания структуры и смысловых единиц

интеллектуальной системы (и всех ее компонентов) с помощью онтологий и логик. Достоинством такого подхода к проектированию является независимость системы от платформы, на которой она реализуется, т.е. разработав базовую семантическую модель интеллектуальной системы, мы имеем возможность ее реализации на различных платформах, существенно не изменяя при этом саму модель.

В данной работе проблема интеллектуализации управления ЖЦ СТС рассматривается в контексте включения в состав систем управления жизненным циклом подсистемы управления знаниями, циркулирующими на всех этапах ЖЦ (см. [Тарасов и др., 2014], [Jun et al., 2007]). При этом особое внимание уделяется знаниям, относящимся к стадии эксплуатации СТС, в частности, этапам технического обслуживания и ремонта (ТОиР). Развивается онтологический подход к созданию и реализации открытых семантических технологий разработки прикладных интеллектуальных систем (с целью создания в дальнейшем интеллектуальной системы управления техническим обслуживанием и ремонтом – см. также [Karray et al., 2013]).

1. Управление знаниями на основе онтологического моделирования

1.1. Управление знаниями

Важнейшей подзадачей управления ЖЦ СТС является управление знаниями, циркулирующими на его этапах и стадиях. Управление знаниями на протяжении жизненного цикла – это инновационная стратегия предприятия, направленная на обмен знаниями, используемыми на разных этапах ЖЦ, получение новых и обновление существующих знаний, что позволяет сотрудникам лучше понимать друг друга, эффективно взаимодействовать и совместно решать их задачи [Попов, 2001], [Wiig, 1997]. В задачах управления знаниями на предприятии термин «знания» выступает и как ресурс, и как объект управления. В целом, он рассматривается в более широком смысле, чем в искусственном интеллекте.

Обычно информационные ресурсы предприятия включают структурированные базы данных, различные документы, неявные знания сотрудников. Управление знаниями означает создание единого информационного пространства предприятия и его постепенное превращение в пространство знаний [Хорошевский, 2008]. Это обеспечивает гибкий распределенный подход к генерации, сбору, представлению, пополнению, распространению, практическому использованию коллективных знаний предприятия и организации доступа к ним.

Существующие общие классификации знаний чаще всего строятся как дихотомии, например: индивидуальные – коллективные, эвристические – алгоритмические, явные (эксплицитные) – неявные (имплицитные), поверхностные (субъективные)

эвристические правила) – глубинные (теории, формальные модели).

Важнейшим аспектом управления знаниями на протяжении как всего ЖЦ СТС, так и его этапов является первоначальное структурирование знаний. Знания на предприятиях могут подразделяться по различным основаниям, например: 1) стадии и этапы жизненного цикла – проектные, производственные, эксплуатационные знания; 2) компоненты деятельности предприятия – знания о продуктах и услугах, о процессах и технологиях, об оборудовании и оснастке, об организационных структурах и персонале предприятия; 3) компоненты окружающей микросреды предприятия – знания о заказчиках, поставщиках, партнерах, конкурентах.

Следуя Т.А.Гавриловой [Гаврилова и др., 2001], предложившей оригинальную классификацию знаний, где тип знаний соотносится с видом профессиональной деятельности, проведем общее структурирование эксплуатационных знаний для решения задач ТО (табл. 1).

Таблица 1 – Виды знаний, циркулирующих в сфере технического обслуживания

№	Тип знания	Вид профессиональной деятельности
1	ЗАЧЕМ-знания	Стратегический анализ: назначение и функции системы ТО
2	СКОЛЬКО-знания	Экономический анализ: затраты на ТО требуемые ресурсы
3	ЧТО-знания	Концептуальный анализ: основные понятия ТО, отношения между ними, понятийная структура
4	КАК-знания	Функциональный анализ: гипотезы и модели, используемые при организации процедур и работ ТО
5	ПОЧЕМУ-знания	Причинно-следственный анализ, диагностика причин и определение соответствующих видов ТО
6	КОГДА-знания	Временной анализ: определение сроков и продолжительности проведения ТО
7	ГДЕ-знания	Пространственный анализ, определение места проведения ТО
8	КТО-знания	Организационный анализ, распределение функций в группе специалистов, выполняющих ТО

База знаний является ключевым компонентом интеллектуальной системы [Гаврилова и др., 2001]. Понятие базы знаний тесно связано с понятием предметной области. В [Голенков и др., 2011] семантическая структура базы знаний трактуется в рамках семантической технологии проектирования баз знаний интеллектуальных систем как иерархическая система взаимосвязанных между собой предметных областей.

Для построения иерархической структуры базы знаний необходимо в рамках предметной области явно указать класс исследуемых объектов, класс вторичных объектов, построенных на основе

исследуемых, класс вспомогательных объектов, через связи с которыми описываются некоторые характеристики исследуемых объектов. Также надо выделить отношения, связывающие исследуемые объекты между собой, а также отношения, которые связывают исследуемые объекты с вторичными и вспомогательными. Речь идет о построении систем онтологий.

1.2. Онтологии: определение, формализация, характеристики, виды

В информатике под онтологией понимается наглядное и формализованное описание структуры некоторой проблемной области (темы). Исходные теоретические представления об онтологиях и онтологическом моделировании изложены в работах [Gruber, 1993], [Guarino, 1995], [Гаврилова и др., 2001] [Добров и др., 2009], [Клещев и др., 2001]. Следуя Т. Груберу, мы рассматриваем онтологию как «спецификацию разделяемой разными людьми концептуализации» [Gruber, 1993]. По сути, построение онтологии обеспечивает формальное представление семантики.

В целом, онтологии отражают соглашения о единых способах построения и использования концептуальных моделей. Они выступают как удобный метод представления и повторного использования знаний, средство управления знаниями и обучения, способ достижения взаимопонимания в некотором сообществе специалистов (например, онтологии жизненного цикла СТС служат основой взаимопонимания между сотрудниками из разных подразделений машиностроительного предприятия) Пример онтологии жизненного цикла сложной технической системы в виде ментальной карты приведен в работе [Тарасов и др., 2014].

В плане отображения характеристик предметной области, формирование онтологии предполагает использование всех доступных моделей знаний, релевантных для данной области (рис. 1).

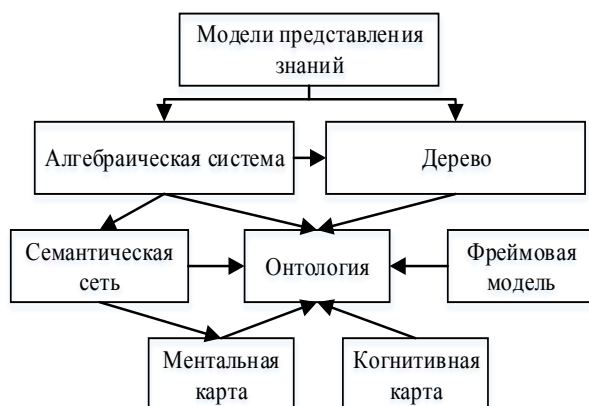


Рисунок 1 – Набор формализмов для представления онтологии

Современные онтологии строятся по большей части одинаково, независимо от языка описания. На

теоретико-множественном языке предметные онтологии задаются четверкой:

$$ONT_{CS} = \langle C, EX, AT, R \rangle, \quad (1)$$

где C – конечное множество понятий предметной области, EX – множество экземпляров, AT – семейство множеств атрибутов понятий (каждое понятие может описываться своим множеством атрибутов), R – конечное множество отношений между понятиями предметной области.

К сожалению, построение единственной понятной и согласованной предметной онтологии часто оказывается невозможным, поэтому обычно строят систему онтологий нижнего и верхнего уровня: на нижнем уровне наряду с предметной онтологией отдельно строятся онтологии задач и приложений, а на верхнем уровне – онтологии базовых категорий, встречающихся в разных предметных областях. Еще выше расположена *метаонтология* (т.е. онтология над онтологиями), которая выступает как основа выбора средств представления, слияния и интеграции онтологий. В нашей работе мы используем два различных вида метаонтологий: сингулярную и гранулярную [Тарасов, 2012]. Так применение сингулярной метаонтологии подразумевает выбор обычного языка теории ориентированных графов и деревьев для онтологии предметной области, а переход к гранулярной метаонтологии означает интервальный характер примитивов онтологии и ее визуализацию с помощью интервалов и гиперграфов. Гранулярные метаонтологии предполагают рассмотрение одних и тех же понятий онтологии на различных уровнях абстрактности. Таким образом, происходят семантизация и прагматизация онтологий.

Следует отметить, что в отечественной литературе два разных понятия – «онтологии верхнего уровня» и «метаонтология» – нередко смешиваются.

Вариант иерархической системы онтологий ТО, представлен на рис. 2. Здесь в качестве онтологии верхнего уровня выступает онтология жизненного цикла СТС.

Для классификации онтологий берут такие основания как цель создания, степень общности, содержимое, глубина проработки, вид базовых отношений и пр. Соответственно онтологии бывают простыми и многоуровневыми, легкими и весомыми, статическими и динамическими, сингулярными и гранулярными, и т.д. [Гаврилова, 2014], [Клещев и др., 2001], [Добров и др., 2009], [Плесневич, 2010], [Тарасов и др., 2012].

Одной из наиболее общих и удобных для постановки задач онтологического моделирования и оценки онтологий является недавно введенная Т.А. Гавриловой базовая классификация онтологий по общему признаку зависимости/независимости от предметной области и следующим 6 критериям:

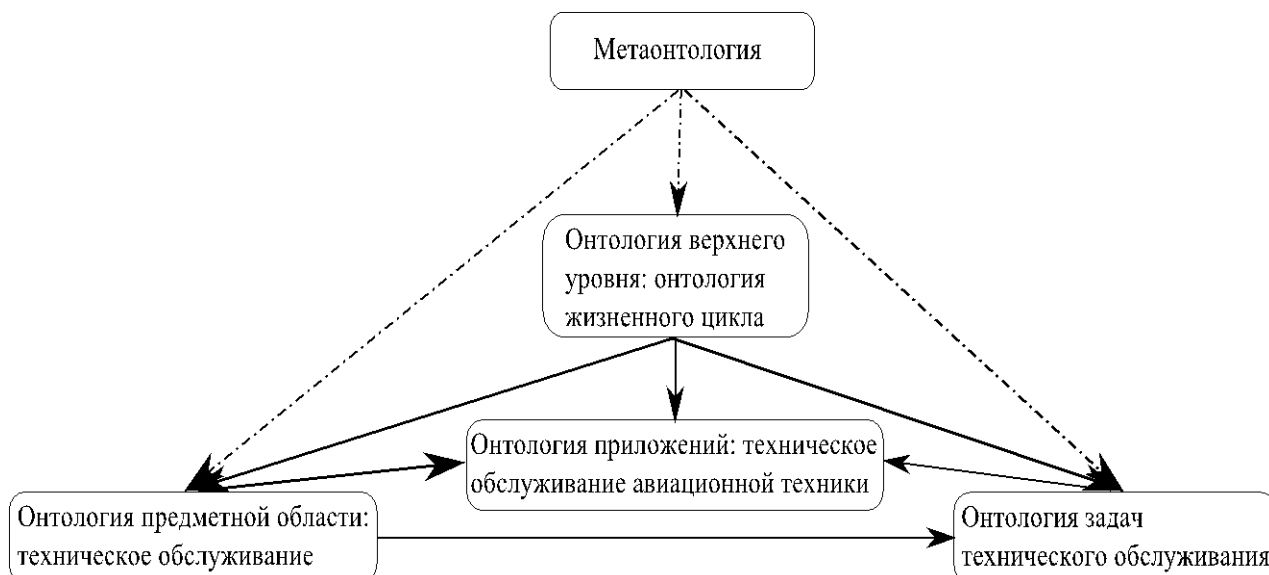


Рисунок 2. – Иерархическая система онтологий технического обслуживания

- Содержание (системная характеристика, определяемая целью построения онтологии, ее функциями, уровнями абстрактности понятий, наличием или отсутствием аксиоматики, логических средств интерпретации и рассуждений);
- Форма (системная характеристика, которая описывает язык представления, глубину онтологии, покрытие или объем, а также ее эргономические характеристики);
- Фокус (возможность постановки акцента на какой-либо ветви);
- Учет времени (наличие динамики);
- Новизна (оригинальность подхода);
- Ошибки (характеризуют синтаксис и семантику онтологии).

Эти шесть критериев позволяют строить набор субъективных классификаций для оценки практических онтологий с учетом их зависимости от предметной области:

- А – зависимые от предметной области,
- В – независимые от предметной области, и
- С – частично зависимые от нее онтологии..

Далее эту классификацию можно продолжить:

А – зависимые от предметной области

А.1– глубина.

А.1.1 – глубинные

А.1.2 – поверхностные (тезаурусы)

А.2 – покрытие (полнота)

А.2.1 – «полные»

А.2.2 – «фрагментарные»

А.3 – эргономичность

А.3.1 – наглядные, удобные для восприятия

А.3.2 – абстрактные

А.4 – корректность

А.4.1 – противоречивые

А.4.2 – непротиворечивые

А.5 – наличие логических средств поддержки

А.5.1 – с логическими средствами

рассуждений

А.5.2 – без логических средств

А.6 – обобщенность

А.6.1 – преобладание общих понятий

А.6.2 – преобладание конкретных понятий

А.6.3 – смешанные

А.7 – фокус

А.7.1 – фокус на отдельных ветвях

А.7.2 – сбалансированные

А.8 учет времени

А.8.1 статические

В.8.2 динамические

В – независимые от предметной области

В.1 – новизна

В.1.1 – заимствованные

В.1.2 – оригинальные

В.2 – ошибки

В.2.1 – исправимые

В.2.2 – неисправимые

В.2.2.1 – ошибки синтаксиса

В.2.2.2 – ошибки семантики

В результате можно строить формулы нужных нам онтологий, например, задачу построения онтологии технического обслуживания самолета можно поставить следующим образом:

A(1.2)(2.1)(3.1)(4.1)(5.2)(7.1)(8.2), т.е.

построить поверхностную, но полную, наглядную, динамическую онтологию без строгого анализа непротиворечивости и логических средств рассуждений с фокусом, например, на ветви «мероприятия технического обслуживания».

Нетрудно видеть, что онтологии нижнего уровня являются предметно зависимыми, а онтологии верхнего уровня и метаонтологии – частично или полностью независимыми от предметной области.

В следующем разделе мы подробнее изложим методику построения онтологий предметной области и задач технического обслуживания самолетов.

2. Онтологии предметной области и задач технического обслуживания (на примере ТО самолетов)

2.1. Основные понятия и атрибуты предметной области «Техническое обслуживание»

Построение онтологии предметной области начинается с определения списка и содержания основных понятий этой области. Онтологии ТО до сих пор остаются малоразработанными (одними из редких источников являются [Горюнова,2009], Kагау et al., 2013]). Поэтому ниже рассмотрим основные понятия и атрибуты ТО и дадим необходимые определения.

К числу основных понятий в сфере технического обслуживания относятся: система ТО, объект ТО, вид ТО, процесс ТО, средства ТО, метод (способ) ТО, объем ТО, цикл ТО, программа ТО.

Система технического обслуживания и ремонта обеспечивает заданный уровень готовности СТС для использования по назначению и поддержанию/восстановлению ее работоспособности в процессе эксплуатации. Здесь главными критериями являются минимальные затраты времени, труда и денежных средств на ТОиР.

Согласно ГОСТ Р 53863-2010, система ТОиР представляет собой совокупность взаимосвязанных средств, документации технического обслуживания и ремонта, а также исполнителей, необходимых для поддержания и восстановления качества изделий, входящих в эту систему. Объектом технического обслуживания (ремонта) называется сложная техническая система (например, самолет Ту-214), обладающая потребностью в операциях ТО и приспособленностью к выполнению этих операций.

В соответствии с компонентной методологией, будем определять онтологию четверкой (1). Здесь *понятия* (concepts) или *классы* (classes) – это абстрактные группы, совокупности или наборы объектов. Они могут включать в себя другие классы, экземпляры, либо же сочетания и того, и другого. Например, понятие-класс «летательный аппарат» включает в себя понятие «самолет». *Экземпляры* (instances) или *индивиды* (individuals) – основные, низкоуровневые компоненты онтологии. Здесь речь идет о конкретных самолетах или процедурах ТО (например, Ту-214, борт №41). Чем является понятие «самолет» – вложенным понятием, или экземпляром зависит от вида онтологии.

Атрибуты (свойства) задают специфическую для объекта информацию. Каждый атрибут имеет, по крайней мере, имя и значение и используется для хранения информации. Например, самолет Ту-214 имеет такие атрибуты, как: (название: Ту-214); (количество кресел: {210, 182, 170}); (двигатель: ПС-90А, 2 х 16000 кгс); (максимальная дальность полета: 6500 км). Основными атрибутами, системы ТОиР, являются объем, трудоемкость, стоимость периодичность работ. Эти атрибуты прямо влияют на эксплуатационные расходы и продолжительность эксплуатации СТС.

Отношения характеризуют зависимости между объектами онтологии. Различные отношения в онтологиях подразделяются на парадигматические (например, таксономические отношения, которые встречаются практически во всех предметных областях) и синтагматические (или ситуативные, отношения, встречающиеся в конкретных ситуациях в определенной предметной области).

Классификация видов ТО дана на рис. 3.

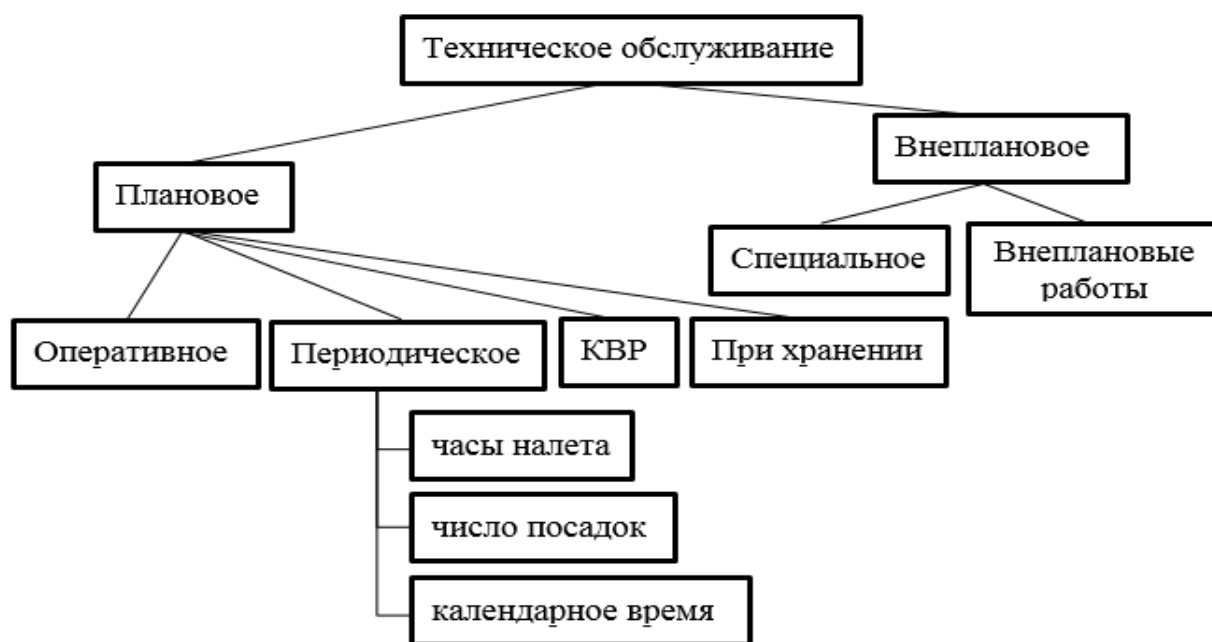


Рисунок 3 – Виды технического обслуживания

2.2. Онтология технического обслуживания

Построение онтологии ТО связано с такими справочниками как: справочник объектов, справочник свойств, справочник работ. Нами рассматривается трехуровневая иерархия понятий – «мероприятие», «процедура», «работа» – относящихся к ТО в целом и его компонентам (рис. 4). Мероприятие по техническому обслуживанию есть целенаправленная деятельность

(родовое понятие – объект верхнего уровня предметной онтологии), связанная с проведением определенных видов ТО. Оно включает набор повторяющихся форм (процедур) ТО, где процедура есть объект среднего уровня. В свою очередь процедура состоит из множества работ (объект нижнего уровня). У каждого компонента онтологии «Работа» есть свой набор свойств, которые часто зависят от свойств входящих в него компонентов.

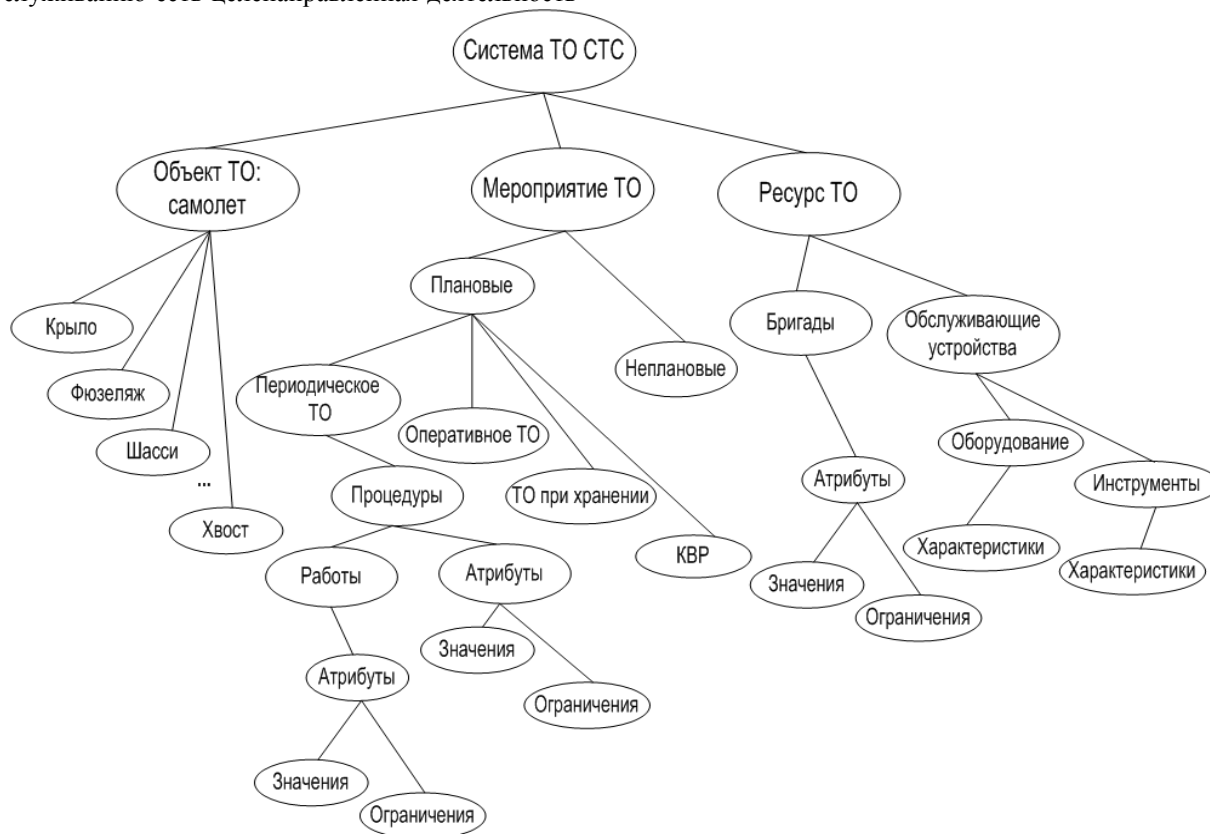


Рисунок 4 – Онтология верхнего уровня для технического обслуживания изделия

Ниже приведем фрагменты онтологии ТО авиационной техники как онтология приложения.

Объекты предметной области:

<ТО, Процедура, Работа, Оборудование, СТС>

Техническое обслуживание: ТО;

Множество процедур $ПР = \{ПР_1, ПР_2, \dots, ПР_n\}$;

Множество работ $РБ = \{РБ_1, РБ_2, \dots, РБ_k\}$;

Оборудование $ОБ = \{ОБ_1, ОБ_2, \dots, ОБ_g\}$;

СТС: $СТС = И$.

Объекты и их свойства:

Объект «ТО» имеет следующие свойства (атрибуты):

ТО.Трудоемкость,

ТО.Затраты;

Объект «ПР» имеет следующие свойства:

ПР.Трудоемкость,

ПР.Затраты,

ПР.Тип;

Объект «РБ» имеет следующие свойства:

РБ.Трудоемкость,

РБ.Затраты,

РБ.Тип,

РБ.Совместная_выполнимость;

Объект «И» имеет следующие свойства:

И.Кол-во_отказов,

И.Время_наработки_на_отказ,

И.Время_наработки,

И.Время_простоя;

Объект ОБ имеет следующие свойства:

ОБ. Время_наработки,

ОБ. Кол-во_отказов,

ОБ. Время_простоя,

ОБ. Время_наработки_на_отказ.

Перечень требований (ограничений) ТР:

{ТР}.Затраты_доп,

{ТР}.Надежность,

{ТР}.Затраты_пр_мин,

{ТР}.Трудоемкость_пр_мин,

{ТР}.Трудоемкость_пр_макс,

{ТР}.Время_наработки_макс,

{ТР}.Трудоемкость_доп,

{ТР}.Время_простоя_макс,

{ТР}.Затраты_пр_макс,

{TR}.Время_наработки_мин,
 {TR}.Время_простоя_макс,
 {TR}.Время_простоя_мин,
 {TR}.Время_выполнения_макс,
 {TR}.Время_выполнения_мин,
 {TR}.Время_макс_м_раб.

На рисунке 5 а) и б) показаны структуры технического обслуживания с различной степенью детализации.

2.3. Онтология задач технического обслуживания

Онтология задач технического обслуживания связана с построением иерархии задач для различных процедур и работ ТО [Kargay et al., 2013].

Например, при ТО самолета выполняются такие задачи как: осмотр, демонтаж, монтаж, очистка, диагностика, замена компонентов системы (рис. 6). Задачи технического обслуживания решаются, опираясь на руководства по технической эксплуатации, а также требования, правила, нормы и стандарты авиакомпании, либо авиаремонтного предприятия. Для осуществления данных работ необходимы персонал, инструменты, расходные материалы и запасные части.

Каждую задачу ТО (работу) можно рассмотреть более детально, декомпозировав ее на операции. Проведем, например, декомпозицию задачи А.2 Демонтаж компонента системы.

В качестве компонента авиационной системы рассмотрим закрылок (рис. 7). Сначала необходимо отклонить закрылок в крайнее положение и убедиться, что он выпущен полностью, а затем сравить давление в гидросистемах и отключить электропитание самолета. Используя переносной светильник, стремянку и торцевой ключ, отвернуть болты и снять обтекатели опор закрылка. Далее следует отвернуть гайки болтов крепления рычагов редукторов-шарниров к качалкам, снять шайбы и вынуть болты опор закрылка. Затем вывести качалки закрылка из проушин балок корневой и концевой опор движением закрылка против направления полета и перпендикулярно оси вращения и снять закрылок. Потом положить осторожно закрылок на подготовленное место. Для проведения данных операций также необходимы вспомогательное оборудование и инструменты: ложемент (стеллаж) для хранения снятого закрылка, переносной светильник, стремянка, набор инструмента техника самолета, вставка для отвертки под болты крепления обтекателя, торцевой ключ для снятия переключки металлизации и силовая отвертка под ключ. Данные задачи решаются техническим персоналом, который руководствуется документацией по технической эксплуатации, а также требованиями, правилами, нормами и стандартами.

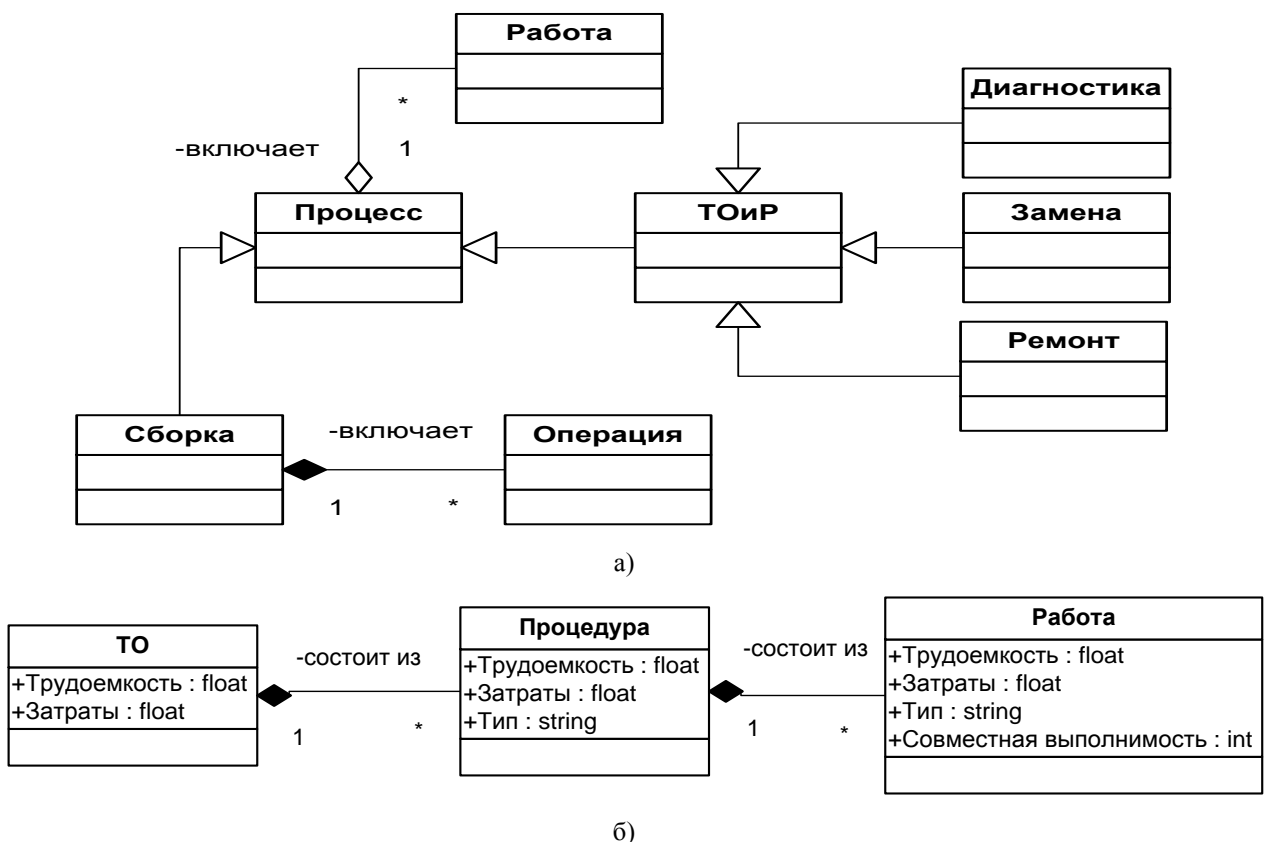


Рисунок 5 – Структурная схема технического обслуживания:
 а) общая схема в терминах диаграммы классов;
 б) фрагменты общей схемы и их свойства

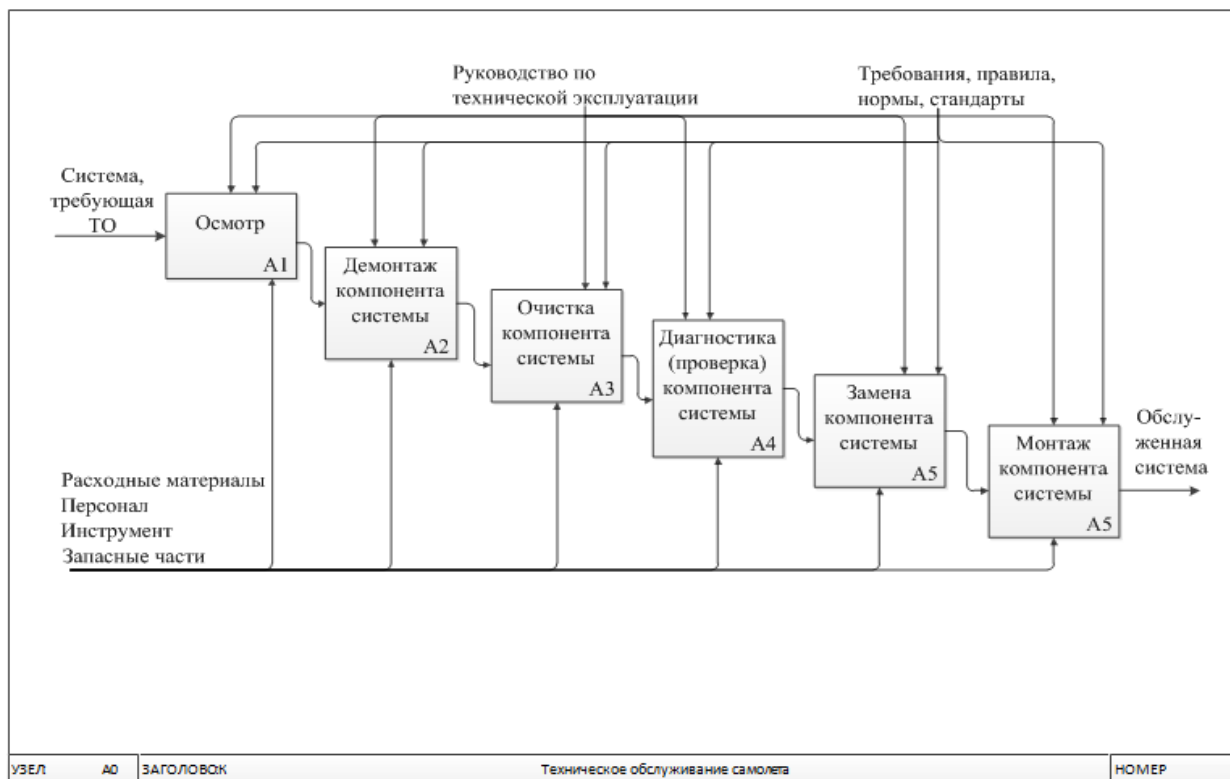


Рисунок 6 – Онтология задач технического обслуживания (диаграмма IDEF 0)

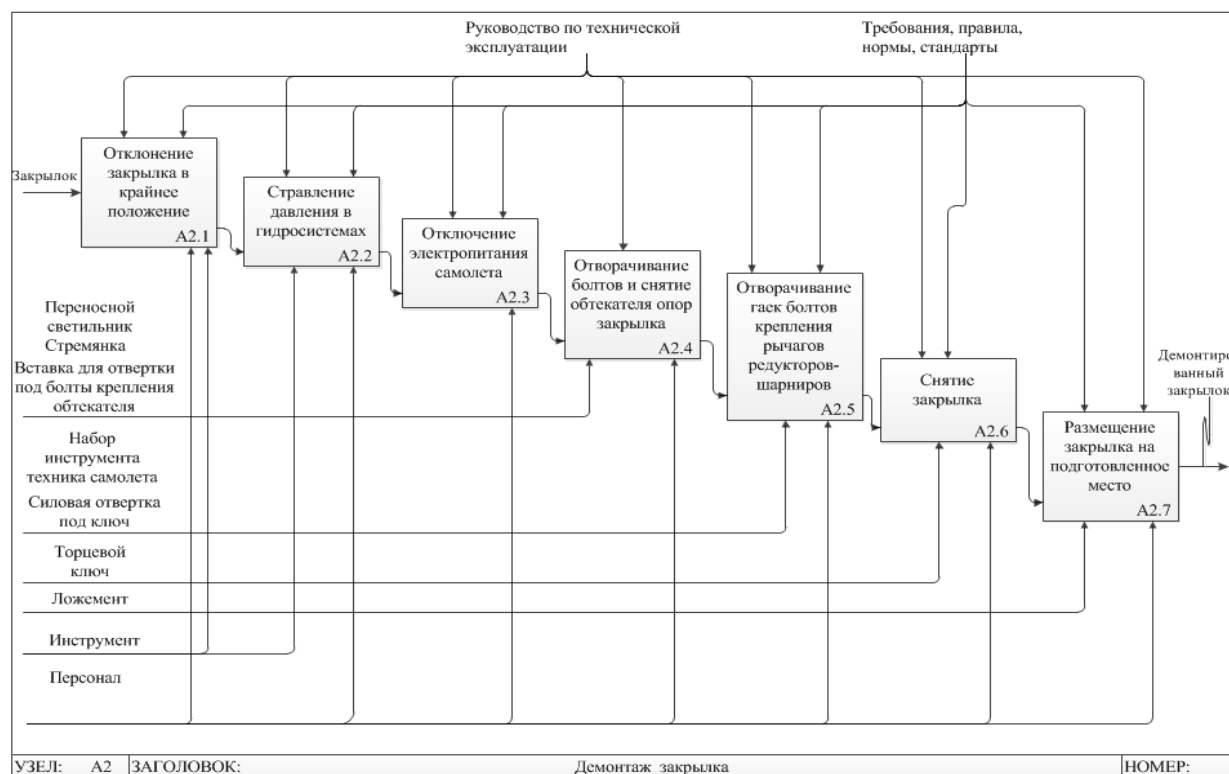


Рисунок 7 – Задачи демонтажа закрылка (диаграмма IDEF 0)

Заключение

В статье рассмотрен онтологический подход к созданию и реализации открытых семантических технологий для разработки интеллектуальной системы управления жизненным циклом сложной

техника. Главное место занимает онтологическое моделирование важнейшего этапа стадии эксплуатации СТС – технического обслуживания.

Дальнейшие перспективы работы связаны с разработкой интеллектуальной справочной системы (ИСС) в области управления жизненным циклом

сложной техники. К числу основных функций ИСС относятся [Давыденко, 2012]:

- обеспечение персонифицированного интерфейса пользователя путем предоставления ему возможности навигации в семантическом пространстве предметной области;
- понимание и интерпретация различных вопросов пользователя, поиск необходимой информации и предоставление ее в удобной для пользователя форме;
- визуальная поддержка постановки и решения задач пользователя с помощью средств когнитивной и коммуникативной графики;
- анализ качества работы пользователя для формирования индивидуальной траектории его обучения.

Подобная система должна опираться на передовые компьютерные вопросно-ответные технологии и обеспечивать пользователю возможность задавать широкий спектр вопросов в предметной области. В частности, в ней следует построить наглядное отображение различных моделей жизненного цикла, прямых и обратных связей между различными стадиями и этапами ЖЦ, основных работ и процедур, выполняемых на конкретных этапах ЖЦ.

Исследование выполнено при поддержке РФФИ, проект №15-07-05623а «Интеллектуальные технологии управления жизненным циклом сложных технических систем».

Библиографический список

[Голенков и др., 2011] Голенков, В.В. Принципы построения массовой семантической технологии компонентного проектирования интеллектуальных систем/ В.В. Голенков, Н.А. Гулякина// Открытые семантические технологии проектирования интеллектуальных систем. Материалы международной научно-технической конференции (OSTIS-2011, Минск, 10-12 февраля 2011 г.). – Мн.: Изд-во БГУИР, 2011. – С.21-59.

[Гаврилова, 2003] Гаврилова, Т.А. Онтологический подход к управлению знаниями при разработке корпоративных информационных систем / Т.А. Гаврилова // Новости искусственного интеллекта, 2003, №2, С.24-30.

[Гаврилова, 2014] Гаврилова, Т.А. Субъективная классификация предметных онтологий / Т.А. Гаврилова // Труды XIV-й национальной конференции по искусственному интеллекту с международным участием (Казань, 24-27 октября 2014г.). – Т.3. – Казань: Изд-во РИЦ «Школа», 2014. – С 93-99.

[Гаврилова и др., 2001] Гаврилова, Т.А. Базы знаний интеллектуальных систем. Учебник/ Т.А. Гаврилова, В.Ф.Хорошевский. – СПб.: Изд-во «Питер», 2001/

[Горюнова, 2009] Горюнова, В.В. Онтологический подход к проектированию систем технического обслуживания/ В.В. Горюнова// Автоматизация и современные технологии. 2009, №12, с.24-29.

[Давыденко, 2012] Давыденко, И.Т. Комплексная методика проектирования семантических моделей интеллектуальных систем/ И.Т. Давыденко// Открытые семантические технологии проектирования интеллектуальных систем. Материалы II-й международной научно-технической конференции (OSTIS-2012, Минск, 16-18 февраля 2012 г.). – Минск: Изд-во БГУИР, 2012. – С.457-466.

[Добров и др., 2009] Добров, Б.В. Онтологии и тезаурусы: модели, инструменты, приложения/ Б.В. Добров, В.В. Иванов, Н.В. Лукашевич, В.Д. Соловьев. – М.: БИНОМ, 2009.

[Заливако и др., 2012] Заливако, С.С. Семантическая технология компонентного проектирования интеллектуальных решателей задач / С.С. Заливако, Д.В. Шункевич // Материалы международной научно-технической конференции «Открытые семантические технологии проектирования интеллектуальных систем» – Минск, 2012

[Клещев и др., 2001] Клещев, А.С. Математические модели онтологий предметных областей. Часть 1. Существующие подходы к определению понятия "онтология"/ А.С. Клещев, И.Л. Артемьева // Научно-техническая информация. Серия 2 "Информационные процессы и системы", 2001, № 2, с.20-27.

[Плесневич, 2011] Плесневич, Г.С. Анализ простых онтологий/ Г.С. Плесневич// Интеллектуальные системы. Коллективная монография/ Под ред. В.М. Курейчика. – М.: Физматлит, 2010. – С.206-221.

[Попов, 2001] Попов, Э.В. Корпоративные системы управления знаниями/ Э.В. Попов// Новости искусственного интеллекта, 2001, №1, с.14-25.

[Тарасов и др., 2012] Тарасов, В.Б. Гранулярные, нечеткие и лингвистические онтологии для обеспечения взаимопонимания между когнитивными агентами/ В.Б. Тарасов, А.П. Калущая, М.Н. Святкина// Открытые семантические технологии проектирования интеллектуальных систем. Материалы II-й Международной научно-технической конференции (Минск, БГУИР, 16-18 февраля 2012 г.). – Минск: Изд-во БГУИР, 2012. – С.267-278.

[Тарасов и др., 2014] Тарасов, В.Б. Онтологии жизненного цикла сложной технической системы/ В.Б. Тарасов, А.В. Федотова, Н.В. Черепанов// Открытые семантические технологии проектирования интеллектуальных систем Материалы IV-й международной научно-технической конференции (OSTIS-2014, Минск, БГУИР, 20-22 февраля 2014 г.). – Минск: Изд-во БГУИР, 2014. – С.471-482.

[Федотова, 2013] Федотова А.В. Интеллектуальная система планирования технического обслуживания и ремонта оборудования в управлении жизненным циклом продукции высокого значения/ А.В. Федотова// Интеллектуальные системы и технологии: современное состояние и перспективы. Сборник научных трудов Международной летней школы-семинара по искусственному интеллекту для студентов, аспирантов и молодых ученых (Тверь – Протасово, 1-5 июля 2013 г.) – Тверь: Изд-во ТвГТУ, 2013. – С.153-161.

[Хорошевский, 2008] Хорошевский, В.Ф. Пространства знаний в сети Интернет и Semantic Web (Часть 1) / В.Ф. Хорошевский // Искусственный интеллект и принятие решений, 2008, № 1, с.80-97.

[ЭЗОП, 2007] Web-сервер онтологий системы ЭЗОП [Электронный ресурс]. Минск., 2010. – Режим доступа: <http://ezop-project.ru/drupal5/>. – Дата доступа: 14.11.2011.

[Gruber, 1993] Gruber, T.R. A Translation Approach to Portable Ontologies/ T.R. Gruber// Knowledge Acquisition, 1993, vol.5, №2, p.199-220.

[Guarino, 1995] Guarino, N. Formal Ontology, Conceptual Analysis and Knowledge Representation/ N. Guarino// International Journal of Human-Computer Studies, 1995, vol.43, №5-6, p.625-640.

[Jun et al, 2007] Jun, H.B. A Primitive Ontology Model for Product Lifecycle MetaData in the Closed-Loop PLM/ H.B. Jun, D. Kiritsis, P. Xirouchakis// Enterprise Interoperability II. New Challenges and Approaches/ Ed. by R.J. Goncalves, J.P. Muller, M. Mertins. M. Zelm. – Heidelberg: Springer-Verlag, 2007. – P. 729-740.

[Karray et al., 2013] Karray, M.H. A Formal Ontology for Industrial Maintenance/ M.H. Karray, B. Chebel-Morello, N. Zerhouni // Applied Ontology, 2012, vol.7, №3, p. 269-310.

[OSTIS, 2011] Проект OSTIS [Электронный ресурс]. Минск, 2011. – Режим доступа: <http://ostis.net/>. – Дата доступа: 11.12.2011.

[Stark, 2011] Stark, J. Product Lifecycle Management: 21st Century Paradigm for Product Realization, 2nd ed. – London: Springer-Verlag, 2011.

[Wiig, 1997] Wiig, K. Knowledge Management: An Introduction and Perspective/ K. Wiig// Journal of Knowledge Management, 1997, vol.1, №1, p.6-14.

ONTOLOGICAL MODELING OF MAINTENANCE

Fedotova A.V. *, Davydenko I.T. **

* *Bauman Moscow State Technical University,
Moscow, Russia*

afedotova.bmstu@gmail.com

** *Belarusian State University of Informatics and
Radioelectronics, Minsk, Republic of Belarus*

ir.davydenko@gmail.com

The problem of lifecycle knowledge management is considered in the context of making intelligent product lifecycle management (PLM) systems. A special attention is paid to structuring and modeling knowledge related to maintenance stage (by using an example of aircraft maintenance). An ontological approach to maintenance knowledge management is developed. A hierarchical structure of maintenance ontologies is built, low-level ontologies are designed. A variant of formal specification of requirements for ontology development is suggested on the basis of a modified Gavrilova's multi-criteria ontology classification. .

Introduction

The main strategy of XXI-st century enterprise is to manage the lifecycle (LC) of complex technical systems (CTS), which covers the stages of design, production and use. Here any LC model may be viewed as a meta-model that integrates enterprise data, processes, organizational structures and personnel to enable the formation of computer-integrated organizations.

The most important stage of lifecycle is the stage of use where the quality of complex technical systems, such as aircraft, is implemented, maintained and restored.

Modern trends in lifecycle management systems consist in developing integrated intelligent systems. One of the preferred choices for the development of intelligent product lifecycle management systems is the adoption of mass semantic technology of component (modular) design.

In this paper, the problem of building intelligent PLM systems is considered in the context of incorporating into PLM system the knowledge management subsystem, where any knowledge circulating at all stages of lifecycle is taken into consideration. In particular, maintenance knowledge that pierces through the use stage is of primary concern.

Main Part

The important aspect of knowledge management throughout the lifecycle of CTS and all its stages is the initial structuring of knowledge. Knowledge in enterprises can be classified by various criteria, such as: 1) the stages and phases of the lifecycle – design, production, use knowledge; 2) the components of enterprise activities – the knowledge on the products

and services, processes and technologies, equipment and accessories, organizational structure and personnel; of the enterprise; 3) the components of the enterprise microenvironment – knowledge about customers, suppliers, partners, competitors.

The paper gives a general structuring of maintenance knowledge depending on professional activities.

The knowledge base is a key component of any intelligent system. A semantic structure of knowledge base is closely related to a hierarchy of ontologies. Since the construction of a single clear and coherent ontology is often impossible, we need to consider low and upper level ontologies as well as a meta-ontology. In this paper the main attention is paid to low level maintenance ontologies such as domain ontology, tasks ontology, application ontology. The pluralism principle for ontological languages is formulated.

A useful modification of Gavrilova's multi-criteria ontological classification is discussed in the context of specifying formal requirements for ontologies. Main concepts and attributes of maintenance domain are analyzed. Some examples of tasks ontologies and application ontologies in case of aircraft maintenance are considered.

Conclusion

Some basic problems of constructing intelligent PLM systems are faced. The concept of lifecycle knowledge management based on open semantic technologies is viewed as the core of intelligent PLM. An ontological approach to the creation and implementation of open semantic technologies for the development of intelligent PLM systems is discussed. Nowadays the problem of ontological modeling of CTS's use stage still remains open (to compare with design and production stages of LC). Our paper is focused on a very important maintenance phase that is crucial for the use stage. The hierarchical system of maintenance ontologies is suggested that is the first step on the way to intelligent maintenance consulting systems.



OSTIS-2015

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822:514

СЕМАНТИЧЕСКИЕ АСПЕКТЫ ПРЕДСТАВЛЕНИЯ И ОБРАБОТКИ ПОИСКОВЫХ ЗАПРОСОВ В СИСТЕМЕ КОРПУС-МЕНЕДЖЕР

Невзорова О.А. *, Мухамедшин Д.Р. **, Билалов Р.Р. **

* Научно-исследовательский институт «Прикладная семиотика» АН Республики Татарстан,
Казанский (Приволжский) федеральный университет, г. Казань, Россия

onevzoro@gmail.com

** Институт вычислительной математики и информационных технологий Казанского
(Приволжского) федерального университета, г. Казань, Россия

damirmuh@gmail.com

akibro@yandex.ru

В статье рассматриваются семантические аспекты представления и обработки поисковых запросов, применяемые в системе управления лингвистическими данными (корпус-менеджер), предназначенной для работы с электронным корпусом текстов татарского языка.

Ключевые слова: корпус-менеджер; поисковые запросы; морфологические формулы.

Введение

Представление и обработка поисковых запросов в системах, поддерживающих лексико-морфологический поиск в электронном корпусе текстов, предполагает анализ формальной и содержательной семантики поисковых запросов. Основными этапами обработки данных, содержащих семантические аспекты, являются разбор лексической и морфологической составляющих запроса, выявление синтаксических и логических ошибок в запросе, определение классов, содержащих необходимые данные, преобразование поискового запроса в запрос к хранилищу данных. В статье будет рассмотрено два из указанных выше этапов: выявление ошибок и определение классов.

Многие поисковые системы, работающие с национальными лингвистическими корпусами, используют готовые ядра, такие как «Яндекс.Сервер», используемый в поисковой системе Национального корпуса русского языка¹. Подобные системы представляют собой комплекс из нескольких подсистем, что позволяет производить быстрый и многофункциональный поиск. Система «Яндекс.Сервер» является проприетарной, а её полная версия распространяется на коммерческой основе.

Корпус-менеджер², обсуждаемый в настоящей статье, разработан специально для работы с лингвистическими корпусами. Функционал, предлагаемый системой корпус-менеджер, включает в себя поиск лексических единиц, морфологический поиск, лексико-морфологический поиск, поиск синтаксических единиц, поиск n-грамм с учетом грамматики и др. В основе системы корпус-менеджер заложена семантическая модель представления данных корпуса татарского языка. Поиск производится при помощи общедоступных инструментов с открытым исходным кодом: система управления базой данных MariaDB и хранилище данных Redis. Разработанный корпус-менеджер ориентирован в первую очередь на поддержку электронных корпусов тюркских языков, что является весьма актуальным для активно развивающегося направления тюркской корпусной лингвистики.

1. Представление данных в системе корпус-менеджер

В системе корпус-менеджер данные представляются в виде семантической структуры классов, которая представлена на рисунке 1. Базовым классом является класс Документы, включающим в себя подкласс Контексты, который в свою очередь включает в себя подкласс Разборы.

¹ <http://www.ruscorpora.ru/>

² <http://corpus.antat.ru/search/>

Последний подкласс содержит в себе около 55,7 млн. элементов, делится на несколько подклассов по морфологическим признакам, а также связан с классами Словоформы, Леммы, Морфологические признаки. Между элементами классов Разборы, Контексты и Документы имеются связи «часть-целое» (на рисунке сплошная линия), между элементами класса Словоформы (и Леммы) и Разборы – связи «часть-целое», между элементами классов Разборы и Морфологические признаки – множественные связи «hasGrammaticalFeature» («имеет морфологический признак», на рисунке квадратные точки), между элементами классов Словоформы и Морфологические признаки – множественные связи «usedWithGrammaticalFeature» («употребляется с морфологическим признаком», на рисунке штрих-пунктир), между элементами класса Морфологические признаки и подклассами класса Разборы – связи «isFoundIn» («встречается в», на рисунке круглые точки).

Такая структура позволяет производить поиск с учётом многих лексических и морфологических параметров, а также находить логические ошибки в пользовательских поисковых запросах. Каждый элемент кроме представленных связей имеет определенные свойства, такие как название, позиция в контексте, позиция в документе и т.п. Эти свойства не используются непосредственно в процессе поиска, но могут быть выведены пользователю наряду с остальными.

2. Представление поисковых запросов

Пользовательский поисковый запрос к системе корпус-менеджер представляет собой упорядоченный кортеж, состоящий из кортежей длины 7 (в общем случае), включающий следующие компоненты:

- Вид поиска;
- Лексическая составляющая запроса;
- Морфологическая составляющая запроса;

- Направление поиска;
- Минимальное расстояние между запросами;
- Максимальное расстояние между запросами;
- Тип поиска.

Например, кортеж Q , указанный в (1), является примером поискового запроса. Здесь Q_1 – первый кортеж поискового запроса, состоящий из следующих компонентов: *lemma* – вид поиска (поиск по лемме), *китап* – лексическая составляющая запроса (*китап (Tat)/книга*), $N|V$ – морфологическая составляющая запроса (морфологическая формула, которая означает «имя существительное ИЛИ глагол»), *right* – направление поиска (вправо по отношению к предыдущему кортежу запроса), 1 – минимальное расстояние от предыдущего кортежа с учётом направления, 10 – максимальное расстояние от предыдущего кортежа с учётом направления, *exact* – тип поиска (точный); Q_2 – второй кортеж поискового запроса, где изменены следующие компоненты: *бар (Tat)/«есть»*, «*имеется*» – лексическая составляющая запроса, $V|ADV$ – морфологическая составляющая запроса (морфологическая формула, которая означает «глагол ИЛИ наречие»), другие компоненты подобны компонентам в кортеже Q_1 . Кортеж Q состоит из кортежей Q_1 и Q_2 .

$$Q_1 = (lemma, \text{китап}, N|V, right, 1, 10, exact)$$

$$Q_2 = (lemma, \text{бар}, V|ADV, right, 1, 10, exact) \quad (1)$$

$$Q = (Q_1, Q_2)$$

Вид, направление, тип поиска, минимальное и максимальное расстояние между запросами имеют детерминированный характер и не требуют дополнительного разбора, тогда как лексическая и морфологическая составляющие запроса имеют стохастический характер, и, чтобы выявить

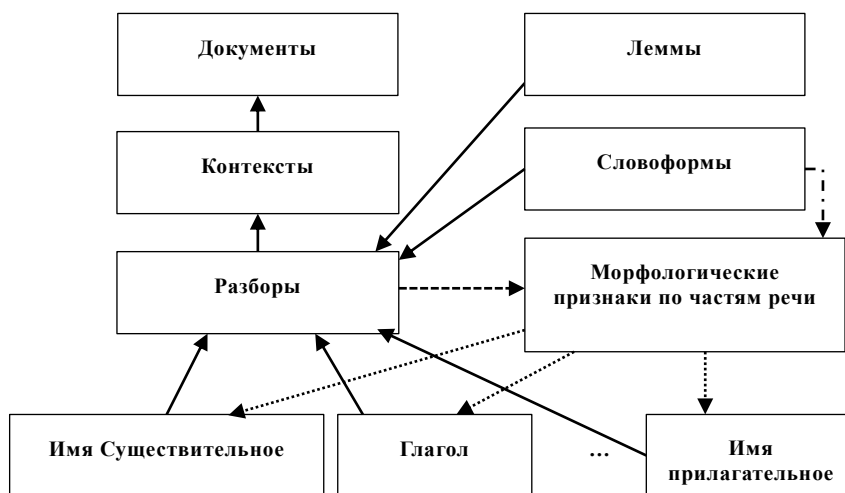


Рисунок 1 – Связи между элементами в системе

семантику этих компонент, необходимо произвести их разбор и обработку.

2.1. Лексическая составляющая запроса

В системе корпус-менеджер лексическая составляющая запроса может представлять собой не только отдельную словоформу или лемму (например, «*китап*» (*Tat*)/*книга*), но и их части (например, «*кита**») или правила с исключениями (например, «*китап* - «*китаплар*»») при поиске по леммам, что означает «все словоформы, имеющие лемму “*китап*”, за исключением словоформы “*китаплар*», «*китаплар*» (*Tat*)/*книги*). Такой расширенный синтаксис лексических составляющих запроса требует со стороны системы правильной интерпретации семантики запроса, заложенной пользователем.

При обработке запроса сначала производится поиск лексической составляющей в классе Словоформы или Леммы в зависимости от вида поиска. Элементы класса Словоформы представляют собой двуместный кортеж вида (2), элементы класса Леммы – двуместный кортеж вида (3).

(словоформа, идентификатор) (2)

(лемма, идентификатор) (3)

Поиск производится для всех частей лексической составляющей, разделенных пробельными символами, по первому элементу кортежа. В результате поиска формируются правила для поиска результатов среди элементов класса Разборы. Например, для запроса «*кита**» при поиске по словоформам будут найдены все словоформы, начинающиеся с «*кита*» («*китап*», «*китаплар*», «*китабы*» и т.д.), которые представляют собой подмножество искомым словоформ W . Для этого примера правилом для поиска станет (4), где w – компонент «словоформа» в кортеже элемента класса Разборы.

$w \in W$ (4)

Если значимая (не исключаящая) часть лексической составляющей запроса не существует в соответствующем классе, система сообщает пользователю о синтаксической ошибке.

2.2. Морфологическая составляющая запроса

Определение семантики морфологической составляющей поискового запроса является хоть и тривиальной, но весьма сложной задачей. В системе возможно использовать морфологические формулы, составленные из морфологических обозначений признаков и операций конъюнкции, дизъюнкции и отрицания. Формула может быть вида конъюнкции переменных (5), дизъюнкции переменных (6), отрицания дизъюнкции переменных (7) и произвольной (8).

N, DIR, SG (5)

$N|DIR|SG$ (6)

$!(N|DIR|SG)$ (7)

$!N|DIR, SG$ (8)

В приведенных примерах морфологические формулы имеют следующую семантику:

- (5) – «имя существительное И направительный падеж И единственное число», то есть по этой формуле должны быть найдены все словоформы, которые имеют одновременно морфологические признаки «имя существительное», «направительный падеж» и «единственное число»;

- (6) – «имя существительное ИЛИ направительный падеж ИЛИ единственное число», то есть по этой формуле должны быть найдены все словоформы, которые имеют хотя бы один из морфологических признаков «имя существительное», «направительный падеж», «единственное число»;

- (7) – «НЕ имя существительное ИЛИ направительный падеж ИЛИ единственное число» (может быть преобразовано в «НЕ имя существительное И НЕ направительный падеж И НЕ единственное число»), то есть по этой формуле должны быть найдены все словоформы, которые не имеют ни одного из морфологических признаков «имя существительное», «направительный падеж», «единственное число»;

- (8) – «НЕ имя существительное ИЛИ направительный падеж И единственное число», то есть по этой формуле должны быть найдены все словоформы, которые не имеют морфологического признака «имя существительное», либо имеют одновременно морфологические признаки «направительный падеж» и «единственное число».

В зависимости от типа формулы, обработка поискового запроса производится по различным алгоритмам.

3. Обработка поисковых запросов

Для формул вида (5), (6), (7) задача определения семантики является тривиальной. Для этих типов формулы общий алгоритм обработки отображен на рисунке 2.

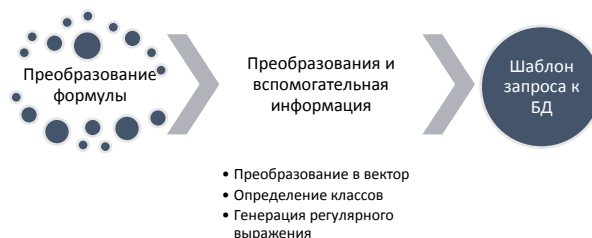


Рисунок 2 - Общий алгоритм обработки морфологических формул

Шаги алгоритма следующие:

1. Преобразование формулы в кортеж морфологических обозначений (разбиение по знаку операции «,» или «|»), здесь же удаляются дополнительные символы: скобки, знаки «!» и «~»).

2. Преобразования и вспомогательная информация.

а. Морфологические обозначения преобразуются в битовый вектор, пригодный для использования в запросе к базе данных.

б. На основе морфологических обозначений определяются классы, где возможно нахождение искомого данных.

с. Генерируется регулярное выражение для отметки подходящих сущностей в контекстах.

3. Формирование запроса к базе данных.

3.1. Обработка произвольных формул

Задача правильного определения семантики произвольной формулы требует наиболее общего решения, покрывающего любые сочетания операций в формуле. Таким решением стали поочередные замены частей формулы на простые составляющие вида (5), (6) или (7) в порядке приоритета операций. Замены производятся пока не будет получена простая формула вида (5), (6) или (7), в которых выполнена замена морфологических обозначений признаков на специальные переменные.

Каждая итерация начинается с процесса преобразований. Это попытка выявить соответствие переменной существующему типу простых формул (с возможностью упрощений, где это может оптимизировать скорость запроса к базе данных, например, $!(N,!V) = N|V$) и заменить эту переменную сразу. Попытка преобразования предпринимается, если в значении переменной не используются другие переменные. Если попытка преобразования завершилась успешно, текущая итерация прерывается.

Если в формуле есть переменные или формула оказалась произвольного типа, части формулы типа (5), (6) или (7) заменяются на переменные, каждая из которых представляется в формуле в виде «REPLACED_S_N», где N – номер замены. Очередность замен соответствует приоритету логических операций.

Затем в формуле из переменных производится поиск частей типа (5). Такие части заменяются на новые переменные, которые содержат в себе все свойства части (пересечение классов, условия, регулярные выражения).

Последним этапом итерации является замена частей типа (6) (или отдельных переменных). На этом этапе все переменные в формуле заменяются на условия запроса к базе данных, объединяются классы и формируются регулярные выражения.

Итерации выполняются для всех первичных замен со скобками. Результаты последней итерации

являются окончательными для сложной формулы и содержат в себе список классов, где возможно нахождение искомого данных, условие для запроса к базе данных, а также регулярное выражения для выделения сущностей в контексте.

Конечный результат работы алгоритмов для обработки всех типов формул представляет собой кортеж, состоящий из преобразованной формулы для работы с базой данных и упорядоченного кортежа, который содержит классы, в которых возможно нахождение искомого результатов.

3.2. Определение классов для лексической составляющей запроса

Чтобы определить множество классов, в которых возможно нахождение данных по искомой лексической составляющей запроса, необходимо воспользоваться связями «usedWithGrammaticalFeature» между элементами классов Словоформы и Морфологические признаки и связями «isFoundIn» между элементами класса Морфологические признаки и подклассами класса разборы. Операция поиска производится для всех частей лексической составляющей запроса и записывается в S_i , где S – множество частей лексической составляющей запроса, i – номер части. В общем случае, получив классы для всех частей лексической составляющей запроса, их необходимо объединить, как показано в (8). Здесь n – это количество частей лексической составляющей запроса, SL – множество классов, в которых возможно нахождение искомого данных.

$$SL = \bigcup_{i=0}^{n-1} S_i \quad (8)$$

Заметим, что исключающие части лексической составляющей в формировании множества классов участия не принимают, так как из их исключения из запроса не следует исключение классов, в которых могут находиться искомые данные.

3.3. Определение классов для морфологической составляющей запроса

Определение секций для морфологической составляющей запроса уже было показано выше для произвольных формул. Для формул типов (5), (6), (7) определение классов проводится по следующему алгоритму:

1. Множество SM представляет собой множество всех возможных классов, если формула имеет тип (4), иначе задается как пустое множество.

2. Если в формуле существует i -е морфологическое обозначение, определяется номер (далее N) для i -го морфологического обозначения, иначе алгоритм завершает работу.

3. Если морфологическое обозначение определяет морфологический класс, то множество s_i заполняется такими натуральными числами x , что $N * 10 \leq x < (N + 1) * 10$, переход к шагу 5, иначе шаг 4.

4. Поиск множества классов по связи «isFoundIn» между элементом класса Морфологические признаки, соответствующем морфологическому обозначению, и подклассами класса Разборы. Найденное множество записывается как s_i .

5. Если формула имеет тип (6), множество SM объединяется с s_i : $SM = SM \cup s_i$, переход к следующей итерации (шаг 2), иначе шаг 6.

6. Если формула имеет тип (5), множество SM пересекается с s_i : $SM = SM \cap s_i$, переход к следующей итерации (шаг 2), иначе шаг 7.

7. Множество SM задается как дополнение к множеству s_i (по отношению к множеству всех возможных классов): $SM = \bar{s}_i$, переход к следующей итерации (шаг 2).

Общее множество классов S определяется как пересечение множества классов для лексической составляющей запроса и множества классов для морфологической составляющей запроса, как показано в (9).

$$S = SM \cap SL \quad (9)$$

3.4. Выявление логических ошибок

Если множество S пустое, это означает, что пользователь допустил логическую ошибку в запросе, и по этому запросу не могут быть найдены никакие данные, так как не существует ни одного класса, в котором возможно нахождение искомых данных. Логическая ошибка может быть выявлена на любом этапе работы системы, если любое из множеств SL и SM является пустым множеством.

В ходе экспериментов в рамках нагрузочного тестирования и тестирования правильности, 6,88% всех запросов содержало логические ошибки, но они не были выявлены системой. По результатам экспериментов была изменена схема представления данных в системе, в которой на момент проведения экспериментов не было связей между элементами класса Морфологические признаки и подклассами класса Разборы. Внесённые изменения позволили существенно улучшить эффективность алгоритма выявления логических ошибок и сократить количество запросов с логическими ошибками, которые не были выявлены системой.

Примером запроса с логической ошибкой может служить запрос, в котором присутствует морфологическая составляющая (10).

$$!(N|V), INF_1 \quad (10)$$

Из морфологической формулы (10) видно, что её первая часть «!(N|V)» имеет вид (7) и означает «НЕ имя существительное или глагол» или «НЕ имя существительное И НЕ глагол». Для этой части результаты могут быть найдены в любом классе, кроме «имя существительное» и «глагол». Вторая часть запроса «INF_1» означает «инфинитив, оканчивающийся на аффикс -ьрга», для этой части классом, в котором могут быть найдены результаты,

является только класс «глагол». Между частями формулы стоит операция конъюнкции. Это означает, что множество классов, содержащих результаты для формулы, является пересечением множеств классов, содержащих результаты для каждой части формулы. Такое множество является пустым множеством, а значит формула содержит логическую ошибку.

Заключение

Предложенные в настоящей статье методы представления и обработки поисковых запросов реализованы в поисковом модуле системы управления лингвистическими данными, работающей с электронным корпусом татарского языка.

Чтобы проверить пригодность, правильность, согласованность, характер изменения во времени, было проведено комплексное тестирование системы. Тестирование пригодности показало, что предложенные методы полностью решают поставленные задачи. Тестирование правильности, основанное на сравнении с другим эталонным методом представления и обработки запросов, показало, что предложенные методы работают правильно. Тестирование согласованности и характера изменения во времени показало, что предлагаемый синтаксис лексической и морфологической составляющей запроса верно интерпретируется системой, а время, необходимое для обработки и выполнения поискового запроса системой, не превышает 0,05 сек. в 98,71% случаев для лексического поиска, в 77,71% случаев для морфологического поиска и в 98,08% случаев для лексико-морфологического поиска. Во многом, таких результатов удалось добиться благодаря представлению данных в виде семантической сети и предложенным методам представления и обработки поисковых запросов.

Использование семантических сетей для представления данных лингвистических корпусов позволяет покрыть широкий спектр задач. В статье описаны лишь некоторые из них. В перспективе предполагается использование этой же схемы для реализации морфологического анализатора и решения задачи снятия омонимии.

Общий подход к решению задач в поисковой системе для лингвистического корпуса позволяет использовать разработанную систему не только для работы с электронным корпусом текстов на татарском языке, но и с корпусами других языков, без существенных изменений в системе.

Библиографический список

- [Suleymanov et al., 2013] Dzhavdet Suleymanov, Olga Nevzorova, Ayrat Gatiatullin, Rinat Gilmullin, Bulat Khakimov National corpus of the Tatar language "Tugan Tel": Grammatical Annotation and Implementation // Procedia - Social and Behavioral Sciences (2013), vol. 95, pp. 68-74.
- [Nevzorova, Salimov, 2012] Nevzorova O., Salimov F. Model of Lexicographical Database: Structure, Basic Functionality,

Implementation // International Journal “Information Models and Analyses”. Vol.1. Number 1, 2012. - P.21-27.

[Cormen et al., 2009] Cormen, Thomas H.; Leiserson, Charles E.; Rivest, Ronald L.; Stein, Clifford (2009). Introduction to Algorithms (3rd ed.). Massachusetts Institute of Technology. Pp. 253–280.

[Davies 2009] Mark Davies The 385+ million word Corpus of Contemporary American English (1990–2008+): Design, architecture, and linguistic insights // International Journal of Corpus Linguistics. Volume 14. Number 2, 2009, pp. 159-190(32)

[Han 2011] Jing Han Survey on NoSQL database // Pervasive Computing and Applications (ICPCA), 2011 6th International Conference on, 26-28 Oct. 2011, pp. 363-366.

[Oracle Corporation 2015] MySQL 5.6 Reference Manual: 19 Partitioning, Oracle Corporation. Web: <http://dev.mysql.com/doc/refman/5.6/en/partitioning.html>

[Аброскин 2009] Аброскин А. А. Поиск по корпусу: проблемы и методы их решения // Национальный корпус русского языка: 2006-2008. Новые результаты и перспективы. СПб.: Нестор-История, 2009

[Захаров 2002] Захаров В.П. Корпусная лингвистика. Учебно-метод. пособие. – СПб., 2005. – 48 с.

SEMANTIC ASPECTS OF SEARCH REQUEST REPRESENTATION AND PROCESSING IN CORPUS-MANAGER SYSTEM

Nevzorova O.A.*, Mukhamedshin D.R.**,
Bilalov R.R.**

* *Research Institute of Applied Semiotics of the Academy of Sciences of Tatarstan Republic, Kazan (Volga Region) Federal University, Kazan, Russia*
onevzoro@gmail.com

** *Institute of Computer Mathematics and Information Technologies, Kazan (Volga Region) Federal University, Kazan, Russia*
damirmuh@gmail.com
akibro@yandex.ru

The article provides semantic aspects of search request representation and processing applied in management system of linguistic data (corpus manager), designed to work with corpus of texts of the Tatar language.

Introduction

Corpus manager discussed in this article designed specifically for working with linguistic corpora. Functionality available within the corpus manager system includes a search of lexical units, morphological search, lexical and morphological search, search of syntactic units, search n-grams with the grammar etc. The corpus manager system based on semantic model data representation of corpus of the Tatar language. Designed corpus manager focuses primarily to support electronic corpora of Turkic languages, which is highly relevant to the rapidly developing areas of Turkic corpus linguistics.

Main part

Data in the corpus manager system presented in the form of the semantic structure of classes. Such structure allows searching considering many lexical and morphological parameters, as well as finding logical errors in user search queries.

User search query to the corpus manager system represented an ordered tuple consisting of tuples of length 7 (generally), which includes the following components: type of search, lexical component of the request, morphological component of the request, search direction, minimum distance between requests, maximum distance between requests, search type.

When processing a request, first searched for lexical component in the Word forms class or Lemmas class depending on the type of search. As a result of the search, system generates rules for the search for results among the elements of the Parsing results class.

Depending on the type of formula, the search query processing performed by various algorithms. For simple formulas, steps of algorithm are as follows: conversion formula to the tuple of morphological signs, conversions and obtaining supporting information for the query to the database, forming query to the database. The problem of the correct defining the semantics of a complex formula requires the most common solutions covering any combinations of operations on a formula. This solution became alternately replacement parts of the formula into simple components in order of operations priority.

To define a set of classes in which is possible to find data on the required lexical component of the request, you must use the relations “usedWithGrammaticalFeature” between elements of the Word forms class and the Grammatical features class and relations “isFoundIn” between elements of the Grammatical features class and subclasses of the Parsing results class. The total set of classes S defined as the intersection of a set of classes for the lexical component of the request and a set of classes for the morphological component of the request.

If the set S is empty, it means that user has made a logical error in the request and this request can not be found any data, since there is no class where possible to find the required data.

Conclusion

The use of semantic networks for data representation in linguistic corpora allows covering a wide spectrum of problems. This article describes a few of them. In the future, it is planned to use the same scheme for the implementation of the morphological analyzer and solving the problem of disambiguation.

A general approach to solving problems in the search engine on linguistic corpora may allow the use of this system not only with electronic corpus of texts of the Tatar language, but also with corpora of other languages without significant changes in the system.



OSTIS-2015

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822

ТЕХНОЛОГИЯ АВТОМАТИЗИРОВАННОГО СИНТЕЗА ИНФОРМАЦИОННЫХ СИСТЕМ С ПОМОЩЬЮ СЕМАНТИЧЕСКИХ МОДЕЛЕЙ ПРЕДМЕТНОЙ ОБЛАСТИ

Бикмуллина И.И.

Казанский национальный исследовательский технический университет

им. А.Н. Туполева, г.Казань, Россия

elsiyar-b@yandex.ru

В работе рассмотрена проблема отсутствия автоматизированного структурного синтеза информационных систем на основе семантических отношений предметной области. Предложен искусственно интеллектуальный подход к автоматизированной разработке информационных систем, усовершенствование современной технологии за счет автоматизированного синтеза диаграмм классов UML на основе семантических моделей.

Ключевые слова: UML; синтез диаграммы классов; автоматизированный синтез структурных моделей; информационная система; разработка.

Введение

Современное общество уже немыслимо без информационных систем (ИС), которые пронизывают почти все области деятельности и постоянно развиваются и совершенствуются. Улучшение качества ИС неразрывно связано с процессом их разработки. Каждый маленький элемент этого, теперь уже огромного, процесса играет очень большое значение [Якунин, 2008].

В 60-е гг. прошлого столетия научное направление «Искусственный интеллект» в нашей стране еще только формировалось, но уже возникло четкое понимание того, что без концептуальной структуризации знаний, без их формального представления создать искусственный интеллект невозможно. Добиться этого интуитивными эвристическими приемами не удалось. Опираясь на взаимосвязи лингвистики с семиологией, прагматикой, теорией информации, белорусский ученый В.В. Мартынов предложил способ «исчисления языковых смыслов» [Массель и др., 2014].

В последние годы наиболее распространенной методологией создания программных систем является объектно-ориентированная (ОО) методология, появившаяся в 80-е годы, по которой окружающий нас мир состоит из объектов и отношений между ними.

Согласно В. Далю [Даль, 1975] объект (предмет) - это все, что представляется чувствам (объект

вещественный) или уму (объект умственный). Таким образом, *объект* воплощает некоторую сущность и имеет некоторое состояние, которое может изменяться со временем как следствие влияния других объектов, находящихся с первым в каких-либо отношениях. Он может иметь внутреннюю структуру: состоять из других объектов, также находящихся между собой в некоторых отношениях. Исходя из этого, можно построить иерархическое строение мира из объектов. В процессе познания или изменения окружающего нас мира мы всегда принимаем в рассмотрение ту или иную упрощенную модель мира (*модельный мир*), в которую включаем объекты и отношения некоторых интересующих нас классов из окружающего нас мира. Таким образом, окружающий нас мир, можно рассматривать (в некотором приближении) как иерархическую структуру модельных миров [Кольцов и др., 2005].

Так в 90-е годы бурно развивается CASE-технология разработки программных систем и связанные с ней формальные методы спецификации программ. CASE (Computer Aided Software Engineering) или программная инженерия с компьютерной поддержкой – это система средств автоматизированного конструирования программного обеспечения (ПО).

Составными частями процесса разработки ИС с использованием CASE - средств являются:

- разработка словаря предметной области;
- разработка диаграмм с помощью языка представления данных UML [Рамбо и др., 2002]

(унифицированный язык моделирования);
- синтез программ.

Используя CASE-мышление, программист уже делает свои программы лучше, поскольку представление программных объектов в диаграммах UML [Рамбо и др., 2002] позволяет наглядно увидеть ошибки и недоработки в полученной иерархии, легко этой иерархией манипулировать, что при ручном кодировании программист вряд ли может осуществить. Однако, установившийся, в настоящее время, стереотип «кодирование, а затем лишь моделирование» не позволяет в полной мере использовать мощные средства UML. А наличие ПО с последовательным, верным, автоматизированным процессом разработки ИС помогло бы облегчить труд разработчика.

В России широко используемыми CASE-средствами, поддерживающими ОО методы, являются Rational Rose Enterprise Edition 2000/2002 (фирмы Rational Software Corporation), Oracle Developer Suite 2000 (фирмы Oracle) и др.

Но для того чтобы перейти к созданию и сопровождению кода при помощи CASE-средства, поддерживающего язык UML (например, Rational Rose), программист должен перестроить свое представление о создании программ. Так как необходимо мыслить уже в терминах языка UML, мыслить диаграммами, а переход к такому типу мышления требует примерно такого же усилия, как переход от процедурного программирования к ОО. Потому что в процессе разработки ПО главной сложностью является формализация предметной области, к которой относится решаемая задача. Для этого необходима хорошая предметно-ориентированная модель, проникающая значительно дальше поверхностного взгляда на проблему. Если в такой модели удастся правильно отразить внутреннюю структуру предметной области, то разработчики ПО получают именно тот инструмент, в котором они нуждаются.

Проблема исследования заключается в необходимости автоматизации процесса разработки (синтеза) диаграммы классов UML.

Поэтому актуальной можно признать цель данного исследования – повышение качества проектной проработки ИС за счет автоматизации процесса синтеза структурных моделей. Из множества структурных моделей информационной технологии выбрана модель диаграммы классов [Якунин, 2008] UML.

1. Анализ современной технологии разработки программы

Разработка интеллектуальных систем и систем обработки сложно-структурированной информации, является трудоемким процессом, но еще более сложным и трудоемким является сопровождение таких систем. Поэтому исследование методов, средств, технологии разработки и сопровождения

остается одной из актуальных задач [Грибова и др., 2013].

В настоящее время наблюдается тенденция все большего вовлечения экспертов в разработку ИС с целью ее ускорения. Представляется очевидным, что чем меньше будет разрыв между знаниями и представлениями эксперта и средствами представления знаний инструментальной системы, тем легче ими будет пользоваться эксперту [Загоруйко, 2013].

К числу основных условий, необходимых для создания в ближайшем будущем эффективных технологий проектирования ИС, можно отнести [Голенков и др., 2013]:

(1) Осознание того, что основным практическим результатом ИИ являются не сами ИС, а мощные и эффективные технологии их разработки.

(2) Осознание того, что без активного и массового привлечения молодых участников создания нового поколения проектирования ИС, указанные технологии не только не будут созданы, но и не смогут в дальнейшем динамично развиваться.

1.1. Определение этапов современной технологии разработки программы

Традиционно автоматизация сопровождается разработкой программного обеспечения практически под конкретную задачу, под класс задач.

Создание прикладного программного обеспечения (ПО) — это сложный процесс. Наиболее критичной его частью является непосредственно программирование приложения, так как на этом этапе осуществляется мысленное отображение понятий прикладной предметной области в используемый язык программирования, что требует знания как языка предметной области, так и языка программирования [Новиков и др., 2009]. На основе анализа технологий создания программных систем получена схема современной технологии создания программной системы (рисунок 1).



Рисунок 1 – Этапы современной технологии разработки ИС

Современная технология разработки программы (рисунок 1) в идеале предполагает:

1 этап. Считается, что эксперт самостоятельно, вручную или в текстовом редакторе разрабатывает онтологию, глоссарий предметной области. Однако, на данном этапе всегда участвуют: эксперт рассматриваемой предметной области (например, машиностроение) и сам разработчик. При этом разрабатываемая система может опираться на несколько предметных областей, что влечет привлечение к разработке нескольких экспертов. В

этом случае разработчик помимо своего дела должен освоить еще ряд специальностей и добиться понимания с экспертами. А эксперту, в процессе разработки онтологии предметной области, трудно выражать свои мысли в процедурном виде [Джарратано и др., 2006], поэтому эти описания в значительном виде должны иметь декларативный характер [Зюзьков, 2003]. Декларативный характер описания близок к языку такому, на котором привык эксперт выражать свои мысли, описывать предметную область. При этом обратное декларативных описаний наиболее соответствует декларативный язык программирования, например, Пролог [Зюзьков, 2003], язык позитивно образованных формул [Васильев и др., 2000]. И все же автоматизация первого этапа в настоящее время изучена явно недостаточно.

2 этап. В идеале, на данном этапе эксперт и разработчик совместными усилиями проектируют диаграмму UML. Однако, чаще всего моделирование осуществляется разработчиком самостоятельно, вручную или составляется им в приложении какого-либо ПО.

Эта модель отражает физическую реализации системы и описывает создаваемый продукт на уровне классов и компонентов. В отличие от модели анализа, модель проектирования имеет явно выраженную зависимость от условий реализации, применяемых языков программирования и компонентов. Для максимально точного понимания архитектуры системы, эта модель должна быть максимально формализована, и поддерживаться в актуальном состоянии на протяжении всего жизненного цикла разработки системы [Трофимов, 2003]. Однако стремление разработчиков как можно быстрее получить программный код приводит к тому, что моделирование диаграмм UML разработчики используют лишь, в крайнем случае, когда проектированию достаточно сложного программного изделия зашло в тупик, а это бывает достаточно запоздалое использование данной технологии. Что ведет к снижению качества программного продукта, а также в необходимости дополнительного ведения документации с целью сопровождения ПО.

3 этап. Разработчик при автоматизированном проектировании ПО на основе разработанных моделей UML с помощью Rational Rose или Visual Studio автоматизированно генерирует программный исходный код.

Таким образом, так как вмешательство разработчика необходимо на каждом этапе проектирования, то увеличивается как нагрузка разработчика, так и время создания программного изделия. Поэтому разработка диаграмм UML целиком зависит от квалификации программиста, знании данного программиста в области UML технологии, а также времени отведенного как на разработку диаграмм UML, так и на вникание в область рассматриваемой предметной области. Если автоматизировать процесс проектирования

диаграмм классов UML, то интерес программистов к использованию инструмента моделирования UML может существенно возрасти.

1.2. Язык представления данных UML

Наиболее распространённым и развитым средством визуального моделирования является UML [Object Management Group, 2014] – язык графического описания для объектного моделирования в области разработки программного обеспечения. Методология UML позволяет производить дальнейшую разработку проекта с определением объекта автоматизации и построением для него информационной модели на физическом уровне вплоть до создания схемы базы данных и генерации кода заголовочных файлов на различных ОО языках [Ивашенко и др., 2004]. Составной частью ОО методологии является задача синтеза структурной модели информационной технологии. Частным случаем задачи синтеза структурной модели является получение диаграммы классов UML.

Класс (тип class языков программирования) описывает множество однотипных объектов с одним и тем же набором атрибутов и функций [Медведев, 2005].

2. Предлагаемое решение проблемы

Решение поставленной в данном исследовании проблемы заключается в модификации современной технологии разработки ИС, создании ПО, позволяющего автоматизировано синтезировать UML диаграмму классов. Разрабатываемый программный продукт как бы будет являться надстройкой над программным продуктом, Rational Rose.

Решение проблемы, недостаточной проработки программного продукта в результате нарушения технологии разработки ПО, заключается в расширении интеллектуальных возможностей автоматизированного проектирования ИС с использованием семантических структурных отношений предметной области.

2.1. Усовершенствованная технология разработки ИС

В предложенной технологии разработки ИС (рисунок 2) наряду с автоматизированной генерацией кода автоматизируется и этап синтеза диаграммы UML. В данной работе из-за невозможности «объять необъятное» синтез диаграммы UML рассматривается на примере синтеза диаграммы классов UML. Также в предлагаемой технологии во время описания онтологии предметной области экспертом, вмешательство программиста становится минимальным, а эксперт получает более высокий статус – статус разработчика.

Усовершенствованная технология разработки ИС (рисунок 2) состоит из следующих этапов:

1 этап: эксперт в приложении автоматизированного синтеза диаграмм UML (разрабатываемого ПО) на основе требований программы синтеза описывает (самостоятельно) онтологию, глоссарий предметной области.

2 этап: программа автоматически синтезирует UML диаграмму, в данной работе автоматизированный синтез рассматриваться будет на примере диаграммы классов.

3 этап: получение на основе полученной на вход диаграммы классов UML «исходника» программы (сгенерированного программного кода, например, в Rational Rose).



Рисунок 2 – Предлагаемая технология создания программной системы

Итак, разница между усовершенствованной и современной технологиями разработки ИС заключается в автоматизации второго этапа - синтеза диаграммы классов UML.

Так если в современной технологии разработки ИС (рисунок 1) чаще всего онтологию предметной области создает разработчик, используя эксперта лишь в виде консультанта, после чего разработчик (или привлекая эксперта) разрабатывает UML диаграмму. То в усовершенствованной технологии создания ИС (рисунок 2) эксперт разрабатывает онтологию предметной области с помощью дружественного интерфейса программы, которая автоматически синтезирует диаграмму классов UML.

Таким образом, эксперты в предлагаемой технологии создания ИС должны сами стать полноправными разработчиками программной системы наравне с разработчиком-программистом.

2.2. Предлагаемая методика автоматизированного процесса проектирования

Предлагается ИИ подход к разработке UML модели на примере построения диаграмм классов.

Синтез диаграмм классов включает:

- 1) построение всевозможных моделей диаграммы классов;
- 2) построение оптимальной модели.

Содержанием задачи автоматического синтеза диаграммы классов является выделение классов и назначение структурных отношений между классами. Как правило, при проектировании информационной технологии могут быть построены несколько структурных моделей, поэтому в нашем случае решение задачи может быть вариантным. Выбор подходящего варианта из представленных диаграмм классов UML остаётся за проектировщиком.

Общая схема процесса автоматизированного синтеза структурной модели, диаграммы классов UML приведена на рисунке 3, где представлен и процесс разработки экспертом онтологии предметной области (1 блок рисунка 3), и формализация предметной области (1 этап и 2 блок рисунка 3) в приложении усовершенствованной технологии проектирования ИС, и синтез, который приведет к структурному решению как описание процесса проектирования, и вывод геометрического представления диаграммы (последний блок рисунка 3).



Рисунок 3 – Укрупненная схема процесса проектирования

Отсюда, предлагаемая методика включает в себя (рисунок 3):

1. Формирование описания модели предметной области экспертом. На основе онтологии, словаря предметной области (1 блок) эксперт заполняет приложение ПО автоматизированного синтеза UML диаграммы.

Формализация задачи моделирования семантики структурной модели программы, заключается в определении структурной модели программы с помощью добротной системы понятий.

Ниже приведены поля в приложении автоматизированного синтеза диаграммы классов UML, которые необходимо заполнить эксперту. Где применяются следующие обозначения:

a – наименование (с маленькой буквы) понятия (concept);

B – наименование (с большой буквы) атрибута (свойства объекта);

b – значение атрибута (свойства объекта);

C – наименование (с большой буквы) функции;

c – значения функции;

D – ограничение области применения (воздействия данного объекта).

Понятие *a*

Атрибуты

B: {*b*}.

Функция *C* (*B*: {*b*}): вещественное.

Значение *c*

Свойство *D*

Конец *a*

Рассмотрим на примере: одной из задач в банковских информационных технологиях является получение прогноза функционирования банков. Прогнозирование осуществляется с помощью статистических характеристик банковских процессов. Для простоты выберем следующие

статистические характеристики: среднее арифметическое значение (оценка математического ожидания) и дисперсию. Необходимо решить задачу синтеза диаграммы классов. Семантическое описание задачи выглядит следующим образом:

Раздел исходные данные

Понятие массив_экспериментальных_данных

Атрибуты

Размер : {целое}.

Данные : {вектор [Размер] из вещественное}.

Функции Заполнить_вектор(Данные : {вектор [Размер] из вещественное}).

Значение

_Ввод_вещественного_вектора(Данные).

Свойство Размер > 0.

Конец массив_экспериментальных_данных

Конец раздела исходные данные

Раздел основные статистические характеристики

Понятие среднее_значение

Атрибуты

Данные : {массив_экспериментальных_данных}.

Значение : {вещественное}.

Функции Получить_значение(Данные : {вектор из вещественное}): вещественное.

Значение _Сумма_вектора(Данные) / Размер из Данные.

Свойство нет

Конец среднее_значение

Понятие дисперсия

Атрибуты

Данные : {массив_экспериментальных_данных}.

Среднее_значение : {среднее_значение}.

Значение : {вещественное}.

Функции Получить_значение(Данные : {вектор из вещественное}, Среднее_значение : {среднее_значение}): вещественное.

Значение

_Сумма_Квadrата_вектора(Данные) / Размер из Данные - _Квadrат(Среднее_значение).

Свойство нет

Конец дисперсия

Конец раздела

основные статистические характеристики

На данном этапе закладываются в семантический язык статистические характеристики:

(1) Спецификация свойств и особенностей класса, в нашем случае, прогноз банков состоит из разделов. Так спецификация «массив экспериментальных данных» содержит раздел: «исходные данные». А раздел «основные статистические характеристики» содержит такие спецификации, как «дисперсия» и «значение среднего».

(2) Спецификация профессионального раздела представляет собой систему описаний понятий исследования. Определение отдельного понятия имеет структуру: **Понятие** <название> <тело понятия> **Конец** <название>.

(3) Описание тела понятия является основой для автоматизированной обработки и может содержать следующие разделы: «Параметры», «Состав», «Атрибуты», «Внешние», «Свойства».

(4) Раздел описания «Атрибуты» определяет атрибутные свойства понятия и представляет собой бинарное отношение «денотат характеризуется ...» между определяемым понятием и элементами раздела. Описание и использование раздела «Атрибуты» строится по тем же правилам, что и описание раздела «Состав».

2. Синтез структурной модели. Содержанием задачи автоматического синтеза диаграммы классов является выделение классов и назначение структурных отношений между классами. Таким образом, разрабатывается система аксиом, которая может быть использована в качестве правил принятия решений в автоматизированной системе синтеза диаграмм классов UML.

Ввиду простоты нашего примера выделение классов является элементарным: структура классов соответствует структуре понятий. Поэтому основное внимание сосредоточим на синтезе структурных отношений. Проиллюстрируем, какие диаграммы классов можно синтезировать в нашем случае (перечень неполный). На рисунке 4 изображено шесть вариантов диаграммы классов, в которых используются отношение обобщения (рис. 4а), отношение композиции (рис. 4б), отношение агрегации (рис. 4в), отношение композиции и обобщения (рис. 4г, рис. 4д), отношение клиент-сервер (рис. 4е).

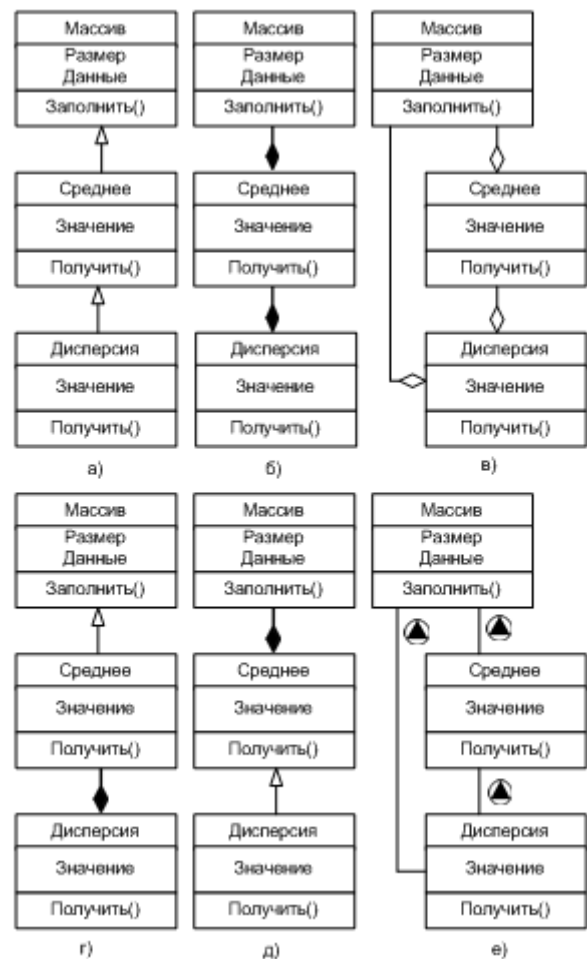


Рисунок 4 – Варианты диаграммы классов

Следовательно, создается формальная система, которая в качестве механизма структуризации данных включает в себя правила их построения, отражающие логику свойств структурной модели предметной области. В результате получим именно ту систему, которая будет использоваться для автоматизированного проектирования диаграммы классов, т.е. систему обработки описаний с четким теоретическим фундаментом.

Заключение

В работе представлено усовершенствование современной технологии разработки ИС за счет автоматизированного проектирования диаграмм классов UML на основе семантических моделей.

Автоматизация синтеза диаграммы классов дает возможность уменьшить трудоемкость проектирования структурных моделей ИС, повысить качество программного проекта за счет возможности многовариантного анализа структурных моделей, стимулировать разработчика на предварительное продумывание структуры прорабатываемой системы, изменить характер труда программиста в направлении от ручной проработки диаграмм UML к выбору оптимального варианта из множеств автоматически полученных вариантов.

Библиографический список

- [Якунин, 2008] Якунин Ю.Ю. Технологии разработки программного обеспечения. Версия 1.0: электрон. учеб. пособие / Ю. Ю. Якунин. – Красноярск : ИПК СФУ, 2008.
- [Массель и др., 2014] Массель Л.В., Массель А.Г. Ситуационное управление и семантическое моделирование в энергетике/ Л.В. Массель, А.Г. Массель// OSTIS, 2014, С. 111-116
- [Даль, 1975] Толковый словарь русского языка/Даль. В. – М.: Советская энциклопедия, 1975
- [Кольцов и др., 2005] Кольцов А.С., Федорков Е.Д. Технология программирования: Учеб. пособие. Воронеж: Воронеж. гос. техн. ун-т, 2005. -272 с.
- [Рамбо и др., 2002] Рамбо Дж., Якобсон А., Буч Г. UML: специальный справочник– СПб.: Питер, 2002. - 656 с.
- [Грибова и др., 2013] Грибова В.В., Клещев А.С. Облачные семантические технологии проектирования интеллектуальных сервисов/ В.В. Грибова, А.С. Клещев// OSIS, 2013, С. 25-30
- [Загорюлько, 2013] Загорюлько Ю.А. Технологии разработки интеллектуальных систем, основанные на интегрированной модели представления знаний/ Ю.А. Загорюлько// OSIS, 2013, С. 31-42
- [Голенков и др., 2013] Голенков В.В., Гулякина Н.А. Открытый проект, направленный на создание технологии компонентного проектирования интеллектуальных систем/ В.В. Голенков, Н.А. Гулякина// OSIS, 2013, С.55-77
- [Новиков и др., 2009] Новиков Ф. А., Тихонова У. Н. Автоматный метод определения проблемно-ориентированных языков (часть 1)/ Ф. А. Новиков, У. Н. Тихонова // Информационно-управляющие системы, 2009, №6, С.34-40.
- [Джарратано и др., 2006] Джарратано Дж., Райли Г. Экспертные системы: принципы разработки и программирование : Пер. с англ./ Дж. Джарратано [и др.]; — М. : Вильямс, 2006, С.779—851
- [Зюзов, 2003] Зюзов В.М. Математическое введение в декларативное программирование/ В.М. Зюзов// 2003. - 83 с.
- [Васильев и др., 2000] Васильев С.Н., Жерлов А.К., Федосов Е.А., Федунко Б.Е. Интеллектуальное управление динамическими системами : монография/ С.Н. Васильев [и др.]; — М.: Физико-математическая литература, 2000. - 352 с.
- [Трофимов, 2003] Трофимов С. Рабочие процессы RUP и диаграммы UML/ С. Трофимов // <http://www.caseclub.ru/>, 2003

[Object Management Group, 2014] UML 2.4.1 / Object Management Group//<http://www.omg.org/spec/UML/2.4.1/>, 2014

[Ивашенко и др., 2004] Ивашенко А.В., Сталькин А.А., Кальшенко У.М. Применение методологии UML при автоматизации управления бизнес-процессами/ А.В. Ивашенко, А.А. Сталькин, У.М. Кальшенко //Исследовано в России : эл. журнал Самарского ГАУ <http://zhurnal.apc.relarn.ru/articles/2004/>, 2004

[Медведев, 2005] Медведев В.И. Программирование на C++,C++.NET/C#:учеб. пособие.- Казань: Мастер Лайн, 2005. - 270 с.

TECHNOLOGY OF AUTOMATED SYNTHESIS THE INFORMATION SYSTEMS USING SEMANTIC MODELS OF SUBJECT AREA

Bikmullina I.I.

*Kazan National Research Technical University
named after A. N. Tupolev (KAI), Kazan, Russia*

elsiyar-b@yandex.ru

The paper considers the problem of the lack of automated structural synthesis of information systems based on the semantic relations of subject area. We propose an artificially intelligent approach to automated information systems development, improvement of modern technology at the expense of automated synthesis the UML class diagrams on based on semantic models.

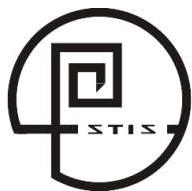
Introduction

Creating the application software (software) is a complex process. The most critical part of him is the directly programming the application, so as on this stage used the mental image the concepts of the application subject area in the programming language, and this requires knowledge as of the language a subject area, as the programming language. The problem of the study is the need to automate the development process (synthesis) class diagrams UML. Therefore, the significant objective of this study is improving the quality of the design study of information system on account of automating the process of synthesis of structural models. From the set of structural models of information technology chosen model class diagram.

Conclusion

The paper presents the improvement of modern information systems development technology through the automated synthesis of UML class diagrams based on semantic models.

Automation of synthesis of a class diagram makes it possible to reduce the complexity of the development of structural models of information systems, improve the quality of software project due to the possibility of multivariate analysis of structural models, allow us to stimulate the developer on preliminary thinking through study the structure of the system, allow us to change the nature labor of the programmer in the direction from the manual study of UML diagrams in the direction to the choice of the optimal variant from the sets automatically received options.



OSTIS-2015

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822:514

ВЕБ-ИНТЕРФЕЙС ДЛЯ СНЯТИЯ МОРФОЛОГИЧЕСКОЙ МНОГОЗНАЧНОСТИ В КОРПУСЕ ТАТАРСКОГО ЯЗЫКА

Гильмуллин Р.А. Гатауллин Р.Р.,

*Казанский Федеральный университет,
Институт «Прикладной семиотики» Академии Наук Республики Татарстан,
г. Казань, Российская Федерация*

rinatgilmullin@gmail.com

ramil.gata@gmail.com

В работе описывается веб-инструментарий, который предназначен для ручного снятия морфологической многозначности в корпусе татарского языка. Отмечены цели и задачи проекта. Описан основной функционал инструментария.

Ключевые слова: разрешение морфологической многозначности, электронный корпус языка, татарский язык.

Введение

Для решения многих лингвистических задач используются электронные коллекции текстовых документов, так называемые электронные корпуса. При этом наиболее информативными и полезными являются размеченные корпуса, в которых текстовым единицам приписана лингвистическая информация. Обычно под такой информацией подразумеваются морфологические, синтаксические и семантические характеристики языка. Корпус может быть использован как в решении исследовательских задач, так и в разработке прикладных приложений, использующие лингвистические модели языка.

В настоящее время специалистами научного-исследовательского института «Прикладная семиотика» Академии наук Республики Татарстан активно ведутся работы по созданию электронного корпуса татарского языка «Туган тел» [Сулейманов и др., 2011]. В данный момент корпус содержит порядка 40 млн. словоформ. Осуществлена полная автоматическая морфологическая разметка корпуса [Сулейманов и др., 1997].

Недостатком данной автоматической разметки является анализ слова в отрыве от контекста, порождая в виде результата всевозможные варианты разбора слова, только один из которых актуален в конкретном контексте. Таким образом, появляется морфологическая многозначность, разрешение которой является следующим этапом в разметке корпуса.

Согласно анализу данных морфологической разметки, доля слов с морфологической многозначностью в корпусе текстов составляет порядка 31,33% [Хакимов, 2014]. В основном к проблеме разрешения морфологической многозначности в татарском языке относится проблема разрешения функциональной омонимии. Функциональная омонимия – омонимия, когда слова совпадают в написании лишь в определенных формах, являясь при этом разными частями речи.

Как показывает анализ существующих методов, проблема снятия морфологической многозначности решалась исследователями разными способами. Первые алгоритмы были основаны на правилах. Позже для решения задачи элиминации многозначности начали применяться статистические алгоритмы. У каждого подхода есть как преимущества, так и недостатки. В условиях татарского языка применение лишь одного подхода не представляется до конца возможным, вследствие чего предлагается применять некий сплав методов, перекрывая недостатки одного метода преимуществами другого [Гатауллин и др., 2014] [Гатауллин, 2014][Зинькина и др., 2005].

Как для статистического подхода нужна обучающая выборка, так и для разработки контекстных правил необходима вручную размеченная часть корпуса со снятой многозначностью для последующего анализа. Вследствие чего было принято решение о разработке инструментария для удобной разметки корпуса, которая в первую очередь будет использоваться как обучающая выборка для

ID	Название	Всего	Проверено	Ошибочных
[ID=817]	N PART	764	0	0
[ID=1058]	N V	331	0	0
[ID=2489]	N PS V	622	0	0
[ID=4038]	N V+ADV+ACC	909	0	0
[ID=1138]	NTM PS AB	116	0	0
[ID=3717]	PART V V+PASS	250	0	0
[ID=1137]	N PS Pres_Sing	743	0	0
[ID=4867]	N PS Pres_Sing	1000	0	0
[ID=4346]	NTM+POSS_SNG+ACC POST	1004	0	0
[ID=491]	Adv N PART POST	38	0	0
[ID=10328]	Adv+POSS_SNG+ACC Adv V V+REFL	1001	0	0
[ID=282]	N AB	877	0	0
[ID=1042]	N V+POSS_SNG	432	0	0

Рисунок 3 – Окно «Моя статистика» отображает полную статистику по разрешениям для данного конкретного пользователя.

Второй режим предполагает разрешение многозначностей по предложениям. Данная функция больше приурочена к подготовке полностью размеченного корпуса со снятыми многозначностями.

Также каждый зарегистрированный пользователь может следить за результатами своих разрешений во вкладке «Моя статистика» (см. Рис.3), а также сравнивать свои результаты с результатами других пользователей, тем самым включается соревновательный момент, немаловажный пункт в «краудсорсинговых» (англ. crowdsourcing, crowd — «толпа» и sourcing — «использование ресурсов») приложениях.

Поиск по словоформе

Результаты поиска по словоформе "алма"

Текст	Номер предложения	Предложение
A_Проект_2824.txt	37449	Видим ли Урмане юкче узак, шери галча кулч терелен теш дэричмеш, шери дэричмешле кылае кылаем, [https://www.youtube.com/watch?v=...]
A_Проект_2824.txt	37489	Диз юкче ала шери аманаклар ар шеришле ар башкалар таште шери, юкче, [https://www.youtube.com/watch?v=...]
A_Проект_2824.txt	37834	Минем [https://www.youtube.com/watch?v=...], шери шеришле аманаклар кулч, шеришле шеришле аманаклар кулч, шеришле кулч, шеришле шеришле аманаклар кулч шеришле аманаклар кулч шеришле аманаклар кулч.
A_Проект_2824.txt	37842	Кылае болып, кылае шеришле шеришле аманаклар кулч [https://www.youtube.com/watch?v=...], шеришле, кылае аманаклар.

Рисунок 4 – Корпус-менеджер. Окно поиска с вкладками «Поиск по словоформе», «Поиск по лексеме», «Поиск по морфеме».

Из дополнительного функционала можно отметить функцию довольно простого корпус-менеджера (см. Рис.4). Реализованы поиск по словоформе, поиск по лексеме, поиск по морфеме и поиск по типу морфологической многозначности.

Словарь

ID [Asc]	Лексема [Asc]	Тип [Asc]	Действия
[ID=7162]	ич дэрт	NTM	Найти в корпусе >>
[ID=23864]	ичмак	N	Найти в корпусе >>
[ID=24170]	ичбармак	N	Найти в корпусе >>
[ID=23256]	ичтыш	N	Найти в корпусе >>
[ID=27744]	ичтеп	POST	Найти в корпусе >>
[ID=396]	ичтепелен	Adv	Найти в корпусе >>
[ID=4648]	ичтепелеш	Adv	Найти в корпусе >>
[ID=473]	ичтепелеш	Adv	Найти в корпусе >>
[ID=11323]	ичтепеш	N	Найти в корпусе >>
[ID=391]	ичтышмак	Adv	Найти в корпусе >>
[ID=391]	ичтышмак	V	Найти в корпусе >>

Рисунок 5 – Корпус-менеджер. Окно словаря. Реализован «поиск по лексеме».

Также на сайте присутствует информация о типах многозначностей, примеры по ним, и

количество экземпляров данного типа, представленного в корпусе татарского языка (см. Рис.6). Данная информация может быть полезна исследователям, занимающимся данной проблематикой.

Типы многозначных разборов

ID [Asc]	Тип [Asc]	Примеры	Количество [ссылка]	Действия
[ID=712]	V+PCP_PS V+PST_DDEF	кайтерелен, ташкызын, чалаткан	533272	Найти в корпусе >>
[ID=2489]	N PS V	бу, бура	232386	Найти в корпусе >>
[ID=8867]	N PS Pres_Sing	ул	232346	Найти в корпусе >>
[ID=1038]	N V	бул, кыя, ту	193324	Найти в корпусе >>
[ID=9469]	V+PST_DDEF V	иде	174759	Найти в корпусе >>
[ID=93]	V+PST_DDEF+DIR V+DIF_L V+PCP_PST+DIR	сөләмәтләр, сөләмәтләр, арманаклар	173922	Найти в корпусе >>
[ID=1138]	NTM PS AB	бура	166758	Найти в корпусе >>
[ID=1237]	V AB	кызыр, кулч, чалч	162388	Найти в корпусе >>
[ID=32]	N+ACC+POSS_SNG N+GEN	турыбашканы, курайшы, тушканы	143923	Найти в корпусе >>
[ID=811]	N PART	кызыр, бура, бу	123087	Найти в корпусе >>

Рисунок 6 – Корпус-менеджер. Окно «Типов морфологических многозначностей». Реализован «поиск по типу многозначности».

Данный инструментариий разрабатывался, используя открытые программные обеспечения, такие как СУБД Postgresql для управления базами данных и веб-фреймворк Django (язык программирования Python) для серверной части инструмента.

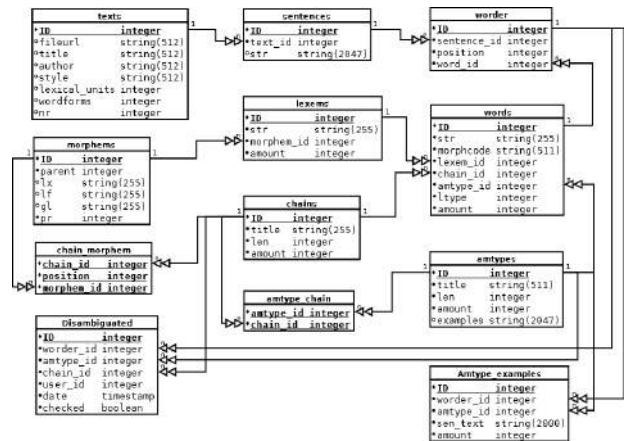


Рисунок 7 – Структура базы данных корпуса.

Заключение

В настоящее время разработан веб-интерфейс, который реализован в виде краудсорсинговой платформы с возможностью привлечения большого количества участников проекта для снятия морфологической многозначности в корпусе татарского языка. На сайте зарегистрировано 32 пользователя, которые являются студентами отделения татарской филологии и межкультурной коммуникации им. Г.Тукая Института филологии и межкультурной коммуникации Казанского Федерального университета. За пять месяцев функционирования сайта суммарное количество разрешенных контекстов достигло 97471, из них уникальными являются 29768 контекста, классифицированных по 22 типам морфологическим многозначностям.

Следующим этапом работ является развитие веб-интерфейса за счет интеграции контекстных правил для автоматического снятия морфологической

многозначности [Гатауллин и др., 2014а], с возможностью непосредственного тестирования правил на корпусных данных.

Необходимо также продвижение проекта через социальные сети для привлечения большего количества участников.

Библиографический список

[Сулейманов и др., 2011] Сулейманов, Д.Ш. Корпус татарского языка: концептуальные и лингвистические аспекты / Д.Ш. Сулейманов, Б.Э. Хакимов, Р.А. Гильмуллин // Вестн. ТГГПУ. – 2011. - № 4 (26). – С.211-216.

[Сулейманов и др., 1997] Сулейманов, Д.Ш. Двухуровневое описание морфологии татарского языка / Д.Ш. Сулейманов, Р.А. Гильмуллин // Тезисы Международной научной конференции "Языковая семантика и образ мира". Казань: Изд-во Казан. гос. ун-та., 1997. Книга 2. С. 65-67.

[Хакимов и др., 2014] Хакимов, Б.Э. Разрешение грамматической многозначности в корпусе татарского языка / Б.Э. Хакимов, Р.А. Гильмуллин, Р.Р. Гатауллин // Ученые записки Казанского университета (в печати).

[Гатауллин и др., 2014а] Гатауллин Р.Р. Программный инструмент для разрешения морфологической многозначности в татарском языке / Р. Р. Гатауллин, Д. Ш. Сулейманов, Р. А. Гильмуллин // Открытые семантические технологии проектирования интеллектуальных систем OSTIS-2014 Open Semantic Technologies for Intelligent Systems МАТЕРИАЛЫ IV МЕЖДУНАРОДНОЙ НАУЧНО-ТЕХНИЧЕСКОЙ КОНФЕРЕНЦИИ (Минск, 20-22 февраля 2014 года), - Минск. : БГУИР, 2014. - С. 503-508

[Гатауллин, 2014б] Гатауллин Р. Р. Веб-инструментарий для снятия морфологической многозначности в текстовом корпусе татарского языка / Р. Р. Гатауллин // Сохранение и развитие родных языков в условиях многонационального государства: проблемы и перспективы: материалы V Международной научно-практической конференции (Казань, 19-22 ноября 2014 г.). – Казань: Отечество, 2014. - С. 71-73

[Зинькина и др., 2005] Ю.В. Зинькина, Н.В. Пяткин, О.А. Невзорова, Разрешение функциональной омонимии в русском языке на основе контекстных правил. // Труды междунар. конф. Диалог'2005. – М.: Наука, 2005. С. 198-202.

[Бочаров и др., 2011] Бочаров, В. В. Программное обеспечение для коллективной работы над морфологической разметкой корпуса / В. В. Бочаров, Д. В. Грановский // Труды международной конференции «Корпусная лингвистика – 2011». 27–29 июня 2011 г., Санкт-Петербург. — СПб.: С.-Петербургский гос. университет, Филологический факультет, 2011.

WEB-SITE FOR HANDY MORPHOLOGICAL DISAMBIGUATION IN TATAR LANGUAGE CORPUS

Gilmullin R.R., Gataullin R.R.

Research Institute of Applied Semiotics of the Tatarstan Academy of Sciences, Kazan Federal University, Kazan, Russia

rinatgilmullin@gmail.com

ramil.gata@gmail.com

Paper presents the easy in use instrument for handy elimination of morphological ambiguities in Tatar language corpus. Talks about main tasks. Shows basic functionalities.

Introduction

Language corpora are very useful informational structures in NLP. For successful uses it must be correctly annotated, including morphological, syntactical and semantical features.

For last years, scientists from Research Institute of Applied Semiotics of the Tatarstan Academy of Sciences have been developing Tatar language corpus, which contains by these days more than 40 million word usages. Morphological features are automatically annotated, but the problem of morphological ambiguity has not been solved yet. For the further researches it is necessary to get the part of corpus, annotated by hand. In the end it will be used to teach the automatic morphological disambiguator based in machine learning.

Main Part

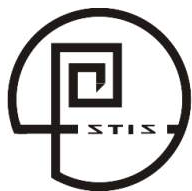
Since the corpus contains very big amount of texts, it is necessary to involve into disambiguation process as many people as possible. Nowadays, it is much easier, if we will use Internet technologies for this task. So, the purpose was to develop the easy in use web-site, where the every could help to annotate the corpus.

As a result, the instrument functions now. Except the main functionality of handy disambiguation, simple corpus manager is also developed. And it is available by Internet at <http://tatcorp.antat.ru/>.

Conclusion

For 5 months of functioning, 32 users had been registered, most of them attends Kazan Federal university, and 29768 unique context had been disambiguated.

Future plan is a promotion of the project through social social network.



OSTIS-2015

(Open Semantic Technologies for Intelligent Systems)

УДК 519.816

МНОГОКРИТЕРИАЛЬНЫЙ ПОДХОД К ФОРМИРОВАНИЮ ПОЛИТИКИ НАУЧНОГО ФОНДА И ЭКСПЕРТНОГО СООБЩЕСТВА

Бойченко В.С.*, Заболева-Зотова А.В.**, Петровский А.Б.***

** Российский гуманитарный научный фонд, Москва, Россия*

boychenko@rfh.ru

*** Российский фонд фундаментальных исследований, Москва, Россия*

zabzot@rfr.ru

**** Институт системного анализа РАН, Москва, Россия*

pab@isa.ru

В работе предлагается методология формирования политики научного фонда и экспертного сообщества, основанная на применении методов группового вербального анализа решений.

Ключевые слова: научная политика; научные фонды; экспертное сообщество; многокритериальный подход; групповой вербальный анализ решений.

Введение

Основным направлением деятельности государственных научных фондов является конкурсный отбор заявок на проведение исследований. Конечным результатом конкурсного цикла является выделение из всего множества поступивших заявок проектов, принимаемых к финансированию. В соответствии с уставами фондов, все правила и процедуры, регламентирующие их деятельность, определяются главной целью – максимально стимулировать инициативу и творческую активность учёных. Обеспечение качества экспертизы, напрямую связанного с прозрачностью и объективностью конкурсных процедур, определено сегодня на государственном уровне как важнейшее требование к работе фондов.

Вместе с тем экспертиза в существующем виде не обеспечивает необходимого уровня обоснованности выбора проектов в соответствии с последовательной профессиональной политикой. «Обоснованность» является синонимом «эффективности» и должна рассматриваться как исчерпывающая характеристика качества работы фондов, как необходимое условие выстраивания отношений между государством и научным сообществом, основанных на взаимных интересах и

ответственности. Основная причина неадекватности сложившейся практики экспертизы, по нашему мнению, в недостаточном аналитическом, методическом и информационном обеспечении процесса рассмотрения поступивших заявок, ограниченности инструментария, используемого для обобщения и интерпретации экспертных заключений, формирования совокупности проектов, удовлетворяющих заданным научным и ресурсным ограничениям.

Для анализа больших массивов экспертных оценок и их интерпретации существуют специальные методические процедуры, которые, в частности, позволяют отбирать объекты в соответствии с априори заданными требованиями к их характеристикам. Это особенно актуально для научных фондов, куда поступает тысячи заявок на гранты. Необходимо упорядочить и структурировать процедуры экспертизы, способы выявления мнений экспертов и интерпретации их оценок, встроить их в единую систему, которая позволила бы органу, принимающему решения, задавать свои предпочтения (политику) и обеспечивать отбор конкретных проектов из разных областей науки, в максимальной степени соответствующих этим предпочтениям.

В работе рассматривается методология формирования политики научного фонда,

предпочтений экспертного совета фонда и коллективных предпочтений экспертного сообщества, основанная на применении методов группового вербального анализа решений.

1. Особенности конкурсного отбора проектов

Предлагаемый подход к выбору совокупности проектов, рекомендуемых к финансированию, из исходного множества рассмотренных заявок основан на использовании формализованных процедур группировки заявок с учетом предпочтений руководства, экспертных советов и экспертов фонда. При разработке подхода учитывались условия проведения конкурсов в фондах, специфика их организации и информационного обеспечения:

- число поступающих на конкурс заявок измеряется многими сотнями, что исключает возможность их непосредственной сравнительной оценки;

- существует возможность унифицированного по структуре представления всех рассматриваемых заявок, отражающего основные аспекты предлагаемых исследований, используется единый формат экспертных заключений;

- основанием для выбора проектов являются индивидуальные оценки независимых экспертов – единственного источника компетенции в фундаментальной науке;

- для рассмотрения экспертных заключений и подготовки предложений по перечню проектов, предлагаемых к финансированию с учётом дополнительных условий (например, ресурсных ограничений), предусмотрен второй уровень экспертной инфраструктуры – экспертные советы с включёнными в их состав секциями, сфера ответственности и полномочия которого чётко регламентированы;

- однозначно определены типы решений по итогам конкурсов, предложения для которых подготавливаются в ходе экспертизы.

Подход строится на следующих методических посылках.

Для фондов типичны ситуации, когда из тысяч заявок необходимо выбрать ограниченное число «наиболее достойных». Структуры, подготавливающие предложения по отбору проектов (экспертные советы), и орган, принимающий окончательное решение по итогам конкурса (совет Фонда), не в состоянии детально и содержательно проанализировать всё множество заявок. Их роль, по сути, сводится к утверждению рекомендаций экспертов. Тем самым окончательный выбор во многом предопределён уже на первой стадии экспертизы. Этим обусловлена объективная потребность в средствах формализованной группировки больших массивов заявок по различным наборам их признаков (характеристик) и представления информации,

степень агрегации и содержание которой определяется уровнем и характером принимаемых решений.

При использовании единой системы критериев оценки заявок в различных областях науки, достаточно подробно и всесторонне отражающих наиболее существенные аспекты предлагаемых исследований, совокупность экспертных оценок с развёрнутой аргументацией эксперта в пользу выбранного им «уровня качества» по каждому из критериев является унифицированным «информационным портретом» проекта, описывающим наиболее существенные для принятия решений аспекты (характеристики) и дающим возможность автоматизировать структурирование массивов проектов любой размерности.

Между рекомендациями эксперта «поддержать» или «отклонить» проект и наборами оценок по критериям существует определенная зависимость, которую можно выявить с помощью специально разработанных средств. Для фундаментальных исследований, к которым относится большинство поступающих в фонды заявок, характерна высокая степень неопределённости перспективности проектов и вида конечного результата. Выводы экспертов в значительной мере основаны на интуиции и общем впечатлении о рассматриваемых заявках и не могут быть полностью обусловлены оценками по критериям, которые, как и любая модель, не в состоянии отразить все нюансы столь сложных объектов. Поэтому существующая зависимость рекомендации эксперта от оценок по критериям не носит характера взаимно-однозначного соответствия. Всегда существует фактор непредсказуемости, который ограничивает возможности формализовать экспертные процедуры.

Таким образом, в процессе экспертизы изначально заложены два противоречия: между однозначным характером принимаемого решения (принять или отклонить проект) и неформализуемым восприятием его экспертом, а также между большой размерностью массива заявок и необходимостью их содержательного анализа. Для методически корректного преодоления этих противоречий в предлагаемом подходе предусмотрено сочетание инструментов формализованной обработки больших массивов экспертных оценок и содержательной интерпретации их комбинаций, не поддающихся формализации.

Рассмотрим способы формализованного отбора проектов, использующие их описание в виде наборов оценок по критериям и предусматривающие формирование на этой основе различных схем представления предпочтений руководства фонда и экспертного сообщества, в соответствии с которыми и осуществляется отбор.

2. Формирование научной политики

Научная политика фонда формируется руководством и экспертными советами фонда, которые представляют научное сообщество в лице ведущих учёных, обладающих не только профессиональными знаниями в отдельных областях и широкой научной эрудицией, но и опытом руководства научными коллективами и принятия решений.

Научная политика задаётся в виде наборов вербальных характеристик проектов по многим критериям и является, по сути, комплексной многоаспектной нечисловой оценкой проекта, аналогичной по смыслу общему или интегральному «баллу». При использовании порядковых вербальных шкал более высокая оценка по каждому критерию означает более высокий уровень соответствующего качества, что даёт ряд преимуществ по сравнению с использованием единственного интегрального числового показателя, в котором «смешаны» количественные оценки по многим частным критериям.

Требования к проектам задаются в достаточно общих и универсальных категориях, применимых для оценки исследований в разных областях и адекватных восприятию основных аспектов состояния науки на высших уровнях принятия решений. Поэтому формирование единой политики на достаточно широком спектре дисциплин не требует знания сугубо специальных вопросов, изучаемых в рамках конкретных проектов. Это – задача экспертов в конкретных направлениях, которые интерпретируют содержание отдельных заявок, задавая оценки по критериям.

Имеется возможность достаточно гибко варьировать как общие требования, так и акценты в рамках отдельных областей, различных видов конкурсов. Очевидно, что «научная значимость результата», «актуальность темы», «комплексность предлагаемого исследования» необходимы для заявок практически во всех областях. Но в различных науках могут быть расставлены дополнительные акценты, например, в отдельных направлениях сегодня важны работы методологического плана, в то время как в других наиболее интересны и востребованы новые концептуальные подходы при описании изучаемых явлений. В случае необходимости возможен итеративный процесс корректировки политики путём изменения комбинаций требований.

Предлагаемая схема в сочетании с тематическими приоритетами (принадлежность к заданной тематической области может рассматриваться как дополнительное требование) может служить полезным методическим инструментом, позволяющим формализовать с минимальными «потерями содержания» политику фондов и выявить в исходном массиве те заявки, которые в наибольшей степени отвечают этой политике. Процедура отбора, основанная на

сопоставлении оценок проектов с априори заданными требованиями, обеспечивает прозрачность и логичность принимаемых решений. Становится более ясным, по каким критериям конкретный проект соответствует требованиям, а по каким отклоняется.

Предпочтения экспертных советов фонда формируются ежегодно до проведения экспертизы заявок в виде набора требований на шкалах критериев. Такие требования, отражающие представление о необходимом уровне качества («научном стандарте») проектов, достойных поддержки, определяются на основе анализа состояния и тенденций развития исследований в соответствующих областях науки и представляют собой своеобразный «эталон», с которым сравниваются экспертные оценки поступивших заявок. Заявки, оценки которых по критериям равны или превышают исходные требования, признаются заслуживающими поддержки.

Разбиение исходного множества проектов на два подмножества может быть осуществлено путём задания некой «пограничной» комбинации оценок: проекты, у которых оценки выше или ниже заданных уровней, составляют, соответственно, первое и второе подмножества. При этом каждый проект из первого подмножества «лучше» любого проекта из второго.

Значения пограничной комбинации оценок образуют нижний порог качества, допустимый для поддержки проектов («порог поддержки») и могут интерпретироваться как набор требований, отражающий предпочтения органа, принимающего решения: удовлетворение этим требованиям служит необходимым исходным условием принятия заявки. Такое формальное разбиение, естественно, не может быть однозначно верным и окончательным. Во-первых, оценки экспертов субъективны и, во-вторых, любая формализация в сфере фундаментальной науки не учитывает многие содержательные нюансы конкретных проектов, которые невозможно отразить на шкалах критериев. Однако можно утверждать, что в число отобранных по данной схеме проектов с большой вероятностью попадут именно те проекты, которые по совокупности признаков, характеризующих научный уровень и отражённых в системе критериев, превосходят проекты, получившие оценки, не соответствующие заданным требованиям.

Коллективные предпочтения экспертного сообщества для отбора проектов формируются с использованием комбинаций экспертных оценок по критериям путём сопоставления с заданными требованиями.

Согласно регламенту проведения экспертизы заявок по ряду конкурсов и РФФИ, и РГНФ, эксперт наряду с оценением отдельных аспектов каждой заявки должен высказать своё мнение о целесообразности поддержки или отклонения

проекта. Явно выраженная «результатирующая рекомендация» эксперта фактически устанавливает зависимость между комбинацией оценок соответствующего проекта по критериям и его отнесением (согласно рекомендации эксперта) к одному из классов, заслуживающих поддержки или отклонения. При такой постановке задачи выбора используется в определённом смысле «обратная» логика: пороговая комбинация оценок определяется не относительно к конкретным проектам в виде некоторого заранее заданного «эталона», единого для всех заявок для данного вида конкурсов, а складывается у эксперта применительно к конкретному проекту в процессе его оценивания.

Агрегированные определённым образом индивидуальные, в том числе и противоречивые рекомендации экспертов в совокупности с пороговыми комбинациями вербальных оценок по критериям представляют «решающие правила», которые, по мнению убедительного большинства экспертов, достаточны для поддержки или отклонения проектов. Такие решающие правила можно рассматривать как выражение консолидированных предпочтений высококвалифицированных исследователей, привлекаемых к оценке заявок. Их можно также интерпретировать как сформированную «снизу» научную политику многочисленной группы экспертов. В результате будут получены альтернативные подмножества проектов, рекомендуемых и не рекомендуемых для поддержки. Для решения подобной задачи может быть использован метод группового вербального анализа решений [Петровский, 2009], [Petrovsky, 2008].

3. Построение решающих правил для отбора проектов

Приведем в качестве примера процедуру, которая позволяет сформировать предпочтения экспертного совета фонда в виде набора решающих правил для рекомендаций по поддержке и отклонению конкурсных заявок. Процедура охватывает два последовательных этапа процесса экспертизы: многокритериальную экспертную оценку поступивших заявок и подготовку на основе экспертных заключений предложений по перечню заявок, рекомендуемых для поддержки фондом.

На первом этапе проводится оценка заявок по критериям, которые отражают различные аспекты предлагаемой проблемы и исследования (научная значимость, новизна, актуальность, и пр.) и основные составляющие потенциала исполнителей, определяющего возможность получения заявленного результата (научная квалификация, аналитический инструментарий, который предусматривается использовать при проведении исследования и т.п.). Каждый критерий имеет порядковую шкалу, состоящую из трёх градаций, которые описывают последовательно убывающие

уровни «качества» по данному критерию, например, «научной новизны»: высший (оценка «а»), средний (оценка «б»), низший (оценка «в»), означающий отсутствие данного аспекта. Для каждой градации позиции на шкалах всех критериев приводятся развёрнутые формулировки, раскрывающие смысловую трактовку соответствующего уровня качества.

В соответствии с регламентом конкурса каждая заявка рассматривается тремя экспертами. Эксперт отмечает на шкале каждого критерия градацию, в наибольшей степени соответствующую, по его мнению, уровню качества рассматриваемой заявки по данному критерию. Кроме того, эксперт сопровождает каждую проставленную оценку развёрнутым комментарием с аргументацией выбора именно такой оценки. Помимо аргументированных оценок эксперт пишет сводное заключение, в котором обосновывается одна из трёх рекомендаций:

«А» – проект безусловно поддержать;

«Б» – проект поддержать при определённых условиях (например, при выделении дополнительных средств для данного научного направления);

«В» – проект отклонить.

Экспертные заключения, включающие набор из оценок эксперта по критериям с комментариями и сводное заключение с рекомендуемым экспертом решением, поступает в «Логический фильтр 1», где выявляются «нелогичные» сочетания оценок по критериям, не согласующиеся с рекомендацией эксперта (рис.1). Примерами могут служить следующие сочетания:

- оценка «б» по всем критериям не сочетается с рекомендацией «А»;

- оценка «в» хотя бы по одному критерию не сочетается с рекомендацией «А»;

- рекомендация «А» возможна только, если по критерию «Методика исследования» выбрана оценка «а», а по остальным критериям оценка «б»;

- оценка «в» по критерию «Методика исследования» при любом сочетании оценок по другим критериям с полным основанием влечёт рекомендацию «В»;

- оценка «в» по трём и более критериям с полным основанием влечёт рекомендацию «В».

- рекомендация «В» нелогична, если ни по одному критерию не проставлена оценка «в».

Такие «нелогичные» сочетания задаются при рассмотрении экспертных заключений, поступающих в фонд, и обсуждаются с экспертами с целью исключения технической ошибки или недоразумения. Если эксперт настаивает на своём мнении, заявка передаётся в секцию соответствующего экспертного совета для содержательного рассмотрения и принятия решения.

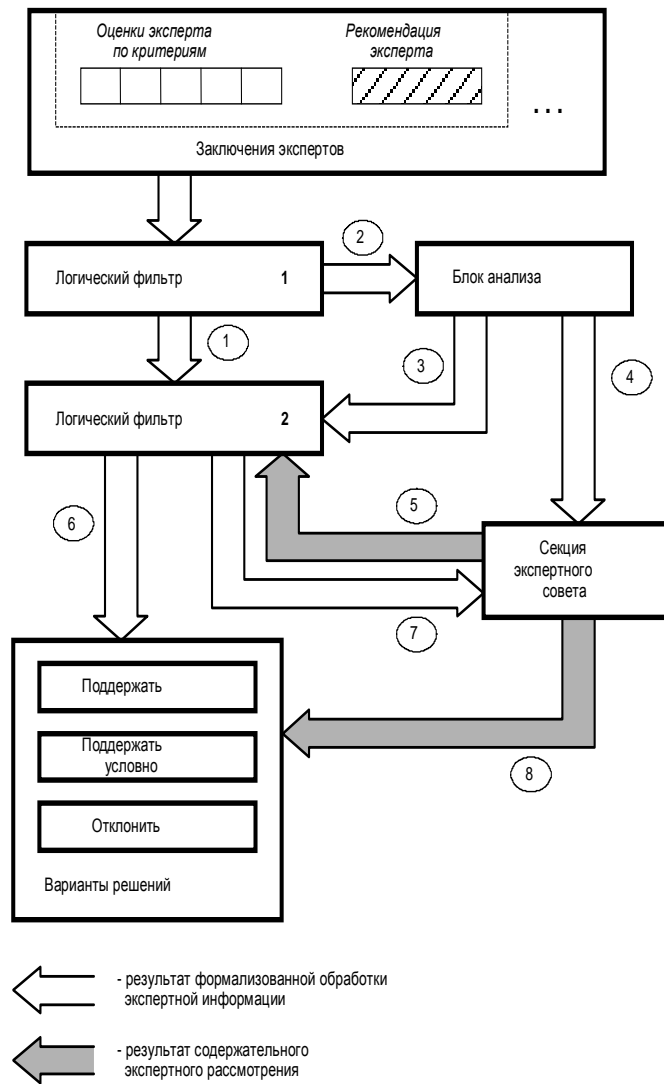


Рисунок 1 - Процедура формирования предпочтения экспертного совета фонда

Обозначения:

1. Экспертные заключения, в которых рекомендация эксперта логически соответствует набору оценок по критериям.
2. Экспертные заключения, в которых нарушена логика соответствия рекомендации эксперта и набора оценок по критериям.
3. Экспертные заключения, в которых рекомендации эксперта приведены в соответствие с набором оценок по критериям для устранения технической ошибки или недоразумения.
4. Экспертные заключения, в которых рекомендации логически не соответствуют набору оценок по критериям, но подтверждены экспертом.
5. Экспертные заключения, в которых рекомендации эксперта содержательно рассмотрены и утверждены членами экспертного совета.
6. Экспертные заключения, которые могут быть отнесены к одному из вариантов решений в соответствии с комбинацией заключений трёх экспертов.
7. Экспертные заключения, которые не могут быть отнесены к одному из вариантов решений из-за логического противоречия в комбинации заключений трёх экспертов.
8. Экспертные заключения, отнесённые к одному из вариантов решений в результате содержательного рассмотрения членами экспертного совета.

На втором этапе на основе комбинации трех экспертных заключений формируются обобщающие рекомендации – решающие правила: П₁. «Поддержать проект», П₂. «Поддержать проект условно», П₃. «Отклонить проект». В «Логическом фильтре 2» задаются комбинации экспертных заключений, которые считаются обоснованием для отнесения соответствующих заявок к одной из выделенных категорий.

Можно выделить несколько типов таких комбинаций заключений, задающих решающие правила для отбора проектов, например:

- все три эксперта единодушны в своих заключениях (комбинации ААА, БББ и ВВВ), отнесение заявок к соответствующим категориям очевидно;

- в комбинации есть две одинаковы рекомендации (ААБ, ААВ, АББ, ББВ, АВВ, ВВВ), тогда применяется правило большинства голосов и «обобщающей» считается рекомендация, выбранная двумя экспертами из трёх.

В результате формируются классы проектов, получивших рекомендации «А», «Б» и «В».

Класс П₁. Поддерживаются заявки, получившие комбинации рекомендаций экспертов: ААА и ААБ.

Класс П₂. Условно поддерживаются заявки, получившие комбинации рекомендаций экспертов: БАА и ББА.

Класс П₃. Отклоняются заявки, получившие комбинации рекомендаций экспертов: АВВ и ВВВ.

В отдельный класс П₄ попадают заявки, которые не могут быть формально отнесены к одному из перечней, имеющие комбинации заключений экспертов: АБВ, ААВ, ББВ. Две последние комбинации подпадают под формальную классификацию, но рекомендация «В» одного из экспертов исключает возможность автоматического отнесения соответствующих заявок к первому или второму классам. Заявки, получившие такие комбинации заключений, требуют индивидуального профессионального рассмотрения.

Заключение

Предложенная процедура может служить полезным методическим инструментом конкурсного отбора проектов в научных фондах. Её корректное использование будет способствовать обеспечению прозрачности экспертных процедур и, соответственно, обоснованности принимаемых решений.

Благодарности

Работа частично поддержана Российским гуманитарным научным фондом (проект 15-03-12014), Российским фондом фундаментальных исследований (проекты 14-07-00916, 14-29-05025).

Библиографический список

[Панченко, 2012] Панченко В.Я. Фундаментальные задачи фундаментальной науки // В мире науки. 2012. №1. С 4-10.

[Петровский, 2009] Петровский А.Б. Теория принятия решений. – М.: Образовательно-издательский центр «Академия», 2009.

[Петровский и др, 2010] Петровский А.Б., Ройзензон Г.В., Тихонов И.П., Балышев А.В. Применение метода «МАСКА» для групповой экспертной классификации научных проектов по многим критериям // Natural and Artificial Intelligence / Ed. by K. Markov, V. Velychko, O. Voloshin. No. 17. – Sofia: ITNEA, 2010. Pp. 56–67.

[Петровский, Лобанов, 2014] Петровский А.Б., Лобанов В.Н. Многокритериальный выбор в пространстве признаков большой размерности: мультиметодная технология ПАКС-М // Искусственный интеллект и принятие решений, 2014. № 3. С.92-104.

[Петровский, Тихонов, 2009] Петровский А.Б., Тихонов И.П. Фундаментальные исследования, ориентированные на практический результат: подходы к оценке эффективности // Вестник РАН. 2009. Том 79. №11. С.1006-1011.

[Фридлянов, 2014] Фридлянов В.Н. Отвечая на вызовы времени. // Вестник Российского гуманитарного научного фонда. 2014. № 3(76). С. 5-22.

[Petrovsky, 2008] Petrovsky A. Group Verbal Decision Analysis // Encyclopedia of Decision Making and Decision Support Technologies. – Hershey: IGI Global, 2008. P.418-425.

MULTICRITERIA APPROACH TO FORMING POLICY OF SCIENTIFIC FOUNDATION AND EXPERT COMMUNITY

Boychenko V.S. *, Zaboleeva-Zotova A.V. **,
Petrovsky A.B. ***

* *Russian Foundation for Humanities,
Moscow, Russia*
boychenko@rfh.ru

** *Russian Foundation for Basic Research,
Moscow, Russia*
zabzot@rfbr.ru

*** *Institute for Systems Analysis RAS,
Moscow, Russia*
pab@isa.ru

This paper proposes a methodology of forming policy of scientific foundation and expert community, based on the use of methods of group verbal decision analysis.
Keywords: scientific policy; scientific foundation; expert community; multicriteria approach; group verbal decision analysis.



OSTIS-2015

(Open Semantic Technologies for Intelligent Systems)

УДК 621.311:658.26

ОНТОЛОГИЧЕСКИЙ ПОДХОД ПРИ ПРОЕКТИРОВАНИИ ТЕХНОЛОГИЧЕСКИХ ПРОЦЕССОВ

Бурдо Г.Б., Воробьева Е.В.

Тверской государственной технической университет

г. Тверь, Россия

gbtms@yandex.ru

evgeniya.vor813@gmail.com

В данной статье рассмотрена проблема создания системы автоматизированного проектирования технологического процесса (САПР ТП). Выявлена и обоснована необходимость построения САПР ТП с использованием онтологического подхода при проектировании. Разработаны и приведены структурированные схемы предметной онтологии детали и технологического процесса изготовления детали. Выявлен перечень функций онтологии задач при проектировании технологического процесса изготовления детали. Показана возможность использования семантических сетей при онтологическом подходе проектирования.

Ключевые слова: автоматизированное проектирование технологического процесса изготовления детали; развивающаяся база знаний; онтологии предметной области и онтологии задач; семантические сети.

Введение

В современном машиностроительном предприятии можно выделить такую разновидность, как многономенклатурное производство. Использование систем автоматизированного проектирования (САПР) различного уровня позволяет более оперативно осуществлять технологическую подготовку производства. Однако, одного данного направления развития недостаточно для помощи инженеру в процессе конструирования.

Обычно на многономенклатурном предприятии имеется огромное количество информации, относящейся к технологическому процессу изготовления детали, однако не всегда организован оперативный доступ к информации.

Автоматизация анализа спроектированных ранее технологий и синтез новых правил проектирования технологического процесса (ТП) в условиях конкретного производства является очень актуальным.

Разработка системы обобщения и накопления информации приводит к созданию систем автоматизированного проектирования технологического процесса (САПР ТП) с развивающейся базой знаний.

Онтологический подход

Для решения поставленной задачи необходимо проанализировать все элементы, влияющие на проектирование ТП, определить информационные связи, возникающие между этими элементами при разработке ТП.

Для организации системы накопления и обобщения информации необходимо структурировано представить все элементы, влияющие на проектирование ТП, описать информационные связи между ними и представить необходимые варианты решений задач, которые позволят привести к выбору выгодного в заданных производственных условиях ТП.

Имеется большое количество различных методов и способов описания данных, однако все большую популярность последнее время приобретают онтологии [Гаврилова, 2003].

Существует множество определений в различных областях использования данного термина. Согласно Груберу [Gruber., 1993] понятие онтология имеет следующее определение: онтология – эксплицитная (явная) спецификация концептуализации знаний. [Гаврилова, 2003]

Концептуализация – это структура реальности, рассматриваемая независимо от словаря предметной области и конкретной ситуации.

Некоторые авторы дают понятию онтология определение, в котором описывают способ построения и использования онтологии: онтология – это иерархически структурированное множество терминов, описывающих предметную область, которая может быть использована как исходная структура для базы знаний. [Добров и др., 2009]

Для того чтобы использовать онтологию при создании технологического процесса необходимо ввести еще одно определение – предметная онтология.

Предметная онтология - представление понятий предметной области с описанием отношений между этими понятиями и множеством функций интерпретации, заданных на понятиях и (или) отношениях предметной онтологии [Евгенов, 2009].

Иными словами предметная онтология представляет собой систему понятий, относящихся к определенной предметной области, и отношений между этими понятиями.

Онтологии дают возможность оперативного доступа к информации в рамках предметной области.

Понятие предметная онтология ($O^П$) можно представить моделью [Евгенов, 2009, Гаврилова и др., 2000]:

$$O^П = \langle C, R, F \rangle, \quad (1)$$

где C – конечное множество понятий предметной области, которую представляет предметная онтология ($O^П$), R - конечное множество отношений между понятиями заданной области, F - конечное множество функций интерпретации, заданных на отношениях предметной онтологии ($O^П$).

Предметную онтологию использую при построении семантической сети детали. Семантические сети в свою очередь необходимы для построения интеллектуальных САПР.

Структура онтологии представляется семантической сетью – ориентированным графом, вершинами которого являются понятия, а ребра отражают семантические отношения между понятиями [Кузнецов и др., 2010].

Чтобы построить онтологию детали, необходимо определить множество понятий, относящихся к этой предметной области. Результаты исследования по определению таких понятий приведены в [Бурдо и др., 2014].

С точки зрения технологии изготовления детали такими понятиями будут следующие. Понятия, относящиеся непосредственно к технологическому процессу: технологический процесс, операция, установ, переход, вспомогательный и рабочий ход, режимы резания.

Понятия, относящиеся непосредственно к детали: заготовка, материал, форма поверхностей,

их взаимное расположение, допуск, точность, шероховатость.

Понятия, описывающие техническое оснащение: оборудование, станок, оснастка, приспособление, режущий, вспомогательный и измерительный инструмент.

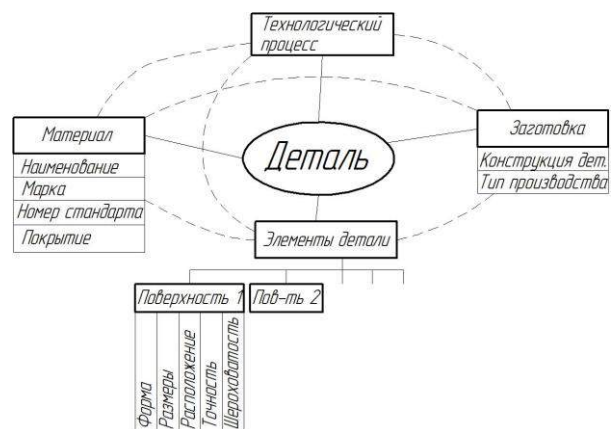


Рисунок 1 – Онтология детали

После определения понятий были определены связи между понятиями (на рисунке 1 и 2 связи между понятиями показаны пунктирными линиями). Следующий этап - была построена онтология детали и онтология технологического процесса (рисунок 1 и 2 соответственно).

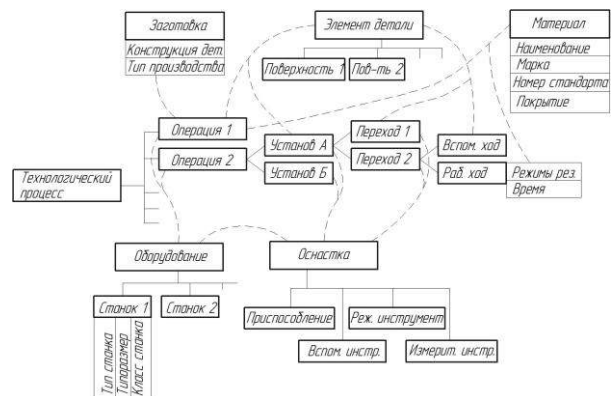


Рисунок 2 – Онтология технологического процесса

Связи между понятиями были выявлены и представлены в статье САПР ТП с развивающейся базой знаний [Бурдо и др., 2014]. Разработанные онтологические модели были построены на основе анализа источников [Кузнецов и др., 2010, Бурдо и др., 2014, Дмитриенко и др., 2013, Нитипатова и др. 2011].

Построение онтологии предметной области недостаточно для полного формирования модели предметной области. В связи определением онтологии (эксплицитная (явная) спецификация концептуализации знаний) вводится еще одно понятие – онтологическая система [Евгенов, 2009].

Модель онтологической системы \sum^O представляется в виде совокупности трех составляющих:

$$\sum^o = \langle O^M, O^П, O^З \rangle \quad (2)$$

где O^M – онтология верхнего уровня, $O^П$ и $O^З$ – множество предметных онтологий и онтологий задач предметной области.

Онтология верхнего уровня представляет собой метаонтологию, которая оперирует общими понятиями и отношениями, которые не зависят от рассматриваемой предметной области.

Метаонтология включает в себе совокупность понятий и отношений между понятиями, необходимые для формирования предметной онтологии, а также онтологии задач.

Совместное взаимодействие предметной онтологии и онтологии задач может обеспечить формирование на их основе операциональной модели [Евгеньев, 2009].

Операциональная модель САПР представляет собой синтез структурированных понятий предметной области и онтологии задач. В данном симбиозе предметная онтология выступает в виде системы данных, отражающих структуру предметной области, а онтология задач выступает в виде элемента, преобразующего эти данные.

Онтология задач представляет собой перечень функций, содержащих смысловое описание понятий, входящих в предметную онтологию.

Для рассматриваемой предметной области онтология задач представляет собой следующий набор функций: анализ материала проектируемой детали, анализ конструктивных элементов детали, выбор метода получения заготовки, разработка технологического процесса изготовления детали.

Разработка технологического процесса изготовления детали раскладывается на следующие функции: формирование последовательности операций, разработка операции, подбор оборудования, разделение операции на установки, выбор оснастки, назначение переходов, расчет рабочих и вспомогательных ходов, расчет режимов резания, расчет основного и вспомогательного времени.

Формирование последовательности операций зависит от вида заготовки и материала, подбор оборудования для выбранной последовательности операций также зависит от обрабатываемого материала детали.

Задача разделения операции на установки применяется при необходимости в зависимости от обрабатываемых поверхностей. Выбор оснастки зависит от конструктивных элементов обрабатываемой детали. Задача назначения переходов зависит как от конструктивных элементов, так и от выбранной ранее оснастки.

Расчет рабочих и вспомогательных ходов производится в зависимости от элементов детали, расчет режимов резания от элементов детали,

материала, выбранного оборудования и оснастки. Расчет основного и вспомогательного времени проводят на основе принятых режимов резания.

Набор выявленных функций онтологии задач представлены в виде схемы (рисунок 3).

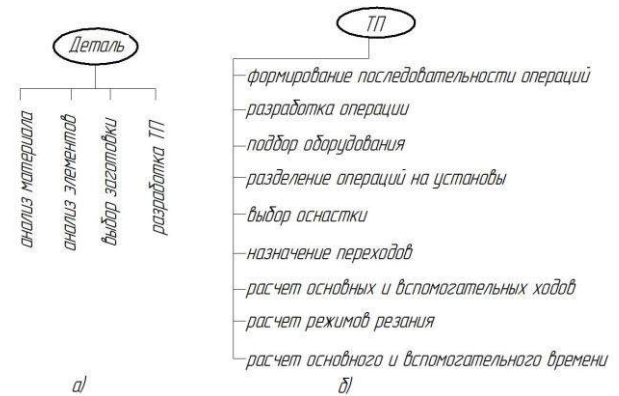


Рисунок 3 – Онтология задач: а) детали, б) технологического процесса

Онтологии делают доступ к информации оперативным, дают возможность лучше понять информацию и произвести ее широкий анализ.

В настоящее время онтологии становятся более распространенными и находят применение в области семантических сетей, управления знаниями и экспертных системах.

Одной из перспективных областей применения онтологий является моделирование и проектирование. В искусственном интеллекте онтологии используются для формальной спецификации понятий и отношений, которые характеризуют определенную область знаний.

Заключение

Онтология детали и технологического процесса представленные визуальным подходом, позволяют наглядно изобразить и объяснить структуру взаимосвязей между понятиями, описывающими деталь с точки зрения технологии изготовления.

Применение онтологического подхода при представлении предметной области позволяет достичь системности представленных понятий и связей между ними.

Представленный онтологический подход может быть использован для разработки новых правил проектирования технологических процессов разного уровня (маршрутной и операционной технологий). На основе выведенных правил и связей может быть построена САПР ТП с развивающейся базой знаний.

Работа выполнена при финансовой поддержке РФФИ, проект № 14-01-00324.

Библиографический список

[Гаврилова, 2003] Гаврилова Т.А. Онтологический подход к управлению знаниями при разработке корпоративных систем

автоматизации / Новости искусственного интеллекта №2, 2003

[Gruber., 1993] Gruber. T.R. A translation approach to portable ontologies. Knowledge Acquisition, 5(2):199-220, 1993.

[Добров и др., 2009] Добров Б.В., Иванов В.В., Лукашевич Н.В., Соловьев В.Д. Онтологии и тезаурусы: модели, инструменты, приложения: учебное пособие. – М.: Интернет-Университет Информационных Технологий; БИНОМ. Лаборатория знаний, 2009

[Евгеньев, 2009] Евгеньев Г.Б. Интеллектуальные системы проектирования: учеб. пособие / Г.Б. Евгеньев. – М. : Изд-во МГТУ им. Н.Э. Баумана, 2009

[Гаврилова и др., 2000] Гаврилова Т.А., Хорошевский В.Ф. Базы знаний интеллектуальных систем. СПб.: Питер, 2000.

[Кузнецов и др., 2010] Кузнецов О.П., Суховеров В.С., Шилина Л.Б. Онтология как систематизация научных знаний: структура, семантика, задачи / Институт проблем управления РАН им. В.А. Трапезникова / Конференция «Технические и программные средства систем управления, контроля и измерения», 2010

[Бурдо и др., 2014] Г.Б. Бурдо, Б.В. Палюх, Е.В. Воробьева. САПР ТП с развивающейся базой знаний/ Открытые семантические технологии проектирования интеллектуальных систем. OSTIS-2014/ Материалы IV международной научно-технической конференции. Минск, 2014

[Дмитриенко и др., 2013] Дмитриенко В.Д., Хавина И.П. Гибридная иерархическая нейронная сеть для хранения знаний технологического процесса механообработки / Вестник НТУ «ХПИ» №39, 2013

[Нитипанова и др., 2011] Нитипанова Г.П., Смирнов С.В. Онтологический анализ предметной области. Задачи базирования детали / Институт проблем управления сложными системами РАН // ПУМСС, 2011

ONTOLOGICAL APPROACH THE DESIGN PROCESS

Burdo G.B., Vorobyeva E.V.

*Tver State Technical University (TvSTU), Tver,
Russia*

gbtms@yandex.ru

evgeniya.vor813@gmail.com

In this article the problem of creating a computer-aided design process (CAD P). Identified and the necessity of CAD P using ontological approach in the design. Developed and are given a structured ontology scheme details and process of manufacturing parts.

Identified a list of functions ontology tasks when designing technological process of manufacturing parts. The possibility of using semantic networks with ontological approach design.

Key words: computer-aided design process of manufacturing parts; developing the knowledge base; ontology and ontology tasks; semantic networks.

Introduction

To solve this problem have been identified all the elements that affect the design process, found information communication occurring between these elements in the development process of manufacturing parts.

For the organization of the system of accumulation and synthesis of information found elements are influencing the design process are described information links between them and submitted the necessary solutions to the problem, which allow you to select profitable in the given industrial process

conditions. Determine the need for a set of functions for the given task ontology. List of identified problems ontologies represented in the form of a diagram.

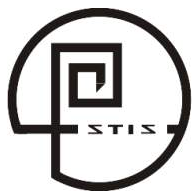
Main Part

Analyzed all the elements that influence the design process, defined information communications occurring between these elements. For the organization of the system of accumulation and synthesis of information through the use of domain ontology have been identified all of the elements described information links between them and submitted the necessary solutions to the problem that will lead to the selection of low-cost process.

Conclusion

Ontology components and process presented a visual approach that allows visually depict and explain the structure of relationships between concepts describing the detail in terms of manufacturing technology.

Application of the ontological approach in presenting the subject area allows for system representation of concepts and relationships between them. Presented ontological approach can be used to develop new rules the design process at different levels (routing and operational technologies). Based on the derived rules and relationships can be built CAD TP developing knowledge base.



OSTIS-2015

(Open Semantic Technologies for Intelligent Systems)

УДК 519.859

ИДЕНТИФИКАЦИЯ МОДЕЛИ ПРЕДСТАВЛЕНИЙ АГЕНТА О ПРЕДМЕТНОЙ ОБЛАСТИ В СИТУАЦИИ ВЫБОРА

Виноградов Г.П., Борзов Д.А.

*Тверской государственной технической университет,
г. Тверь, Россия*

Wgp272ng@yandex.ru

Рассматривается проблема идентификации представлений агента о предметной области в ситуации выбора. Введено понятие субъективного представления о предметной области. Для формализации представлений предложено использовать нечеткую продукционную модель. Приведено описание структуры представлений и один из вариантов ее адаптации. Рассмотрены способы оценки полезности модели представлений.

Ключевые слова: нечеткое множество; принятия решений; нечеткие предпочтения; модели нечеткого выбора.

ВВЕДЕНИЕ

Человек (далее агент) живет и действует в объективно существующем мире. Но все свои действия он осуществляет в соответствии со своими представлениями о: предметной области, способах действия, целях и получаемых результатах, демонстрируя тем самым так называемое активное поведение. В работе [Виноградов, 2011] показано, что рациональность является центральным понятием в задачах изучения и моделирования интеллектуального поведения. Концепция рациональности должна применяться с точки зрения субъективных целей, норм, ценностей и представлений агента о ситуации выбора. Это позволяет с единых позиций рассматривать поведение различных агентов в различных средах. Результаты, которые получает агент от своих действий, зависят в значительной степени от того, как агент воспринимает процессы в предметной области и как он строит программы своего поведения. Основой формирования таких программ являются: 1) представления агента о компонентах ситуации выбора, которые следует рассматривать как модели, и 2) способность агента формировать такие модели. С их помощью агент осуществляет взаимодействие с окружающим миром путем внутреннего программирования на языке описания представлений. Функциональное назначение представлений – получить возможность моделировать возможные результаты в зависимости от выбираемых действий и состояния внешней среды.

Реальность агент познает также только с помощью им же созданных моделей при данном уровне своих знаний. В ситуации неполной информации представления имеют гипотетико-дедуктивную структуру. Гипотезы выступают попытками разрешить проблемы, дедукция позволяет провести проверку содержания гипотез фактами.

Методы моделирования должны быть направлены на выявление психических стратегий конкретного человека путем анализа его поведения, речевых паттернов, невербальных реакций в ситуациях целеустремленного состояния, выбора и реализации способов действия. Такой подход предполагает интерактивное взаимодействие исследователя и агента. В процессе этого взаимодействия, которое носит также и итерационный характер, выявляются специфичные когнитивные, лингвистические и поведенческие навыки, которые субъект использует в ситуациях выбора для достижения желаемых результатов. Они, в свою очередь, отражают его систему ценностей, норм, способностей, убеждений и знаний. Эти структуры выражаются в процедурах формирования решений и реализации их в поведении. В этой связи представляет интерес разработка математических моделей, учитывающих поведение агента на основе теории нечетких систем и теории отношений.

1. ФОРМАЛИЗАЦИЯ ПРЕДСТАВЛЕНИЙ

В модели выбора представление – это та часть окружения агента, которая им воспринимается и осознается. Оно является результатом фильтрации

его уникальным опытом, культурой, языком, убеждениями, интересами, предположениями и неврологией. Фильтрация позволяет упростить результат восприятия, чтобы можно было осмыслить воспринимаемое. Степень упрощения определяется целями и зависит от способностей. Качество представлений определяется **не точностью** или **адекватностью**, а **полезностью** при достижении желаемых состояний. Представления следует рассматривать как результат преобразования опыта субъекта (его еще иначе называют экспертным знанием), теоретического (книжного) опыта и результатов измерения параметров ситуации целеустремленного состояния.

Описание представлений средствами языка образует модель, которая может «отчуждаться» от ее создателя, «читаться», «пониматься» другими агентами при коммуникации. Языковые возможности субъекта являются фильтрами при описании представлений. Модели не рассчитаны на отражение либо конструирование единственной объективной реальности, их задача заключается в том, чтобы воспроизвести какой-либо аспект возможной реальности. Поэтому не имеет значения «истинность» модели, учитывается лишь «убежденность в ее полезности». Полезность модели зависит от степени, в которой она позволяет совершить эффективный переход из одного целеустремленного состояния в другое.

Для вербального описания представлений агента широко применяется нечеткое отношение моделирования $R: E \times M \rightarrow [0,1]$ [Заде, 1978]. Его использование позволяет описать причинно-следственную связь между наблюдаемым состоянием компонентов или параметров ситуации целеустремленного состояния и внутренним представлением агента о ней в виде лингвистических переменных. Для описания влияния выделенных агентом факторов на результаты $o_i^k, i = \overline{1, m}$ будем использовать аппроксимацию представлений в виде нечетких продукционных правил, которые имеют вид:

Если x_1 есть A_{r1}^k и если x_2 есть A_{r2}^k и ... и если

$$x_N \text{ есть } A_{rN}^k,$$

то

$$o_i^k = f_{ir}^k(x_1, x_2, \dots, x_N), r = \overline{1, R}, i = \overline{1, m}, \quad (1)$$

где R – количество продукционных правил, r – номер текущего продукционного правила,

$o_i^k = f_{ir}^k(x_1, x_2, \dots, x_N)$ – четкая функция, отражающая представление агента о функциональной связи входных факторов с возможными результатами для r -го правила (r -ая

частная модель); A_{ri}^k – нечеткие переменные, определенные на $X^k = \{x_i^k, i = \overline{1, N}\}$.

В качестве функции $f_{ir}(\bullet)$ могут использоваться, например, формальные модели, модели, словесное описание. Функции $f_{ir}(\bullet)$ могут быть заданы графиком, таблицей, алгоритмом вычисления т.д.

2. ОПИСАНИЕ СТРУКТУРЫ ПРЕДСТАВЛЕНИЙ АГЕНТА

Пусть в ситуации выбора участвует множество $N = \{1, 2, \dots, n\}$ агентов. Если присутствует неопределенный параметр $\theta \in \Omega$ (будем считать, что множество Ω является общим знанием), то структура представлений I_i i -го агента включает в себя следующие элементы:

1. представление i -го агента о параметре θ – обозначим его $\theta_i, \theta_i \in \Omega$;
2. представления i -го агента о представлениях других агентов о параметре θ – обозначим их $\theta_{ij}, \theta_{ij} \in \Omega$;
3. представления i -го агента о представлении j -го агента о представлении k -го агента – обозначим их $\theta_{ijk}, \theta_{ijk} \in \Omega, j, k \in N$ и так далее, что образует иерархию представлений i -го агента.

Структура представлений I_i i -го агента задается набором всевозможных значений вида $\theta_{ij_1-j_l}$, где l

пробегает множество целых неотрицательных чисел, $j_1, \dots, j_l \in N$, а $\theta_{ij_1-j_l} \in \Omega$. Для

формирования своих представлений агент использует следующие источники информации о состоянии внешней среды и окружения [Новиков, 2008]:

1) априорная частная информация $\omega_k(\theta) \subseteq \Omega$ (этот вид информации в литературе называется знанием, опытом);

2) действия других агентов: наблюдая их и предполагая, что оппоненты действуют рационально, агент может (при выполнении предположения об общем знании) оценивать информацию $q(c)$ о состоянии природы, на основании которой был выполнен выбор способа действия i -м агентом ($i \neq k$);

3) множество g результатов выбора агентов – на основании этой информации агент может сделать вывод о тех состояниях природы, при которых выбранный способ действия приводит к наблюдаемым выигрышам;

4) множество $\rho \subseteq \Omega$ состояний природы, при которых наблюдаемый вектор действий агентов приводит именно к данному наблюдаемому значению o результата системы:

$$\sigma(c,o)=\{\theta \in \Omega / G(\theta(c))=o\}. \quad (2)$$

2.1. Модели адаптации представлений агента

Специфика формирования представлений агентом состоит, в частности, в том, что он в качестве информации для корректировки своих представлений о неопределенном параметре может использовать не только результаты наблюдения за внешней средой, но и результаты наблюдения за действиями и результатами деятельности других агентов, пытаясь «объяснить», почему они выбрали именно эти действия.

Рассмотрим описание модели ситуации выбора, включающей в себя $n \in N = \{1, 2, \dots, n\}$ агентов. Пусть $\theta \in \Omega$ – *состояние природы*, описывающее все существенные характеристики внешней среды. Агент с номером $i \in N$ имеет интервальную информацию $\omega_i(\theta) \subseteq \Omega$ о состоянии природы, причем эта информация не противоречит истинному положению дел, то есть $\forall \theta \in \Omega, \forall i \in N, \theta \in \omega_i(\theta)$.

Результат $o = G(\theta, x)$ агента зависит от вектора $c = (c_1, c_2, \dots, c_n) \in C' = \prod_{i \in N} C_i$ действий других агентов связанных с агентом принимающим решение, где $c_i \in C_i$, и θ – состояния природы. Будем считать, что каждый агент наблюдает вектор действий всех агентов, общий результат и выигрыши всех агентов.

Предположим, что удельная ценность выбора каждого агента зависит от состояния природы θ и результата o всех участников ситуации выбора: $EV_k(o) = f_k(\theta, G(c, \theta))$, $k \in K$, причем множество агентов K , их результаты $\{o_k(\bullet)\}$, множества допустимых способов действия $\{C_k\}$, а также множество Ω возможных значений состояний природы, функция $G(\cdot)$ и факт наблюдения как результата и выигрышей, так и всего вектора действий каждым членом команды являются среди них общим знанием, что позволяет для описания их поведения использовать аппарат теории игр.

Обозначим множество параметрических (параметром является значение состояния природы, которое определяет структуру и уровень представлений агента) равновесий Нэша через

$$E_N(\theta) = \{ \{c_k\}_{k \in K} \in C' / \forall k \in K, \forall c_k \in C_k \quad (2) \\ f_k(\theta, G(\theta(c_1, \dots, c_K))) \geq \\ \geq f_k(\theta, G(\theta, c_1, \dots, c_{k-1}, y_k, \dots, c_{k+1}, \dots, c_K)) \}.$$

Если множество Ω_0 возможных значений состояний природы является общим знанием среди агентов, то, предполагая, что они устраняют неопределенность вычислением максимального

гарантированного результата, получим следующее множество равновесий их игры:

$$E(\Omega_0) = \{ \{c_k\}_{k \in K} \in C' / \forall k \in K, \forall y_k \in C_k \\ \min_{\theta \in \Omega_0} f_k(\theta, G(\theta, c_1, \dots, c_K)) \geq \\ \min_{\theta \in \Omega_0} f_k(\theta, G(\theta, c_1, \dots, c_{k-1}, y_k, \dots, c_{k+1}, \dots, c_K)) \}.$$

Обозначим $q(c) \subseteq \Omega$ – множество состояний природы, при которых наблюдаемый агентами вектор их действий является равновесием:

$$q(c) = \{ \theta \in \Omega / \exists \Omega_0 : \theta \in \Omega_0 \}. \quad (3)$$

Пусть $g = (g_1, \dots, g_n) \in \mathfrak{R}^n$ – наблюдаемый агентами вектор значений их целевых функций, тогда множество тех значений состояний природы, при которых (наряду с наблюдаемым результатом o) могут реализоваться наблюдаемые выигрыши агентов g , имеют вид

$$\eta(g, o) = \{ \theta \in \Omega / f_j(\theta, o) = g_j, j \in K \}. \quad (4)$$

Поскольку вектор $g = (g_1, g_2, \dots, g_n) \in \mathfrak{R}^n$ определяется через EV_k , то вектор $V = \{v_1, \dots, v_k\} \in \Theta^n$, будет определять требуемый уровень представлений агентов в ситуации выбора.

3. ПРЕДСТАВЛЕНИЯ КАК СУБЪЕКТИВНАЯ МОДЕЛЬ, СВЯЗЫВАЮЩАЯ СПОСОБЫ ДЕЙСТВИЯ И РЕЗУЛЬТАТ

Агент при формировании своих представлений в той или иной форме использует аппарат аргументации для построения последовательности гипотез, сходящихся к субъективно-истинной по убеждению.

Необходимость учета при моделировании поведения агента его убежденности в адекватности своих представлений о ситуации выбора предполагает введение в модель выбора лингвистической переменной «убежденность». Интервал $[0, 1]$ используется как универсальное множество для задания лингвистической переменной «убежденность» с терминами «убежден» и «не убежден» с функциями принадлежности, предложенными Л. Заде [Заде, 1976].

Убежденность – это объект, который существует только в сознании агента. Агент использует его для выражения своего отношения к своим представлениям об объекте. То есть он занимает определенную позицию по отношению к своим представлениям. Будем обозначать такое отношение как *Убежденность(агент, x)*, где x – это представление. Оно может существовать, например, в форме (1).

Если существует преобразование *объект* \rightarrow *представление*, то существует и обратное

преобразование *представление* \rightarrow *объект*, которое следует положить в основу описания способности агента рассуждать о своих убеждениях. Правомерность такого утверждения следует из того факта, что субъект (при рациональной форме поведения) стремится к изоморфности представлений к объекту. Для этого он оценивает адекватность представлений некоторым набором параметров $K = \{h_i, i = \overline{1, n}\}$.

Пусть $h_j : K \rightarrow [0, 1]$ – частная оценка j -го показателя. Она может вычисляться как субъективно, так и объективно. Пусть нечеткая мера для $(K, 2^K)$ является субъективной мерой, выражающей степень важности подмножеств из K . Например, $g(h_1)$ выражает степень важности показателя h_1 при оценке агентом соответствия представлений объекту. Тогда интеграл

$$\int_v h(k) \circ g(\cdot) = \sup_{\alpha \in [0, 1]} \{\alpha \wedge g(M(\alpha))\}. \quad (4)$$

позволяет вычислить обобщенную оценку качества представлений. Здесь $M_\alpha = \{x \mid \mu(x) \geq \alpha\}$ – уровневое множество. Очевидно, что существует $\varphi : J \rightarrow [0, 1]$, которое позволяет вводить в модель субъективную оценку убежденности. Использование нечеткого интеграла позволяет вычислять нечеткое ожидаемое значение результата. Как показано в [Бочарников, 2000] с помощью нечеткого интеграла можно моделировать принятие решений по средневзвешенной оценке (линейная свертка). Это связано с тем, что в задачах многокритериального выбора нечеткий интеграл обеспечивает получение решения, соответствующего медиане, которая в порядковых шкалах является аналогом среднего. Поскольку в этом случае ослабляется условие суммы для коэффициентов важности критериев и вводится формализация, основанная на монотонности оценок, то полученная интегральная оценка будет практически совпадать с интуитивными ожиданиями человека. Приведенные рассуждения позволяют предложить следующий алгоритм вычисления степени убежденности.

Пусть описанные выше показатели качества построенной модели представлений образуют вектор $Z = \{z_j, j = \overline{1, k}\}$. Тогда очевидно, что

$Z = \{z_j, j = \overline{1, k}\}$ будет зависеть от вектора параметров функций принадлежности $W = \{w_i, i = \overline{1, m}\}$, которые для агента образуют

вектор управляющих переменных, выбором которых он может обеспечить требуемый уровень прогностической эффективности модели представлений.

Будем предполагать, что агент обладает достаточно высокой квалификацией и опытом, то

его шкала ценностей определена таким образом, что различные наборы показателей $Z = \{z_j, j = \overline{1, k}\}$ имеют для него неодинаковое

значение. Это позволяет предположить существование у него непрерывного монотонно возрастающего по каждому показателю квазивогнутого индикатора предпочтений $U(Z)$ такого, что

$$Z(W^{(1)}) \succ Z(W^{(2)}) \leftrightarrow U(Z(W^{(1)})) > U(Z(W^{(2)}))$$

$$Z(W^{(1)}) \sim Z(W^{(2)}) \leftrightarrow U(Z(W^{(1)})) = U(Z(W^{(2)}))$$

где $W^{(1)}, W^{(2)} \in \Omega_W$ (здесь Ω_W – множество допустимых значений управляющих переменных).

Сделанное предположение относительно функции $U(Z(W))$ позволяет определить решение задачи векторной оптимизации, как множество точек $\{w_i^0\}$, максимизирующих функцию $U(Z(W))$, таких что:

$$W^0 = \{w_i^0, i = \overline{1, m}\} \in \Omega_W$$

и

$$\text{Arg max } U(z_1(W), \dots, z_m(W)).$$

Для найденных значений $W^0 \in \Omega_W$ должно выполняться условие оптимальности по Парето и поиск решения должен проходить по паретовой границе множества $Z(W)$.

Функция $U(Z(W))$ в явном виде, как правило, неизвестна, поэтому для определения оптимальных величин целесообразно использовать интерактивные процедуры. Для этого выбирается некоторое решение $W^{(1)}$, с использованием информации, получаемой от агента, определяется поведение $U(Z(\bullet))$ в окрестности точки $W^{(1)}$ и на этой основе строится последовательность решений $\{W^{(i)}\}$, которая при определенных условиях сходится к W^0 .

Однако, часто множество $E(Z(W))$ невыпуклое и поиск в пространстве решений сопряжен со значительными трудностями. Поэтому паретову границу целесообразно параметризовать элементами более простого множества A . Из известных процедур параметризации для целей оптимизации прогностических свойств модели представлений наиболее подходящей является процедура ассортиментной параметризации [Vinogradov, 2011]:

$$U(Z(W)) = U(\alpha, Z(W)) = \langle \alpha, W \rangle,$$

где $\langle \bullet \rangle$ — означает скалярное произведение,

$$\alpha \in A, \{\alpha_j \geq 0, \sum_{j=1}^k \alpha_j = 1\}. \text{ При этом}$$

выполняются условия:

$$\forall W \in \Pi_W, \exists \alpha(W) \in A:$$

$$W(\alpha) = \text{Arg max } U(\alpha(W), Z(W)) = W^0,$$

$$\forall \alpha \in A, \exists W(\alpha) \in \Pi_W,$$

где Π_W — область Парето.

Пусть V^* — совокупность предпочтительных с точки зрения агента показателей $Z(W)$, причем $V^* \neq \emptyset$ и $V^* \in E(Z(W))$, тогда согласно принятой процедуре параметризации, V^* можно представить как $V^* = \sigma(A^*)$, где A^* — множество максимальных элементов отношения \succ , определяемых предпочтениями агента на множестве параметров A , по правилу

$$\alpha_1 \succ \alpha_2 \Leftrightarrow \sigma(\alpha_1) \geq \sigma(\alpha_2); \alpha_1, \alpha_2 \in A.$$

Тогда задача принятия решения по выбору оптимальных структуры и параметров модели представлений может быть записана в виде:

$$U^*(\alpha) \rightarrow \max, \alpha \in A, . \quad (6)$$

где $U^* = U \circ \sigma$.

Таким образом, произведена параметрическая декомпозиция экстремальной задачи $U(Z(W))$, $W \in \Omega_w$, $Z(W) \in \Omega_z$ на задачу вычисления σ и задачу $\max U(\sigma(\alpha))$, $\alpha \in A$. Такая декомпозиция распределяет роли в человеко-машинном диалоге следующим образом:

- на ЭВМ вычисляется параметризация σ , которая для ассортиментной параметризации имеет вид $\max \Psi$ при $Z(W) \geq \alpha Z$;
- агент участвует в решении задачи (6).

В качестве формальной основы диалоговой процедуры построения модели представлений следует воспользоваться методами прямого поиска, не требующие информации о производных целевой функции, так как латентными факторами выступают качественные признаки, и кроме того, функция $U(\bullet)$ предпочтений агента, в общем случае, не является дифференцируемой. Наибольший эффект следует ожидать от применения методов случайного поиска.

При организации диалога с агентом использовалась следующая модель его реакции на предъявленное решение. По двум решениям $Z(W^{(1)})$ и $Z(W^{(2)})$ агент сообщает вектор с компонентами:

$$\xi_i \quad i = \overline{1, k} \text{ такой, что:}$$

$$\xi_i = \begin{cases} +1, \text{ изменение } i\text{-го критерия} \\ \text{предпочтительно;} \\ -1, \text{ в противном случае;} \\ 0, \text{ изменение } i\text{-го критерия безразлично.} \end{cases}$$

Общая структура алгоритма случайного поиска для задачи построения модели представлений имеет следующий вид:

$$\alpha(s+1) = \alpha(s) + \mathcal{G}(s+1),$$

где s — номер обращения к агенту, $\mathcal{G}(s+1)$, —

вариация вектора α , определяется в пространстве случайных векторов в зависимости от модели реакции агента.

При $\xi_i = 0$ вариацию α_i следует положить равной 0. В остальных случаях целесообразно использовать алгоритм с поощрением случайностью:

$$\mathcal{G}_i(s+1) = \begin{cases} \gamma(s+1)r_i(s+1) \text{ при } \xi_i \geq 0 \\ \gamma(s)r_i(s) \text{ при } \xi_i < 0, \end{cases}$$

где $\gamma(s+1)$ — скаляр, выбранный из условий сходимости. Например, в случае, если в результате двух шагов $U(\alpha, Z(W(\alpha)))$ возрастает, тогда $\gamma(s) = d \cdot \gamma(s-1)$, где d — параметр акселерации, $d > 1$, $r(s+1)$ — случайный вектор нормируемый следующим образом:

$$r_i(s+1) = \frac{C_i(s+1)}{\sqrt{q}}, \text{ где}$$

$$d = \sum_{i \in I(s+1)} [C_i(s+1)]^2 + \sum_{j \in I(s+1)} [C_j(s+1)]^2 + \sum_{k \in I(s+1)} [C_k(s+1)]^2,$$

$C_i(s+1)$ — случайный вектор, распределенный на единичной сфере

Учет ограничений $\alpha(s+1) \in S_\alpha$ производится следующим образом:

$$\sum_{i \in I(s+1)} r_i(s+1) + \sum_{k \in I(s+1)} r_k(s+1) = 1 - \sum_{j \in I(s+1)} r_j(s+1)$$

Ускорение сходимости описанного выше алгоритма возможно за счет более полного учета информации о направлении поиска в пространстве решений, получаемой от агента.

4. ИНФОРМАЦИОННАЯ СИСТЕМА СБОРА, ХРАНЕНИЯ И ОБРАБОТКИ ДАННЫХ ДЛЯ ДИАГНОСТИКИ ЗАБОЛЕВАНИЙ ПАЦИЕНТОВ

Описанная схема формирования представлений агента о ситуации выбора реализована в информационной системе сбора, хранения и обработки данных для диагностики заболеваний пациентов. Она выполнена в соответствии с клиент-серверной архитектурой, так как такая архитектура системы позволяет полностью решить возложенные на неё задачи. Уровень сервера состоит из центральной базы данных и центрального сервера для обработки запросов. Уровень клиента состоит из панели администратора, которая обслуживает систему, и множества клиентов.

Центральная БД является хранилищем для постоянно поступающих данных о пациентах, заболеваниях и проявлении симптомов у пациентов. Она содержит в себе функциональности для

выполнения описанных выше процедур расчета. Центральный сервер предназначен для обслуживания запросов на чтение и редактирование БД, поступающих от рабочей станции и клиентов. На сервере развернуты две службы, чтобы обеспечить обработку запросов от панели администратора и клиентов. Уровень «Клиенты» предназначен для формирования собственного списка пациентов и проявлений у них симптомов с последующей диагностикой заболеваний, используя данные для клиентских модулей расчета. Эти данные будут записаны в БД как «сырые», а через панель оператора можно будет исследовать данные от клиента и часть из них выделить в «эталонные». После этого будут обновлены данные для клиентских модулей расчета.

Процедура формирования модели представлений построена на использовании методов мягких вычислений (МВ). Она состоит из семи блоков: самоорганизующейся сети, нелинейной регрессионной модели, классификатора, модуля сравнения и обучения, машины нечеткого логического вывода (НЛВ), расчета оценок убежденности, базы знаний (БЗ).

Каждый элемент этой модели представлен одним из методов парадигмы мягких МВ. Её особенность заключается в выполнении параллельных прогнозов нелинейной регрессионной моделью и машиной НЛВ с последующим синтезом единого прогноза.

Заключение

Предложенный подход может быть положен в основу построения интеллектуальных систем и предполагает:

- 1) разработку принципов построения эволюционных адаптивных баз данных и знаний;
- 2) разработку методов обработки данных на основе применения "активной" логической сети правил, управляемой потоком данных;
- 3) разработку методов быстрого поиска маршрута логического вывода на основе построения многополюсной сети правил и поиска ее минимального разреза;
- 4) разработку методов распараллеливания потоковой обработки взаимозависимых данных путем построения виртуальных потоковых баз данных.

Библиографический список

- [Виноградов и др., 2011] Виноградов, Г.П. Моделирование поведения агента с учетом субъективных представлений о ситуации выбора / Виноградов, Г.П., Кузнецов В.Н. // Искусственный интеллект и принятие решений. № 3. с. 58-72.
- [Заде, 1978] Заде, Л.А. Основы нового подхода к анализу сложных процессов принятия решений / Заде Л.А. // Математика сегодня. – М.: Знание, 1978, - С. 5-40.
- [Vinogradov, 2011]. Vinogradov, G. Decision Making based of Subjective Conceptions of Decision situation / Vinogradov G. // Interactive Systems and Technologies: the Problems of Human-Computer Interaction, - Collection of Scientific papers. Uljanovsk, 2011, pp. 403-414.

[Новиков, 2008] Новиков, Д. А. Математические модели формирования и функционирования команд / Новиков Д.А.; – М.: Физматлит, 2008.

[Бочарников, 2000] Бочарников, В.П. Fuzzy-технология: Математические основы. Практика моделирования в экономике / Бочарников В.П.; – Санкт-Петербург: Наука, РАН, 2000. – 328с.

[Заде, 1976] Заде, Л.А. Понятие лингвистической переменной и его применение к принятию приближенных решений / Заде Л.А.; – М.: Мир, 1976

IDENTIFICATION MODEL REPRESENTATIONS OF THE AGENT ABOUT THE SUBJECT IN A SITUATION OF CHOICE

Vinogradov, P.*, Borzov D. A.**

* *Tver state technical University,
Tver, Russia
Wgp272ng@mail.ru*

** *Tver state technical University,
Tver, Russia
Wgp272ng@yandex.ru*

Considers the problem of identifying the representations of the agent about the subject in a situation of choice. The notion of subjective ideas about the subject. To formalize the notions proposed to use fuzzy production model. Describes the structure of the representations and one of the options for its adaptation. Learn how to evaluate the usefulness of the model views.

INTRODUCTION

Person (the agent) lives and works in objectively existing world. But all his actions he performs in accordance with their views about: the subject area, methods of action, goals and results, demonstrating the so-called active behavior. Shows that rationality is a Central concept in the objectives of the study and simulation of intelligent behavior. The concept of rationality should be applied from the point of view of subjective goals, norms, values and beliefs of the agent about the situation of choice. This allows a unified approach to consider the behavior of different agents in different environments.

CONCLUSION

The proposed approach can be used as a basis for building intelligent systems and includes:

- 1) the development of principles of evolutionary adaptive data and knowledge bases;
- 2) the development of data processing methods based on application of the "active" Boolean network rule-driven data flow;
- 3) the development of methods for fast route lookup logical conclusion based on the construction of a multi-polar network of rules and search for the minimum cut;
- 4) the development of methods of parallelization streaming interdependent data by building virtual streaming databases.



OSTIS-2015

(Open Semantic Technologies for Intelligent Systems)

УДК 004.021:004.424.4:004.51

ИССЛЕДОВАНИЕ ЭФФЕКТИВНОСТИ ГЕНЕТИЧЕСКОГО АЛГОРИТМА ДЛЯ ТЕМАТИЧЕСКОГО ДОКУМЕНТАЛЬНОГО ПОИСКА

Иванов В.К., Палюх Б.В.

Тверской государственный технический университет, г. Тверь, Россия

mtivk@tstu.tver.ru

pboris@tstu.tver.ru

В статье приведены результаты экспериментальных исследований эффективности генетического алгоритма, примененного для формирования эффективных поисковых запросов и отбора релевантных результатов при выполнении документального тематического поиска. Исследования проводились с целью сравнительного анализа семантической релевантности и качества ранжирования документов, найденных в Интернет различными способами. Показано, что разработанная технология расширяет возможности семантического поиска и увеличивает число релевантных результатов.

Ключевые слова: генетический алгоритм; поисковый запрос; ранжирование; релевантность.

Введение

Выполнение тематического поиска в документальных хранилищах [Голицына, 2009], является известной процедурой. Однако, как показывает практика, ее эффективное выполнение продолжает оставаться не такой простой задачей, как может показаться (см., например [Попов, 2009]). Одна из основных сложностей – семантически корректная формулировка поисковых запросов, обеспечивающих приемлемые показатели точности поиска. Актуальность разработок новых подходов в этом направлении очевидна, учитывая широкую область применения тематического документального поиска.

В статье представлены результаты экспериментальных исследований эффективности генетического алгоритма, примененного для формирования эффективных поисковых запросов и отбора релевантных результатов поиска. Исследования проводились с целью сравнительного анализа семантической релевантности и качества ранжирования документов, найденных в Интернет различными способами.

1. Особенности тематического документального поиска

Тематический документальный поиск ставит целью отбор документов, содержащих координированную информацию (взаимосвязанные факты, их ретроспективу и перспективу) в

тематическом сегменте или по заданному объекту. Результатом такого поиска является множество документов, максимально релевантных заданной тематике в целом, а не просто сведения об отдельно взятых событиях, предметах или явлениях. Области применения тематического поиска довольно многообразны: поиск инновационных решений, определение новых направлений бизнеса, сбор информации о клиентах, конкурентный анализ и разведка, обзоры источников научно-технической информации, работа конкурсных комиссий и экспертиза проектов, поиск в патентных исследованиях, подбор учебных материалов.

При выполнении тематического поиска пользователи неизбежно сталкиваются со следующими объективными проблемами:

- Сложность подбора ключевых понятий для формулировки поисковых запросов; невозможность использования в запросах всех доступных критериев отбора документов одновременно.
- Ограничения поисковой системы по составу, структуре и сложности запросов.
- Разрозненность и неоднородность сведений, часто находящихся на стыке смежных областей; наличие альтернатив с сопоставимой релевантностью.
- Отсутствие эффективных систем кластеризации и классификации информации по направлениям тематического поиска.

Решая указанные проблемы, следует правильно интерпретировать результаты поиска, имея в виду

совместную оценку релевантности документов, найденных разными запросами, корректность ранжирования поисковой системы, доступность для оценки всех релевантных результатов, наличие эффективных решений в других областях для успешного использования в данной области.

2. Реализация генетического алгоритма для фильтрации и ранжирования результатов поиска

В проекте интеллектуальной распределенной системы информационной поддержки инноваций в науке и образовании [Палюх и др., 2013], [Ivanov et al, 2014] предлагается технология генерации поисковых запросов, фильтрации и ранжирования результатов поиска. Эта технология должна быть использована для создания хранилищ инновационных решений образовательного и научного назначения.

Основная идея генерации поисковых запросов – организация с помощью специального генетического алгоритма эволюционного процесса, формирующего в поисковой системе устойчивую и эффективную популяцию запросов для получения высокорелевантных результатов. В ходе этого процесса кодированные запросы, последовательно подвергаются генетическим изменениям и выполняются в поисковой системе. Далее оценивается семантическая релевантность промежуточных результатов поиска, вычисляются значения целевой функции и осуществляется отбор наиболее пригодных запросов.

В [Иванов и др., 2014] представлен прототип программной реализации генетического алгоритма для формирования эффективных поисковых запросов и отбора релевантных результатов при выполнении документального тематического поиска. В частности, описаны основные шаги и параметры алгоритма, компоненты программного обеспечения и результаты предварительных исследований алгоритма. Этот прототип реализован в виде программного продукта Genetic Algorithm Framework (GAF).

В исследовании, описанном в настоящей статье, для проведения экспериментов GAF использовался в следующем составе:

- Пользовательский интерфейс.
- Основная библиотека GAF.
- Модуль морфологического анализа и лемматизации.
- Модуль семантического анализа сходства текстов.
- Модуль поиска для Bing.
- Модуль управления базой данных.
- Модуль управления метаданными

3. Подготовка исходных данных

Для участия в исследованиях были приглашены

12 экспертов-исследователей в следующих областях знаний: электронное обучение, базы данных, строительные материалы и изделия, трение и износ в машинах, автоматизация химико-технологических процессов, технологии и средства механизации и технического обслуживания в сельском хозяйстве, медицина, астрофизика.

Каждый эксперт предоставил текстовый материал (автореферат кандидатской или докторской диссертации, монографию, одну или несколько научных статей), наиболее адекватно отражающий ту часть области знаний, в которой он (эксперт) является специалистом.

4. Поисковые запросы и параметры их выполнения

На основе материалов, представленных экспертами, были подготовлены запросы двух типов:

- Тип 1. Набор ключевых слов из заглавия предоставленного материала. Длина запроса – от 5 до 11 термов.
- Тип 2. Набор ключевых слов из текста предоставленного материала. Длина запроса – 50 термов. В качестве термов запроса использовались слова материала, имеющие наибольшие веса, вычисленные в соответствии с мерой $TF*IDF$, но без второго компонента (обратной частоты документа).

Запросы обоих типов формулировались в двух вариантах:

- Вариант 1. Запрос с фиксированными термами (термы в кавычках).
- Вариант 2. Запрос с использованием лемм термов с возможностью словоформ.

Запрос типа 1 выполнялся в поисковой системе Bing со стандартными настройками. Запрос типа 2 выполнялся в среде GAF с модулем поиска для Bing (далее среда GAF/Bing). Идентификаторы запросов, используемые далее в статье, представлены в табл. 1. Отметим, что квалификатор Q (quoted) используется для запросов с вариантом формулировки 1 (термы в кавычках).

Таблица 1 – Идентификаторы типов запросов

Идентификатор типа запроса	Тип запроса	Вариант формулировки
Bing.Q	1	1
GAF/Bing.Q	2	1
Bing	1	2
GAF/Bing	2	2

Используемые параметры GAF/Bing:

- Количество запросов в генерируемых популяциях $g_2 = 8$.
- Количество ключевых слов в каждом генерируемом запросе $g_3 = 6$.

- Количество результатов поиска, возвращаемых запросом $f_1 = 20$, либо популяцией запросов $f_2 = 20$, либо суммарно всеми популяциями $f_3 = 20$.
- Коэффициент для учета расположения документов на одном сервере $f_4 = 0,75$.
- Весовые коэффициенты для аргументов p , r и s соответственно при расчете ранга результата поиска $f_5 = 0,33, f_6 = 0,33, f_7 = 0,34$ (о функции пригодности и ее аргументах см. далее раздел "Сравнительный анализ качества ранжирования").
- Вероятность мутации запроса $m_1 = 1$.
- Число проходов алгоритма (или число генерируемых популяций) $e_1 = 10$.

5. Обработка результатов экспертами

Результаты выполнения всех запросов по каждой тематике (списки адресов документов в Интернет, упорядоченные по неизвестному экспертам алгоритму) передавались экспертам для оценки их релевантности.

Под релевантностью предлагалось понимать соответствие содержания материала теме исследований, проведенных или проводимых экспертом. Должно было оцениваться соотношение объёма информации, полезной для разработки темы, к общему объёму информации в материале. Для оценки релевантности документа предлагалось использовать следующую шкалу:

- 0 – материал не содержит информации, полезной для разработки темы.
- 1 – материал содержит незначительное количество информации, полезной для разработки темы.
- 2 – материал содержит достаточное количество информации, полезной для разработки темы.
- 3 – материал содержит значительное количество информации, полезной для разработки темы.

Оценку каждого найденного документа следовало выполнить:

- С позиций высококвалифицированного специалиста по теме исследований эксперта (релевантность специалиста).
- С позиций специалиста, только *начинающего* научную работу по теме исследований эксперта (релевантность новичка).

Экспертам рекомендовалось учитывать, что полезность найденных документов может определяться похожими исследованиями, справочными данными, ссылками на другие материалы, описаниями аналогичных применяемых моделей, методов и технологий. При сомнениях в выборе одного из двух значений баллов шкалы предлагалось использовать меньшее значение. Также, релевантность информационных материалов

из справочников, словарей, энциклопедий рекомендовалось, как правило, оценивать баллами 0 или 1 (исключения – уникальная справочная информация).

Не рекомендовалось рассматривать как полезные рекламные объявления, коммерческие предложения, "визитные карточки" предприятий и т.п.

Для сохранения всех исходных данных и результатов исследований была разработана специализированная база данных.

6. Сравнительный анализ релевантности результатов поиска

На рис. 1 и 2 показана средняя релевантность документов – результатов поиска, определенная экспертами для специалистов и новичков соответственно.

В случае использования в запросах лемм термов с возможностью словоформ средняя релевантность документов, найденных Bing, на 3...9% выше, чем у документов, отобранных GAF/Bing. Причем это касается релевантности, определенной как для специалистов, так и для новичков. Для запросов с фиксированными термами Bing дает еще лучшую (на 11...13%) релевантность обоих типов для отобранных документов.

В целом, соотношения средней релевантности документов для специалистов и новичков для различных типов запросов сохраняются.

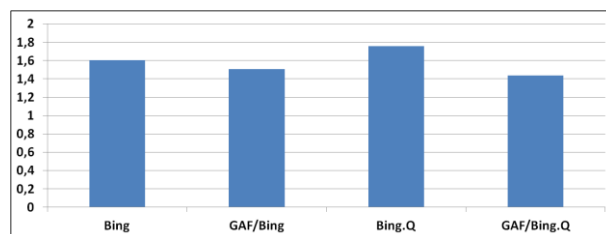


Рисунок 1 – Средняя релевантность результатов поиска (для специалистов)

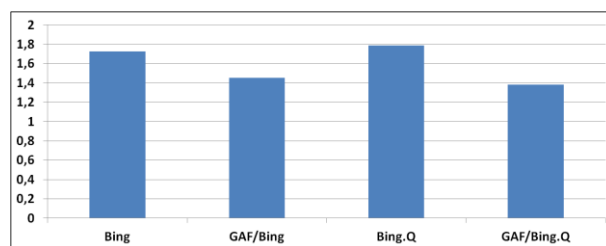


Рисунок 2 – Средняя релевантность результатов поиска (для новичков)

На рис. 3 и 4 показана точность поиска - доля релевантных документов в результатах поиска, определенная экспертами для специалистов и новичков соответственно.

В случае использования в запросах лемм термов доля релевантных результатов для специалистов в результатах поиска выше для Bing на 5%. Для новичков доля релевантных результатов выше для

GAF/Bing на 7%. Для запросов с фиксированными терминами доля релевантных результатов для специалистов примерно одинакова для Bing и для GAF/Bing, но для новичков доля релевантных документов, отобранных GAF/Bing выше на 4%.

В целом можно отметить более высокую долю релевантных документов, найденных с применением GAF/Bing.

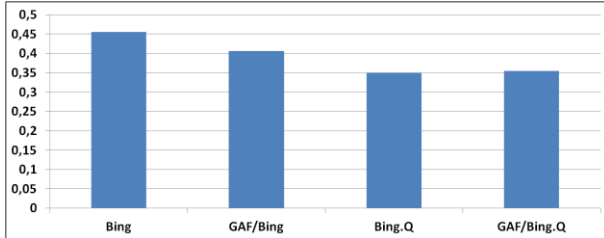


Рисунок 3 – Точность поиска - доля релевантных результатов в результатах поиска (для специалистов)

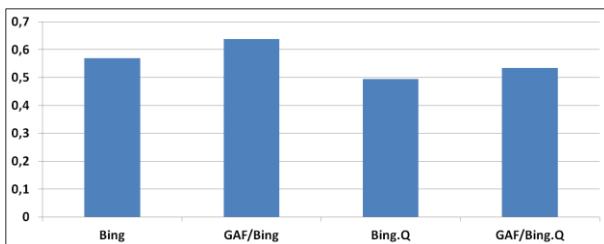


Рисунок 4 – Точность поиска - доля релевантных результатов в результатах поиска (для новичков)

7. Сравнительный анализ качества ранжирования

Ранжирование результатов поиска в GAF осуществляется в ходе вычисления значений функции пригодности \bar{W} при выполнении генетического алгоритма. Значение \bar{W} определяет качество запросов (пригодность особей популяции); генетический алгоритм ищет максимум \bar{W} :

$$\bar{W} = \frac{1}{N} \sum_{j=1}^N \bar{w}_j \rightarrow \max \quad (1)$$

где

\bar{W} – функция пригодности популяции;

N – количество запросов в популяции;

\bar{w}_j – функция пригодности j -го запроса популяции.

$$\bar{w} = \frac{1}{P} \sum_{i=1}^P w_i(f, p, s, a) \quad (2)$$

где

P – количество результатов поиска в запросе;

w_i – функция пригодности i -го результата запроса.

Для ранжирования результатов поиска используется значение $w_i(f, p, s, a)$. Для каждого i -го результата запроса значение w_i зависит от следующих аргументов: f – определяется позицией результата в ранжированном списке результатов, построенного используемой поисковой системой;

p – определяется вхождением данного результата в списки результатов большинства запросов; s – определяется семантической близостью к эталонным текстам, адаптивно формируемым в ходе выполнения алгоритма; a – определяется параметрами пользователя и является фактором среды (значения f, p, s, a нормированы на диапазон от 0 до 1). Для ранжирования использовалась средневзвешенная сумма f, p, s и a . Отметим, что предлагаемый способ ранжирования уточняет подходы, принятые в поисковых системах, которые, в свою очередь, основаны на сочетании в разной степени семантической релевантности и авторитетности документов, а также на поведенческих особенностях пользователей.

Для оценки качества ранжирования результатов поиска использовались метрики DCG@n и NDCG@n [Järvelin et al., 2002], [Агеев и др., 2010]. Эти метрики оценивают качество первых n документов в результатах поиска. Для документов, упорядоченных по значению функции пригодности w_i , вычислялись значения

$$DCG = \frac{1}{N} \sum_{p=1}^P \frac{2^{gr(p)} - 1}{\log_2(2 + p)} \quad (3)$$

где

$gr(p)$ – средняя экспертная оценка релевантности, выставленная документу, расположенному на позиции p в списке результатов;

$gr \in [0,3]$, причем 3 означает "релевантный", 0 – "нерелевантный", 1 и 2 – "частично релевантный": "релевантный(+)" или "релевантный(-)";

$1/\log_2(2+p)$ – дисконт за позицию документа (документы в начале списка имеют больший вес).

На рис. 5 показано соотношение значений метрик DCG для идеального ранжирования, то есть ранжирования, выполненного экспертами (DCG*Expert), ранжирования, выполненного поисковой системой (DCG*Bing), и ранжирования, выполненного GAF (DCG*GAF/Bing).

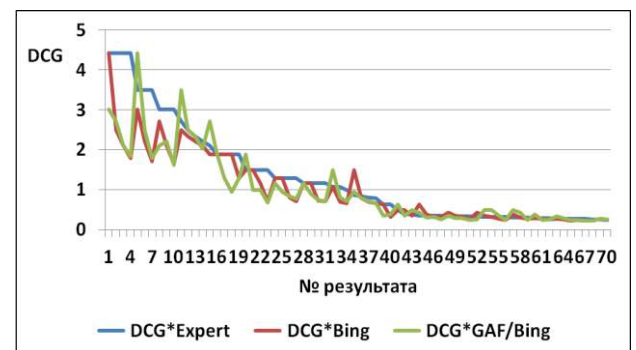


Рисунок 5 – Значения метрики DCG релевантных результатов поиска (для специалистов)

В экспериментах использовались запросы Bing (см. табл. 1) со стандартными параметрами поисковой системы Bing и запросы GAF/Bing (см. табл. 1) с параметрами: $g_2 = 1, g_3$ – соответствует запросу Bing, $f_1 = f_2 = f_3 = 20, f_4 = 0,75, f_5 = 0,33,$

$f_6 = 0,33, f_7 = 0,34, m_1 = 1, e_1 = 10$. Описание параметров приведено выше в разделе "Поисковые запросы и параметры их выполнения".

В целом графики на рис. 5 показывают хорошее визуальное совпадение значений DCG для рассматриваемых способов ранжирования.

Для точной количественной оценки сходства значений из пар последовательностей $\rho_{ExBing} = (DCG*Expert, DCG*Bing)$ и $\rho_{ExGAF} = (DCG*Expert, DCG*GAF/Bing)$ были рассчитаны значения коэффициента взаимной корреляции ρ_{12} :

$$\rho_{12}(j) = \frac{r_{12}(j)}{\frac{1}{N} \left[\sum_{n=0}^{N-1} x_1^2(n) \sum_{n=0}^{N-1} x_2^2(n) \right]^{1/2}} \quad (4)$$

где

ρ_{12} – коэффициент взаимной корреляции значений из последовательностей 1 и 2;

j – сдвиг значения, $j=0$ в рассматриваемом случае;

n – номер значения в последовательностях;

N – число значений в каждой последовательности;

x_1, x_2 – значения из последовательностей 1 и 2, в рассматриваемом случае это значения из $DCG*Expert$ и значения из $DCG*Bing$ либо из $DCG*GAF/Bing$;

r_{12} – взаимная корреляция значений из последовательностей 1 и 2:

$$r_{12}(j) = \frac{1}{N} \sum_{n=0}^{N-1} x_1(n)x_2(n+j) \quad (5)$$

При проведении расчетов использовалась методика из [Айфичер, 2004].

На рис. 6 представлена сравнительная оценка значений ρ_{12} , рассчитанных для значений из пар ρ_{ExBing} и ρ_{ExGAF} . Принимались во внимание оценки релевантности для специалистов и новичков. Диаграммы показывают, что качество ранжирования Bing незначительно выше, чем дает GAF/Bing - значение ρ_{12} для пары ρ_{ExBing} выше на 1,3...1,4%, чем значение ρ_{12} для пары ρ_{ExGAF} . При оценке релевантности результатов для новичков значения ρ_{12} выше и для ρ_{ExBing} и для ρ_{ExGAF} .

Далее для результатов поиска Bing и GAF/Bing вычислялись нормализованные значения $NDCG = \frac{DCG}{Z}$, где Z –максимально возможное значение DCG для случая идеального ранжирования в соответствие с оценками эксперта. Показатель $NDCG$ принимает значения от 0 до 1.

Соотношение значений $NDCG$ для результатов поиска Bing ($NDCG*Bing$) и GAF/Bing ($NDCG*GAF$) с детализацией по релевантности для специалистов и новичков представлено на рис. 7.

Видно, что нормализованная метрика DCG в целом дает оценку качества ранжирования GAF/Bing выше (0,8%), чем ранжирования Bing.

Особенно это проявляется при оценке релевантности документов с позиций новичков.

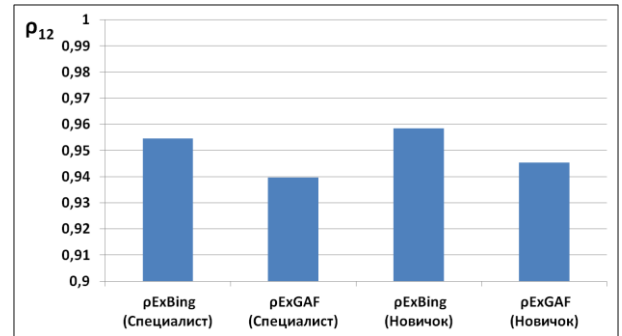


Рисунок 6 – Коэффициенты взаимной корреляции значений метрик DCG релевантных результатов поиска (для специалистов и новичков)

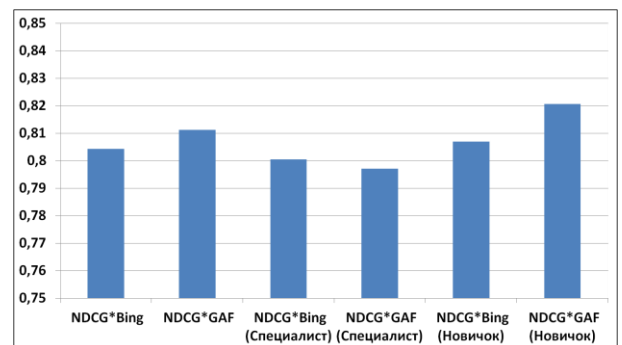


Рисунок 7 – Нормализованные значения метрики DCG

8. Выводы по результатам экспериментов

1. Точность поиска или доля релевантных результатов, найденных GAF/Bing, выше, чем у Bing в случае определения релевантности для новичков. Отметим, что точность поиска является очевидным наиболее значимым критерием оценки результатов поиска.

2. Процент совпадений результатов (адресов найденных документов) при поиске через GAF/Bing и Bing – 2%. Очевидно, что технология GAF/Bing позволяет находить новые оригинальные документы.

3. Средняя релевантность документов, найденных с использованием предлагаемой технологии GAF/Bing, практически не уступает средней релевантности документов, найденных Bing. Причем разрыв уменьшается для запросов, состоящих из лемм с возможностью словоформ.

4. Качество ранжирования результатов поиска GAF/Bing соответствует качеству ранжирования Bing. Значения коэффициента взаимной корреляции DCG для ранжирования GAF/Bing и экспертного ранжирования сопоставимы. То же можно сказать о значениях нормализованной метрики $NDCG$.

Заключение

Анализ результатов исследования

эффективности генетического алгоритма для тематического документального поиска показывает, что наибольший эффект от применения разработанной технологии достигается при поиске источников информации по новой для специалистов теме на начальных этапах ее изучения и освоения. При этом существенно увеличивается число уникальных и достаточно релевантных результатов поиска.

В дальнейших исследованиях авторы предполагают провести тестирование разработанной технологии на дорожках TREC, таких как поиск в Web [Collins-Thompson et al, 2014] и KBA [Frank et al, 2014].

Авторы выражают благодарность коллегам, принявшим непосредственное участие в этом исследовании в качестве экспертов: Гусарову А.А., Егеревой И.А., Зоренко Д.А., Кузьмину Д.Г., Образцову И.В., Панасенкову А.П., Петропавловской В.Б., Румянцеву Ю.П., Шматову Г.П.

Работа выполнена при финансовой поддержке РФФИ в рамках проекта № НК13-07-00342 "Интеллектуальная распределенная система информационной поддержки инноваций в науке и образовании".

Библиографический список

[Голицына, 2009] Голицына, О.Л. Информационные системы : учеб. пособие для вузов / О.Л. Голицына, Н.В. Максимов, И.И. Попов. – М. : ФОРУМ, 2009. – 496 с.

[Попов, 2009] Попов, С.В. Тематический поиск в интернете: назад в будущее / С.В. Попов // Информационное общество, 2009, № 6, с. 54-56.

[Палюх и др., 2013] Палюх, Б.В. Архитектура интеллектуальной системы информационной поддержки инноваций в науке и образовании / Б.В. Палюх, В.К. Иванов, А.Н. Сотников // Программные продукты и системы. – 2013. – № 4. – С. 197-202.

[Ivanov et al] Approaches to the Intelligent Subject Search / V. K. Ivanov, B. V. Palyukh, A.N. Sotnikov // Federated Conference on Computer Science and Information Systems (FedCSIS'2014) (September 7-10, 2014, Warsaw, Poland). Annals of Computer Science and Information Systems. Volume 3. Position Papers, DOI 10.15439/978-83-60810-57-6. - Warszawa, 2014. - P. 13-20.

[Иванов и др., 2014] Иванов, В.К. Прототип программной реализации генетического алгоритма для документального поиска / Б.В. Палюх, В.К. Иванов, П.И. Мескин // Четырнадцатая национальная конференция по искусственному интеллекту с международным участием КИИ-2014 (24-27 сентября 2014 г., г. Казань, Россия). Труды конференции. Т. 3. – Казань, 2014. – С. 191-199.

[Järvelin et al., 2002] Järvelin, K. Cumulated gain-based evaluation of IR techniques / K. Järvelin, J. Kekäläinen // ACM Transactions on Information Systems (TOIS) TOIS. - 2002. - Vol. 20, issue 4, October 2002. - P. 422-446. - DOI: <http://doi.acm.org/10.1145/582415.582418>.

[Агеев и др., 2010] Агеев, М. Российский семинар по Оценке Методов Информационного Поиска. Труды РОМИП 2010. Приложение А. Официальные метрики / М. Агеев, И. Кураленок, И. Некрестьянов. - Казань, 2010. - С. 172-187. - Режим доступа: http://romip.ru/romip2010/20_appendix_a_metrics.pdf.

[Айфичер, 2004] Айфичер, Э.С. Цифровая обработка сигналов: практический подход. / Э.С. Айфичер, Б.У. Джервис // Второе издание. Пер. с англ. - М.: Изд. Дом "Вильямс", 2004. – 992 с.

[Collins-Thompson et al, 2014] Collins-Thompson, K. TREC 2013 Web Track Overview / K. Collins-Thompson, P. Bennett,

F. Diaz, C. L. A. Clarke, E.M. Voorhees // The Twenty-Second Text REtrieval Conference (TREC 2013) Proceedings (January 30, 2014). - URL: <http://trec.nist.gov/pubs/trec22/papers/WEB.OVERVIEW.pdf>.

[Frank et al, 2014] Frank, J.R. Evaluating Stream Filtering for Entity Profile Updates for TREC 2013 (KBA Track Overview) / J.R. Frank, S.J. Bauer, M. Kleiman-Weiner, D.A. Roberts, N. Tripuraneni, C. Zhang, C. Ré, E. Voorhees, I. Soboroff // The Twenty-Second Text REtrieval Conference (TREC 2013) Proceedings (January 30, 2014). –

URL: <http://trec.nist.gov/pubs/trec22/papers/KBA.OVERVIEW.pdf>.

STUDY THE EFFECTIVENESS OF GENETIC ALGORITHM FOR DOCUMENTARY SUBJECT SEARCH

Ivanov V.K., Palyukh B.V.

Tver State Technical University, Tver, Russia

mtivk@tstu.tver.ru

pboris@tstu.tver.ru

This article presents results of experimental studies the effectiveness of the genetic algorithm that was applied to effective queries creation and relevant document selection. Studies were carried out to the comparative analysis of the semantic relevance and quality ranking of the documents found on the Internet in various ways. Analysis of the results shows that the greatest effect of presented technology is achieved by finding new documents for skilled users in the initial stages of the study of the topic. Additionally, the number of unique and relevant results is significantly increased.

Introduction

One of the main difficulties during the subject search in documentary data warehouses is correct search queries formulating that should provide acceptable accuracies level. This article presents results of experimental studies the effectiveness of the genetic algorithm that was applied to effective queries creation and relevant document selection.

Main Part

The genetic algorithm for filtering and ranking the search results has been implemented previously. Source data for the experiments were prepared with twelve experts together.

Next, the initial search queries were formulated, algorithm parameters were assigned and large series of query generations and search operations were performed. The results were processed by involving experts.

In conclusion the comparative analysis the relevance of search results and ranking quality were performed.

Conclusion

Analysis of the results shows that the greatest effect of presented technology is achieved by finding new documents for skilled users in the initial stages of the study of the topic. Additionally, the number of unique and relevant results is significantly increased.



УДК 004.822:514

МОДЕЛИРОВАНИЕ РАБОТЫ АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ УПРАВЛЕНИЯ КАЧЕСТВОМ СЕТЯМИ ПЕТРИ

Бурдо Г. Б., Виноградов Г. П., Сорокин А. Ю.

*Тверской государственной технической университет,
г. Тверь, Россия*

gbtms@yandex.ru

wgp272@mail.ru

alexeySOROKIN13@mail.ru

В данной работе рассмотрена проблема моделирования автоматизированной системы управления качеством сетями Петри. Рассмотрены логико-временные алгоритмы процессов управления качеством на стадиях жизненного цикла изделия. Показана возможность использования расширенных сетей Петри для моделирования параллельных процессов системы.

Ключевые слова: жизненный цикл изделия, критерии качества изделия, сети Петри.

ВВЕДЕНИЕ

Качество продукции является важнейшим фактором, влияющим на конкурентоспособность предприятия. Ранее, для достижения желаемого уровня качества продукции, было достаточно контролировать его на отдельных этапах жизненного цикла изделия (ЖЦИ). В настоящее время для обеспечения качества продукции необходимо управлять качеством [Колчин А.Ф. и др. 2002] на каждом этапе и стадии ЖЦИ (рис.1).

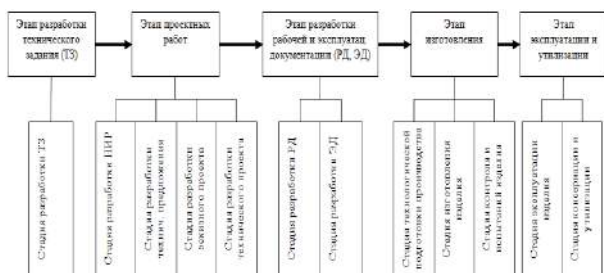


Рисунок 1 - Основные этапы и стадии ЖЦИ

С целью управления качеством продукции на всех этапах жизненного цикла изделия в каждом предприятии создается *система управления качеством (СУК)* продукции, функционирование которой определяется стандартами серии ИСО 9000.

Поэтому, учитывая необходимость оперативного внесения корректив в качественные показатели продукции и широкий спектр выпускаемой

продукции современным машиностроительным предприятием, создание автоматизированных систем управления качеством является актуальной задачей.

1. ОСНОВНАЯ ЧАСТЬ

В автоматизированной системе управления качеством продукции (АСУ КП) оценка качества изделия ведется на основе количественных и качественных показателей качества и экспертных оценок. Эта информация может быть нечеткой и недостаточно определенной для того, чтобы быть выраженной математическими зависимостями. Адекватная обработка неопределенной информации, где входные данные не являются точными, невозможна традиционными методами, так как не позволяет учесть присущую этим данным неопределенность. При этом нечеткая логика и теория нечетких множеств являются эффективным подходом в решении данной проблемы. Поэтому для описания функционирования АСУ КП эффективно применить нечеткие сети Петри.

Нечеткие сети Петри позволяют дать адекватное представление и анализ структуры динамических дискретных моделей сложных систем и логико-временных особенностей процессов их функционирования.

На основе ранее разработанных теоретико-множественная модели системы управления качеством, и системы критериев оценки качества

[Бурдо Г. Б. и др. 2014] была смоделирована сеть Петри для АСУ КП. (рис. 2).

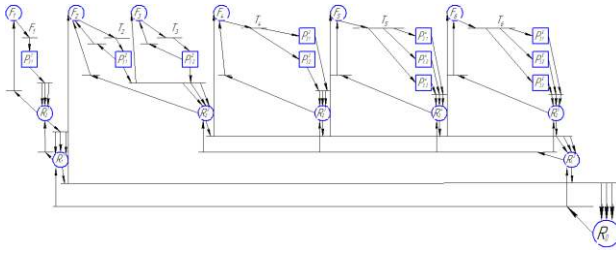


Рисунок 2 - Нечеткая сеть Петри на этапах разработки технического задания и проектных работ в АСУ КП

Здесь нечеткая сеть Петри C_f , в соответствии с [Блюмин С. Л. и др. 2010] и [Борисов В. В.], определяется как $C_f = (N, \hat{M}_0, \kappa)$, где:

1. $N = (P, T, I, O)$ – структура нечеткой сети Петри, для которой: $P = \{F, K, R\}$ – непустое конечное множество позиций, условия для активации процессов; T – непустое конечное множество переходов, протекающие процессы в системе; $I: P \times T \rightarrow \{0,1\}$ – входная функция переходов; $O: T \times P \rightarrow \{0,1\}$ – выходная функция переходов.

2. \hat{M}_0 – матрица начальной маркировки ($n \times m$). Каждый её элемент $m_{ij}^0 = [0,1] (\forall i \in \{1,2,\dots,n\}, \forall j \in J)$ равен значению функции принадлежности наличия $(j-1)$ числа маркеров в позиции p_i на момент начала запуска нечеткой сети Петри.

3. $\kappa = [\kappa_1, \kappa_2, \kappa_3]$ – множество правил, модификация которых отражает специфику введенной нечеткости в начальную маркировку.

Динамика изменения маркировок нечеткой сети Петри типа $C_f = (N, \hat{M}_0, \kappa)$ после момента её запуска подчиняется следующим правилам, отражающим специфику введенной нечеткости в маркировку.

1) κ_1 : правило определения текущего состояния.

Любое текущее состояние нечеткой сети Петри C_f определяется некоторой его маркировкой, т.е. матрицей \hat{M}_0 размерности ($n \times m$), каждый компонент которой $m_{ij} \in [0,1]$, и интерпретируется как значения функции принадлежности нечеткого наличия $\{0,1,2,\dots,d\}$ маркеров в соответствующих позициях $p_i \in P$ нечеткой сети Петри C_f . Начальное состояние НСП C_f определяется вектором начальной маркировки \hat{M}_0 .

2) κ_2 : правило (условие) активности перехода.

Переход $t_i \in T$ нечеткой сети Петри C_f называется активным (разрешенным, возбужденным) при некоторой текущей маркировке \hat{M} , если выполнено следующее условие:

$$\sigma_i \geq I(p_i, t_k), i=1, \dots, n$$

где $\sigma_i = \max_{(j \in J) \wedge (m_j > 0)} (\{j\} - 1), i=1, \dots, n$

То есть σ_i равно индексу максимального, отличного от нуля, значения степени принадлежности наличия маркеров в позициях $p_i \in P$ нечеткой сети Петри. При этом некоторый переход $t_i \in T$ является активным, если имеется нечеткое количество маркеров, больше или равное числу дуг, соединяющих соответствующие входные позиции с рассматриваемым переходом нечеткой сети Петри.

3) κ_3 : правило нечеткого срабатывания перехода.

Если переход $t_k \in T$ нечеткой сети Петри C_f является активным при некоторой текущей маркировке \hat{M} , т.е. выполняется правило активности переходов κ_2 . То нечеткое срабатывание данного перехода приводит к новой маркировке, элементы которой определяются следующим образом:

1. для каждой из входных позиций $p_i \in P$, для которых $I(p_i, t_k) > 0$:

$$m_{i1}^{l+1} = \max_{(j \in \{1,2,\dots,l(p_i, t_k)+1\})} \{m_{ij}\}$$

$$m_{ij}^{l+1} = m_{i, j+1(p_i, t_k)}, j=2, \dots, m$$

2. для каждой из выходных позиций $p_j \in P$, для которых $O(t_k, p_i) > 0$:

$$m_{ij}^{l+1} = \min \{m_{ij}, 1 - q_k\}, j=1, 2, \dots, O(t_k, p_i)$$

$$m_{i1}^{l+1} = \max \left\{ \min_{(\forall j \in J) \wedge (j > O(t_k, p_i))} [m_{ij}, 1 - q_k], \min [m_{i, j-O(t_k, p_i)}, q_k] \right\}$$

где q_k – степень принадлежности, характеризующая возможность нечеткого срабатывания перехода t_k , рассчитываемая следующим образом:

$$q_k = \min_{(i \in \{1,2,\dots,n\})} \left\{ \max_{(j \in J) \wedge (j > I(p_i, t_k))} [m_{ij}] \right\}, (\forall t_k \in T)$$

3. если некоторые позиции $p_i \in P$ являются одновременно входными и выходными для активного перехода t_k , то для них элементы матрицы новой маркировки рассчитываются последовательно, сначала для входных позиций, затем для выходных.

ЗАКЛЮЧЕНИЕ

Представленная расширенная сеть Петри для автоматизированной системы управления качеством продукции позволяет представить алгоритм работы системы. Показаны логико-временные особенности процессов управления качеством на стадиях жизненного цикла изделия. Расширенная сеть Петри позволяет дать анализ структуры критериев динамической модели системы управления качеством. Следующим шагом становится разработка методики расчета значений комплексных критериев на основе нечетких множеств.

(Работа выполнена при финансовой поддержке РФФИ, проект № 14-01-00324).

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

[Блюмин С. Л. и др. 2010] Окрестностное моделирование сетей Петри: монография / С.Л. Блюмин, А.М. Шмырин, И.А. Седых, В.Ю. Филоненко. - Липецк: ЛЭГИ, 2010. - 124 с.

[Борисов В. В.] Борисов В. В., Круглов В. В., Федулов А. С. Нечеткие модели и сети. – М.: Горячая линия – Телеком, 2007. – 284с.: ил.

[Колчин А.Ф. и др. 2002] Колчин А. Ф., Овсянников М. В., Стрекалов А. Ф., Сумарков С. В. Управление жизненным циклом продукции. – М.: Анархист, 2002. – 304 с.

[Бурдо Г. Б. и др. 2014] Бурдо Г. Б., Сорокин А. Ю. Модели и критерии автоматизированной системы управления качеством наукоемкой машиностроительной продукции / Бурдо Г. Б., Сорокин А. Ю. // Гибридные и синергитические интеллектуальные системы. Материалы 2 международного Поспеловского симпозиума. С. 78-86.

MODELING OF AUTOMATED QUALITY CONTROL BY PETRI NETS

Burdo G.B., Vinogradov G. P., Sorokin A. Y.

Tver State Technical University (TvSTU), Tver, Russia

gbtms@yandex.ru

wgp272@mail.ru

alexeySOROKIN13@mail.ru

In this paper we consider the problem of modeling the automated system of quality control by Petri nets. Considered logical-time algorithms of quality management processes at all stages of the product life cycle. The possibility of using extended Petri nets for modeling concurrent processes in the system.

Key words: the life cycle of products, quality criteria of the product, the Petri nets.

INTRODUCTION

Product quality is the most important factor affecting the competitiveness of the enterprise. At the present time to provide the quality products you need to control the quality at every stage and phase of the product life cycle. The purpose of quality control at all stages of the product life cycle in every enterprise creates a quality management system (QMS) products, the operation of which is determined by the standards of series ISO 9000.

MAIN PART

In the automated system of quality control (automatic gearbox) assessment of the quality of products is conducted on the basis of quantitative and qualitative indicators of quality and expertise. This information can be vague and not sufficiently specific to be expressed by the mathematical dependencies. Adequate processing of uncertain information, where the input data are not accurate, and impossible by traditional methods, as it allows to take into account the inherent data uncertainty. While fuzzy logic and fuzzy set theory is an effective approach in solving this problem. Therefore, to describe the functioning of the automatic gearbox to effectively apply fuzzy Petri nets.

Fuzzy Petri nets allow to give adequate representation and analysis of the structure of dynamic discrete models of complex systems and logic-time characteristics of the processes of their operation

CONCLUSION

Presents an extended Petri net for the automated quality control of products allows us to represent the algorithm. Shown logico-temporal features of quality management processes at all stages of the product life cycle. Extended Petri net allows you to give an analysis of the structure of the criteria of the dynamic model of quality management system. The next step to become the development of methods of calculation of complex criteria based on fuzzy sets.



OSTIS-2015

(Open Semantic Technologies for Intelligent Systems)

УДК 004.89

ПРИМЕНЕНИЕ НЕЙРОННЫХ СЕТЕЙ ГЛУБОКОГО ДОВЕРИЯ ДЛЯ ВЫДЕЛЕНИЯ СЕМАНТИЧЕСКИ ЗНАЧИМЫХ ПРИЗНАКОВ

Головки В.А., Крощенко А.А.

*Брестский государственный технический университет,
г. Брест, Республика Беларусь*

gva@bstu.by

kroschenko@gmail.com

В работе приводятся основные принципы построения и обучения нейронной сети глубокого доверия для выделения семантически значимых признаков на основе выборки CIFAR-10. Для предобучения нейронной сети глубокого доверия применяется разработанный подход, базирующийся на минимизации ошибки реконструкции видимых и скрытых образов для ограниченной машины Больцмана (RBM).

Ключевые слова: Семантическая сеть, нейронная сеть глубокого доверия, ограниченная машина Больцмана, ошибка реконструкции, метод обратного распространения ошибки

Введение

Семантические и нейронные сети являются моделями представления знаний. При этом если семантическая сеть хранит знания об объектах и процессах окружающего мира в виде структуры с связями и отношениями, то нейронная сеть оперирует их численными свойствами. Возникает важный вопрос: каким образом возможно использовать аппарат нейронных сетей в построении семантических моделей? Ответить на данный вопрос помогут новые идеи и подходы, появившиеся в теории нейронных сетей в последнее десятилетие, связанные с нейронными сетями глубокого доверия.

Начиная с 2006 г., благодаря работам [Hinton et al., 2006a], [Hinton, 2002], [Hinton et al., 2006b], [Hinton, 2010] Хинтона (J. Hinton), начался обратный отсчет в развитии многослойных перцептронов под новым названием - нейронные сети глубокого доверия (deep belief neural networks). Такие сети в общем случае представляют собой дальнейшее развитие многослойных перцептронов и интегрируют различные парадигмы обучения нейронных сетей. Благодаря своей многослойной архитектуре, они позволяют обрабатывать и анализировать большой объем данных, а также моделировать когнитивные процессы в различных областях. В настоящее время большинство высокотехнологичных компаний в США (Microsoft, Google и т.д.) используют нейронные сети глубокого доверия для проектирования различных

интеллектуальных систем. По версии ученых Массачусетского технологического института (США), нейронные сети глубокого доверия (deep belief neural networks), входят в список 10 наиболее прорывных высоких технологий, способных в недалеком будущем в значительной степени преобразить повседневную жизнь большинства людей на нашей планете. В настоящее время нейронные сети глубокого доверия считаются революционным шагом в области интеллектуальной обработки данных.

В данной статье рассмотрена одна из основных моделей обучения нейронных сетей глубокого доверия, базирующаяся на ограниченной машине Больцмана (restricted Boltzmann machine (RBM)). Предложен новый метод для обучения ограниченной машины Больцмана и показано, что правило обучения ограниченной машины Больцмана является частным случаем предложенного метода обучения, который базируется на минимизации суммарной квадратичной ошибки восстановления информации. Доказана эквивалентность максимизации вероятности распределения данных в ограниченной машине Больцмана и минимизация суммарной квадратичной ошибки восстановления информации в слоях (RBM).

Рассматривается применение нейронных сетей глубокого доверия для решения задачи выделения семантически значимых признаков на примере выборки CIFAR-10.

1. Нейронные сети глубокого доверия

Нейронная сеть глубокого доверия содержит множество скрытых слоев (рисунок 1) и осуществляет глубокое иерархическое преобразование входного пространства образов.

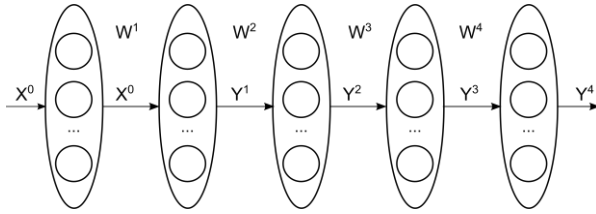


Рисунок 1 - Нейронная сеть глубокого доверия

Выходное значение j -го нейрона k -го слоя определяется следующим образом:

$$y_j^k = F(S_j^k), \tag{1}$$

$$S_j^k = \sum_{i=1} w_{ij}^k y_i^{k-1} + T_j^k, \tag{2}$$

где F – функция активации нейронного элемента, S_j^k – взвешенная сумма j -го нейрона k -слоя, w_{ij}^k – весовой коэффициент между i -ым нейроном ($k-1$)-го слоя и j -м нейроном k -го слоя, T_j^k – пороговое значение j -го нейрона k -го слоя.

Для первого (распределительного) слоя

$$y_i^0 = x_i. \tag{3}$$

В матричном виде выходной вектор k -го слоя

$$Y^k = F(S^k) = F(W^k Y^{k-1} + T^k), \tag{4}$$

где W – матрица весовых коэффициентов, Y^{k-1} – выходной вектор ($k-1$)-го слоя, T^k – вектор пороговых значений нейронов k -го слоя. Если нейронная сеть глубокого доверия используется для классификации образов, то выходные значения сети часто определяются на основе функции активации **softmax**:

$$y_j^F = \text{softmax}(S_j) = \frac{e^{s_j}}{\sum_l e^{s_l}} \tag{5}$$

Процесс обучения нейронных сетей глубокого доверия в общем случае состоит из двух этапов: предобучение нейронной сети методом послойного обучения, начиная с первого слоя (pre-training). Данное обучение осуществляется без учителя. Настройка синаптических связей всей сети (fine-tuning) при помощи алгоритма обратного распространения ошибки или алгоритма «бодрствования и сна» (wake-sleep algorithm).

Важным этапом обучения нейронных сетей глубокого доверия является предобучение слоев нейронной сети. Один из основных подходов к

предобучению – метод, базирующийся на представлении каждого слоя нейронной сети в виде ограниченной машины Больцмана (RBM).

1.1. Ограниченная машина Больцмана

Ограниченная машина Больцмана состоит из двух слоев стохастических бинарных нейронных элементов, которые соединены между собой двунаправленными симметричными связями (рисунок 2). Входной слой нейронных элементов называется видимым (слой X), а второй слой называется скрытым (слой Y). Нейронную сеть глубокого доверия можно представить как совокупность ограниченных машин Больцмана. Ограниченная машина Больцмана может генерировать (представить) любое дискретное распределение, если используется достаточное количество нейронов скрытого слоя [Bengio, 2009].

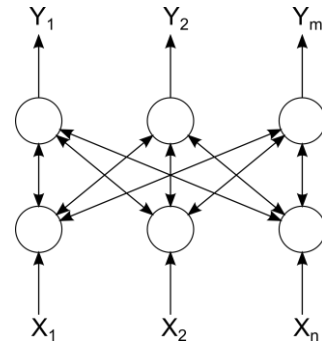


Рисунок 2 - Ограниченная машина Больцмана

Данная сеть является стохастической нейронной сетью, в которой состояния видимых и скрытых нейронов меняются в соответствии с вероятностной версией сигмоидной функции активации:

$$p(y_j | x) = \frac{1}{1 + e^{-S_j}}, S_j = \sum_i w_{ij} x_i + T_j \tag{6}$$

$$p(x_i | y) = \frac{1}{1 + e^{-S_i}}, S_i = \sum_j w_{ij} y_j + T_i \tag{7}$$

Состояния видимых и скрытых нейронных элементов принимаются независимыми:

$$P(x | y) = \prod_{i=1}^n P(x_i | y)$$

$$P(y | x) = \prod_{j=1}^m P(y_j | x)$$

Таким образом, состояния всех нейронных элементов ограниченной машины Больцмана определяются через распределение вероятностей. В RBM нейроны скрытого слоя являются детекторами признаков, которые сохраняют закономерности входных данных. Основная задача обучения состоит в воспроизведении распределения входных данных на основе состояний нейронов скрытого слоя как можно точнее. Это эквивалентно максимизации функции правдоподобия путем модификации синаптических связей нейронной сети. Рассмотрим

это подробнее. Вероятность нахождения видимого и скрытого нейрона в состоянии (x, y) определяется на основе распределения Гиббса:

$$P(x, y) = \frac{e^{-E(x, y)}}{Z}$$

где $E(x, y)$ – энергия системы в состоянии (x, y) , Z – параметр, который определяет условие нормализации вероятностей, то есть, чтобы сумма вероятностей равнялась единице. Данный параметр определяется следующим образом:

$$Z = \sum_{x, y} e^{-E(x, y)}$$

Вероятность нахождения видимых нейронов в определенном состоянии равняется сумме вероятностей конфигураций $P(x, y)$ по состояниям скрытых нейронов:

$$P(x) = \sum_y P(x, y) = \sum_y \frac{e^{-E(x, y)}}{Z} = \frac{\sum_y e^{-E(x, y)}}{\sum_{x, y} e^{-E(x, y)}}$$

Для нахождения правила модификации синаптических связей необходимо максимизировать вероятность воспроизведения состояний видимых нейронов $P(x)$ ограниченной машиной Больцмана. Для того, чтобы определить максимум функции правдоподобия распределения данных $P(x)$ будем использовать метод градиентного спуска в пространстве весовых коэффициентов и пороговых значений сети, где в качестве градиента применим функцию логарифмического правдоподобия:

$$\ln P(x) = \ln \sum_y e^{-E(x, y)} - \ln \sum_{x, y} e^{-E(x, y)}$$

Тогда градиент равен

$$\frac{\partial \ln P(x)}{\partial w_{ij}} = \frac{\partial}{\partial w_{ij}} \ln \sum_y e^{-E(x, y)} - \frac{\partial}{\partial w_{ij}} \ln \sum_{x, y} e^{-E(x, y)}$$

Преобразуя последнее выражение, получим

$$\begin{aligned} \frac{\partial \ln P(x)}{\partial w_{ij}} &= - \frac{1}{\sum_y e^{-E(x, y)}} \sum_y e^{-E(x, y)} \frac{\partial E(x, y)}{\partial w_{ij}} + \\ &+ \frac{1}{\sum_{x, y} e^{-E(x, y)}} \sum_{x, y} e^{-E(x, y)} \frac{\partial E(x, y)}{\partial w_{ij}} \end{aligned}$$

Так как

$$P(x, y) = P(y | x)P(x)$$

то

$$P(y | x) = \frac{P(x, y)}{P(x)} = \frac{(1/Z)e^{-E(x, y)}}{(1/Z)\sum_y e^{-E(x, y)}} = \frac{e^{-E(x, y)}}{\sum_y e^{-E(x, y)}}$$

В результате можно получить следующее выражение:

$$\begin{aligned} \frac{\partial \ln P(x)}{\partial w_{ij}} &= - \sum_y P(y | x) \frac{\partial E(x, y)}{\partial w_{ij}} + \\ &+ \sum_{x, y} P(x, y) \frac{\partial E(x, y)}{\partial w_{ij}} \end{aligned}$$

В данном выражении первое слагаемое определяет позитивную фазу работы машины Больцмана, когда сеть работает на основе образов из обучающей выборки. Второе слагаемое характеризует негативную фазу функционирования, когда сеть работает в свободном режиме независимо от окружающей среды. Рассмотрим энергию сети RBM. С точки зрения энергии сети задача обучения состоит в том, чтобы на основе входных данных найти конфигурацию выходных переменных с минимальной энергией. В результате на обучающем множестве сеть будет иметь меньшую энергию по сравнению с другими состояниями. Функция энергии бинарного состояния (x, y) определяется аналогично сети Хопфилда:

$$E(x, y) = - \sum_i x_i T_i - \sum_j y_j T_j - \sum_{i, j} x_i y_j w_{ij} \quad (8)$$

В этом случае

$$\frac{\partial E(x, y)}{\partial w_{ij}} = -x_i y_j$$

и

$$\frac{\partial \ln P(x)}{\partial w_{ij}} = \sum_y P(y | x) x_i y_j - \sum_{x, y} P(x, y) x_i y_j$$

Так как математическое ожидание равняется:

$$E(x) = \sum_i x_i P_i$$

то

$$\frac{\partial \ln P(x)}{\partial w_{ij}} = E[x_i y_j]_{data} - E[x_i y_j]_{model}$$

Аналогичным образом можно получить градиенты для пороговых значений:

$$\frac{\partial \ln P(x)}{\partial T_i} = E[x_i]_{data} - E[x_i]_{model}$$

$$\frac{\partial \ln P(x)}{\partial T_j} = E[y_j]_{data} - E[y_j]_{model}$$

Как следует из последних выражений, первое слагаемое характеризует работу сети на основе данных из обучающей выборки, а второе слагаемое – на основе данных модели (данные генерируемые сетью), то есть в свободном режиме независимо от окружающей среды. Так как вычисление математического ожидания на основе RBM сети

является очень сложным, Хинтон предложил использовать аппроксимацию данных слагаемых, которую он назвал контрастным расхождением (contrastive divergence (CD)) [Hinton et al., 2006a]. Такая аппроксимация основывается на дискретизаторе Гиббса (Gibbs sampling). В этом случае первые слагаемые в выражениях для градиента характеризуют распределение данных в момент времени $t = 0$, а вторые слагаемые характеризуют реконструированные или генерируемые моделью состояния в момент времени $t = k$. Исходя из этого, CD-k процедура может быть представлена следующим образом:

$$x(0) \rightarrow y(0) \rightarrow x(1) \rightarrow y(1) \rightarrow \dots \rightarrow x(k) \rightarrow y(k)$$

В результате можно получить следующие правила для обучения RBM сети. В случае применения CD-1, $k=1$ и учитывая, что в соответствии с методом градиентного спуска

$$w_{ij}(t+1) = w_{ij}(t) + \alpha \frac{\partial \ln P(x)}{\partial w_{ij}(t)}$$

Можно получить, что

$$w_{ij}(t+1) = w_{ij}(t) + \alpha(x_i(0)y_j(0) - x_i(1)y_j(1))$$

$$T_i(t+1) = T_i(t) + \alpha(x_i(0) - x_i(1))$$

$$T_j(t+1) = T_j(t) + \alpha(y_j(0) - y_j(1)).$$

Аналогичным образом, для алгоритма CD-k

$$w_{ij}(t+1) = w_{ij}(t) + \alpha(x_i(0)y_j(0) - x_i(k)y_j(k))$$

$$T_i(t+1) = T_i(t) + \alpha(x_i(0) - x_i(k))$$

$$T_j(t+1) = T_j(t) + \alpha(y_j(0) - y_j(k)).$$

Из последних выражений видно, что правила обучения ограниченной машины Больцмана минимизируют разницу между оригинальными данными и данными генерируемыми моделью. Генерируемые моделью данные получаются при помощи сэмплирования Гиббса.

1.2. Обучение нейронной сети глубокого доверия на основе RBM

Обучение нейронной сети глубокого доверия происходит на основе «жадного» алгоритма послойного обучения (greedy layer-wise algorithm). В соответствии с ним вначале обучается первый слой сети как RBM машина. Для этого входные данные поступают на видимый слой нейронных элементов и используя CD-k процедуру вычисляются состояния скрытых $p(y|x)$ и видимых нейронов $p(x|y)$. В процессе выполнения данной процедуры (не более 100 эпох) изменяются весовые коэффициенты и пороговые значения RBM сети, которые затем фиксируются. Затем берется второй слой нейронной сети и конструируется RBM машина. Входными данными для нее являются данные с предыдущего слоя. Происходит обучение и процесс продолжается для всех слоев нейронной сети [Hinton, 2009]. В результате такого обучения без учителя можно получить подходящую

начальную инициализацию настраиваемых параметров сети глубокого доверия. На заключительном этапе осуществляется точная настройка параметров всей сети при помощи алгоритма обратного распространения ошибки или алгоритма «бодрствования и сна» (wake-sleep algorithm).

1.3. Альтернативный взгляд на ограниченную машину

В данном разделе рассматривается альтернативный взгляд на ограниченную машину Больцмана как автоассоциативную нейронную сеть, которая может функционировать с любыми данными, как бинарными, так и числовыми. Предлагается новый метод для получения правила обучения ограниченной машины Больцмана [Golovko et al., 2014]. Он базируется на минимизации ошибки реконструкции видимых и скрытых образов, которую можно получить, используя итерации сэмплирования Гиббса. По сравнению с традиционным подходом, основанным на энергии методе (energy-based method), который базируется на линейном представлении нейронных элементов, предложенный метод позволяет учитывать нелинейную природу нейронных элементов. Рассмотрим ограниченную машину Больцмана, которую будем представлять в виде трех слоев нейронных элементов [Golovko et al., 2012]: видимый, скрытый и видимый (рисунок 3).

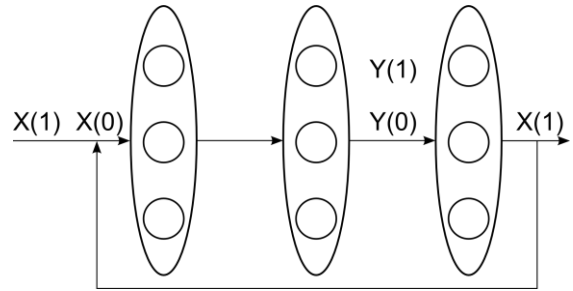


Рисунок 3 - Представление RBM в виде рециркуляционной сети

Сэмплирование Гиббса заключается в следующей процедуре. Пусть $x(0)$ входной вектор, который поступает на видимый слой в момент времени 0. Тогда выходные значения нейронов скрытого слоя:

$$y_j(0) = F(S_j(0)), \quad (9)$$

$$S_j(0) = \sum_i w_{ij} x_i(0) + T_j. \quad (10)$$

Инверсный (последний) слой реконструирует входной вектор на основе данных со скрытого слоя. В результате получается восстановленный вектор $x(1)$ в момент времени 1:

$$x_i(1) = F(S_i(1)), \quad (11)$$

$$S_i(1) = \sum_j w_{ij} y_j(0) + T_i. \quad (12)$$

Затем вектор $x(1)$ поступает на видимый слой, и вычисляются выходные значения нейронов скрытого слоя:

$$y_j(1) = F(S_j(1)), \quad (13)$$

$$S_j(1) = \sum_i w_{ij} x_i(1) + T_j. \quad (14)$$

Продолжая данный процесс, можно получить на шаге k

$$y_i(k) = F(S_j(k)), S_j(k) = \sum_i w_{ij} x_i(k) + T_j.$$

$$x_i(k) = F(S_i(k)), S_i(k) = \sum_j w_{ij} y_j(k-1) + T_i.$$

Целью обучения ограниченной машины Больцмана является минимизация суммарной квадратичной ошибки реконструкции данных на скрытом и восстанавливающем слое, которая в случае CD- k определяется следующим образом:

$$E_s = \frac{1}{2} \sum_{l=1}^L \sum_{j=1}^m \sum_{p=1}^k (y_j^l(p) - y_j^l(p-1))^2 + \frac{1}{2} \sum_{l=1}^L \sum_{i=1}^n \sum_{p=1}^k (x_i^l(p) - x_i^l(p-1))^2$$

В случае CD-1 суммарная квадратичная ошибка

$$E_s = \frac{1}{2} \sum_{l=1}^L \sum_{j=1}^m (y_j^l(1) - y_j^l(0))^2 + \frac{1}{2} \sum_{l=1}^L \sum_{i=1}^n (x_i^l(1) - x_i^l(0))^2$$

где L количество входных образов. Как следует из приведенных выше выражений ошибка состоит из двух частей: ошибки восстановления информации на скрытом и выходном слое.

Теорема 1. Максимизация функции правдоподобия распределения данных $P(x)$ в пространстве синаптических связей ограниченной машины Больцмана эквивалентна минимизации суммарной квадратичной ошибки сети в том же пространстве при использовании линейных нейронов.

Следствие 1. Линейная ограниченная машина Больцмана с точки зрения обучения эквивалентна автоассоциативной нейронной сети при использовании в ней при обучении сэмпирования Гиббса.

Следствие 2. Для нелинейной ограниченной машины Больцмана правило модификации синаптических связей в случае CD- k будет следующим:

$$w_{ij}(t+1) = w_{ij}(t) -$$

$$-\alpha \left(\sum_{p=1}^k (y_j(p) - y_j(p-1)) x_i(p) F'(S_j(p)) + (x_i(p) - x_i(p-1)) y_j(p-1) F'(S_i(p)) \right)$$

$$T_j(t+1) = T_j(t) - \alpha \left(\sum_{p=1}^k (y_j(p) - y_j(p-1)) F'(S_j(p)) \right),$$

$$T_i(t+1) = T_i(t) - \alpha \left(\sum_{p=1}^k (x_i(p) - x_i(p-1)) F'(S_i(p)) \right),$$

Следствие 3. Для нелинейной ограниченной машины Больцмана правило модификации синаптических связей в случае CD-1 будет следующим:

$$w_{ij}(t+1) = w_{ij}(t) -$$

$$\alpha ((y_j(1) - y_j(0)) F'(S_j(1)) x_i(1) + (x_i(1) - x_i(0)) F'(S_i(1)) y_j(0)),$$

$$T_i(t+1) = T_i(t) - \alpha (x_i(1) - x_i(0)) F'(S_i(1)),$$

$$T_j(t+1) = T_j(t) - \alpha (y_j(1) - y_j(0)) F'(S_j(1)).$$

Если используется групповое обучение (batch learning), то в этом случае метод градиентного спуска записывается следующим образом:

$$w_{ij}(t+1) = w_{ij}(t) - \alpha \frac{\partial E_s}{\partial w_{ij}(t)} \quad (15)$$

$$T_i(t+1) = T_i(t) - \alpha \frac{\partial E_s}{\partial T_i(t)} \quad (16)$$

$$T_j(t+1) = T_j(t) - \alpha \frac{\partial E_s}{\partial T_j(t)} \quad (17)$$

В данном разделе получены правила обучения для ограниченной машины Больцмана, которые базируются на минимизации квадратичной ошибки восстановления информации в скрытом и видимом слоях. Предложенный метод позволяет учитывать нелинейную природу нейронных элементов. Показано, что классические выражения для обучения ограниченной машины являются частным случаем предложенного метода. Доказана теорема об эквивалентности максимизации функции правдоподобия распределения данных $P(x)$ в пространстве синаптических связей и минимизации суммарной квадратичной ошибки сети в том же пространстве для линейной ограниченной машины Больцмана.

2. Применение нейронных сетей глубокого доверия

2.1. Постановка задачи и условия проведения эксперимента

Для анализа эффективности обучения разработанных алгоритмов была выбрана известная выборка CIFAR-10. Эта выборка состоит из 60000 изображений размерностью 32X32 различных объектов, принадлежащих 10 основным классам: 1.

самолет, 2. автомобиль, 3. птица, 4. кот, 5. олень, 6. собака, 7. лягушка, 8. лошадь, 9. корабль и 10. грузовик.

Для обучения распознавания объектов данных классов использовалась нейронная сеть глубокого доверия архитектуры 3072 – 1500 – 750 – 10. Из полной выборки было взято 20000 изображений для формирования выборки обучения. Основные параметры обучения: скорость обучения – 0,03 (на этапе предобучения и fine-tuning), размер mini-batch – 200, количество эпох предобучения – 10, fine-tuning – 50.

После выполнения обучения данной нейронной сети глубокого доверия на выходном слое были оценены расстояния между векторами весовых коэффициентов.

2.2. Постановка задачи и условия проведения эксперимента

Данные о расстояниях между весовыми коэффициентами на последнем (выходном) слое нейронной сети были упорядочены, прогнандированы и сведены в следующую таблицу псевдорасстояний:

Таблица 1 – Псевдорасстояния для классов CIFAR

0	37	16	20	2	17	21	7	1	34
	0	42	36	41	38	45	44	26	15
		0	33	11	24	18	23	13	30
			0	8	3	10	32	29	35
				0	4	5	12	6	40
					0	14	22	28	39
						0	27	25	43
							0	19	31
								0	9
									0

Исследуя полученную таблицу, можно заметить, что близкими семантическими признаками обладают такие классы как, например, самолет и корабль, самолет и олень, кот и собака, олень и собака, олень и лягушка. При этом наименьшими общими признаками обладают классы автомобиль и лошадь, автомобиль и лягушка.

Заключение

С развитием областей искусственного интеллекта наблюдается появление различных возможностей для интеграции идей и их совместного использования в рамках общих теорий. К примеру, вполне возможно использование нейронных сетей, а в частности, нейронных сетей глубокого доверия (как обладающий потенциалом в обработке больших объемов данных) в задачах выделения семантически значимых признаков и их

анализа. Как правило, при этом получаемые данные способствуют определению скрытых, зачастую плохо выявляемых при ручном анализе, общих признаков.

В данной статье предлагается новый подход к предобучению нейронных сетей глубокого доверия, а также приводится пример его использования для предобучения сети классификации образов. Параметры полученной сети анализируются с целью определения семантически значимых признаков.

Библиографический список

- [Hinton et al., 2006a] Hinton, G. A fast learning algorithm for deep belief nets / G. Hinton, S. Osindero, Y. Teh // *Neural Computation*. – 2006. – Vol. 18. – P. 1527–1554.
- [Hinton, 2002] Hinton, G. Training products of experts by minimizing contrastive divergence // *Neural Computation*. – 2002. – Vol. 14. – P. 1771–1800.
- [Hinton et al., 2006b] Hinton, G. Reducing the dimensionality of data with neural networks / G. Hinton, R. Salakhutdinov // *Science*, 313 (5786). – 2006. – P. 504–507.
- [Hinton, 2010] Hinton, G. A practical guide to training restricted Boltzmann machines // *Tech. Rep. 2010-000*. – Toronto: Machine Learning Group, University of Toronto, 2010.
- [Bengio, 2009] Bengio, Y. Learning deep architectures for AI // *Foundations and Trends in Machine Learning*. – 2009. – Vol. 2(1). – P. 1–127
- [Hinton, 2009] Hinton, G. Greedy layer-wise algorithm // *Journal of Machine Learning Research* 1. – 2009. – P. 1–40.
- [Golovko et al., 2014] Golovko, V. A Learning Technique for Deep Belief Neural Networks / V. Golovko, A. Kroshchanka, U. Rubanau, S. Jankowski // in book *Neural Networks and Artificial Intelligence*. – Springer, 2014. – Vol. 440. *Communication in Computer and Information Science*. – P. 136–146.
- [Golovko et al., 2012] Golovko, V. Neural network model for transient ischemic attacks diagnostics / V. Golovko, H. Vaitsekhovich, E. Apanel, A. Mastykin // *Optical Memory and Neural Networks (Information Optics)*. – 2012. – 21(3). – P. 166–176.

APPLYING DEEP BELIEF NEURAL NETWORKS TO EXTRACTION VALUABLE SEMANTIC FEATURES

Golovko V.A. *, Kroshchanka A.A. *

**Brest State Technical University*

gva@bstu.by

kroschenko@gmail.com

The main principles of construction and learning deep belief neural networks for extraction valuable semantic features are proposed. The proposed approach is based on minimization of reconstruction mean square error, which we can obtain using a simple iterations of Gibbs sampling.



OSTIS-2015

(Open Semantic Technologies for Intelligent Systems)

УДК 004.896

РАЗРАБОТКА ОТКРЫТЫХ ТЕХНОЛОГИЙ ПРОЕКТИРОВАНИЯ ИНТЕЛЛЕКТУАЛЬНЫХ РОБОТОТЕХНИЧЕСКИХ СИСТЕМ

Прокопович Г.А., Сычѳв В.А., Герасюто С.Л.

*сектор робототехники лабораториии моделирования самоорганизующихся систем,
Объединѳнный институт проблем информатики НАН Беларуси, г. Минск, Республика Беларусь*

{Rprakovich, VSychov, Contacts}@robotics.by

Предлагается разработка открытой технологии проектирования интеллектуальных робототехнических систем, необходимых для управления мобильными роботами в реальной человеческой среде, которая позволит сократить время создания и увеличить эффективность работы подобных систем.

Ключевые слова: интеллектуальные робототехнические системы, открытые технологии, разработка.

Введение

Среди новейших активных средств защиты человека всё большее место начинает занимать робототехника, которая способна предложить кардинально новые методы *индивидуальной* защиты: от моторизированных экзоскелетов, предназначенных для защиты и увеличения силы человека, до автономных или полуавтономных мобильных роботов (МР), использующих так или иначе технологии интеллектуальных информационных систем (ИИС).

На сегодняшний день широкое применение получили МР, предназначенные для частичной либо полной замены человека при работе в экстремальных условиях, к которым можно отнести за пределы для человеческого организма значения радиации, температуры, давления, гравитации, времени, влажности и химически активных сред. Как правило, с подобными условиями работы человек сталкивается при исследовании и освоении подводных глубин, дальнего севера и космического пространства, на химическом производстве, атомных станциях, при проведении пожарных, сапѳрных и спасательных работ, мониторинге загрязнѳнных территорий [1].

Тем не менее остаѳтся ещё множество проблемных областей человеческой деятельности, которые из-за своей сложности и вредности для здоровья срочно требуют частичной либо полной автоматизации. К таким областям можно отнести, например, аграрную сферу, утилизацию отходов производства и потребления и т.д. Причѳм немаловажной областью человеческой

деятельности, которая практически не автоматизирована, является сфера обслуживания.

1. Проблемы проектирования ИРС

На сегодняшний день уже можно насчитать несколько десятков МР, основной задачей которых является замещение человека во вредных и опасных условиях труда: проведение сапѳрных работ, помощь при исследовании морских глубин и реализация спасательных работ под водой, демонтаж промышленных строений, уборка улиц и помещений [1,2]. Подобные МР оснащены разнообразными исполнительными системами, высокопроизводительной бортовой вычислительной системой, множеством сенсоров и коммуникационных устройств. Причѳм большинство из них являются дистанционно управляемыми, что значительно сокращает радиус действия и срок проведения работ, а также вносит человеческий фактор.

По этой причине, коллективы учѳных во всѳм мире не прекращают разработки интеллектуальных робототехнических систем (ИРС), способных в реальном времени решать такие задачи, как

- распознавание объектов окружающей среды (дороги, специальные знаки, люди и животные, постройки, линия горизонта и т.д.);
- определение оптимального маршрута при недостаточном количестве исходных данных;
- определение положения робота в пространстве;
- ведение диалога с пользователем;
- моделирование процесса рассуждений на основе имеющейся базы знаний и т.д.

Однако, несмотря на прилагаемые колоссальные усилия, на данный момент не наблюдается значительного прорыва в указанном направлении. С нашей точки зрения, это можно объяснить следующими тремя причинами.

Во-первых, основной проблемой является то, что поведение таких сложных технических объектов управления, как роботы, трудно поддаётся формализации и математическому описанию. Процесс аналитического описания поведения таких объектов усложняется еще и тем, что зачастую не только внешняя среда, но и свойства самих объектов априори неизвестны или изменяются в процессе функционирования.

Во-вторых, значительное отрицательное влияние оказывает всё нарастающая дифференциация научных направлений ИИС, главной целью которых является поиск основных законов и механизмов функционирования человеческого интеллекта для последующего практического использования. Это приводит к появлению большого числа терминов и понятий, хотя относящихся, в сущности, к относительно небольшому числу объектов исследования. К ним можно отнести: «artificial life», «animat», «data mining» и т.д.

И, наконец, в-третьих, текущий уровень интеграции программных и технических средств не позволяет разрабатывать универсальные системы управления, чтобы МР мог выполнять сложные задачи полностью в автономном режиме.

2. Преимущества создания открытых технологий проектирования

Как видно из выше перечисленного, наблюдается разрозненность исследований и расплывчатость описания самого объекта исследований ИИС. Таким образом, для решения указанных проблем и вывода разработки автономных МР на совершенно новый уровень требуется разрабатывать *открытую технологию* проектирования ИРС, которую следует рассматривать как частный случай ИИС. Одним из характерных отличий ИРС является то, что на её вход подаются сенсорные данные от реальных датчиков, имеющих свои пределы точности и зависящие от условий эксплуатации, а выходные сигналы в режиме реального времени должны управлять исполнительной системой, обладающей такими непреодолимыми физическими особенностями как инертность, силы трения и притяжения.

Под технологией проектирования ИРС мы понимаем:

- разработку общей методики проектирования ИРС (от написания технического задания до её сертификации);
- возможность интеграции различных моделей представления и обработки знаний, а также методов распознавания образов и принятия управляющих решений;

- обеспечение модульности как программных, так и аппаратных элементов разрабатываемых ИРС, которая предназначена для решения проблем совместимости различных моделей ИИС с известными устройствами обработки информации и механическими исполнительными системами.

Основной целью предлагаемого проекта по созданию открытой технологии проектирования ИРС является не только объединение усилий специалистов из различных областей, но и привлечение большого числа пользователей, которые одновременно будут выступать не только как потребители, но и как тестировщики. Хорошим примером может служить проект OSTIS, который, в частности, может решить проблему интеграции различных моделей представления и обработки знаний.

Заключение

Рассмотрена востребованность МР для автоматизации широкого круга задач, для управления которыми требуется создание ИРС. Для уменьшения времени разработки, а также повышения эффективности функционирования МР предложено создавать открытые технологии проектирования ИРС.

DEVELOPMENT OF OPEN TECHNOLOGY DESIGNING OF INTELLIGENT ROBOTIC SYSTEMS

Prakapovich R.A., Sychyou U.A., Gerasuto S.L.

*Robotics department of self-organized system
modeling laboratory of United Institute of
Informatics Problems of NAS of Belarus*

{RPrakapovich, VSychyov, Contacts}@robotics.by

To develop the open technology of intelligent robotic systems it is proposed. It is necessary for mobile robots control in the real human environment. This will reduce the time to create and increase the efficiency of such systems.

Introduction

There are many problematic areas of human activity, which, due to their complexity and hazards to health urgently require partial or full automation. Moreover, it needs to use of intelligent robotic systems.

Conclusion

Reviewed demand of mobile robots for automation a wide range of tasks to control which requires the creation of the intelligent robotic systems. To reduce development time and increase the efficiency of mobile robots suggested to development of open technology designing of intelligent robotic systems.



OSTIS-2015

(Open Semantic Technologies for Intelligent Systems)

УДК 004.021

ПОИСК В СТАТИЧЕСКИХ РОЯХ

Воробьев В.В.

*Московский институт электроники и математики НИУ ВШЭ,
г. Москва, Россия
gatus86@mail.ru*

В работе рассматривается решение проблемы организации запросов и поиска в статических роях. Данная проблема вызвана отсутствием какой-либо карты базы данных, знаний о структуре роя и т.д. Для решения данной задачи предлагается использование безадресных запросов с волновым характером их распространения по статическому рою. Это позволит существенно снизить нагрузку коммуникационных каналов и, следовательно, избежать ситуации критических загрузок в сети, образованной узлами статического роя. В основе предложенного решения лежат хорошо известные неинформированные методы поиска.

Ключевые слова: статический рой; алгоритмы поиска; запрос; неинформированный поиск.

Введение

В данный момент одним из быстроразвивающихся направлений современной робототехники является групповая робототехника, объектами изучения которой являются множества взаимодействующих между собой роботов [Карпов, 2013]. Стоит отметить, что основное отличие между этими групповыми образованиями заключается в степени информированности о других членах группы, их общих целях, задачах и т. д. [Карпов, 2011].

Данное направление в робототехнике представлено большим количеством работ, например [Konolige et al., 2006], [Dorigo et al., 2012], [McLurkin et al., 2005], [Roy et al., 2013], в которых рассмотрены различные аспекты роевого поведения, как то: движение в группе, поиск цели и т. д. Пожалуй, самыми интересными из этих работ, является [Kornienko et al., 2006], где авторы пытаются реализовать эффект эмерджентности. Данный эффект заключается в том, что система из множества простых объектов способна решать сложные задачи, которые не способен решать ни один, ни малое количество этих простых объектов.

В [Карпов, 2013] подробно рассмотрен предварительный перечень вопросов, которые необходимо решить для появления таких свойств.

Особый интерес из этого перечня вызывают вопросы организации запросов к базе данных всего роя и поиск по этой базе, которые и будут рассматриваться в этой работе. Данные задачи

неразрывно связаны друг с другом, и имеет смысл рассматривать и решать их совместно.

База данных роя представляет собой накопленный опыт, наблюдаемые факты и т.п. Очевидно, что такое фрагментарное хранение отлично от того, что принято в распределенных базах данных [Карпов, 2013].

Отдельно стоит отметить тот факт, что на рой наложены ограничения, указанные в [Карпов, 2013]:

- Коммуникационные возможности членов роя серьезно ограничены, т. е. он может общаться только с фиксированным числом своих соседей.
- Статический характер роя, т. е. рассматриваются статические структуры, представляющие собой полученные в некоторый момент времени схемы соединения членов роя. Такая совокупность называется *статическим роём*.

В этом случае, ограничение по коммуникационным возможностям может серьезно повлиять на механизм формирования запроса.

Актуальность решения данной задачи заключается в том, что эффективный механизм формирования запроса и поиска по базе данных позволит подойти к решению целого ряда задач, возникающих перед статическим роём, например, к задаче выработки решения, о чем также будет сказано несколько слов.

Общие сведения о задаче и поисковых алгоритмах

Говоря об организации запросов к общей базе данных статического роа, нельзя не отметить, что путей решения данной задачи может быть два [Карпов, 2013]:

- Использование известных сетевых протоколов или их аналогов
- Отправка безадресного сообщения всем членам роа.

Однако первый путь имеет существенные недостатки, например: недостаток, связанный с физической организацией статического роа, а именно с использованием сугубо локальных связей между его членами. В таком случае посылка адресного запроса через своих соседей на периферию сопряжена с существенным возрастанием трафика. Кроме того необходима синхронизация членов роа во времени.

Другой путь заключается в отправке узлом безадресного сообщения своим соседям, которое ретранслируется каждым соседом дальше в том случае, если он не может выполнить запрос. Вычислительные эксперименты показывают, что подобная схема позволяет в асинхронном режиме гарантированно получать результаты запроса с детерминированной максимальной задержкой [Карпов, 2013].

Таким образом, для решения данной задачи лучше подходит отправка безадресного запроса всем членам роа. Однако механизм распространения запроса зависит от метода поиска в пространстве состояний. На данный момент существует множество различных методик поиска, каждая из которых имеет свои преимущества и недостатки. Поэтому имеет смысл рассмотреть данные методики и алгоритмы, которые их реализуют более подробно на предмет применимости их для решения задачи поиска в статическом роа:

- **Информированные методы** (эвристические). Характерной особенностью данных методов является то, что они используют дополнительную информацию (эвристику) о конкретной задаче, что позволяет существенно сократить перебор путем исключения заведомо бесперспективных вариантов. Информация о конкретной задаче формируется в виде *эвристической функции*. Эвристическая функция на каждом шаге перебора оценивает альтернативы на основании дополнительной информации с целью принятия решения о том, в каком направлении стоит продолжать перебор [Рассел и др., 2006].

- **Неинформированные методы** (методы слепого поиска, методы грубой силы). Данные методы не используют никаких дополнительных знаний о задаче и последовательно просматривают все состояния до тех пор, пока не будет найдено решение. Все на что способны неинформированные методы поиска, - выработать приемников и

отличать целевое состояние от нецелевого [Эделькамп и др., 2012].

Рассмотрим некоторые алгоритмы, относящиеся к информированным методам поиска:

- **Поиск по первому наилучшему совпадению.** Алгоритм исследует граф путем расширения наиболее перспективных узлов, выбираемых в соответствии с указанным правилом. Этим правилом является некая эвристическая оценка $f(n)$, где n - узел графа, которая может зависеть от описания этого узла, описания цели, информации, собранной поиском на данный момент или каких-нибудь дополнительных знаний о предметной области [Pearl, 1984]. Выбирается узел с наименьшей оценкой.

- **Алгоритм A^* .** Является разновидностью поиска по первому наилучшему совпадению. В данном алгоритме эвристическая оценка $f(n)=g(n)+h(n)$, где $g(n)$ - стоимость пути от начального узла до узла n , $h(n)$ - оценка стоимости пути от узла n до цели. Временная сложность алгоритма зависит от выбранной эвристики и в худшем случае растет экспоненциально по сравнению с длиной оптимального пути. Кроме того, требует большого объема памяти при самом неблагоприятном исходе, так как приходится запоминать экспоненциальное количество узлов.

- **Алгоритм A^* с итеративным углублением** (iterative deepening A^* IDA*). Данный алгоритм имеет в своей основе идею итеративного углубления, применяемого в алгоритме IDDFS, который относится к неинформированным методам поиска и рассмотрен ниже. Здесь стоит отметить то, что IDA* останавливает развертывание когда оценка пути $f(n)$, которая вычисляется также, как и в алгоритме A^* превышает некую пороговую стоимость bound. Это позволяет существенно снизить потребляемые ресурсы памяти по сравнению с A^* .

- **Рекурсивный поиск по первому наилучшему совпадению** (Recursive Best-First Search, RBFS). Является простым рекурсивным алгоритмом поиска, имитирующий поиск по первому наилучшему совпадению, но с использованием только линейного пространства. Алгоритм контролирует значение $f(n)$ наилучшего альтернативного пути, доступного из любого предка текущего узла. Если значение превышает, то рекурсия отменяется и продолжает с альтернативного пути. RBFS несколько более эффективен нежели IDA*.

- **A^* с ограничением памяти и упрощенный A^* с ограничением памяти.** (memory-bounded A^* , MA* и simplified memory-bounded SMA*). Также как IDA* и RBFS решают проблему потребления большого объема памяти, однако лишены их недостатков, например, использование не всей доступной памяти.

Информированные методы обеспечивают более эффективный поиск, но они не могут применяться для такого рода задач в статическом роа. Это

обусловлено тем, что целевой узел в данной задаче неизвестен, следовательно, включать различного рода эвристики часто не имеет смысла, так как алгоритм, реализующий такой поиск, становится неоптимальным. Кроме того, база данных всего статического роя, в общем случае, не является распределенной [Карпов, 2013], так как не имеет структуры, описывающей, в каких узлах хранятся те или иные данные, т.е. часто заранее неизвестна даже область, в которой необходимо проводить поиск. Данная особенность также влияет на целесообразность использования информированных методов поиска.

Хотя неинформированные методы поиска и решают эту задачу менее эффективно по сравнению с информированными, они являются наиболее подходящими для ее решения. Действительно, так как неизвестно в каком узле статического роя может находиться искомая продукция наиболее оправданной стратегией поиска будет полный перебор всех узлов. Рассмотрим неинформированные методы поиска подробнее:

- **Поиск в ширину** (breadth-first search, BFS). Алгоритм поиска, в котором вначале разворачивается корневой узел, затем все потомки корневого узла, после этого разворачиваются потомки этих потомков и т. д.

- **Поиск по критерию стоимости** (uniform-cost search, UCS). В какой-то мере является расширением алгоритма поиска в ширину, который учитывает стоимость действий. Данный алгоритм будет полным и оптимальным, в случае если стоимости строго положительны.

- **Поиск в глубину** (depth-first search, DFS). В данном алгоритме поиска разворачивается самый глубокий узел в текущей периферии дерева поиска, т. е. анализируется первый преемник текущего узла, затем преемник преемника и т. д.

- **Поиск с ограничением глубины** (depth-limited search, DLS). Является вариантом алгоритма поиска в глубину, где задается максимальная глубина поиска l , что позволяет решить проблему бесконечного поиска. Однако данный алгоритм мало применим, так как он не является полным, в случае если поверхностная цель d лежит за границей поиска $l > d$ и не оптимален в случае $d < l$. Является основой поиска в глубину с итеративным углублением.

- **Поиск в глубину с итеративным углублением** (iterative-deepening depth-first search, IDDFS, DFID). В своей основе имеет алгоритм DLS с возможностью увеличения границы поиска l в случае если цель не была найдена. Сочетает в себе преимущества DFS (пространственная сложность $O(b \cdot l)$) и BFS (полнота и оптимальность при конечном b).

Однако не стоит забывать о принципиальной возможности использования информированных методов поиска после нескольких итераций поиска с помощью методов неинформированных. Таким образом, поисковый запрос в статическом рое

должен формироваться на основе неинформированных методов поиска причем запрос должен распространяться на безадресной основе.

Формирование запроса и поиск в статическом рое

Рассмотрим следующую задачу: существует $N=m \cdot n$ членов статического роя, связанных друг с другом локальными связями (рис. 1).

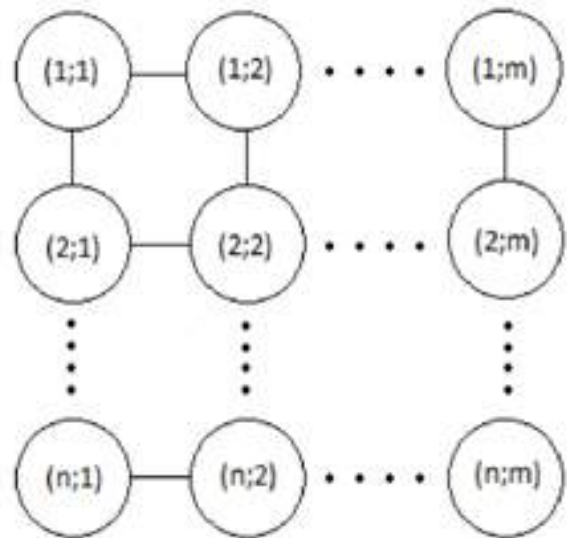


Рисунок 1 – Физическая структура статического роя

Стоит отметить тот факт, что с точки зрения каждого члена роя в его составе находится только он сам и его соседи. Это является следствием ограничения на взаимодействие друг с другом.

Существует иницирующий узел, который, например, был выбран с помощью процедуры, описанной в [Карпов, 2014]. Данный узел либо сам формирует запросы, либо имеет возможность получать их извне. С точки зрения организации процедуры выполнения запроса не имеет значения, откуда он появился. Важно лишь то, что его формат поддерживается данной системой и то, что после получения/формирования запроса, иницирующий узел не может получать/формировать новый до тех пор, пока запрос не будет выполнен или станет понятно, что он невыполним.

Так как уже было сказано, что запрос является безадресным, и на начальном этапе не имеет смысла использовать информированные методы поиска, то, иницирующий узел, в случае неудачной попытки его обработать (ничего не найдено в его БД), передаст его **всем** своим соседям. Соседи также попытаются выполнить запрос, а в случае неудачи, отправят его своим соседям и т.д.. Такой подход наиболее похож на алгоритм **поиска в глубину с итеративным углублением** (IDDFS). Отличие заключается в том, что благодаря тому, что каждый сосед иницирующего узла это отдельный вычислительный блок, мы можем отправить запрос одновременно всем соседям, которые, при необходимости, увеличат глубину поиска.

Однако может появиться такая ситуация, в которой узел попытается передать запрос узлу, который этот запрос уже обработал. Такая коллизия решается путем введения трех правил:

- Отправляющий узел знает количество своих потомков, которым был отправлен запрос, и запоминает его.
- Узел не отправляет запрос узлу, от которого он его получил. То есть реализуется функция, позволяющая запомнить, откуда пришел запрос.
- В случае если узел обработал запрос и, по прошествии некоторого времени, снова получает тот же запрос, он отвечает на него специальным сообщением. Отправляющий узел, приняв это сообщение, уменьшает число своих потомков на единицу.

Таким образом, запрос постепенно уйдет от иницирующего узла на периферию роя, сформировав дерево поиска и не загружая при этом каналы связи, так как назад запрос уйти не может из-за правила №2, а соседям, которые его уже обработали из-за правила №3 пройдет сначала запрос, а потом ответ на него (рис. 2). Иницирующий узел (ИУ) посылает запрос (одинарная стрелка) своим соседям, те посылают его своим соседям и т.д. Двойными стрелками показана отправка запроса узлу, который его уже принял.

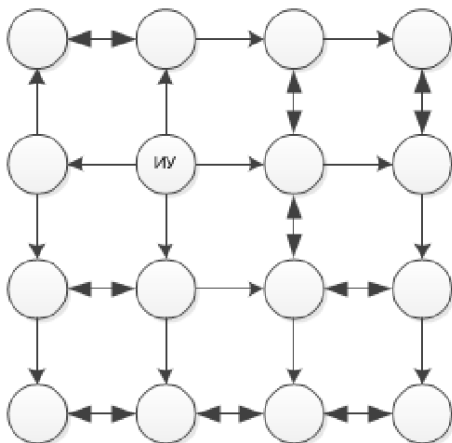


Рисунок 2 – Отправка запроса

В случае если какой-либо узел смог выполнить запрос, он формирует сообщение, механизм распространения которого полностью аналогичен механизму распространения поискового запроса. Это сообщение несет в себе метку о том, что поиск закончен и результат этого поиска. Метка блокирует узлы, которые получили это сообщение для получения поискового запроса, который был выполнен и для получения другого ответа на этот запрос. Когда это сообщение доходит до иницирующего узла, он способен снова получать/формировать новый запрос.

Для того, чтобы избежать ситуации с “зависанием поиска”, которая характерна тем, что запрос не может быть выполнен (например, в базе данных статического роя нет такой информации)

реализуется следующий механизм: в случае если периферийный узел не имеет потомков (исходя из вышеописанных правил), то он отправляет соответствующее сообщение узлу-предку. Как только предок получил такое сообщение от всех своих потомков, он передает такое же сообщение своему предку и т.д. Как только иницирующий узел получил такие сообщения от всех своих потомков, он может принимать/формировать другой запрос.

Однако механизм формирования, распространения и выполнения запроса в статическом рое несет сугубо вспомогательную функцию по отношению к другим задачам, например, задаче выработки общего решения. Полагается, что правило является множеством импликаций, вида $c_1 \& c_2 \& \dots \& c_n \rightarrow f$. Иницирующий узел будет пытаться отработать такое правило и поочередно отправлять запросы в сеть на определение истинности каждого из конъюнктов. Узлы будут искать в своих БД интересные инициатора данные, которые могут быть либо фактами, либо правыми частями неких хранящихся в узлах правил. Реализация такого механизма в духе ПРОЛОГа, не смотря на свою универсальность, ведет к тому, что все ресурсы сети, и особенно коммуникационных каналов, будут заняты исключительно логическим выводом [Карпов, 2013].

Однако использование такого рода запроса, позволяет отчасти решить проблему сильной загрузки канала связи. Кроме того, с таким механизмом формирования запроса достаточно легко реализуется функция управления поиском в ПРОЛОГе с использованием предикатов fail и cut.

Предикат fail реализуется путем добавления к запросу соответствующего аргумента, который позволяет выполнившему запрос узлу не формировать блокирующее сообщение и передать запрос далее.

Механизм работы предиката cut в такой системе полностью аналогичен – в аргументе запроса указан соответствующий параметр, при нахождении которого дальнейший поиск блокируется. Также имеется возможность удалить другие результаты поиска, которые были найдены до того как сработал cut.

Кроме того, если запрос выполнен, но ответ также является конъюнкцией продукций, то иницирующий узел передает управление узлу, содержащему ответ, который в свою очередь также отправляет свой запрос на поиск соответствующих конъюнкций, то есть осуществляется рекурсия.

Характеристики поискового алгоритма при его использовании в статическом рое

Рассмотрим ряд характеристик работы данного алгоритма в условиях статического роя:

- **Полнота.** Так как данный алгоритм является алгоритмом IDDFS, который применяется в условиях статического роя, то он является полным [Рассел и др., 2006].
- **Оптимальность.** Данный алгоритм оптимален, и эта характеристика также вытекает из оптимальности IDDFS [Рассел и др., 2006]. Так как в статическом рое количество коммуникационных каналов его членов ограничено, то он оптимален всегда.
- **Пространственная сложность.** Очевидно, что пространственная сложность данного алгоритма будет не более, чем у IDDFS и будет равна $O(b^*l)$, где b – коэффициент ветвления, l – глубина поиска [Рассел и др., 2006]. При этом b ограничено количеством коммуникационных каналов каждого члена роя.
- **Временная сложность.** Временная сложность в условиях применения в статическом рое будет зависеть исключительно от глубины поиска l , коэффициента связи k и средней величины базы данных узла N , и будет равна $O(l * k \oplus N)$

Коэффициент связи необходим для более точного определения временной сложности, так как время поиска в статическом рое сильно зависит от времени распространения запроса, то есть, от качества каналов связи.

Рассмотрим процедуру поиска на примере. В момент времени $T=1$ запрос приходит к инициирующему узлу (рис. 3). Узел сразу отправляет запрос своим узлам-потомкам и начинает обрабатывать запрос. Узлы-потомки, в свою очередь, производят те же действия.

```
-----Level#0-----
#-1 <3 5 3 2 9 6 2 7 4 0 >Status: received a request
Clock #-1
```

Рисунок 3 – Начало обработки запроса

На рисунке 4 видно, что запрос дошел до периферийных узлов, и искомая информация была найдена.

```
-----Level#0-----
#-1 <3 5 3 2 9 6 2 7 4 0 !0! >>Status: request not found
-----Level#1-----
#-2 <6 1 9 8 0 4 9 3 0 6 !6! >>Status: request not found
#-3 <4 9 5 2 7 4 7 0 4 2 !2! >>Status: request not found
-----Level#2-----
#-4 <0 2 6 7 9 7 8 1 1 8 !8! >>Status: request not found
#-5 <8 0 8 0 4 6 1 0 4 7 !7! >>Status: request not found
#-6 <7 7 6 8 9 6 2 9 8 1 !1! >>Status: request not found
#-7 <5 5 4 7 8 2 7 0 0 3 !3! >>Status: request not found
-----Level#3-----
#-8 <4 0 4 3 8 2 6 3 0 !0! >>Status: request not found
#-9 <9 4 3 2 6 8 0 6 8 !8! >>Status: request not found
#-10 <5 7 9 9 3 1 1 5 7 !7! >>Status: request not found
#-11 <1 6 3 4 8 1 5 3 1 !1! >>Status: request not found
#-12 <7 3 1 5 6 0 3 4 4 !4! >>Status: request not found
#-13 <4 3 4 0 3 4 0 9 7 !7! >>Status: request not found
#-14 <6 2 6 0 2 2 1 3 5 !5! >>Status: request not found
#-15 <3 2 2 3 6 1 7 2 11 >>Status: request is done. Fact is on 3 level
Clock #-7
```

Рисунок 4 – Искомая информация найдена

На рисунке 5 видно как запрос дошел до инициирующего узла.

```
-----Level#0-----Fact is on this level
#-1 <3 5 3 2 9 6 2 7 4 0 >Status: request not found
-----Level#1-----
#-2 <6 1 9 8 0 4 9 3 0 6 >Status: request not found
#-3 <4 9 5 2 7 4 7 0 4 2 >Status: request not found
-----Level#2-----
#-4 <0 2 6 7 9 7 8 1 1 8 >Status: request not found
#-5 <8 0 8 0 4 6 1 0 4 7 >Status: request not found
#-6 <7 7 6 8 9 6 2 9 8 1 >Status: request not found
#-7 <5 5 4 7 8 2 7 0 0 3 >Status: request not found
-----Level#3-----
#-8 <2 4 0 4 3 8 2 6 3 0 >Status: request not found
#-9 <7 9 4 3 2 6 8 0 6 8 >Status: request not found
#-10 <5 7 9 9 3 1 1 5 7 >Status: request not found
#-11 <4 1 6 3 4 8 1 5 3 1 >Status: request not found
#-12 <2 7 3 1 5 6 0 3 4 4 >Status: request not found
#-13 <1 4 3 4 0 3 4 0 9 7 >Status: request not found
#-14 <9 6 2 6 0 2 2 1 3 5 >Status: request not found
#-15 <9 3 2 2 3 6 1 7 2 11 >Status: request is done
Clock #-10
```

Рисунок 5. Запрос выполнен

Заключение

Таким образом, предложенная методика реализации запросов внутри статического роя позволяет решить следующие задачи:

- Производить поиск необходимых элементов внутри общей базы данных статического роя.
- Находить все возможные альтернативы искомого элемента.
- Производить рекурсивный поиск в случае, если искомым элементом оказался не фактом, а, в свою очередь, также множеством импликаций.

Методика имеет в своей основе хорошо известный алгоритм неинформированного поиска IDDFS и формирует запросы на безадресной основе, что позволяет не загружать слабые коммуникационные каналы. Кроме того, такой механизм формирования запросов позволяет вообще не загружать эти каналы, даже в случае рекурсивного поиска. Однако это влечет за собой то, что при достаточно большом числе узлов статического роя или их неудачной конфигурации (например, узлы расположены в линию и инициирующий узел находится в начале этой линии) может возникать ситуация с их простоем, т.е., обработав и передав запрос дальше, узлы будут ждать завершения его выполнения. Данная проблема будет рассматриваться в дальнейшем.

Еще одним немаловажным аспектом работы является то, что такой механизм может позволить подойти к решению проблемы выработки общего решения с использованием ПРОЛОГ. Как уже было упомянуто, одним из препятствий к его использованию является слабость коммуникационных каналов статического роя. Такой механизм формирования запросов может оказаться решением данной проблемы.

Библиографический список:

- [Карпов,2011] Карпов В.Э. Коллективное поведение роботов. Желанное и действительное //Современная мехатроника. Сб. научн. трудов Всероссийской научной школы (г.Орехово-Зуево, 22-23 сентября 2011) -Орехово-Зуево,2011. Т132 с. сс.35-51.
- [Карпов, 2013] Карпов В.Э. Управление в статических роях. Постановка задачи. //VII-я Международная научно-практическая конференция "Интегрированные модели и мягкие вычисления в искусственном интеллекте" (20-22 мая 2013) Сб. научных трудов. В 3-т., М: Физматлит, 2013, Т.2, с.730-739

[Карпов, 2014] Карпов В.Э. Процедура голосования в однородных коллективах роботов //XIV национальная конференция по искусственному интеллекту с международным участием КИИ-2014 (24-27 октября 2014 г., Казань, Россия): Труды конференции. В 3-т., Казань: Изд-во РИЦ «Школа», 2014, Т.2, с.159-167, -341с.

[Рассел и др., 2006] Стюарт Рассел, Питер Норвиг Искусственный интеллект: современный подход // Artificial Intelligence: A Modern Approach.- 2-е изд. - М: Вильямс, 2006. - 1408 с. - ISBN 5-8459-0887-6.

[Dorigo et al., 2012] M. Dorigo, D. Floreano, L.M. Gambardella, F. Mondada et al. Swarmanoid: a novel concept for the study of heterogeneous robotic swarms // IEEE Robotics & Automation Magazine, Pages In press, 2012

[Konolige et al., 2006] K. Konolige, D. Fox, C. Ortiz, et al. Centibots: Very large scale distributed robotic teams // Springer Tracts in Advanced Robotics Volume 21, 2006, pp 131-140

[Kornienko et al., 2006] S. Kornienko, O. Kornienko, P. Levi Swarm embodiment - a new way for deriving emergent behavior in artificial swarms //Autonome Mobile Systeme 2005 (AMS05), 25-32, DOI: 10.1007/3-540-30292-1_4, 2006

[McLurkin et al., 2005] J. McLurkin, D. Yamins Dynamic Task Assignment in Robot Swarms // Robotics: Science and Systems Conference, June 8, 2005

[Pearl, 1984] Pearl J. Heuristics: Intelligent Search Strategies for Computer Problem Solving. - Addison-Wesley, 1984. - . 48.

[Roy et al., 2013] N. Roy, P. Newman, S. Srinivasa Towards A Swarm of Agile Micro Quadrotors.//Robotics:Science and Systems VIII - MIT Press 1, 2013

THE SEARCHING TASK IN THE STATIC SWARM

Vorobiev V.V.

National Research University The Higher
School of Economics, Moscow, Russian
Federation
gatus86@mail.ru

This paper considers the problem of organization and search queries in static swarms. This problem is caused by the absence of any map database of knowledge about the structure of the swarm, etc. To solve this problem we suggest the use of non-address requests to the wave nature of the distribution of the static swarm. This will significantly reduce the load of communication channels, and hence avoid the situation critical loads the network formed by nodes of the static swarm. At the heart of the proposed solutions are well known uninformed search methods

.



OSTIS-2015

(Open Semantic Technologies for Intelligent Systems)

УДК 004.891.3

ПОСТРОЕНИЕ ИНТЕЛЛЕКТУАЛЬНЫХ МУЛЬТИАГЕНТНЫХ КОБОРГ-СИСТЕМ

Соловьев В.И.

*Научно-инженерный Центр Ассоциации содействия Всемирной Лаборатории
г. Москва, Россия*

sicwl@newmail.ru

В работе рассматривается метод построения на базе коборг-технологии интеллектуальных мультиагентных систем диагностирования и управления, включающих интеллектуальные робототехнические комплексы, соединения локальных техно-коборгов, действующих в сетевом пространстве, а также команды (коллективы) людей, выполняющих определенное задание.

Ключевые слова: коборг-технология; органы оперативное диагностирование состояния МА-объектов.

Введение

Основная сущность коборг-технологии для диагностирования и управления сложными организованными объектами (A complicated organized objects-Coborgs) была освещена в [Соловьев В.И., 2011а], [Соловьев В.И., 2011б]. Она заключается в представлении таких объектов как некоторого организованного единства всех согласованно действующих в них процессов, органов, систем или функциональных узлов, не относящиеся к животному или растительному миру, но обладающие основными свойствами живых организмов. К коборгам относятся все сложные организованные объекты в технической, экономической, административной, военной и прочих областях человеческой деятельности. По сути, термин Коборг достаточно точно объясняет природу широко применяемого термина «организм» во всех "неживых" приложениях. Понятие коборга относится как локальным организованным объектам, так и распределенным мульти-агентным объектам различного типа и назначения (в дальнейшем МА-Коборги).

Основными свойствами коборга (МА-коборга) являются:

1. наличие в его составе органов и процессов в них происходящих;
2. наличие внутренних параметров состояния (ВПС) органов и комплексных параметров состояния коборга в целом. Они представляются в виде различных нормативов, заданных диапазонов и траекторий, предельно допустимых минимальных или максимальных величин, определяющих

нормальное (заданное) состояние как отдельного органа, так и коборга в целом;

3. способность принимать и обрабатывать поступающую на его входы информацию в реальном или псевдореальном времени.

4. воспринимать, хранить, использовать и представлять знания в данной предметной области.

5. способность оперативно диагностировать текущее состояние органов и коборга в целом и своевременно выявлять намечающиеся расстройства и заболевания.

6. выявлять причины расстройства или заболевания отдельных органов и коборга в целом.

7. формировать управляющие воздействия по ликвидации намечающегося расстройства или заболевания коборга.

Интеллектуальные системы оперативного диагностирования и управления, построенные на применении коборг-технологии, называются СмартКоборгСистемами (SmartCoborgSystems).

Основная часть

В данном докладе рассмотрена возможность применения коборг-технологии к созданию интеллектуальных систем оперативного диагностирования и управления сложными мультиагентными объектами, которые мы в соответствии с основной концепцией коборг-технологии назовем мультиагентными коборгами (МА-коборги). МА-коборгами могут являться:

- команда, коллектив, подразделение, конкретный персонал людей, выполняющих определенное задание или решающих определенную задачу;

– действующая группа интеллектуальных роботов - агентов (ИРА), в том числе и кибернетических;

– действующие соединения локальных технокорборгов (надводных и подводных судов, самоходных наземных машин, воздушных аппаратов), образующих неоднородное сетцентрическое пространство.

Важнейшим моментом при построении интеллектуальных систем диагностирования текущего состояния и управления МА-корборга является задача его представления адекватной совокупностью органов (процессов, узлов, систем и др.). Наиболее очевидным здесь может быть компоновка органов на функциональном принципе. Например, органами футбольной команды могут являться: оборона, полузащита и нападение. Органами группы по проведению специальной операции могут являться подразделения разведки, исполнения, прикрытия и обеспечения. На рис.1 представлена блочная схема управления мультиагентным корборгом (МА-Корборг Система).

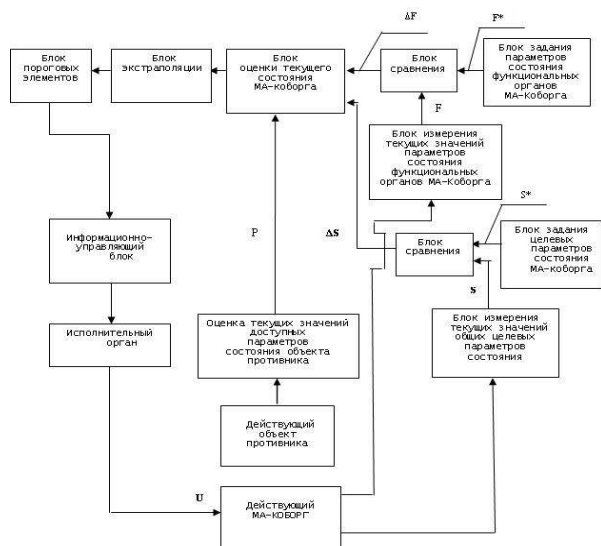


Рисунок 1 - Блок-схема управления МА-Корборгом

Основным отличием мультиагентного корборга от локального заключается в явном отсутствии непрерывно регулируемой выходной координаты Y , а управление происходит по заданным конечным или промежуточным целям, а также конкретным заданиям. В этом случае, когда невозможно задать непрерывную траекторию движения управляемого объекта на весь период времени, управление МА-корборгом осуществляется по внутренним параметрам состояния (ВПС). Применительно к МА-корборгу целесообразно выделять две категории ВПС:

– комплексные параметры состояния (КПС), характеризующие общее состояние «здоровья» МА-корборга по аналогии с общими параметрами состояния человеческого организма (температура тела, проявление болевых ощущений, артериальное давление, химический состав крови и мочи). Такими общими параметрами при проведении спецопераций являются: основной сценарий, время

и продолжительность проведения этапов операции (подготовка, исполнение, отход, завершение), скрытность, связь, сведения о противнике и пр. В деятельности бизнес-фирм, организаций, компаний это могут быть намеченные бизнес-планы работ или решения определенных задач, окончательные и промежуточные сроки (график) выполнения работ, доступные данные о конкурирующих фирмах. В спортивной игре общими командными параметрами могут считаться: схема игры, текущий счет матча, время владения мячом, исполнение "наигранных" типовых тактических ситуаций, доступные данные о сопернике и др.

– параметры состояния функциональных органов команды (ПСФО), характеризующие состояние "здоровья" соответствующего органа МА-корборга. Например, функциональными органами подразделения по проведению спецопераций являются: подразделения разведки, исполнения, прикрытия, обеспечения. В коллективах бизнес-организаций функциональными органами можно считать подразделения продаж, закупок, маркетинга и рекламы и др. В спортивной команде - это органы защиты, полузащиты и нападения.

Информация об отклонениях от заданных и фактических значений с выходов блоков сравнения КПС (ΔS) и ПСФО (ΔF) поступают на блок оценки текущего состояния МА-корборга. Сюда же поступает доступная текущая информация о состоянии объекта противника (соперника, конкурента и т.п.). Отметим, что в реальности на блок оценки текущего состояния поступают отклонения от значений доступных параметров противника или конкурентов и собственных аналогичных параметров МА-корборга (на схеме не показано в целях ее упрощения).

При диагностировании текущего состояния МА-корборга необходимо разрешить следующие принципиальные задачи:

- определить возможный перечень комплексных и внутренних параметров состояния функциональных органов команды;
- определить возможное число конкретных расстройств и заболеваний в соответствующей предметной области на весь период жизненного цикла МА-корборга;
- с учетом знаний экспертов, в том числе руководителей, командиров, главных тренеров и др., а также на основе существующих инструкций и положений, выбранных приоритетов по каждому этапу выполнения задания (работы), установить начальные значения коэффициентов уверенности вывода о развитии конкретного расстройства по значению и тенденции изменения соответствующих КПС и ПСФО для каждого порождающего правила. Здесь же определяется и степень влияния на общую диагностику текущих доступных параметров о противнике(сопернике, конкуренте).

Остановимся более подробно на так называемом онтологическом аспекте задачи диагностирования,

связанным с терминологией расстройств и заболеваний МА-коборга. При этом задача диагностирования определяется некоторой совокупностью модельных характеристик, описывающих процесс функционирования МА-коборга в каждой предметной области.

Принципиальной особенностью такого подхода заключается в том, что диагностируется не только текущее состояние органов МА-коборга на предмет их общего расстройства ("здоров" – "не здоров"), но и раскрываются основные модельные характеристики (свойства) нарушений его нормального функционирования.

На наш взгляд, к таким универсальным модельным характеристикам, отображающих в общем случае текущие состояния функционирования многих мультиагентных объектов (команды, группы, коллектива), можно отнести:

- эффективность действий;
- активность действий;
- разнообразие действий.

Однако, использование полной совокупности этих характеристик при создании МА-коборг систем не всегда возможно из-за специфики конкретной предметной области. Так, если для команд по игровым видам спорта, эти характеристики применимы в полной мере на основе регистрируемых ВПС, то, например, для команды (соединения, отряда) по проведению спецоперации эти модельные характеристики могут быть применены только в отношении характеристик "эффективность" и "активность" функционирования соответствующих органов команды. Использование характеристики "разнообразие" функционирования органов команды здесь достаточно проблематична, поскольку данный процесс требует строгого выполнения всех запланированных действий при основном сценарии выполнения операции. Что касается текущей оценки действий коллектива организации, выполняющего определенную конечную работу (проект, задание), то здесь целесообразно использовать общепринятое понятие только эффективности трудовой деятельности персонала по целям [Магура М.И и др. 2002] на базе соответствующих частных показателей. К ним, например, можно отнести объемы и сроки исполнения этапов и всей работы в целом, число исполнителей, финансовые затраты и др. Такие частные показатели, характеризующие эффективность деятельности органов коллектива, выступают в качестве внутренних параметров состояния МА-коборга и определяются экспертами в базе знаний диагностирования с соответствующими значениями весовых коэффициентов.

Таким образом, мы рассмотрели три варианта применения модельных характеристик для идентификации текущего состояния МА-коборга, используемых в различных предметных областях. Первый вариант диагностирования расстройства,

включающий все три модельные характеристики, применим для команд в игровых видах спорта, второй – для отрядов и соединений, проводящих специальные операции, третий – для коллективов организаций, осуществляющих свою деятельность по поставленным целям. Принципиально заметим, что при создании МА-коборг систем для оперативного диагностирования и управления удобно иметь дело с обратными понятиями приведенных модельных характеристик, отражающих степень расстройства функционирования органов, а именно, с расстройствами вида "неэффективные действия", "пассивные действия" и "однообразные действия" команды (группы, коллектива). В случае, когда все ВПС находятся в заданных интервалах, имеет место нормальное (заданное) состояние МА-коборга. Считается, что каждому конкретному состоянию МА-коборга соответствует свое подмножество производственных правил. Текущее значение соответствующего расстройства определяются по процедуре MYCIN [Buhanan В.Г.и др. 1984], которая осуществляет расчет коэффициентов вероятности развития расстройства CF_j , определяемый экспертными знаниями, промежуточными и конечными результатами доказательств на непрерывном интервале $[-1 +1]$ от соответствующих симптомов, появление которых обуславливается отклонением текущих значений ВПС объекта от установленных диапазонов. База знаний диагностирования текущего состояния органов МА-коборга представляет собой набор продукций, позволяющий сочетать логический вывод и вычисления коэффициентов уверенности вывода о развитии расстройства CF_j . Она имеет многослойную древовидную структуру, послойно включающая "И", "ИЛИ" и "КОМБ" правила.

Отметим, что указанный подход диагностирования МА-коборга по модельным характеристикам равно применим для выше названных действующих групп интеллектуальных (кибернетических) роботов – агентов, (ИРА) и действующих отрядов (соединений) локальных техно-коборгов (надводных и подводных судов, самоходных наземных машин, воздушных аппаратов), образующих неоднородное сетцентрическое пространство.

Одной из важных функций системы является своевременное выявление внешних и внутренних возмущений, действующих на коборг, так как именно они являются основной причиной изменения его состояния. В зависимости от степени расстройства МА-коборга SmartCoborgSystem может формировать следующие управляющие воздействия:

- симптоматические - до момента ликвидации причины расстройства;
- параметрические - ликвидирующие причину расстройства;
- структурные - когда усилия по повышению эффективности управления и применения других

организационно-технических мероприятий не дают результатов.

Заклучение

Введение модельных характеристик расстройств, отражающих процессы неэффективности, пассивности и однообразия действий мультиагентного коборга позволяет:

1. Своевременно вскрывать механизм расстройства функционирования органов МА-коборга.
2. Повысить качество оперативного диагностирования и управления текущего состояния МА-коборга.
3. Создать универсальный подход для построения интеллектуальных систем оперативного диагностирования и управления МА-коборгами различного назначения.

CREATION OF INTELLEAGENT MULTIAGENT COBOR-SYSTEMS

Soloviev V. I.

*Association for World Laboratory Assistance
Science and Engineering Center,
Moscow, Russia
sicwl@newmail.ru*

The paper describes a method of constructing on the basis of Coborg-technologies for intelligent multi-agent systems diagnosis and management provided intelligence for robotic systems, connections local koborgs perating in a network-centric space, and teams (groups) of people performing a certain task.

Introduction

Complicated organized objects consist of a certain unity of all the processes, organs and systems acting inside it. They belong neither to animality nor organistic nature, despite they possess the main characteristics of living organisms. Such objects are called Coborgs (complicated organized objects) and systems which operate them are called Smart Coborg Systems. In substance, the term Coborg explains precisely enough the nature of the widely used term "organism" in "nonliving" application. The concept of Coborg is connected with both local objects and distributed multi-agent objects of different types and used for various purposes (hereinafter MA-Coborgs).

Main Part

For diagnosis and management of MA-Coborg it is crucial to represent it as a set of bodies (processes, components, systems).

The layout of the bodies of MA-Coborg is implemented according to the functional principle of a particular subject area. The current state of each body of MA-Coborg is internal state parameters (ISP). For MA-Coborg is necessary to distinguish two categories of IPS:

– integrated parameters that characterize the general state of "health" MA-Coborg by analogy with the general parameters of human body (body temperature, the

demonstration of pain, blood pressure, chemical composition of blood and urine);

– parameters of the functional status of organs of the team, describing the state of the respective body of MA-Coborg.

The process of diagnostics of operational state agencies of MA-Coborg is determined by a set of model characteristics used for each subject area. The characteristics that define the disorder in the functioning of organs MA-Coborg, include:

- "the inefficiency" of actions;
- "the passivity" of actions;
- "the monotony" of actions.

If all IPS are in predetermined intervals, MA-Coborg is in normal state.

For diagnosis of the current state of MA-Coborg is necessary to solve the following problems:

- to determine the possible range of integrated and internal parameters of the functional bodies;
- to determine the possible number of specific disorders and diseases in a particular subject area;
- to determine the meaning of disorder taking into account the knowledge of experts, and transitional and final results of logical demonstration of the related symptoms.

The degree of influence on the general diagnosis of the current available parametrs of the opponnet is also determined.

It is known that each state of MA-Coborg corresponds to a subset of production rules.

The knowledge base of diagnosis of the current state of MA-Coborg's bodies is a set of productions which combines logical conclusions and calculations of the coefficients of the certainty of the conclusion about the development of the disorder CF. It has a multilayered arborescent structure, layerwise including "AND", "OR" and "COMB" rules.

Knowledge base of the causes of disorders of the functioning of MA-Coborg estimates errors in actions of each intelligent agent.

Depending on the degree of disorder MA-Coborg SmartCoborgSystem can generate the following control actions:

- symptomatic, which is used until the elimination of the causes of the disorder;
- parametric, which eliminates the causes of disorders;
- structural, which are used when efforts to improve management efficiency and other organizational and technical measures do not yield results.

Conclusion

The introduction of model characteristics, which reflect disorders of MA-Coborg, allows you to:

1. reveal the mechanism of the disorder in the functioning of organs MA-Coborg;
2. improve the quality of operational diagnosis and management of the current state of MA-Coborg.
3. create a universal approach for building intelligent systems operative diagnosis and management MA-coborgs for various purposes.



OSTIS-2015

(Open Semantic Technologies for Intelligent Systems)

УДК 004.934.5

АДКРЫТЫЯ КАМПАНАНТЫ WWW.CORPUS.BY ДЛЯ НАТУРАЛЬНА-МАЎЛЕНЧАГА ІНТЭРФЕЙСУ

Гецэвіч Ю.С.*, Лабанаў Б.М.*, Лысы С.І.*, Гюнтар А.В.*, Дзенісюк Д.А.*, Захар'еў В.А.**

** Аб'яднаны інстытут праблем інфарматыкі Нацыянальнай акадэміі навук Беларусі,
г. Мінск, Беларусь*

yury.hetsevich@gmail.com

lobanov@newman.bas-net.by

stanislau.lysy@gmail.com

lena205593@gmail.com

d.denissyuk@gmail.com

*** Беларускі дзяржаўны ўніверсітэт інфарматыкі і радыёэлектронікі, г. Мінск, Беларусь
delfvad@gmail.com*

У дадзеным артыкуле апісаны агульныя падыходы да праектавання, рэалізацыі і выкарыстання адкрытых кампанентаў інтэрнэт-рэсурсу www.corpus.by для натуральна-маўленчага інтэрфейсу з мэтай паляпшэння і ўдасканалення працы сістэмы сінтэзу маўлення па тэксце з адначасовым прадстаўленнем магчымасці інтэграцыі з праектам OSTIS. Прыводзяцца апісанні мэтаў і задачай, спосабаў працы і даступнай функцыянальнасці некаторых рэалізаваных сэрвісаў.

Ключавыя словы: сінтэз маўлення па тэксце, натуральна-маўленчы інтэрфейс, кампанент, інтэрнэт-сэрвіс.

Уводзіны

Дзеля вырашэння задачы ўсталявання ўзаемадзеяння паміж чалавекам і машынай пры дапамозе маўлення навукоўцы ва ўсім свеце працуюць над стварэннем сістэм сінтэзу і распазнавання маўлення. У апошні час сістэмы сінтэзу маўлення па тэксце (ССМТ) ужо дасягнулі пэўнай дасканаласці і шырока выкарыстоўваюцца. Аднак нельга сказаць, што задача сінтэзу маўлення з'яўляецца вырашанай, асабліва ў дачыненні да беларускай мовы. У сувязі з гэтым аўтарамі было прынята рашэнне стварыць інтэрнэт-рэсурс, які б аб'ядноўваў шэраг інтэрнэт-сэрвісаў, кожны з якіх у сваю чаргу вырашаў бы пэўныя свае ўласныя задачы (напрыклад, прыкладныя лінгвістычныя задачы), але ў той жа час дапамагаў распрацоўшчыкам ва ўдасканаленні працы сінтэзатара маўлення па тэксце. Такі падыход дае магчымасць праз ужыванне сэрвісаў вялікай колькасцю карыстальнікаў правесці тэставанне і збор неабходнай інфармацыі, пры дапамозе якіх можа ўдасканальвацца якасць працы сінтэзатара маўлення па тэксце [Лобанов, 2008], [Гецэвіч, 2012].

У дакладзе прадстаўлены агульныя падыходы да праектавання, рэалізацыі і выкарыстання інтэрнэт-сэрвісаў у мэтах паляпшэння працы ССМТ з прадстаўленнем магчымасці інтэграцыі з іншымі праектамі (у прыватнасці з праектам OSTIS) праз інтэрнэт-сайт www.corpus.by [www.corpus.by, 2012]. Прыводзяцца апісанні мэтаў і задачай, спосабаў працы і даступнай функцыянальнасці некаторых рэалізаваных сэрвісаў.

1. Агульныя падыходы да праектавання інтэрнэт-сэрвісаў

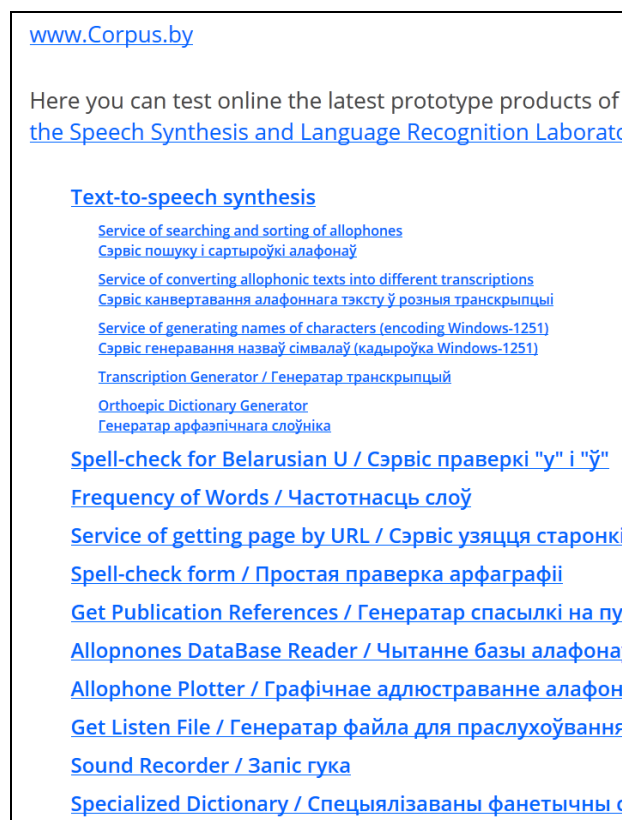
Распрацоўка ССМТ з'яўляецца складанай задачай, якая ахоплівае вялікую колькасць падзадач, а таксама сутыкаецца з шэрагам пабочных прыкладных лінгвістычных задач, якія патрабуюць вырашэння. Задачы, якія паўстаюць пры стварэнні ССМТ, можна ўмоўна падзяліць на тры групы:

- перадапрацоўку ўваходных тэкстаў;
- непасрэдна сінтэз маўлення;
- апрацоўку і выкарыстанне выніковых даных.

Інтэрнэт-рэсурс www.corpus.by мае сваёй мэтай рашэнне як непасрэднай задачы сінтэзавання

маўлення, так і шэрагу дадатковых задач перадапрацоўкі тэксту і апрацоўкі вынікаў працы сінтэзатара маўлення. Дзеля дасягнення дадзенай мэты было прынята рашэнне распрацаваць для гэтага рэсурсу сістэму інтэрнэт-сэрвісаў, кожны з якіх вырашаў бы пэўную ўласную задачу, а ў спалучэнні з іншымі сэрвісамі – неабходныя падзадачы ССМТ. На сённяшні дзень распрацаваны альбо знаходзяцца ў праежжавай стадыі распрацоўкі і даступны бясплатна ў Інтэрнэце наступныя сэрвісы:

- сэрвіс пошуку і сартыроўкі алафонаў;
- сэрвіс канвертавання алафоннага тэксту ў розныя транскрыпцыі;
- сэрвіс генерацыі назваў сімвалаў;
- сэрвіс генерацыі транскрыпцый;
- сэрвіс генерацыі арфаэпічнага слоўніка;
- сэрвіс праверкі “y” і “ў”;
- сэрвіс падліку частотнасці слоў;
- сэрвіс генерацыі спасылкі на публікацыю;
- сэрвіс графічнага адлюстравання алафонаў і алафонных фраз;
- сэрвіс генерацыі файла для праслухоўвання;
- сэрвіс запісу гуку;
- сэрвіс спецыялізаванага фанетычнага слоўніка; і іншыя.



Малюнак 1 – Знешні інтэрфейс інтэрнэт-рэсурсу www.Corpus.by

Знешні інтэрфейс пачатковай старонкі інтэрнэт-рэсурсу www.Corpus.by, на якой знаходзяцца

спасылкі на ўсе пералічаныя вышэй сэрвісы, прадстаўлены на малюнку 1.

2. Фарміраванне патрабаванняў да праектуемых інтэрнэт-сэрвісаў

Мэтай распрацоўкі інтэрнэт-сэрвісаў з’яўляецца павышэнне якасці працы сінтэзатара маўлення па тэксце для беларускай мовы. Аўтарамі артыкула было заўважана, што гэтага можна дасягнуць некалькімі спосабамі: па-першае, праз вынясенне пэўных элементаў існуючай ССМТ для вольнага карыстання ў Інтэрнэце у выглядзе інтэрнэт-сэрвісаў, якія б мелі пэўную мэтавую аўдыторыю; па-другое, праз распрацоўку пэўных інтэрнэт-сэрвісаў, якія б маглі быць у далейшым інтэграваны ў ССМТ, а пакуль удасканалення і адтэставання ў Інтэрнэце. Такім чынам можна атрымаць наступнае:

- непасрэдня вынікі працы кожнага сэрвісу на адвольна зададзеных карыстальнікам уваходных даных. Мноства адпаведнасцяў «уваходныя даныя – выніковыя даныя» можа быць выкарыстана як для аналізу правільнасці працы таго ці іншага элемента ССМТ, так і для стварэння аўтаматызаванага тэставання працы сэрвісу і ССМТ;
- зваротную сувязь ад мэтавай аўдыторыі і экспертаў, у супрацоўніцтве з якімі вядзецца распрацоўка;
- зваротную сувязь ад звычайных зацікаўленых у тэматыцы карыстальнікаў.

Прымаючы да ўвагі пералічаныя вышэй мэты, можна сфармуляваць наступныя асноўныя патрабаванні да праектуемых інтэрнэт-сэрвісаў:

- простасць, зручнасць і інтуітыўная зразумеласць карыстальніцкага інтэрфейсу;
- захаванне ўсіх даных, пададзеных карыстальнікам на ўваход;
- захаванне ўсіх выніковых даных;
- аператыўнае аўтаматызаванае апаўнячэнне распрацоўшчыкаў аб памылках у працы сэрвісаў і сістэмы сінтэзу маўлення.

Праектаванне інтэрнэт-сэрвісаў уключае ў сябе як вызначэнне патрабаванняў, так і апісанне статычнай структуры і ўзаемадзеяння элементаў сістэмы. У якасці графічнай мовы дакументавання выкарыстоўваецца натацыя UML 2.0 (Unified Modeling Language).

На малюнку 2 прадстаўлена схема кампанентаў узаемадзеяння карыстальніка з шэрагам службовых сэрвісаў сістэмы адкрытых кампанентаў для пабудавання маўленчага інтэрфейсу – www.Corpus.by. Пяройдзем да разгляду функцыянальных магчымасцей праектуемых інтэрнэт-сэрвісаў.

3. Вызначэнне варыянтаў выкарыстання інтэрнэт-сэрвісаў

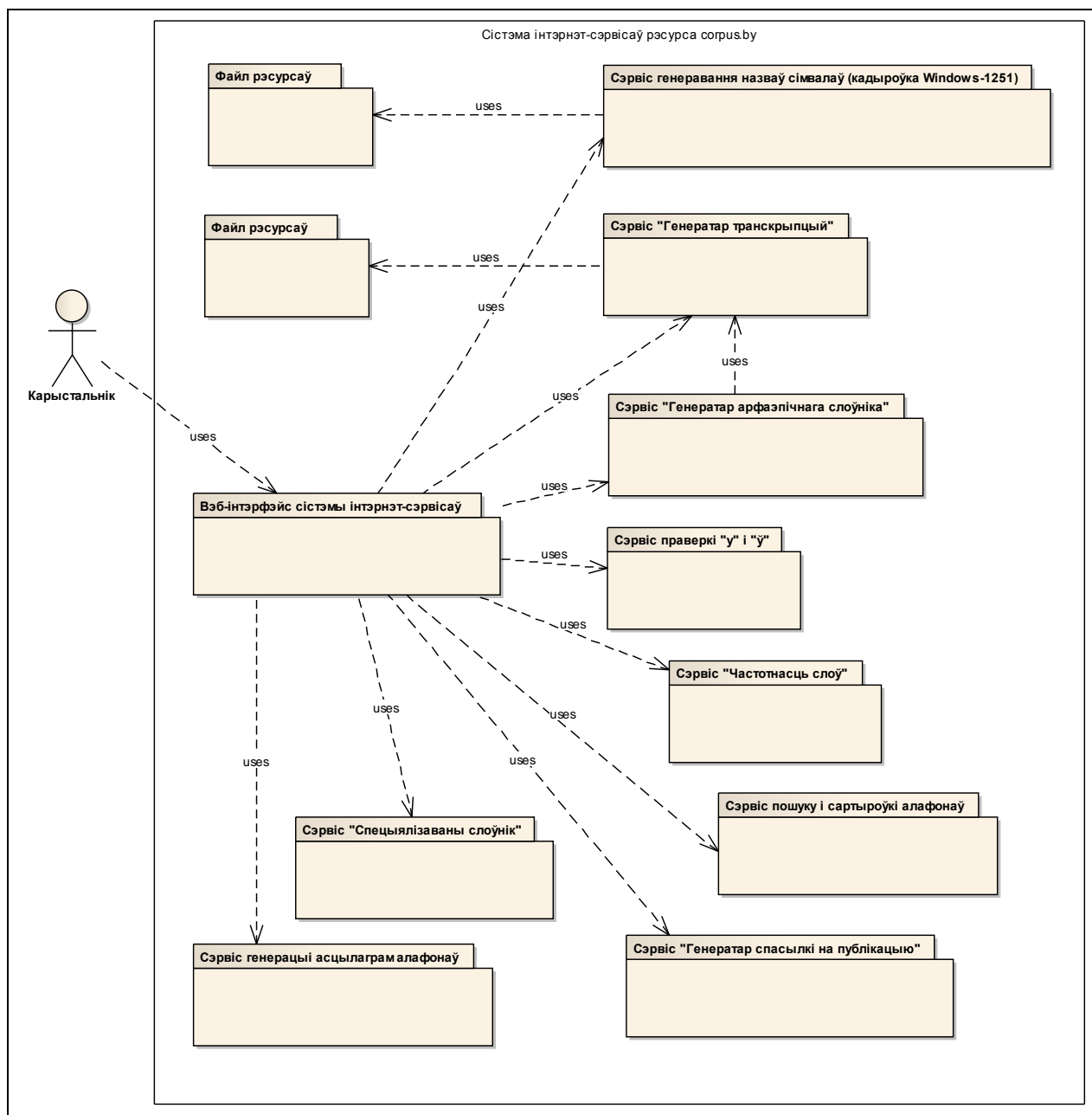
Праектуемая сістэма інтэрнэт-сэрвісаў прадугледжвае выкарыстанне яе як знешнімі

карыстальнікамі, так і распрацоўшчыкамі. Выкарыстанне інтэрнэт-сэрвісаў аднымі і другімі можа як супадаць, так і адрознівацца, бо шэраг распрацоўшчыкаў з’яўляюцца адначасова і карыстальнікамі ў сваіх працоўных мэтах.

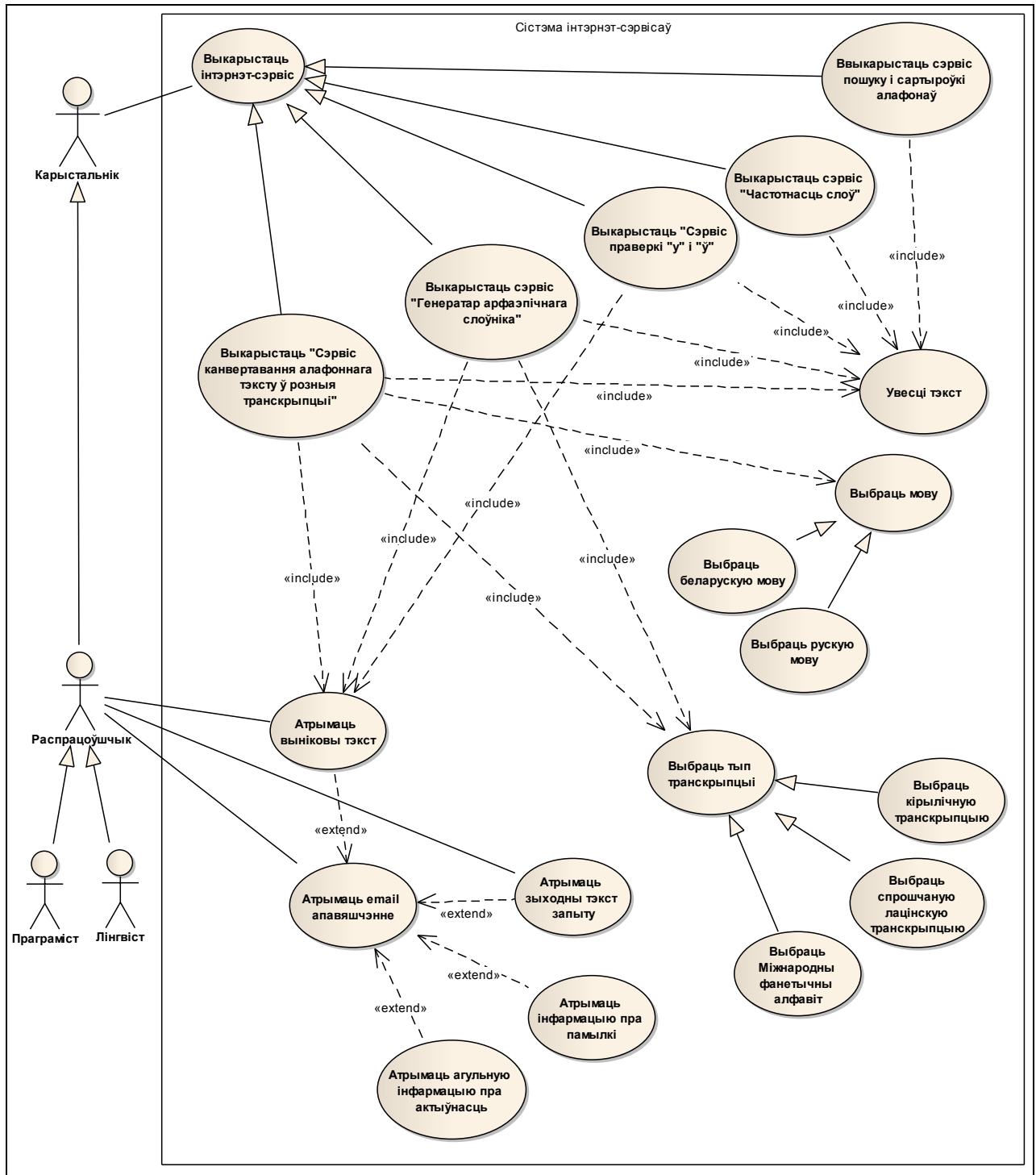
На малюнку 3 прадстаўлена схема варыянтаў выкарыстання сістэмы інтэрнэт-сэрвісаў карыстальнікамі і распрацоўшчыкамі.

На схеме варыянтаў выкарыстання прыведзены наступныя сэрвісы: “Сэрвіс канвертавання алафоннага тэксту ў розныя транскрыпцыі”, “Генератар арфаэпічнага слоўніка”, “Сэрвіс праверкі “у” і “ў”, сэрвіс “Частотнасць слоў”, “Сэрвіс пошуку і сартыроўкі алафонаў”. У кожным

з іх карыстальнік можа падаць на ўваход пэўны тэкст альбо паслядоўнасць сімвалаў і атрымаць на выхадзе неабходны выніковы тэкст у адпаведнасці з задачай, якую вырашае дадзены сэрвіс. У «Сэрвісе канвертавання алафоннага тэксту ў розныя транскрыпцыі» і сэрвісе «Генератар арфаэпічнага слоўніка» карыстальнік можа абраць адзін з трох тыпаў транскрыпцыі: кірылічную, спрошчаную лацінскую ці транскрыпцыю ў фармаце Міжнароднага фанетычнага алфавіту. У «Сэрвісе канвертавання алафоннага тэксту ў розныя транскрыпцыі», як і ў шэрагу іншых, не пазначаных на схеме сэрвісаў, магчымы выбар мовы ўваходнага тэксту: беларускай альбо рускай.



Малюнак 2 – Схема кампанентаў узаемадзеяння карыстальніка з сістэмай інтэрнэт-сэрвісаў www.Corpus.by



Малюнак 3 – Схэма варыянтаў выкарыстання сістэмы інтэрнэт-сэрвісаў www.Corpus.by

Распрацоўшчыкі могуць выкарыстоўваць інтэрнэт-сэрвісы для аналізу працы сэрвісаў і прыняцця рашэнняў па дапрацоўцы і выпраўленні памылак іх працы, а таксама для вырашэння шляхоў развіцця, пошуку сродкаў і спосабаў павышэння эфектыўнасці працы праз назіранне за пэўнымі тэндэнцыямі і заканамернасцямі ў карыстальніцкіх запытах да сэрвісаў.

Дадзеныя сэрвісы праектуюцца як для звычайных карыстальнікаў, так і для адмысловай мэтавай аўдыторыі: лінгвістаў-экспертаў і ўласна

распрацоўшчыкаў праграмных сродкаў: дадзеных сэрвісаў і ССМТ. Такім чынам, распрацоўшчыкі (лінгвісты і праграмісты) з'яўляюцца і прыватным выпадкам карыстальніка, але між іншага валодаюць і сваімі ўласнымі спецыфічнымі магчымасцямі выкарыстання.

Звычайныя карыстальнікі маюць магчымасць бясплатна выкарыстоўваць інтэрнэт-сэрвісы рэсурсу www.Corpus.by у адвольных мэтах. Прывядзём апісанне некалькіх распрацаваных і даступных у інтэрнэце сэрвісаў.

4. Апісанне сэрвісаў: функцыянальныя магчымасці, спосабы выкарыстання

4.1. Генератар файлаў для праслухоўвання

Сэрвіс “Generator of Listen File” ставіць сваёй мэтай прывесці хаця б па адным прыкладзе для адлюстравання кожнага канкрэтнага алафона ці дыфона (для дыфонаў пакуль рэалізаваны толькі выпадак “санорны зычны – галосны”) [Listen file, 2014]. У аснове дадзенага сэрвісу ляжаць спісы слоў, якія былі спецыяльна складзены экспертам. Для зручнасці карыстання і ўспрымання, падабраныя словы згрупіраваны па тэматыках, паколькі адвольны набор слоў, зусім не звязаных між сабой, часта выклікае пэўныя складанасці – карыстальнік падсвядома пачынае суадносіць слова з яго прадметнай вобласцю, што адцягвае ўвагу ад пастаўленай перад ім задачы і запавольвае працу.

Для таго, каб атрымаць спіс слоў для алафонаў, размеркаваных па тэматыках, неабходна націснуць на кнопку “Read Data from Xslx File (Allophones)” (малюнак 4а), адпаведна для дыфонаў – “Read Data from Xslx File (Diphones)” (малюнак 4б). Акрамя прыкладаў карыстальнік таксама атрымае інфармацыю пра колькасць тэматык і слоў, а таксама сярэдняю колькасць слоў у адной тэматыцы.

Функцыянальнасць сэрвісу “Generator of Listen File” дае магчымасць паляпшаць працу сінтэзатара маўлення шляхам начытвання адабраных і разбітых па тэматыках слоў і выразання алафонаў, якіх на дадзены момант няма ў базе сінтэзатара і якія маюць ня вельмі добрую якасць. Акрамя таго дадзены сэрвіс можа выкарыстоўвацца для больш хуткага і зручнага стварэння новых “галасоў” для сінтэзатара.

Generator of Listen File

Generator of Listen File

Read Data from Xslx File (Allophones)

Read Data from Xslx File (Diphones)

Read Import Txt File

Read Import Xslx File

Read Data from Xslx File (Allophones)

Topics = 42, Words = 719, Words per Topic = 17.1

Тэматычны дамен: будоўля..##

аббудава+ць..#
аббі+ць..#
аббіва+нне..#
абпіло+ўванне..#
адту+ліна..#
бу=йнамашта+бных..#
бэ=лечна-кансо+льны..#
гу=каізаля+цыя..#
да+х хі+сткі..#
жыллѐ+..#
не+калькі а+рак..#
пабудава+ць э+лінг..#

Тэматычны дамен: ваенная тэматыка..##

правядзі+ а=нтывае+нную а+кцыю..#
радыёгра+ма..#
раззбрае+нне..#
ружжо+..#

а)

Generator of Listen File

Generator of Listen File

Read Data from Xslx File (Allophones)

Read Data from Xslx File (Diphones)

Read Import Txt File

Read Import Xslx File

Read Data from Xslx File (Diphones)

Topics = 37, Words = 805, Words per Topic = 21.8

Тэматычны дамен: абстрактыныя назо+ўнікі..##

абплята+нне..#
адда+насць..#
аддале+нне..#
адту+ліна..#
аслупяне+нне..#
бясці+лле..#
выбе+льванне..#
ка+шлянне..#
любо+ў..#
рва+нь..#
імкне+нне..#

Тэматычны дамен: будоўля..##

аб'е+ктавы..#
аб'е+мны..#
аб'е+місты..#
аббудава+ць..#
абпіло+ўванне..#

б)

Малюнак 4 – Прыклад адлюстравання слоў для рэпрэзентацыі а) алафонаў; б) дыфонаў

4.2. Генератар спасылкі на публікацыю

Стварэнне бібліяграфічных спасылак – неад’емная частка навуковай працы. Каб спрасціць гэтую задачу для карыстальнікаў, быў створаны сэрвіс “Get a publication reference!” [Publication references, 2014]. Дадзены сэрвіс распрацаваны для аўтаматычнай генерацыі спасылак.

Зайшоўшы на старонку сэрвісу, карыстальніку неабходна пазначыць фармат спасылкі, тып крыніцы (кніга, артыкул часопісу, канферэнцыя, вэб-сайт) і мову афармлення публікацыі (беларуская, руская ці англійская) (малюнак 6).

Наступным крокам з’яўляецца ўвод даных карыстальнікам. Абавязковыя для запаўнення палі пазначаны чырвонай зорчак (*). Таксама прадугледжана магчымасць прагляду прыкладаў запаўнення, для чаго неабходна націснуць на кнопку “Ачысціць усё і паказаць прыклады”. Прыклад запаўнення формы данымі паказаны на малюнку 7.

Малюнак 6 – Пазначэнне фармату, тыпу і мовы публікацыі

Малюнак 7 – Запаўненне формы данымі

Па заканчэнні запаўнення формы неабходна націснуць на кнопку “Атрымаць спасылку на публікацыю”. Вынік з’явіцца ў асобным полі ніжэй (малюнак 8).

На дадзены момант для карыстальнікаў даступны толькі ВАК-фармат публікацый. Плануецца пашырыць магчымасці сэрвісу праз павелічэнне колькасці тыпаў спасылак і далучэнне наступных фарматаў: Chicago, MLA, APA.

Publication references

Гецавіч, Ю.С. Стварэнне сэрвіса арфаэпічнага генератара слоўнікаў / Ю.С. Гецавіч, А.В. Гюнтар, С.І. Лысы [і інш.] // Тезі доповідей міжнародной конференції «Діалекты в синхроніі та дяхроніі : загальнаслов’янський контекст» (Кіев, 2–4 квітня 2014 року) / За ред. П.Ю. Гриценка. Ін-т укр. мови НАН Украіны. Кіев : КММ, 2014. — С. 101-106.

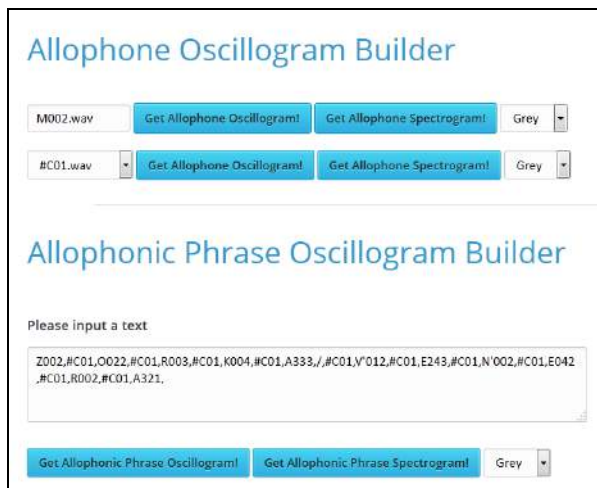
Малюнак 8 – Згенераваная спасылка на публікацыю

Праца з генератарам бібліяграфічных спасылак дазволіла даведацца больш дакладна пра іх змест і структуру, што з’яўляецца добрай асновай для распрацоўкі сістэмы распазнавання спасылак у тэкстах беларускай і рускай моў. Інфармацыя якую можна атрымаць з бібліяграфічных спасылак пры распазнаванні, можа быць карыснай для бібліятэк, выдавецтваў ці навуковых устаноў, бо яна змяшчае звесткі пра аўтараў і рэдактараў, назвы публікацый, назву і месца выдавецтва, год выдання і інш.

4.3. Сэрвіс графічнага адлюстравання алафонаў і алафонных фраз

Сэрвіс графічнага адлюстравання алафонаў і алафонных фраз “Allophone Plotter” – гэта яшчэ адзін са службовых сэрвісаў сістэмы адкрытых кампанентаў для пабудавання маўленчага інтэрфейсу – www.Corpus.by [Allophone Plotter, 2014]. Яго галоўнай задачай з’яўляецца графічнае адлюстраванне фізічнага сігналу ў часавым альбо частотным выглядзе адпаведнай фанетычнай адзінкі – алафона, абранага карыстальнікам сэрвісу, альбо платаванне цэлай фразы па фанетычным тэксце, які мае форму радка з паслядоўна запісаных алафонаў. Для адлюстравання сігналу ў часовай вобласці выкарыстоўваецца графік асцылаграмы – залежнасці амплітуды сігналу ад часу. Для частотнага прадстаўлення выкарыстоўваецца графік спектраграмы – залежнасці амплітуды ці энергіі сігналу ад часу і ад частаты адначасова. У гэтым выпадку графік выглядае як двухмерны каляровы малюнак сігналу з часам, які адкладаецца па восі абсцыс, частатой – па восі ардынат, а ўзровень энергіі сігналу характарызуецца інтэнсіўнасцю колеру на малюнку.

Сэрвіс можа выкарыстоўвацца экспертамі-фанетыстамі, лінгвістамі, студэнтамі філалагічных і педагогічных вузаў ці проста зацікаўленымі асобамі для знаёмства з “выглядам” і фізічнымі характарыстыкамі алафонаў (рэалізацыямі фанем), у працэсе вывучэння беларускай мовы. Напрыклад, спектраграма можа вельмі моцна дапамагчы ў вывучэнні як фанетыкі мовы ў агульным, так і асобных гукаў мовы ў прыватнасці. Кожны алафон мае свае асаблівасці з пункту гледжання яго фізічных параметраў, якія з’яўляюцца вынікамі дзеяння інтра- і экстралінгвістычных фактараў мовы і якія складаюць “жывую непаўторную карціну” гэтага гука, добра бачную на частотна-часавым плане, які сабой уяўляе спектраграма.

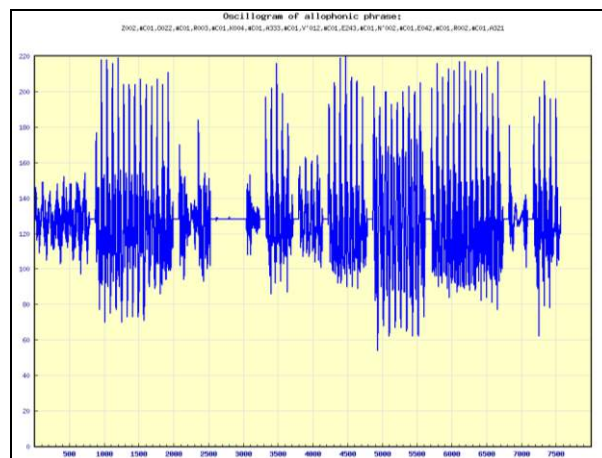


Малюнак 9 – Знешні інтэрфейс “Сэрвісу графічнага адлюстравання алафонаў і алафонных фраз”

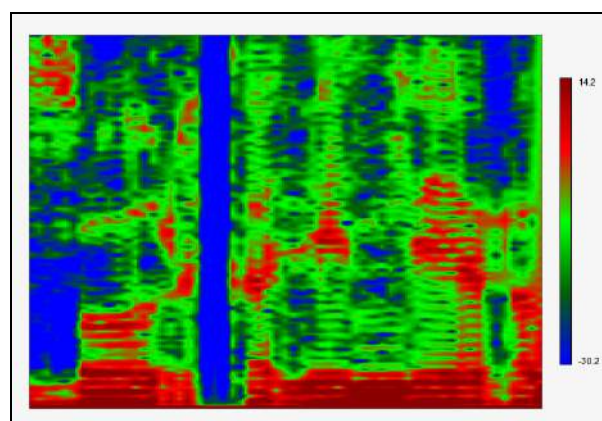
Асцылаграма фанетычнай фразы можа эфектыўна выкарыстоўвацца для вывучэння прасодыкі (інтанацыйнага склада) мовы, ад якой вельмі моцна залежыць успрыманне маўлення чалавекам. Як вядома, прасодыка складаецца з трох кампанентаў: энергетыкі – цяжучай змены сілы гуку, рытмікі – змены працягласці гукаў і паўз і мелодыкі – руху частаты асноўнага тону. Усе гэтыя моўныя з’явы яркава назіраюцца на асцылаграме. Напрыклад, праз адлюстраванне рытмікі асцылаграма можа падказаць карыстальніку, дзе адносна агульнай працягласці фразы, знаходзіцца слоўны ці фразавы націск, таму што ў гэтым месцы на графіку павінна назірацца большая амплітуда сігналу.

На сённяшні момант агульны сцэнар працы з сэрвісам выглядае наступным чынам. Карыстальнік заходзіць на старонку “Сэрвісу графічнага адлюстравання алафонаў” (знешні інтэрфейс старонкі прадстаўлены на малюнку 9). На гэтай старонцы ён можа наплатваць асобны алафон ці алафонную фразу. Асобны алафон можна ўвесці ўручную ў тэкставым полі ці выбраць з фанетычнай базы праз выпадаючае меню. Алафонную фразу карыстальнік павінен уводзіць у спецыяльна адведзенае тэкставае поле, у якім дазволены шматрадковы ўвод. Пасля ўвода асобнага алафона ці алафоннага тэксту карыстальнік павінен націснуць адпаведную кнопку для пабудавання асцылаграмы ці спектраграмы. Пры выбары платавання спектраграмы, у карыстальніка ёсць магчымасць выбару карты колеру: чорна-белай ці каляровай. Далей трэба пачакаць некаторы час пакуль сэрвіс апрацуе запыт і згенеруе адказ у выглядзе html-старонкі, у якую ўбудаваны малюнак сігналу ў фармаце png, і якую браўзер адлюструе карыстальніку. Вынікі працы сэрвісу прадстаўлены на малюнку 10.

Асаблівацю сэрвісу з’яўляецца магчымасць апрацоўкі сігналу і будавання графічных “партрэтаў гукаў жывой мовы on-line”, а таксама яго арыентацыя на беларускую мову.



а)



б)

Малюнак 10 – Вынікі працы “Сэрвісу графічнага адлюстравання алафонаў і алафонных фраз”: а) асцылаграма алафоннай фразы; б) спектраграма алафоннай фразы

На дадзены момант аўтары не маюць інфармацыі аб аналагічных бясплатных on-line сэрвісах для лінгвістаў ці людзей, зацікаўленых у вывучэнні беларускай мовы, якія валодалі б такой жа самай функцыянальнасцю. У далейшым плануецца выкарыстанне гэтага сэрвісу ў якасці дадатковай функцыянальнасці для больш высокаўзроўневых сэрвісаў рэсурсу www.Corpus.by.

Заклучэнне

У выніку праведзенай працы былі распрацаваны агульныя падыходы да стварэння сістэмы інтэрнэт-сэрвісаў www.Corpus.by у мэтах паляпшэння і ўдасканалення працы сістэмы сінтэзу маўлення па тэксце з адначасовай магчымасцю як вырашэння розных прыкладных задач, так і інтэграцыі з іншымі праектамі, у прыватнасці з праектам OSTIS. У дадзеным артыкуле апісаны асаблівасці праектавання, рэалізацыі і выкарыстання сістэмы, прыведзены апісанні некаторых распрацаваных сэрвісаў, а менавіта: сэрвісу “Генератар файлаў для праслухоўвання”, сэрвісу “Генератар спасылкі на публікацыю”, “Сэрвісу графічнага адлюстравання алафонаў і алафонных фраз”.

Бібліяграфічны спіс

[Allophone Plotter, 2014] Allophone Plotter [Electronic resource]. – 2014. – Mode of access : <http://corpus.by/readVoiceDBAllophones/>. – Date of access : 12.12.2014.

[Listen file, 2014] Generator of Listen File [Electronic resource]. – 2014. – Mode of access : <http://corpus.by/genListenFile/>. – Date of access : 12.12.2014.

[Publication references, 2014] Get a publication reference! [Electronic resource]. – 2014. – Mode of access : <http://corpus.by/publicationReference/>. – Date of access : 12.12.2014.

[www.Corpus.by, 2012] Text-to-Speech PHP-Based Synthesizer [Electronic resource]. – 2012. – Mode of access : <http://corpus.by/>. – Date of access : 12.12.2014.

[Гецэвіч, 2012] Гецэвіч, Ю.С. Алгарытмы лінгвістычнай апрацоўкі тэкстаў для сінтэзу маўлен-ня на беларускай і рускай мовах: дыс. ... канд. тэхн. навук / Ю.С. Гецэвіч. – Мінск, 2012. – 193 с.

[Лобанов, 2008] Лобанов, Б.М. Компьютерный синтез и клонирование речи / Б.М. Лобанов, Л.И. Цирульник. – Минск : Белорусская наука, 2008. – 344 с.

WWW.CORPUS.BY: OPEN-SOURCE COMPONENTS FOR NATURAL LANGUAGE INTERFACES

Hetsevich Y.S.*, Lobanov B.M.*, Lysy S.I.*,
Hiuntar E.V.*, Denisyuk D.A.*,
Zakharyeu V.A.**

* *United Institute of Informatics Problems,
National Academy of Sciences, Minsk, Belarus*

**yury.hetsevich@gmail.com,
lobanov@newman.bas-net.by,
stanislau.lysy@gmail.com, lena205593@gmail.com,
d.denisyuk@gmail.com**

** *Belarusian State University of Informatics and
Radioelectronics, Minsk, Belarus*

delfvad@gmail.com

This paper describes general approaches to designing, implementation and usage of open-source components of www.Corpus.by internet-resource for Natural Language Interfaces. These components serve to improve the performance of a text-to-speech system (TTS), while at the same time providing the opportunity of integration with the OSTIS project. The paper also define goals and objectives, ways of operation and available functionality of some services that have been implemented.

INTRODUCTION

In order to solve the problem of speech interaction between people and machines, scientists all over the world are working on creating speech synthesis and recognition systems. Recently, TTS systems have reached certain degree of accuracy and become widely used, but it is impossible to say that the task of text-to-speech synthesis is completely resolved, especially it concerns the Belarusian language. In this regard, the authors decided to create an Internet resource that would bring together a number of Internet-services,

each aimed at solving certain tasks and at the same time helping developers to make the TTS system performance more optimal and precise.

MAIN PART

The tasks facing the developers of a text-to-speech system may be conventionally divided into three groups: preprocessing of input texts, speech synthesis itself and processing of resulting data.

The internet-resource www.Corpus.by is aimed at addressing the challenge of text-to-speech synthesis itself, as well as at solving a range of additional tasks of text-preprocessing and processing of the results of the text-to-speech synthesizer's operation. In order to achieve this aim, the decision was made to develop a system of Internet-services, all of each would solve its own certain task and, when combined with all the rest services, would also solve the essential subtasks of the text-to-speech system. The following services are currently developed or are on the intermediate stage of development and are easily available on the Internet:

- “Service of searching and sorting of allophones”;
- “Service of converting allophonic texts into different transcriptions”;
- “Service of generating names of characters (encoding Windows-1251)”;
- “Transcription Generator”;
- “Orthoepic Dictionary Generator”;
- “Spell-check for Belarusian U”;
- “Frequency of Words”;
- “Get Publication References”;
- “Allophone Oscillogram Builder”;
- “Allophone Plotter”;
- “Get Listen File”;
- “Sound Recorder”;
- “Specialized Dictionary”;
- etc.

CONCLUSION

As a result of the work carried out by the authors, in order to improve and make more precise the performance of the text-to-speech system, and at the same time to provide free electronic services to the population in order to resolve a wide range of applied tasks, general approaches to creation of the system of internet-services www.Corpus.by were developed.



OSTIS-2015

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822:514

РАСПРАЦОЎКА КАМПАНАЕНТА РАСПАЗНАВАННЯ МАЎЛЕННЯ ДЛЯ НАТУРАЛЬНА МАЎЛЕНЧАГА ІНТЭРФЕЙСУ

Нікалаенка К.А.* , Кайгародава Л.І.** , Гецэвіч Ю.С.**

* *Беларускі дзяржаўны ўніверсітэт інфарматыкі і радыёэлектронікі, Мінск, Рэспубліка Беларусь*

anak247@gmail.com

** *Аб'яднаны інстытут праблем інфарматыкі НАН Беларусі, Мінск, Рэспубліка Беларусь*

lesia.piatrouskaya@gmail.com

yury.hetsevich@gmail.com

Апісваецца распрацоўка кампанента для аўтаматычнага распазнавання беларускага маўлення з мэтай кіравання рознымі мабільнымі прыстасаваннямі, у тым ліку робатамі. Сістэма распазнае абмежаваную колькасць каманд вызначаных дыктараў. Вынікі тэставання паказалі, што сістэма можа распазнаваць да 80% каманд. Прыведзены прыклад працы сістэмы распазнавання галасавых каманд, адаптаванай для РНР.

Ключавыя словы: кампанент; распазнаванне маўлення; інтэрфейс; натуральна маўленчы інтэрфейс.

Уводзіны

У аснове кожнай маўленчай тэхналогіі ляжыць так званы «engine» ці ядро праграмы — набор дадзеных і правіл, па якіх ажыццяўляецца апрацоўка дадзеных. У залежнасці ад прызначэння гэтага ядра адрозніваюць TTS і ASR engine. TTS (Text-to-Speech) engine дае магчымасць сінтэзу маўлення па тэксце, а ASR (Automatic Speech Recognition) engine прызначана для распазнавання маўлення. Існуе некалькі буйных вытворцаў, якія займаюцца стварэннем ASR ядзер. Сярод іх самымі даступнымі і папулярнымі з'яўляюцца Sphinx, НТК, Julius і Kaldi. Разгледзім іх.

CMU Sphinx складаецца з серыі распазнавальнікаў маўлення і трэніроўшчыка акустычнай мадэлі. Sphinx – гэта дыктаранезалежны распазнавальнік бесперапыннага маўлення, які выкарыстоўвае Схаваную Маркаўскую мадэль і п-грамную статыстычную моўную мадэль [Sphinx, 2014].

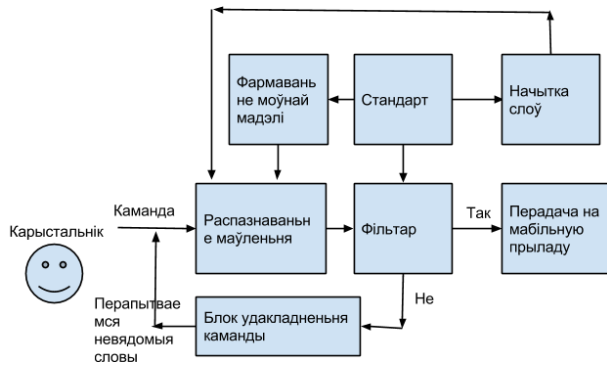
НТК — гэта інструментарый для распазнавання маўлення, што выкарыстоўвае Схаваную Маркаўскую мадэль. Праграмны пакет НТК — праграмае забеспячэнне для апрацоўкі НММ мадэляў. НТК уяўляе сабою набор бібліятэк і інструментаў, якія могуць быць выкарыстаны ў аналізе і працы з маўленчым сігналам [НТК, 2014].

Julius — гэта распазнавальнік бесперапыннага маўлення з вялікім слоўнікам, дэкодар праграмага забеспячэння для даследавання ў вобласці злучанага маўлення і распрацоўкі. Для запуску распазнавальніка маўлення Julius трэба падабраць моўную і акустычную мадэль для мовы. Julius адаптуе акустычную мадэль кадананага фармату НТК ASCII, базу дадзеных вымаўлення фармату НТК, і 3-х узроўневых 2-грам пабудовы моўнай мадэлі стандарта [Julius, 2014].

Kaldi — падобны да НТК з пункту гледжання мэты і сферы ўжывання прадукт. Асноўнай мэтай распрацоўшчыкаў з'яўляецца стварэнне сучаснага і лёгка пераноснага кода, які лёгка мадыфікаваць і пашыраць [Kaldi, 2014].

Існуюць і іншыя, больш спецыфічныя сістэмы распазнавання маўлення, такія як iATROS, RWTH ASR, Simon, а таксама больш марудныя воблачныя сэрвісы Google ASR і Yandex ASR.

Мэта гэтай працы — распачаць стварэнне сістэмы распазнавання беларускага маўлення для кіравання мабільнымі робатамі. Для гэтага патрэбна стварыць акустычную базу і моўную мадэль, якая будзе максімальна блізкай да натуральнай мовы. Для нас важна асобна даследаваць гэтыя дзве задачы, а потым аб'яднаць у адно. Прынцыповая архітэктурна сістэмы распазнавання галасавых каманд паказана на малюнку 1.



Малюнак 1 – Прынцыповая архітэтура сістэмы аўтаматычнага распазнавання маўлення

Для стварэння акустычнай мадэлі нам падыходзіць праграмны пакет НТК, так як ён мае магчымасці настройкі і апрацоўкі НММ мадэляў, а таксама мае вельмі падрабязную дакументацыю. Нездарма на ім будзе акустычная база шматлікіх распазнавальнікаў. Для стварэння моўнай мадэлі будзем выкарыстоўваць інструментарый NooJ — настройвальны лінгвістычны працэсар, які дазваляе будаваць спецыялізаваныя электронныя слоўнікі, правілавыя сінтаксічныя і марфалагічныя граматыкі для апрацоўкі электронных тэкстаў (корпусаў ці тэкставых запытаў) у рэальным часе.

1. Праект моўнай мадэлі сістэмы распазнавання

На пачатку стварэння моўнай мадэлі мы выкарыстаем глыбокі сінтаксічны аналіз, каб атрымаць мадэль, якая будзе проста для ўспрымання, але таксама будзе адлюстроўваць

паўнату мадэлі натуральнай мовы. Будзем ужываць такія канцэпты як 'Суб'ект', 'Дзеянне', 'Аб'ект' і 'Характарыстыка'. Пры выкарыстанні інструментарыя *NooJ Syntactic Grammar* праектуем графавую мадэль для аб'яднання гэтых канцэптаў і лінгвістычных адзінак, з якіх будзе складацца гэтыя канцэпты. Далей мы генерыруем слоўнік для робатаў з дапамогай інструментарыя *NooJ Dictionary*. Некаторыя адзінкі з гэтага слоўніка будзе выглядаць наступным чынам:

Робат_Віцебск прынясі лыжку,
GUID=R1+Action=take+Object=spoon

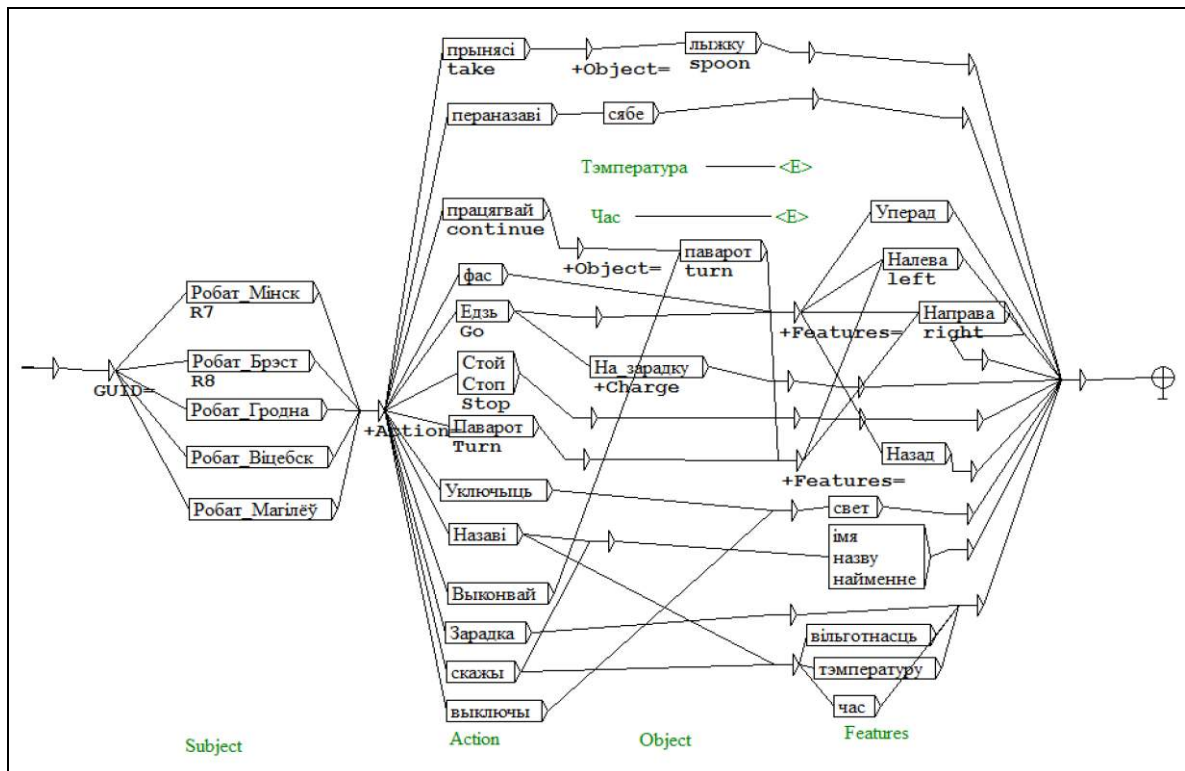
Робат_Гродна Едзь На_зарадку,
GUID=R2+Action=Go+Charge

Робат_Брэст Выконвай паварот Направа,
GUID=R3+Action=turn+Features=right

Робат_Брэст Выконвай паварот Налева,
GUID=R3+Action=turn+Features=left

Тут канцэпт 'Суб'ект' прадстаўляе сабой імя робата, 'Дзеянне' — каманда, якую павінен выканаць робат, 'Аб'ект' — мэтанакіраванне для дзеяння, 'Характарыстыка' — удакладненне для канцэптаў 'Аб'ект' ці 'Дзеянне'.

Выкарыстоўваючы гэтыя канцэпты і інструментарый *NooJ*, у выніку мы атрымалі моўную мадэль для сістэмы ўзаемадзеяння чалавека і робата. Схема мадэлі бачна на малюнку 2.



Малюнак 2 – Схема моўнай мадэлі з выкарыстаннем інструментарыя NooJ

```

C:\Windows\system32\cmd.exe - C:\WebServers\home\corpus1by\www\dialogCalc\HTKwork\tst.bat
Read 30 physical / 30 logical HMMs
Read lattice with 22 nodes / 38 arcs
Created network with 162 nodes / 178 links
File: wavs/ct1.wav
sil robot_minsk sil edz sil uperad sil == [352 frames] -68.1784 [Ac=-23998.8 L
M=0.0] <Act=153.1>

Z:\home\recUby\www\dialogCalc\HTKwork>HUite -o ST -T 1 -l '*' -C config -a -H hm
m7/macros -H hmm7/hmmdefs -i recout7.mlf -p 0.0 -s 5.0 -S test.scp -w wnet dict
.txt monophones.txt
Read 30 physical / 30 logical HMMs
Read lattice with 22 nodes / 38 arcs
Created network with 162 nodes / 178 links
File: wavs/ct1.wav
sil robot_minsk sil edz sil uperad sil == [352 frames] -68.1960 [Ac=-24005.0 L
M=0.0] <Act=153.1>

Z:\home\recUby\www\dialogCalc\HTKwork>HUite -o ST -T 1 -l '*' -C config -a -H hm
m8/macros -H hmm8/hmmdefs -i recout8.mlf -p 0.0 -s 5.0 -S test.scp -w wnet dict
.txt monophones.txt
Read 30 physical / 30 logical HMMs
Read lattice with 22 nodes / 38 arcs
Created network with 162 nodes / 178 links
File: wavs/ct1.wav

```

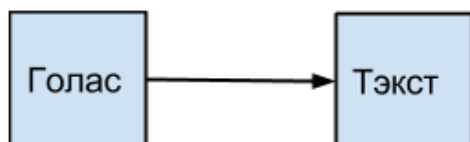
Малюнак 3 – Прыклад працы сістэмы распазнавання маўлення на базе пакета НТК

2. НТК як прылада для распазнавання

На базе НТК было створана ядро сістэмы распазнавання галасавых камандаў для кіравання мабільнымі робатамі на беларускай мове. У сістэме было 17 простых каманд кіравання і 5 эталонных галасоў розных дыктараў, у тым ліку і жаночых. Прыкладамі каманд могуць з’яўляцца словы: уперад, назад, налева, паварот і г.д. Назвы робатаў былі выбраныя адпаведна назвам абласным цэнтрам Рэспублікі Беларусь: Мінск, Гродна, Магілёў, Брэст, Віцебск, Гомель. Прыкладамі каманд-дзеянняў могуць з’яўляцца словы: едзь, стой, падымі, апусці. Пасля навучання і тэставання НММ мадэляў сістэмы НТК дакладнасць распазнавання склала каля 80% для мужчынскіх галасоў і каля 50-75% для жаночых. Прыклад працы сістэмы распазнавання галасавых камандаў на беларускай мове на базе НТК прыведзена на малюнку 3.

3. Прыклад распазнавання маўленчых каманд для мабільнага робата

Ідэя распазнавання маўленчых каманд заключаецца ў тым, каб атрымаць тэкставую інфармацыю з галасу карыстальніка сістэмы і вызначыць у ім каманду для кіравання мабільным робатам (малюнак 4).



Малюнак 4 – Агульны прынцып працы распазнавальніка маўлення

У працэсе працы натрэніраваны набор бібліятэк НТК быў інтэграваны ў мову праграмавання РНР. РНР – скрыптовая мова

праграмавання агульнага прызначэння, якая інтэнсіўна ўжываецца для распрацоўкі вэб-прылад. У выніку быў распрацаваны распазнавальнік маўлення, які адаптуе вывад працы бібліятэк НТК да адмысловага фармату, з якім працягвае працаваць РНР – выдае карыстальніку.

Разгледзім прыклад працы створанага прататыпа тэставай сістэмы, якая распазнае каманды для мабільнага робата.

Крок 1. Выбар аўдыёфайла для распазнавання (малюнак 5). Праз прыладу для запісу галасу запісваецца аўдыёфайл з маўленчай камандай для сістэмы, якая разглядаецца. Потым гэты файл змяшчаецца ў папку з уваходнымі дадзенымі для сістэмы. У выніку, файл робіцца даступным для выбару ў спісе файлаў-прыкладаў каманд. Карыстальнік выбірае файл-прыклад і націскае на кнопку ‘Recognize!’.

Крок 2. Распазнаванне маўлення выбранага аўдыёфайла (малюнак 6). Пры націску па кнопцы ‘Recognize!’ назва выбранага файла перадаецца ў праграмы модуль, які рэалізаваны на РНР, які далей запускаяе пакет НТК. Гэта рэалізавана праз спецыяльны файл-сцэнар працы НТК, які выклікаецца з РНР камандай *exec*. З дапамогай атрыманай раней натрэніраванай сістэмы распазнавання маўлення на базе НТК у пункце 2 з аўдыяфайла атрымліваюцца распазнаныя каманды.

Крок 3. Вывад распазнаных каманд для мабільнага робата (малюнак 7). Сістэма выдае карыстальніку вынікі распазнанай маўленчай каманды ў выглядзе тэксту ў тэрмінах суб’ект, дзеянне, аб’ект, характарыстыка.

Recognizer of commands for mobile robots

choose audio-file:

testcommand.wav

- a1.wav
- a2.wav
- aa1.wav
- aa2.wav
- b1.wav
- b2.wav
- ct1.wav**
- ct2.wav
- ct3.wav
- ct4.wav
- ct5.wav
- ct6.wav
- r1.wav
- r2.wav
- test_edz.wav
- testcommand.wav
- u4.1.test.wav
- u4.2.test.wav
- u4.3.test.wav
- u4.4.test.wav

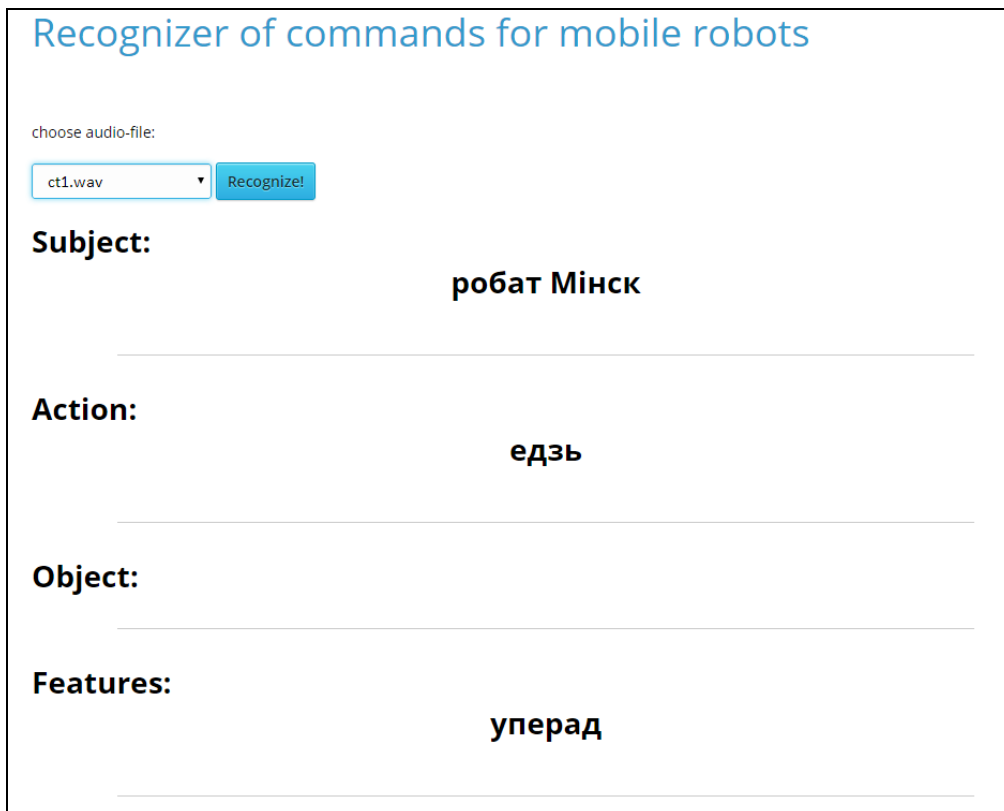
Малюнок 5 – Спіс гукавых файлаў-прыкладаў для падачы на ўваход сістэмы распазнавання маўлення на базе PHP і НТК

```
C:\Windows\system32\cmd.exe - C:\WebServers\home\corpus1by\www\dialogCalc\HTKwork\tst.bat
Read 30 physical / 30 logical HMMs
Read lattice with 22 nodes / 38 arcs
Created network with 162 nodes / 178 links
File: wavs/wavs1.wav
sil robot_minsk sil edz sil uperad sil == [352 frames] -68.1784 [Ac=-23998.8 L
M=0.0] (Act=153.1)

Z:\home\recUby\www\dialogCalc\HTKwork>HUnit -o ST -T 1 -l '*' -C config -a -H hm
m7/macros -H hmm7/hmmdefs -i recout7.mlf -p 0.0 -s 5.0 -S test.scp -w wnet dict
.txt monophones.txt
Read 30 physical / 30 logical HMMs
Read lattice with 22 nodes / 38 arcs
Created network with 162 nodes / 178 links
File: wavs/ct1.wav
sil robot_minsk sil edz sil uperad sil == [352 frames] -68.1960 [Ac=-24005.0 L
M=0.0] (Act=153.1)

Z:\home\recUby\www\dialogCalc\HTKwork>HUnit -o ST -T 1 -l '*' -C config -a -H hm
m8/macros -H hmm8/hmmdefs -i recout8.mlf -p 0.0 -s 5.0 -S test.scp -w wnet dict
.txt monophones.txt
Read 30 physical / 30 logical HMMs
Read lattice with 22 nodes / 38 arcs
Created network with 162 nodes / 178 links
File: wavs/ct1.wav
```

Малюнок 6 – Дэталізаваная справаздача працы функцый распазнавання НТК



Малюнак 7 – Вынік распазнавання каманды ў тэрмінах суб’ект, дзеянне, аб’ект, характарыстыка для мабільнага робата на галасавы запіт карыстальніка

4. Сэрвіс запісу голасу

Прылада запісу голасу была распрацавана як Інтэрнэт-сэрвіс запісу голасу. Гэтая прылада не патрабуе ўстаноўкі ніякіх іншых праграм акрамя Adobe Flash Player і дазваляе наўпрост запісаць голас на сервер праз мікрафон. Дадзеная прылада была рэалізавана з дапамогай Флэш-прылады Wami, якая была інтэгравана ў PHP з дапамогай JavaScript. Знешні інтэрфейс прылады на малюнку 8.

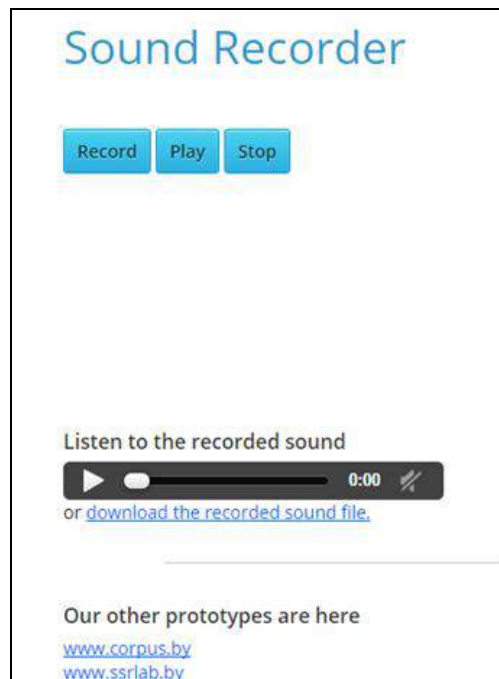
Прылада складаецца з двух частак. Першая частка прадстаўлена трыма кнопкамі, праз якія карыстальнік мае магчымасць пачаць запіс голасу, прайграць запісаны гук і спыніць запіс ці праслухванне. Другая частка інтэрфейсу прадстаўлена фрэймам, які дазваляе прайграць запісаны аўдыёфайл, а таксама скачаць яго.

Заўважым, што лінк на назву файла будзе генеравацца ўнікальна. Для фарміравання ўнікальнасці імя выкарыстоўваецца імя, дата, ай-пі адрас карыстальніка і рандомны лік. Такім чынам, любы карыстальнік зможа выкарыстоўваць такі аўдыёфайл для сваіх іншых мэт па спецыяльнай унікальнай спасылцы.

У будучым плануецца выкарыстоўваць гэты сэрвіс для начыткі галасавых каманд рознымі дыктарамі. Гэтыя дыктары будуць заходзіць на старонку сэрвіса праз свае хатнія камп’ютары і начытваць пэўную паслядоўнасць каманд. Праз гэтыя будуць будавацца ў паўаўтаматычным рэжыме

шматлікія акустычныя мадэлі, што будзе набліжаць нашу сістэму распазнавання маўлення да дыктаранезалежнай.

Зараз гэты сэрвіс даступны ў рэжыме on-line на сайце www.corpus.by [Sound Recorder, 2015].



Малюнак 8 – Знешні інтэрфейс сістэмы запісу гуку

Заклучэнне

Такім чынам, быў створаны першы прататып сістэмы распазнавання маўлення галасавых камандаў для кіравання мабільнымі робатамі на беларускай мове. Дакладнасць распазнавання дастатковая для выкарыстання сістэмы ў рэальных мабільных прыстасаваннях. Заўважым, што колькасць каманд для распазнавання можна змяняць у залежнасці ад мэт той ці іншай сістэмы.

У перспектыве два сэрвісы – сістэма па распазнаванню і сістэма запісу голасу з мікрафона будуць аб'яднаны ў адзіную сістэму, якая павінна стаць інтэрфейсам для галасавога ўводу дадзеных. Такую сістэму можна будзе выкарыстаць як модуль галасавога ўводу ў розных сістэмах, у тым ліку, для сістэмы OSTIS [Гецевіч, 2011] .

Далей плануецца дадаць у распрацаваны прататып моўную мадэль, якую можна распрацаваць, напрыклад, з дапамогай прылады NooJ, для вырашэння наступных задач:

- вызначэнне набору каманд для кіравання мабільнымі робатамі;
- распрацоўка правіл адсячэння неадпаведных каманд для выканання мабільнымі робатамі.

Спіс літаратуры

- [[www.Corpus.by](http://www.corpus.by), 2012] Text-to-Speech PHP-Based Synthesizer [Electronic resource]. – 2012. – Mode of access : <http://corpus.by/>. – Date of access : 12.12.2015.
- [[Sound Recorder](http://soundrecorder.com), 2015] Запіс гука // [Электронны рэсурс]. — 2014. Рэжым доступу : <http://corpus.by/soundRecorder/>. — Дата доступу : 10.01.2015.
- [[HTK](http://htk.eng.cam.ac.uk), 2014] НТК [Электронны рэсурс] – Рэжым доступу: <http://htk.eng.cam.ac.uk> – Дата доступу: 22.06.2014.
- [[Sphinx](http://cmusphinx.sourceforge.net/), 2014] CMU Sphinx [Электронны рэсурс] – Рэжым доступу: <http://cmusphinx.sourceforge.net/> – Дата доступу: 26.10.2014.
- [[KALDI](http://kaldi.sourceforge.net/), 2014] KALDI [Электронны рэсурс] – Рэжым доступу: <http://kaldi.sourceforge.net/> – Дата доступу: 26.10.2014
- [[Julius](http://julius.sourceforge.jp/en_index.php), 2014] Julius [Электронны рэсурс] – Рэжым доступу: http://julius.sourceforge.jp/en_index.php – Дата доступу: 26.10.2014
- [[PHP](http://php.net/), 2014] PHP [Электронны рэсурс] – Рэжым доступу: <http://php.net/> – Дата доступу: 26.10.2014
- [[Гецевіч, 2011](#)] Естэственна-языковыя інтэрфейсы інтэлектуальных впрасно-ответных систем / В.М. Вяльцев, Ю.С. Гецевіч, В.А. Житко, А.А. Кузьмін // Доклады БГУИР. – 2011. – № 8 (62). – С. 80–86.

COMPONENT DESIGN FOR SPEECH RECOGNITION OF NATURAL LANGUAGE INTERFACE

Nikolaenko K.A. *, Kaigorodova L.I. **, Hetsevich Y.S. **

**Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus*

anak247@gmail.com

***United Institute of Informatics Problems, National Academy of Sciences, Minsk, Republic of Belarus*

lesia.piatrouskaya@gmail.com

yury.hetsevich@gmail.com

INTRODUCTION

This work is the start for further design of recognition system for robots-human interaction. The goal of the project is to interact with some number of robots in order to make them perform commands.

MAIN PART

In this article we present the design of some building blocks of the recognition system.

Using deep syntactic analysis and NooJ tools we design the language that would be common and close to every-day language of the humans and that it would be able for machines to 'understand' it.

HTK toolkit was integrated with PHP programming language for our recognition system. In the result syntactic analyzer has been created to gain text information out of voice data.

CONCLUSION

In the result the prototype of recognition system of robots-human interaction has been designed. This prototype can be the basis for two services to be combined: recognition system and dictation system with the use of microphones.



OSTIS-2015

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822:514

РАЗДЕЛЬНОЕ МОДЕЛИРОВАНИЕ РЕЧЕВОГО СООБЩЕНИЯ В ВИДЕ ГОЛОСОВЫХ, ФОНЕТИЧЕСКИХ И ПРОСОДИЧЕСКИХ ПАРАМЕТРОВ

Азаров И.С., Петровский А.А.

*Белорусский государственный университет информатики и радиоэлектроники,
г. Минск, Республика Беларусь*

azarov@bsuir.by

palex@bsuir.by

В работе исследуется возможность отдельного моделирования речевого сообщения (речевого сигнала) путем разделения его на голосовую, фонетическую и просодическую составляющие. В работе предлагается практический способ построения модели сообщения и приводятся результаты ее экспериментального применения к задаче конверсии голоса. Модель использует искусственную нейронную сеть, устроенную по принципу автоматического кодера, устанавливающую соответствие между пространством речевых параметров и пространством возможных фонетических состояний, унифицированным для произвольного голоса.

Ключевые слова: модель речевого сообщения, фонетика, просодика, искусственные нейронные сети

Введение

В настоящее время большую часть существующих задач обработки речевых сигналов можно условно разделить на следующие основные направления: синтез речи по тексту, распознавание речи, создание различных звуковых эффектов (например, конверсия голоса), кодирование речи и улучшение речевых характеристик (например, повышение разборчивости речи и шумоподавление).

Синтез речи по тексту и распознавание речи являются, вероятно, наиболее важными из всех, поскольку в перспективе могут привести к организации полноценного голосового интерфейса между человеком и компьютером. Для качественного синтеза речи по тексту сегодня вместо аллофонного синтеза применяется корпусный, использующий десятки часов речевого материала. Создание такой базы данных для каждого диктора является очень сложным процессом, требующим привлечения большого числа специалистов. Что касается распознавателей слитной речи, то для них при обучении системы распознавания теперь используются намного большие речевые выборки, содержащие различные

Задача изменения голоса (конверсии голоса) появилась сравнительно недавно, тем не менее развитие данного направления происходит очень активно. Целью конверсии голоса является замена

личности говорящего при сохранении содержания исходного речевого сообщения. Решение данной задачи подразумевает установление соответствия между голосом исходного диктора и целевого на основе некоторого обучающего речевого материала. Основной проблемой в конверсии голоса является несоответствие обучающих данных. Самый качественный результат конверсии достигается при использовании параллельных обучающих фраз, т.е. когда исходный и целевой дикторы произносят одни и те же обучающие фразы. Однако, более практичными, являются решения, позволяющие выполнять обучение при помощи произвольных текстовых корпусов не соответствующих друг другу. Далее эти два подхода будем называть «текстозависимым» и «текстонезависимым» соответственно. Последние опубликованные результаты показывают что возможно достигать средней разборчивости и узнаваемости целевого диктора более 75%, что, учитывая сложность задачи, является достаточно высоким показателем.

Кодирование речи является одним из самых первых направлений в цифровой обработке речевых сигналов. Наиболее эффективными здесь являются решения, основанные на параметрическом моделировании речевого сигнала. Современные вокодерные системы обеспечивают удовлетворительное кодирование узкополосного речевого сигнала со скоростью потока 2.4–12кбит/с и широкополосного со скоростью потока выше 8кбит/с. Не решенной в полной мере остается задача

кодирования речи на сверхнизких скоростях потока менее 1.2кбит/с.

Все вышеперечисленные задачи относятся к одному и тому же объекту исследования (речевому сигналу), и, несмотря на все имеющиеся различия, между ними существует внутренняя взаимосвязь. Дальнейший успех в решении каждой из этих задач зависит от того, насколько удачно моделируется речь как феномен в различных его аспектах: представление процесса речеобразования, интерпретации содержания речевого сообщения (в том числе фонетического, смыслового, эмоционального) и процесса восприятия. Потому разработка способов для наиболее адекватного и универсального моделирования речевого сигнала представляется перспективным научным направлением.

В данной работе исследуется возможность создания параметрического описания речевого сигнала, основанного на раздельном моделировании голоса диктора и содержания речевого сообщения. Фундаментальной проблемой является разделение параметров речевого сигнала на индивидуальные характеристики диктора, а также фонетическое и просодическое содержимое. Похожее разделение естественным образом возникает в задаче синтеза речи по тексту, поскольку отдельно существует текст (содержание сообщения) и модель голоса (речевая база данных, соответствующая заданному диктору), которая используется при озвучивании этого текста. В распознавании речи решается похожая задача разделения, однако в обратном направлении: на вход поступает речевой сигнал, на выходе нужно сформировать текст сообщения. В отличие от классических задач преобразования речь->текст и текст->речь в настоящей работе предлагается использовать несколько другую логику разделения параметров речи. Предполагается, что озвучиваемое сообщение содержит не только текстовую информацию и состоит не из последовательности фонем, а скорее из последовательности состояний речевого тракта, выполняющих универсальную фонетическую и просодическую функции для любого голоса. Таким образом, ставится задача реализации пары преобразований $\text{речь} \rightarrow \text{фонемы} + \text{просодика}$ и $\text{фонемы} + \text{просодика} \rightarrow \text{речь}$ с использованием параметрического описания голоса исходного и целевого дикторов. Постановка задачи в таком виде делает возможную область применения модели голоса очень широкой, поскольку во-первых, модель является фонетически мотивированной, однако не подразумевает явного преобразования речи в текст; во-вторых, позволяет сохранять и использовать просодику исходного речевого сообщения; в-третьих, теоретически после преобразования сохраняется возможность восстановления исходного сигнала с субъективно незначительными потерями, что невозможно достичь в результате последовательного выполнения преобразований $\text{речь} \rightarrow \text{текст}$ и $\text{текст} \rightarrow \text{речь}$. Модель речевого сообщения, предлагаемая в данной работе

использует нейронную сеть, построенную по принципу автоматического кодера, реализованного в виде искусственной нейронной сети (autoencoder) [Hinton G.E., 2006]. В работе также приводятся практические результаты применения предлагаемой модели для конверсии голоса.

1. Параметрическое представление речевого сигнала

Для выполнения обработки речевого сигнала необходимо выполнить его параметрическое описание, т.е. представить в виде последовательности характеристических векторов одинаковой размерности. В контексте решаемой задачи речевой сигнал удобно рассматривать как процесс имеющий квазипериодические (детерминированные) и шумовые (стохастические) составляющие [D'Alessandro C., 1995]. Можно считать, что квазипериодические составляющие порождаются периодическими колебательными движениями голосовых связок и характерны для гласных (вокализованных звуков), в то время как шумовые возникают вследствие неперiodических колебаний и характерны для шипящих согласных (невокализованных звуков). Моделирование процесса обеспечивается путем раздельного представления каждой из этих составляющих при помощи разных средств описания. Этот подход широко применяется в современных системах обработки речи [Kawaahra H., 2009]. В данной работе используется аналогичный способ параметрического описания, реализованный при помощи алгоритма оценки параметров периодической модели с многокомпонентным гармоническим возбуждением [Azarov E., 2014].

Речевой сигнал разбивается на перекрывающиеся фрагменты каждый из которых описывается набором параметров: спектральной огибающей, мгновенной частотой основного тона (если фрагмент вокализованный) и типом возбуждения, который может быть вокализованным, невокализованным либо смешанным.

Квазипериодическая составляющая речевого сигнала $s(n)$ представляется в виде суммы синусоид или действительной части комплексных экспонент с непрерывной амплитудой, частотой и фазой, а шумовая как случайный процесс с заданной спектральной плотностью мощности (СПМ):

$$s(n) = \sum_p^P A_p(n) \cos \varphi_p(n) + r(n)$$

где P – число синусоид (комплексных экспонент), $A_p(n)$ – мгновенная амплитуда p -ой синусоиды, $\varphi_p(n)$ – мгновенная фаза p -ой синусоиды, $r(n)$ – аperiodическая составляющая. Мгновенная частота $F_p(n)$, находящаяся в интервале $[0, \pi]$ (π

соответствует частоте Найквиста), является производной от мгновенной фазы. Предполагается, что амплитуда изменяется медленно, что означает ограничение частотной полосы каждой из составляющих.

Характеристический вектор состоит из значения частоты основного тона, спектральной огибающей и признака вокализованности текущего речевого фрагмента. Данный набор параметров, выделяется из речевого сигнала при помощи алгоритма, состоящего из следующих шагов:

- оценка мгновенной частоты основного тона при помощи устойчивого к ошибкам алгоритма слежения за мгновенной частотой основного тона IRAPT (Instantaneous Robust Algorithm for Pitch Tracking) [Azarov E., 2012];
- деформация временной оси сигнала для обеспечения стационарности частоты основного тона;
- оценка мгновенных гармонических параметров речевого сигнала с использованием ДПФ-модулированного банка фильтров – каждая гармоника основного тона вокализованной речи попадает в отдельный канал банка фильтров, где преобразуется в аналитический комплексный сигнал, из которого выделяется мгновенная амплитуда, фаза и частота;
- на основе анализа полученных значений мгновенной частоты различные области спектра классифицируются как периодические и аperiodические;
- гармоники, принадлежащие периодическим областям спектра синтезируются и вычитаются из исходного сигнала;
- остаток переводится в частотную область при помощи кратковременного преобразования Фурье;
- параметры синтезированных гармоник и СПМ остатка объединяются в одну общую спектральную огибающую и переводятся в логарифмическую шкалу;
- смежные спектральные огибающие анализируются для определения способа возбуждения всего анализируемого фрагмента сигнала.

Каждая спектральная огибающая представляется в виде вектора логарифмических значений энергии равномерно расположенных в шкале мелов. Для речевого сигнала с частотой дискретизации 44.1 кГц используется 100-мерный вектор. Размерность вектора определяет компромисс между качеством реконструкции сигнала и вычислительной сложностью. На основе практических экспериментов установлено, что выбранная размерность является достаточной для реконструкции натуральной речи.

2. Модель речевого сообщения с фонетической привязкой на основе нейронной сети

Модель использует нейронную сеть, построенную по принципу автоматического кодера. Автоматический кодер представляет собой многослойную нейронную сеть, которая преобразовывает многомерные данные в коды меньшей размерности и затем восстанавливает их в первоначальном виде [Hinton G.E., 2006].

Понижение размерности данных широко используется в задачах классификации, связи, распознавания и др. Достаточно простым и широко применяемым способом является анализ главных компонент, выделяющий в обучающей выборке направления с максимальной дисперсией и описывающий каждый элемент выборки через координаты по этим направлениям. Показано, что системы понижения размерности данных на основе нейронных сетей обладают гораздо более широкими возможностями, поскольку, в отличие от метода анализа главных компонент позволяют выполнять нелинейные преобразования. Обе сети, кодер и декодер можно обучить вместе, уменьшая разницу между исходными данными и их реконструкцией. Частные производные всех параметров легко вычисляются, используя правило дифференцирования сложной функции, для распространения производной ошибки сперва через сеть декодирования, а затем через сеть кодирования.

В контексте конверсии голоса полезным свойством автоматического кодера является способность самостоятельно классифицировать, упорядочивать и находить похожие данные без "учителя", т.е. каких-либо заранее заданных целевых меток в обучающей выборке. Показано, что при достижении хорошей реконструкции данных, близкие коды пониженной размерности соответствуют схожим входным данным. Это может быть использовано для построения производительных ассоциативных поисковых систем [Nair V., 2010].

В задаче конверсии голоса основной проблемой является поиск соответствия между параметрами исходного голоса и целевого. Проблема возникает из-за того, что параметры голоса выделяются на основе речевых фрагментов, соответствующим разным состояниям исходного и целевого дикторов. Каждый диктор имеет некоторое пространство возможных вариаций при произношении одних и тех же фонетических единиц, обусловленных многообразием речевых оттенков, выражением различных эмоций и интонаций. Таким образом одна и та же фонема, находясь в одном и том же фонетическом контексте может звучать по-разному. Учитывая это, очень сложно найти соответствие между состояниями разных дикторов.

Основная идея применения автоматического кодера в данной работе заключается в использовании фонетически мотивированных кодов

пониженной размерности, (имеющих фонетическую интерпретацию). Предлагается организовать процесс обучения нейронной сети таким образом, чтобы коды, соответствующие одной фонеме компактно располагались в одной определенной области пространства, причем границы, примыкающих к нему областей, обеспечивали переход к другим фонемам. Для того, чтобы обеспечить возможность непрерывного перехода из любой фонемы в любую размерность пространства равна числу используемых фонем. Расположено каждой фонемы в пространстве кодов фиксировано вдоль координат пространства в интервале от 0 до 1.

Использованная конфигурация искусственной нейронной сети показана на рисунке 1. Кодер выполняет функцию отображения

$$H = (w_4 RL(w_3 RL(w_2 RL(w_1 X + b_1) + b_2) + b_3) + b_4) \otimes M$$

где X – характеристический вектор речевого сигнала, H – вектор пониженной размерности, M – вектор фонетической маски, w_{1-4} и b_{1-4} – весовые коэффициенты и смещения соответствующих сигналов сети, \otimes обозначает поэлементное умножение. В сети используется кусочно-линейная функция активации $RL(x) = \max(0, x)$, поскольку показано, что она обеспечивает более эффективное внутреннее представление речевых данных по сравнению с логистической и позволяет ускорить процесс обучения [Zeiler M.D., 2013]. На выходе кодера формируются коды пониженной размерности, на которые накладываются ограничения для того, чтобы выполнить фонетическую привязку. Наложение ограничения выполняется путем перемножения сигнала H на фонемную маску, представляющую собой разреженную матрицу, и формируемую на основе фонетической разметки речевого корпуса.

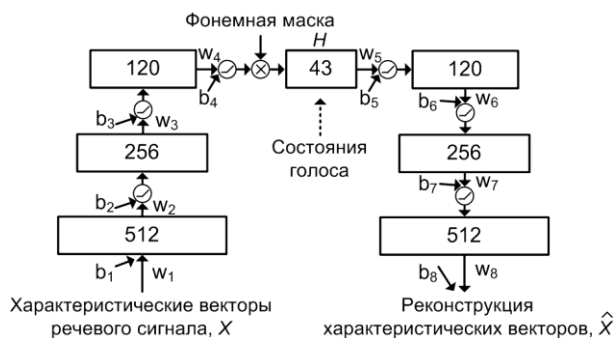


Рисунок 1 - Формирование состояний голоса на основе нейронной сети

Декодер выполняет реконструкцию кодов пониженной размерности в характеристические векторы \hat{X} . Соответствующая функция отображения имеет следующий вид:

$$\hat{X} = (w_8 RL(w_7 RL(w_6 RL(w_5 H + b_5) + b_6) + b_7) + b_8)$$

Использовалось следующее число нейронов в каждом скрытом слое нейронной сети: 512-256-120-

43-120-256-512. Обучение сети включает несколько этапов, которые кратко описаны ниже.

2.1. Предварительная сегментация обучающего речевого корпуса на фонемы

Для того чтобы при обучении нейронной сети было возможно осуществить фонетическое разграничение состояний голоса, необходимо установить соответствие каждого обучающего характеристического вектора определенной фонеме. С этой целью выполняется сегментация обучающего речевого корпуса на фонемы, используя фонемную транскрипцию. Задача определения границ фонем в речевом сигнале имеет классическое решение, основанное на использовании скрытой марковской модели [Rabiner L.R., 1993]. Для повышения точности анализа может применяться предварительная ручная (частичная либо полная) разметка.

2.2. Инициализация параметров сети и предварительное обучение

Поиск оптимальных коэффициентов многослойного кодера является сложной задачей, поскольку для того, чтобы алгоритм обратного распространения ошибки был эффективным, требуется хорошее начальное приближение. При использовании больших начальных коэффициентов процесс обучения обычно сходится к плохому локальному минимуму, использование малых коэффициентов приближает градиент в начальных слоях к нулю, что делает невозможным обучение сети с большим количеством слоев. Известен метод раздельного предварительного обучения слоев при помощи ограниченной машины Больцмана, успешно использованный в различных задачах машинного обучения [Hinton G.E., 2006]. В настоящей работе применяется схема предварительного обучения, основанная на частичном обнулении данных. Каждая пара матриц коэффициентов инициализируется случайными числами, и тренируется отдельно в виде нейронной сети с одним скрытым слоем. На вход сети подаются данные, часть из которых случайным образом обнуляется, на выходе сети восстанавливаются полные исходные данные. Причем обеспечивается равенство соответствующих матриц кодера и декодера $w_1 = w_8^T$, $w_2 = w_7^T$, $w_3 = w_6^T$, $w_4 = w_5^T$. Обучение выполняется при помощи алгоритма обратного распространения ошибки с накоплением градиента (momentum). При обучении матриц коэффициентов $w_4 = w_5^T$ частные производные по внутренним сигналам вычисляются с дополнительным слагаемым, обеспечивающим повышение активности в точках, обозначенных единицами в фонетической маске и понижение в точках, обозначенных нулями. Фонетическая маска формируется на основании предварительной сегментации речевого корпуса. Пространство кодов состояний голоса и маска

имеют размерность 43 (равную числу фонемам русского языка). Каждая фонема соответствует отдельной координате. Маска содержит единицы в координатах, соответствующих фонеме каждого вектора состояния, как показано на рисунке 2. Промежуточный переход между двумя соседними фонемами задается двумя единицами.

2.3. Обучение системы кодера/декодера

После предварительного определения коэффициентов каждого слоя выполняется подгонка параметров всей модели при помощи модификации алгоритма обратного распространения ошибки RPROP (Resilient back PROPagation) [Осовский С.]. При перемножении внутреннего сигнала на маску каждый отдельный вектор попадает либо в определенную плоскость пространства, соответствующую двум смежным фонемам, либо на координатную ось, соответствующую одной фонеме. В процессе подгонки параметров происходит уменьшение ошибки реконструкции и упорядочение характеристических векторов каждой фонемы вдоль осей пространства – рисунок 2.

		Номер вектора состояния														
		1	2	3	4	5	6	7	8	9	10	11	12	13		
Фонемы	а	0	0	1	1	1	1	1	0	0	0	1	1	1	...	1
	э	0	0	0	0	0	0	0	0	0	0	0	0	0	...	2
	и	0	0	0	0	0	0	0	0	0	0	0	0	0	...	3
	о	0	0	0	0	0	0	0	0	0	0	0	0	0	...	4
	⋮													⋮		
	б'	0	0	0	0	0	0	0	0	0	0	0	0	0	...	41
	м	1	1	1	1	0	0	1	1	1	1	1	0	0	...	42
	м'	0	0	0	0	0	0	0	0	0	0	0	0	0	...	43
		/ м		а		м		а/								

Рисунок 2 - Маска кодов пониженной размерности для осуществления фонетической привязки

Процедура предварительного обучения создает достаточно хорошее начальное приближение и характеристические векторы в начале процедуры уже определенным образом упорядочены. Однако, как показано на рисунке 11, после выполнения некоторого числа итераций происходит перестановка векторов, уменьшающая ошибку реконструкции и разницу между соседними векторами. В результате обучения формируется модель сообщения, которая включает модель голоса диктора и модель каждой отдельной фонемы и возможных переходов между ними, содержащихся в обучающей выборке.

3. Результаты экспериментов

Целью выполненных экспериментов является оценка практической применимости предложенной модели для решения задачи конверсии голоса с текстонезависимым обучением. Обученные модели голосов использовались для поиска соответствия между характеристическими векторами исходного и целевого дикторов. Модели обучались независимо

друг от друга. Конверсия речевого сигнала выполнялась с использованием ручной фонетической разметки, в которой выделялся характерный «центральный» характеристический вектор каждой фонемы. На основании разметки автоматически определялись «переходные» характеристические векторы, относящиеся к границам между фонемами. Соответствие между центральными и переходными характеристическими векторами исходного и целевого диктора устанавливалось следующим образом. На вход кодера исходного диктора подавался входной характеристический вектор и вычислялся вектор пониженной размерности, умноженный на фонемную маску. Из обучающей выборки целевого диктора извлекался характеристический вектор, наиболее близкий к полученному в пространстве кодов пониженной размерности. Все необходимые для синтеза характеристические векторы вычислялись путем интерполяции между конвертированными центральными и переходными векторами.

3.1. Речевая база и оценка качества конверсии

Для экспериментальной оценки качества конверсии использовалась речевая база на русском языке. База содержит широкополосную речь 6-и дикторов (3 мужчин и 3 женщины), записанную с частотой дискретизации 44,1 кГц. Для каждого из дикторов в базе содержится по 26 фраз для обучения и по 4 фразы для конверсии.

В ходе экспериментов выполнено сравнение двух методов: 1) изложенный выше метод конверсии с текстонезависимым обучением на основе автоматического кодера (далее обозначается как 'ТН') и метод с текстозависимым обучением на основе нейронной сети с кусочно-линейной функцией активации (далее обозначается как 'ТЗ') [Azarov E., 2013].

Для обоих методов использовались одинаковые алгоритмы параметризации и синтеза речевого сигнала, одинаковые обучающие последовательности исходного и целевого дикторов, одинаковые сигналы возбуждения и одинаковые целевые контуры основного тона. Результаты конверсии оценивались субъективно в терминах узнаваемости целевого диктора и натуральности звучания реконструированной речи с использованием средних значений оценок экспертов MOS (mean opinion score). В ходе эксперимента задействовано четверо слушателей, которые ставили оценки (по шкале от 1 до 5). Усреднение оценок выполнено отдельно по четырем группам в зависимости от направления конверсии мужчина-мужчина, мужчина-женщина, женщина-мужчина, женщина-женщина (обозначенных как "мм", "мж", "жм" и "жж" соответственно) для каждого из методов конверсии. Средние оценки показаны на рисунке 3.

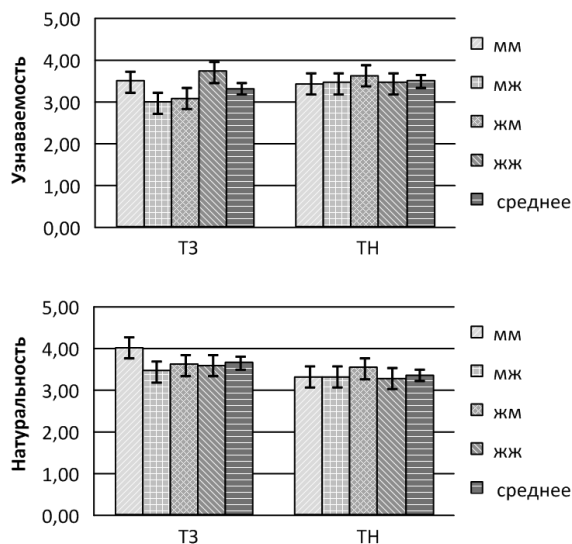


Рисунок 3 - Субъективная оценка узнаваемости и натуральности конвертированной речи. Средние значения оценок экспертов (доверительный интервал 95%).

На основании прослушивания и анализа оценок можно сделать вывод, что метод на основе автоматического кодера обеспечивает немного более высокую узнаваемость целевого диктора, однако несколько проигрывает по натуральности звучания. Повышение средней узнаваемости скорее всего обусловлено тем, что описанный способ, позволяет ослабить эффект усреднения спектральной огибающей, характерный для систем с текстозависимым обучением. Понижение натуральности обусловлено в первую очередь ошибками полуавтоматической сегментации речевого корпуса и тем, что использовалась простая модель сегментации, выделяющая только границы и центр каждой из фонем. Следует также отметить, что в методе T3 используется разделение параметров огибающей на высокочастотные и низкочастотные, а так же последовательность состояний диктора, генерируемых автоматически на основе текущих значений основного тона и признаков вокализованности. Таким образом, на вход системы конверсии T3 поступает намного больше характеристических признаков. В тоже время, необходимо учитывать, что в методе TH использована ручная фонемная разметка, которая значительно упрощает поиск соответствия между исходными и целевыми данными.

Заключение

В работе исследуется возможность создания модели речевого сообщения с фонетической привязкой на основе искусственной нейронной сети, построенной по принципу автоматического кодера. Приводятся результаты экспериментального применения данного подхода к решению задачи конверсии голоса с текстонезависимым обучением. Показано, что формирование унифицированных состояний в виде кодов пониженной размерности позволяет установить соответствие между

различными голосами и может использоваться в системах синтеза речи по тексту и конверсии голоса. Особенностью полученной модели является относительная инвариантность к характеру произношения, что достигается за счет привязки внутренних состояний к фонетическому содержанию, что может использоваться в различных системах обработки речи, таких как системы автоматического распознавания и кодирования.

Библиографический список

- [Hinton G.E., 2006] Hinton G.E., Salakhutdinov R.R. Reducing the Dimensionality of Data with Neural Networks, Science // Vol. 313 no. 5786 pp. 504-507, 28 July, 2006.
- [D'Alessandro C., 1995] D'Alessandro C., Yegnanarayana B., Darsinos V. Decomposition of speech signals into deterministic and stochastic components // ICASSP-95., vol.1, pp.760-763, 9-12 May 1995.
- [Kawaahra H., 2009] Kawaahra H., Nisimura R., Irino T., Morise M., Takahashi T., Banno B. Temporally variable multi-aspect auditory morphing enabling extrapolation without objective and perceptual breakdown // Proc. ICASSP, Taipei, Taiwan, April 2009.
- [Azarov E., 2014] Azarov E., Vashkevich M., Petrovsky A. Guslar: a framework for automated singing voice correction // The 39th International Conference on Acoustics, Speech and Signal Processing (ICASSP) 2014, Florence, Italy, May 2014. – P. 7969-7973.
- [Azarov E., 2012] Azarov E., Vashkevich M., Petrovsky A. Instantaneous pitch estimation based on RAPT framework // Proc. EUSIPCO'12, Bucharest, Romania, Aug. 2012, pp. 2787-2791.
- [Nair V., 2010] Nair V., Hinton G.E. Rectified linear units improve restricted Boltzmann machines // Proc. ICML, Haifa, Israel, June 2010.
- [Zeiler M.D., 2013] Zeiler M.D., Ranzato M., Monga R., Mao M., Yang K., Le Q.V., Nguyen P., Senior A., Vanhoucke V., Dean J., Hinton G. On Rectified Linear Units for Speech Processing // Proc. ICASSP, Vancouver, Canada, May 2013.
- [Rabiner L.R., 1993] Rabiner L.R., Juang B-H. Fundamentals of speech recognition // Pearson Education, 1993, P. 507.
- [Осовский С.] Осовский С. Нейронные сети для обработки информации // Москва: "Финансы и статистика", 2002, 344 с.
- [Azarov E., 2013] Azarov E., Vashkevich M., Likhachov D., Petrovsky A. Real-time Voice Conversion Using Artificial Neural Networks with Rectified Linear Units // Proc. INTERSPEECH Lyon, France, Aug. 2013, pp. 1032-1036.

SEPARATE MODELING OF SPEECH USING VOICE, PHONETICAL AND PROSODIC CHARACTERISTICS

Azarov E., Petrovsky A.

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

azarov@bsuir.by

palex@bsuir.by

The paper investigates possibility of speech modeling using separate voice, phonetical and prosodic components. The paper presents a practical way of building such model and some experimental results of applying the model to voice conversion. The model uses an artificial neural network organized as autoencoder that establishes correspondence between space of speech parameters and space of possible phonetic states, unified for any voice.



УДК 004.822:514

ПРИМЕНЕНИЕ МЕТОДОВ ИНТЕЛЛЕКТУАЛЬНОГО АНАЛИЗА ВРЕМЕННЫХ РЯДОВ ДЛЯ ЗАДАЧИ КЛАСТЕРИЗАЦИИ ПОЛЬЗОВАТЕЛЕЙ ПО ГОЛОСУ

Вагин В.Н., Ганишев В.А.

*Национальный исследовательский университет «МЭИ»,
г. Москва, Российская федерация*

vagin@appmat.ru

v.ganishev@gmail.com

В данной работе рассматривается применение методов интеллектуального анализа временных рядов для задачи кластеризации пользователей по голосу. В качестве модели пользователя используется набор мел-частотных кепстральных коэффициентов. Использование метода нейросетевого сжатия данных позволяет сократить размерность вектора признаков. Кластеризация выполняется с помощью самоорганизующихся карт Кохонена.

Ключевые слова: кластеризация, временные ряды, мел-частотные кепстральные коэффициенты, самоорганизующиеся карты

Введение

Кластеризация пользователей по голосу – автоматическое соотношение каждой голосовой записи из некоторого набора к отдельному, как правило, анонимному пользователю. Чаще всего процесс кластеризации проходит без непосредственного контроля. Этот процесс часто является составной частью задач распознавания пользователей по голосу и распознаванию речи.

При решении данной задачи ставится вопрос не “что было сказано?”, а “кто это сказал?”. Кластеризация пользователей по голосу находит применение при анализе теле- и радио-трансляций, записей конференций и телефонных переговоров. Создание отдельной модели каждого пользователя является весьма ресурсоемким, когда речь идет о массовых событиях и системах, содержащих тысячи записей сотен пользователей: системы в таком случае смогут работать лишь на предопределенном множестве пользователей, что лишает их гибкости. С другой стороны, создание отдельной модели для пользователя, редко использующего эти системы, является неоправданным с точки зрения экономии ресурсов.

В последние годы решение данной задачи является одним из приоритетных направлений таких областей исследований, как анализ сигналов, компьютерная безопасность и искусственный интеллект.

Для решения этой задачи часто применяются следующие модели:

- смеси гауссовских распределений (*GMM, Gaussian Mixture Model*) [Han et al., 2008];
- скрытые марковские модели (*HMM, Hidden Markov Model*) [Ajmera et al., 2003];
- гистограммные модели (*Histogram Model*) [Rodriguez Fuentes et al., 2004];
- алгоритм спектральной кластеризации Ына-Джордана-Вайса (*Ng-Jordan-Weiss spectral clustering algorithm*) [Ning et al., 2004];
- алгоритм байесовской адаптации [Faltlhauser et al., 2006]

Считается, что данные модели более подходят для описания поведения, характерного именно для голосового сигнала. Тем не менее, перспективным представляется применение инструментов интеллектуального анализа временных рядов.

В данной работе предлагается использовать другой подход, основанный на применении классического алгоритма кластеризации для характеристик, выделенных из сигнала и уникальных для каждого пользователя. В качестве таких характеристик используются мел-частотные кепстральные коэффициенты.

1. Модель пользователя

1.1. Причины использования мел-частотных кепстральных коэффициентов

Звуковой сигнал является одним из средств взаимодействия человека с окружающей средой и людей между собой. Голос зависит от многих физиологических параметров говорящего и является по своей сути индивидуальной характеристикой каждого человека. Тем не менее, голос не является постоянной характеристикой, он изменяется в течение жизни человека, на него также влияют состояние здоровья и эмоции.

Современные средства записи позволяют представить звуковой сигнал в виде временного ряда, показывающего изменение частоты во времени. Спектр сигнала, его представление в частотном пространстве является более информативным для анализа, чем сигнал сам по себе. Для вычисления спектра часто используется быстрое преобразование Фурье, алгоритм которого является достаточно простым для реализации и имеет сложность $O(N \log_2 N)$, меньшую, чем сложность классического алгоритма дискретного преобразования Фурье $O(N^2)$ [Cooley et al., 1965]. Люди реагируют на частотные изменения, поэтому при решении задач, связанных с анализом человеческого голоса, часто используют «кепстр» (*cepstrum*) [Bogert et al., 1963] – результат применения преобразования Фурье к спектру сигнала.

Также в процессе эволюции звуки в более низком частотном диапазоне содержали в себе больше полезной информации, чем находящиеся в более высоком частотном диапазоне. С учетом этих особенностей человеческого слуха были разработаны мел-частотные кепстральные коэффициенты («мел» является сокращением английского слова «melody» - мелодия) [Vyas et al., 2013]. С помощью данных коэффициентов более тщательно анализируется информация, получаемая из низкочастотного диапазона, а влияние высокочастотных составляющих, обычно содержащих посторонний шум, на результат распознавания уменьшается.

Вся голосовая запись разделяется на небольшие интервалы, длительностью $\sim 10-30$ мс (время квазистационарности сигнала), называемые фреймами. Для каждого фрейма отдельно рассчитывается набор мел-частотных кепстральных коэффициентов, который в дальнейшем будет использоваться для кластеризации.

1.2. Вычисление мел-частотных кепстральных коэффициентов

Алгоритм вычисления мел-частотных кепстральных коэффициентов можно разбить на следующие этапы [Molau et al., 2001]:

а. разбиение сигнала на фреймы;

б. применение весовой функции (окна) к каждому фрейму;

в. применение преобразования Фурье;

г. использование мел-частотного фильтра;

д. вычисление кепстра.

а. Разбиение сигнала на фреймы

Звуковой сигнал в общем случае не является стационарным, т.е. их амплитуда и спектр изменяются во времени, что приводит к невозможности применения многих техник анализа. Но отдельно взятый короткий интервал порядка 10-30мс можно считать стационарным. Часто применяют следующую методику деления сигнала на фреймы: сигнал разделяется на интервалы длиной N мс следующим образом: начало первого фрейма совпадает с началом записи, второй фрейм начинается через M мс интервалов ($M < N$), соответственно он на $N-M$ мс перекрывает первый фрейм. На Рис. 1 показан случай для $N = 20$ мс и $M = 16$ мс.

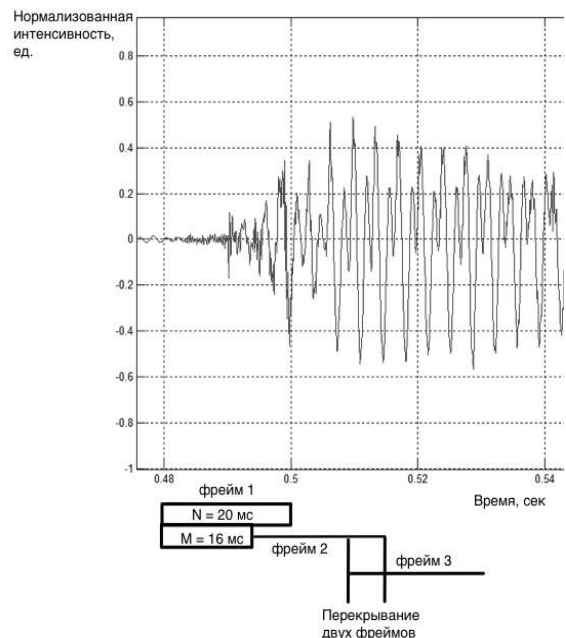


Рисунок 1 – Пример разбиения на фреймы

Несмотря на стационарность, такое представление сигнала не позволяет использовать преобразование Фурье. Если частоты гармоник (частотных составляющих) сигнала не совпадают с базисными частотами преобразования Фурье, то в спектре будут возникать «лишние» гармоники, которые будут лишь «зашумлять» полученное представление. Данный эффект носит название «размытие спектра» или «спектральная утечка».

б. Применение весовой функции (окна)

Одним из возможных вариантов решения возникшей проблемы является применение к сигналу весовой функции специального вида:

$$\omega(n), 0 \leq n \leq N - 1 \quad (1)$$

Результат применения весовой функции к каждому фрейму выглядит следующим образом (Рис. 2а):

$$y(n) = x(n) \cdot \omega(n), 0 \leq n \leq N - 1 \quad (2)$$

где $x(n)$ – значение временного ряда в точке n , а $y(n)$ – взвешенное значение временного ряда в точке n .

Наиболее предпочтительным является применение «мягких» весовых функций, которые сводят значения на границах фрейма к нулю. Эта операция называется «сглаживанием». Наиболее часто используемой является весовая функция Хэмминга, которую можно представить следующей формулой [Bhatnagar et al., 2012]:

$$\omega(n) = 0.53836 - 0.46165 \cdot \cos\left(\frac{2\pi n}{N-1}\right) \quad (3)$$

Преобразование Фурье примененное к «взвешенному» временному ряду дает более четкий спектр (Рис. 2б).

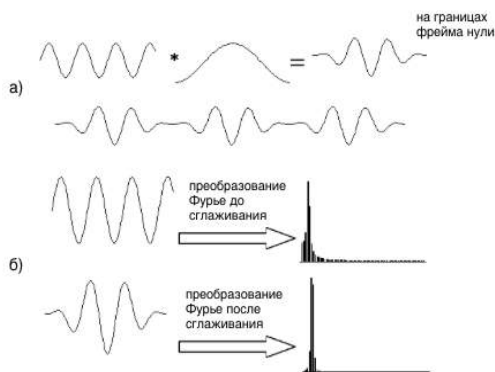


Рисунок 2 – Сглаживание сигнала:
а) Применение весовой функции к фрейму
б) Применение преобразование Фурье

в. Преобразование Фурье

На следующем этапе необходимо применить преобразование Фурье, которое переведет сигнал из временного пространства в частотное. На практике чаще всего применяется быстрое преобразование Фурье, который имеет следующий вид [Cooley et al, 1965]:

$$Y_n = \sum_{k=0}^{N-1} y_k \cdot e^{-\frac{2\pi jkn}{N}}, 0 \leq n \leq N - 1, j = \sqrt{-1} \quad (4)$$

где y_k – взвешенное значение временного ряда в точке k ,

Y_n – комплексная амплитуда n -той гармоники сигнала, представляемого временным рядом.

Результатом данного этапа является спектр сигнала.

г. Использование мел-частотного фильтра

На данном этапе к спектру сигнала применяется специального вида фильтр. Каждому значению частоты, полученному на предыдущем шаге ставится в соответствие значение на мел-частотной шкале. Значения данной шкалы для частот ниже

1000 Гц точно соответствуют спектру сигнала, полученному при преобразовании Фурье, частоты выше 1000 Гц – логарифмируются. В результате получается модифицированный энергетический спектр сигнала $mel(f)$ для каждой гармоники частоты f , для вычисления которого используется следующая приближенная формула [Molau et al., 2001]:

$$mel(f) = 2595 \cdot \lg\left(1 + \frac{f}{700}\right) \quad (5)$$

К данному спектру применяется фильтр специального вида, ставящий в соответствие каждой частоте определенный набор мел-коэффициентов $\tilde{S}_k, 1, \dots, K$, где K – количество мел-коэффициентов, на практике часто выбирают значение от 12 до 24.

д. Использование мел-частотного фильтра

На предыдущем шаге алгоритма полученные коэффициенты \tilde{S}_k необходимо перевести в мел-кепстальное пространство. Для этого удобно использовать дискретное косинусоидальное преобразование, которое описывается следующей формулой [Chen et al., 1977]:

$$\tilde{C}_n = \sum_{k=1}^K \lg(\tilde{S}_k) \cdot \cos\left[n\left(k - \frac{1}{2}\right)\frac{\pi}{K}\right], 0 \leq n \leq K \quad (6)$$

где \tilde{C}_n – полученные мел-частотные кедральные коэффициенты.

1.3. Акустические векторы

Данный алгоритм применяется к каждому фрейму, в результате чего последнему соответствует набор мел-коэффициентов, который используется в большинстве работ как модель пользователя для кластеризации и называется акустическим вектором.

Но изменение мел-коэффициентов также содержит определенную информацию о пользователе. Основным отличием данной работы от предыдущих является расширение акустического вектора путем учета динамики изменения мел-коэффициентов δ_i , которая выражается разностью мел-частотных кедральных коэффициентов, данного фрейма и предыдущего:

$$\delta_i(\tilde{C}_k[i]) = \tilde{C}_k[i-1] - \tilde{C}_k[i] \quad (7)$$

Данный подход позволяет анализировать изменения мел-частотных кедральных коэффициентов, что также несет в себе информацию, идентифицирующую пользователя.

При данном подходе первый фрейм не может использоваться для кластеризации, так как изменение мел-частотных кедральных коэффициентов будет нулевым. А L – количество элементов акустического вектора x – увеличивается вдвое: $L = |x| = |[\tilde{C}_1, \dots, \tilde{C}_K, \delta(\tilde{C}_1), \dots, \delta(\tilde{C}_K)]| = 2 \cdot K$.

2. Применение метода нейросетевого сжатия для акустических векторов

Большая размерность акустического вектора является проблемой, которая снижает скорость кластеризации. Для сокращения размерности акустического вектора предлагается использовать метод нейросетевого сжатия данных.

Данный метод реализуется с помощью трехслойной нейронной сети следующего вида [Tishby et al., 1999]:

- входной и выходной слои сети идентичны и соответствуют набору элементов акустического вектора;
- скрытый слой содержит меньшее количество нейронов.

После обучения сети коэффициенты скрытого слоя будут представлять собой новый акустический вектор меньшей размерности. В общем случае данный алгоритм может применяться несколько раз для дальнейшего сокращения размерности до тех пор, пока среднеквадратичная ошибка E не превышает заранее заданный порог:

$$E = \sum_{k=1}^n E(k) = \frac{1}{2} \sum_{k=1}^L (y_k - x_k)^2 < \varepsilon \quad (8)$$

где $E(k)$ – среднеквадратичная ошибка для k -го акустического вектора записи;

L – количество элементов акустического вектора;

y_k – полученное значение акустического вектора на выходе нейронной сети;

x_k – значение акустического вектора на входе нейронной сети;

ε – некоторый заранее определенный порог.

3. Самоорганизующиеся карты

Задача определения пользователя по акустическому вектору относится к классу задач распознавания по шаблону. Использование нейронных сетей Кохонена для решения задачи кластеризации пользователей по голосу было выбрано в силу точности кластеризации и ее скорости [Mogi et al., 2001].

В данной работе используются самоорганизующаяся карта Кохонена [Kohonen, 1995]. Она представляет собой нейронную сеть с двумя слоями, причем нейроны первого (распределительного) слоя соединены со всеми нейронами второго (выходного) слоя, которые расположены в виде двумерной решетки. Количество нейронов в выходном слое определяет максимальное количество групп, на которые система может разделить входные данные.

Для обучения сети Кохонена используется соревновательный метод [Kohonen, 1995]. На каждом шаге обучения из исходного набора данных случайно выбирается один вектор. Затем производится поиск нейрона выходного слоя, для

которого расстояние между его вектором весов и входным вектором - минимально.

Алгоритм обучения сети Кохонена выглядит следующим образом [Головки, 2001]:

1. Инициализация малыми случайными значениями на отрезке $[-1, 1]$ матрицы весов сети W размерности $L \times T$, где T – количество записей, которые необходимо кластеризовать;
2. организация акустических векторов в очередь в случайном порядке. Все вектора помечены как необработанные;
3. выбор первого необработанного элемента x из очереди;
4. для каждого выхода сети j вычисляются расстояния d_j между его вектором весов w_j и входным акустическим вектором. В данной работе используется квадрат евклидова расстояния:

$$d_j = \rho(w_j, x), \quad (9)$$

где

$$\rho(w_j, x) = \sum_{i=1}^n (w_{ji} - x_i)^2 \quad (10)$$

5. поиск выходного нейрона j_m с минимальным расстоянием d_{j_m} :

$$j_m = \arg \min (d_j) \quad (11)$$

6. вычисление изменения весов $\Delta W = \{\Delta w_j\}$ для всех нейронов j выходного слоя:

$$\Delta w_j = (w_j - x) \cdot h(u, c, t) \cdot \eta \quad (12)$$

где η – коэффициент скорости обучения;

c – номер нейрона победителя j_m в двумерной решетке второго слоя;

j – номер нейрона в двумерной решетке второго слоя;

w_j – вектор весовых коэффициентов связи входного слоя и нейрона с номером j ;

x – акустический вектор на входе сети;

$h(u, c, t)$ – функция окрестности. В данной работе использована функция Гаусса:

$$h(u, c, t) = e^{-\frac{\rho(c, u)}{\sigma(t)}} \quad (13)$$

где t – параметр времени;

σ – радиус окрестности h :

$$\sigma(t) = \frac{1}{e^{t-2}} \quad (14)$$

7. корректировка матрицы весов W нейронной сети:

$$W := W - \Delta W \quad (15)$$

8. элемент x входной очереди помечается как обработанный;

9. если в очереди имеются акустические векторы, помеченные как необработанные, то переход к п.3;

10. если критерий останова обучения не достигнут, то переход к п.2. В данной работе в

качестве критерия останова используется проверка стабилизации выходов сети: когда акустические векторы на последующих этапах обучения перестают переходить между кластерными элементами. Математически это выражается в том, что определитель матрицы изменений весов меньше некоторого порогового значения.

11. окончание алгоритма.

После применения данного алгоритма одному пользователю будет соответствовать несколько кластеров. Считаем, что голосовой сигнал принадлежит пользователю в том случае, если более половины фреймов были ассоциированы с этим пользователем. На Рис.3 показан результат кластеризации для двух пользователей. Акустические вектора первого пользователя обозначены кругом, второго – треугольником. Сверху представлены акустические вектора одной записи. На изображении видно, что они ассоциированы с разными кластерами.

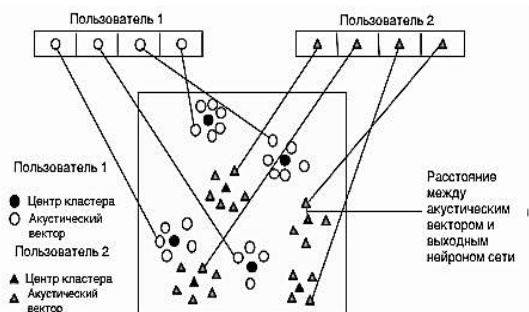


Рисунок 3 – Пример результата кластеризации

4. Практическая реализация данного метода

Для вычисления мел-частотных кепстральных коэффициентов используются средства свободно распространяемого фреймворка Sphinx 4, разработанной в университете Карнеги-Меллон [Walker et al., 2004]. Данный комплекс реализует множество функций, необходимых для распознавания пользователей по голосу, и обладает простым интерфейсом.

Как уже было сказано ранее, большая размерность акустического вектора является проблемой, поэтому для ее решения был предложен метод нейросетевого сжатия.

Программный комплекс имеет следующую структуру:

- Блок преобразования входного голосового сигнала. На данном этапе входной голосовой сигнал в формате .wav приводится к удобной для обработки форме представления в виде временного ряда.
- Блок обработки сигнала. На данном этапе на основе сигнала вычисляются мел-частотные кепстральные коэффициенты.
- Блок предобработки акустического вектора. В данном блоке реализуется сжатие акустического вектора до меньшего размера.

- Блок кластеризации. Здесь непосредственно применяется алгоритм кластеризации, описанный выше.

- Блок принятия решения. На данном этапе принимается решение о принадлежности записи определенному шаблону пользователя.

Система реализована в виде библиотеки C++, так как данный язык обладает кроссплатформенностью и высокой производительностью вычислений.

В настоящее время ведется исследование возможности усовершенствования алгоритма кластеризации для автоматического переобучения при добавлении голосовых записей новых пользователей.

Заключение

Данная работа рассматривает кластеризацию пользователей по голосу. В качестве характеристик пользователя предлагается использовать расширенный акустический вектор каждого фрейма голосовой записи, состоящий из мел-частотных кепстральных коэффициентов, а их изменения относительно прошлого фрейма.

Для сокращения размерности акустического вектора в данной работе предложен метод нейросетевого сжатия данных, который позволяет сократить размерность исходных данных для задачи кластеризации.

В качестве алгоритма кластеризации используются самоорганизующиеся карты Кохонена, так как использование нейронных сетей для кластеризации позволяет учитывать неочевидные закономерности в голосовых характеристиках, такие как характер изменений частоты голоса и т.д.

Предложенный метод учитывает дополнительные особенности голосовых характеристик каждого пользователя, такие как скорость изменения частоты голоса.

Библиографический список

[Ajmera et al., 2003] Ajmera J. [et al.] A Robust Speaker Clustering Algorithm // IEEE Workshop on Automatic Speech Recognition and Understanding, 2003, 2003, С. 411-416

[Bhatnagar et al., 2012] Bhatnagar A.C. [et al.] Analysis of Hamming window using advance peak windowing method // International Journal of Scientific Research Engineering & Technology (IJSRET) Vol.1 Issue 4, 2012, С. 15-20

[Bogert et al., 1963] Bogert B.P. [et al.] The Quefrency Analysis of Time Series for Echoes: Cepstrum, Pseudo Autocovariance, Cross-Cepstrum and Saphe Cracking // Proceedings of the Symposium on Time Series Analysis (M. Rosenblatt, Ed) Chapter 15, New York: Wiley, 1963, С. 209-243

[Chen et al., 1977] Chen W.-H. [et al.] A Fast Computational Algorithm for the Discrete Cosine Transform // IEEE Transactions of Communications, Vol.Com-25, No.9, 1977, С.1004-1009

[Cooley et al., 1965] Cooley J.W. [et al.] An Algorithm for the Machine Calculation of Complex Fourier Series // Mathematics of Computation, 1965, С. 297-301

[Falthausen et al., 2001] Falthausen R. [et al.] Robust Speaker Clustering in Eigenspace // IEEE Workshop on Automatic Speech Recognition and Understanding, 2001, С. 57-60

[Han et al., 2008] Han K.J. [et al.] Agglomerative Hierarchical Speaker Clustering using Incremental Gaussian Mixture Cluster Modeling // Proceedings of InterSpeech, 2008, C. 20-23

[Kohonen, 1995] Kohonen T. Self-Organizing Maps // Springer, 1995

[Linde et al., 1980] Linde, Y. [et al.] An Algorithm for Vector Quantizer Design // IEEE Transactions on Communications 28, 1980, C. 84-95

[Molau et al., 2001] Molau S. [et al.] Computing mel-frequency cepstral coefficients on the power spectrum // IEEE International Conference on Acoustics, Speech, and Signal Processing Vol.1, 2001, C.73-76

[Mori et al., 2001] Mori K. [et al.] Speaker change detection and speaker clustering using VQ distortion for broadcast news speech recognition // Acoustics, Speech, and Signal Processing, 2001. Proceedings. (ICASSP '01). 2001 IEEE International Conference on (Volume:1), 2001, C. 413-416

[Ning et al., 2006] Ning H. [et al.] A Spectral Clustering Approach to Speaker Diarization // Proc. ICSLP, 2006, C.

[Tishby et al., 1999] Tishby N., Pereira F., and Bialek W. The Information Bottleneck Method // The 37th annual Allerton Conference on Communication, Control, and Computing, 1999, с.368-379

[Rodriguez Fuentes, 2004] Rodriguez Fuentes L.J. [et al.] A Speaker Clustering Algorithm for Fast Speaker Adaptation in Continuous Speech Recognition // Text, Speech and Dialogue: Lecture Notes in Computer Science Volume 3206, 2004, C. 433-440

[Vyas et al., 2013] Vyas G. [et al.] Speaker Recognition System Based on MFCC and DCT // International Journal of Engineering and Advanced Technology(IJEAT) Vol. 2, Issue 5, 2013

[Walker et al., 1975] Walker W. [et al.] Sphinx-4: A flexible open source framework for speech recognition // Technical Report , 2004

[Вагин и др., 2008] Вагин В.Н. [и др.] Достоверный и правдоподобный вывод в интеллектуальных системах, 2-ое издание, исправленное и дополненное // – М. : ФИЗМАТЛИТ, пол ред. Вагина В.Н. и Поспелова Д.А., 2008, 712 с..

[Головко, 2001] Головко В.А. Нейронные сети: обучение, организация, применение // М. : ИПРЖР, 2001

APPLICATION OF TIME SERIES ANALYSIS FOR SPEAKER CLUSTERING

Vagin V.N., Ganishev V.A.
*National Research University «MPEI»,
Moscow, Russia*

vagin@appmat.ru
v.ganishev@gmail.com

The purpose of this paper is the introduction of time series analysis methods to the problem of speaker clustering. User's model used for clustering is based on the mel-frequency cepstral coefficients. We consider the use of methods of neuro-network data compression to reduce the dimensionality of the feature vector. Clustering is performed using self-organizing Kohonen maps.

Introduction

Speaker clustering is an automatic classification of voice recordings on some patterns of users, often without direct supervision. This process is often an integral part of the user recognition and speech recognition tasks.

By solving this problem the main question is not "What was said?" but "Who said that?". Speaker clustering is used in the analysis of television and radio broadcasts, recordings of telephone conversations and

conferences. Creation of separate model for each user is very resource-intensive in context of mass events and systems and makes the system over-fitted and not flexible.

Main Part

In this paper it is proposed to use speaker model based on mel-frequency cepstral coefficients (MFCC). It views the input voice signal in terms of model, that is close to perception of the human ear. It provides more detailed analysis of low-frequency component of the recording that contains more information needed for clustering.

The main contrast of this article from a huge variety of previous works is the usage not only the computed MFCC as features for clustering, but also the dynamic difference between MFCC of the considered frame and the previous.

The expansion of the acoustic vector follows to complicating and deceleration at the clustering phase. For solving this problem, the use of bottleneck method for feature compression is proposed.

The concept of self-organizing Kohonen maps is used for clustering. This method provides the clear and fast algorithm that is suitable for speaker clustering, because each user will have several clusters. That helps to overview different phonemes said by user in different situations.

Conclusion

This paper considers the problem of speaker clustering. As the features for clustering it is proposed to use advanced acoustic vector for each frame of a voice recording, consisting of mel-frequency cepstral coefficients and their dynamic changes over the previous frame.

To reduce the dimension of the acoustic vectors the bottleneck method for data compression in terms of a neural net is used.

The self-organizing Kohonen map is applied as a clustering algorithm.



OSTIS-2015

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822:514

ГОЛОСОВАЯ ИДЕНТИФИКАЦИЯ ПОЛЬЗОВАТЕЛЯ В СИСТЕМАХ КОНТРОЛЯ ДОСТУПА

Меньшаков П.А., Мурашко И.А.

*Гомельский государственный технический университет имени П.О. Сухого
Гомель, Республика Беларусь*

pmenshakov@gmail.com

iamurashko@tut.by

На данный момент основным средством контроля доступа являются пропускные пункты, оборудованные различными средствами контроля доступа. Но большинство из средств контроля доступа имеют высокую цену. Причем большая часть расходов приходится на выделение персонального средства идентификации каждому пользователю. Решением данной проблемы может стать голосовая идентификация. Использование биометрии позволяет отказаться от чипов и карт доступа, исключить потерю средства идентификации и его кражу. А использование голоса позволит отказаться от дорогостоящего оборудования для считывания данных.

Ключевые слова: голосовая идентификация; биометрия; контроль доступа.

Введение

В настоящее время, голосовая идентификация, как и биометрия в целом, уже получила широкое распространение. Самый простой пример – сканеры отпечатков пальцев, установленные почти на каждом ноутбуке.

Биометрия предполагает систему распознавания людей по одной или более физических или поведенческих черт. В области информационных технологий биометрические данные используются в качестве формы управления идентификаторами доступа и контроля доступа. Также биометрический анализ используется для выявления людей, которые находятся под наблюдением [1]. Так же биометрия предусматривает и поведенческий анализ объекта. К ним относятся ходьба, жесты и т.п.

Процесс авторизации, при использовании биометрии, довольно прост. При помощи устройства, предназначенного для получения той или иной характеристики, сканируются текущие данные идентифицируемого и сравниваются с полученными ранее данными. Биометрические системы имеют ряд серьезных преимуществ: биометрия использует свойства человеческого тела и его поведение, что делает эти данные уникальными (для того, чтобы, при помощи своего собственного отпечатка пальцев, предоставить чужой отпечаток пальца или сделать радужную оболочку своего глаза похожей на чью-то другую

требуются довольно редкие и сложные технологии); в отличие от бумажных идентификаторов (паспорт, водительские права, удостоверение личности), от пароля или персонального идентификационного номера (ПИН), биометрические характеристики не могут быть подвергнуты краже, не могут быть утеряны или забыты. Довольно длительное время отпечатки пальцев используются для идентификации преступников и предотвращения воровства или мошенничества. Некоторые люди умеют имитировать голоса, но, это требует особых навыков, которые не часто повстречаешь в обыденной жизни [2].

1. Принцип голосовой идентификации

Сам процесс голосовой идентификации не требователен к ресурсам, и состоит из двух этапов. Сперва, необходимо получить голосовой отпечаток пользователя и преобразовать к виду, в котором его можно будет сравнить с другими. Вторым шагом является сравнение голосовых отпечатков при помощи обученной нейронной сети. Для реализации процесса преобразования необходимо произвести определенный порядок действий.

При помощи микрофона получается запись голоса идентифицируемого и отправляется на ЭВМ. Наиболее оптимальным является получение WAV файла, в виду простоты работы с ним.

Полученную запись голоса необходимо разделить на кадры. Разделение на кадры

представлено на рисунке 1. Данное действие необходимо для более простой работы с записанной звуковой дорожкой.

Далее все вычисления будут производиться с каждым кадром в отдельности.

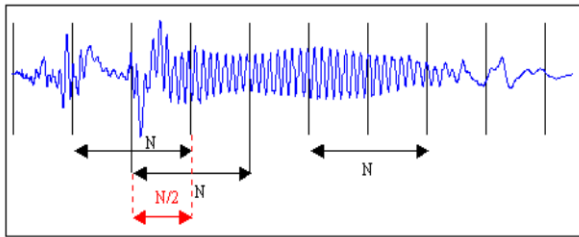


Рисунок 1 – График звуковой волны

Следующим этапом является устранение нежелательных эффектов и шумов. Это необходимо для того, чтобы записи, полученные в разное время соответствовали друг другу независимо от сторонних факторов. Существует множество способов, при помощи которых можно уменьшить шумовые эффекты. Мною использовалось умножение каждого кадра на особую весовую функцию "Окно Хемминга":

$$\omega(n) = 0.53836 - 0.46164 * \cos\left(\frac{2\pi n}{N-1}\right). \quad (1)$$

где n – порядковый номер элемента в кадре, для которого вычисляется новое значение амплитуды,

N – длина кадра (количество значений сигнала, измеренных за период).

Полученные кадры преобразуются в их частотную характеристику при помощи прогонки через "Быстрое Преобразование Фурье":

$$X_k = \sum_{i=0}^{N-1} x_n e^{-\frac{2\pi i}{N} kn}. \quad (2)$$

где N – длина кадра (количество значений сигнала, измеренных за период),

x_n – амплитуда n -го сигнала,

X_k – N -комплексных амплитуд синусоидальных сигналов, слагающих исходный сигнал.

На сегодняшний день наиболее успешными являются системы распознавания голоса, использующие знания об устройстве слухового аппарата. Они базируются на том, что ухо интерпретирует звуки не линейно, а в логарифмическом масштабе. В виду данных особенностей необходимо привести частотную характеристику каждого кадра к «мелам». Зависимость представлена на рисунке 2.

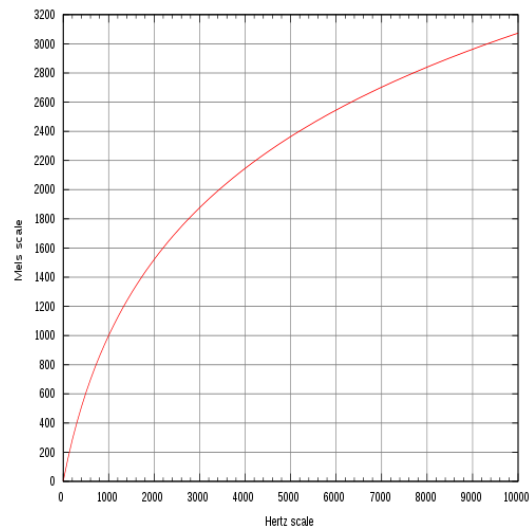


Рисунок 2 – График зависимости высоты звука (в мелах) от его частоты

Для перехода к «мел» характеристике используется следующая зависимость:

$$m = 1127 \log_e \left(1 + \frac{f}{700}\right). \quad (3)$$

где m – частота в мелах,

f – частота в герцах.

Это последнее действие, необходимое для последующего преобразование в вектор характеристики, который, впоследствии, сравнивается с базой голосовых записей. Вектор будет состоять из мел-кепстральных коэффициентов, получить которые можно по следующей формуле:

$$c_n = \sum_{k=1}^K (\log S_k) \left[n \left(k - \frac{1}{2} \right) \frac{\pi}{K} \right]. \quad (4)$$

где c_n – мел-кепстральный коэффициент под номером n ,

S_k – амплитуда k -го значения в кадре в мелах,

K – наперед заданное количество мел-кепстральных коэффициентов $n \in [1, K]$.

Полученный вектор характеристик добавляется в базу данных, для последующего сравнения с ним.

Однако более оптимальным вариантом является использование нескольких записей одного и того же голоса. Заранее определенное количество образцов голоса можно использовать для обучения нейронной сети.

В работе использовалось обучение без учителя, так как оно является намного более правдоподобной моделью обучения в биологической системе. Развитая Кохоненом и многими другими, она не нуждается в целевом векторе для выходов и, следовательно, не требует сравнения с предопределенными идеальными ответами, а обучающее множество состоит лишь из входных векторов. Обучающий алгоритм подстраивает веса сети так, чтобы получались согласованные

выходные векторы, т.е. чтобы предъявление достаточно близких входных векторов давало одинаковые выходы. Процесс обучения, следовательно, выделяет статистические свойства обучающего множества и группирует сходные векторы в классы. Предъявление на вход вектора из данного класса даст определенный выходной вектор [3]. Схематически данная сеть изображена на рисунке 3.

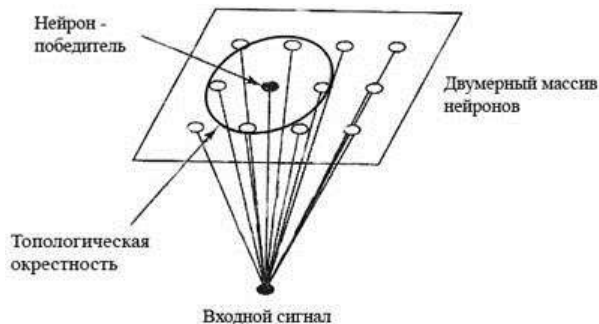


Рисунок 3 – Схема сети Кохонена

Распространение сигнала в такой сети происходит следующим образом: входной вектор нормируется на 1.0 и подается на вход, который распределяет его дальше через матрицу весов W . Каждый нейрон в слое Кохонена вычисляет сумму на своем входе и в зависимости от состояния окружающих нейронов этого слоя становится активным или неактивным (1.0 и 0.0). Нейроны этого слоя функционируют по принципу конкуренции, т.е. в результате определенного количества итераций активным остается один нейрон или небольшая группа. Этот механизм называется латеральным. Так как отработка этого механизма требует значительных вычислительных ресурсов, в моей модели он заменен нахождением нейрона с максимальной активностью и присвоением ему активности 1.0, а всем остальным нейронам 0.0. Таким образом, срабатывает нейрон, для которого вектор входа ближе всего к вектору весов связей.

Если сеть находится в режиме обучения, то для выигравшего нейрона происходит коррекция весов матрицы связи по формуле:

$$w_n = w_n + \alpha(x - w_n). \quad (5)$$

где w_n – новое значение веса;

w_m – старое значение;

α – скорость обучения;

x – величина входа.

Геометрически это правило иллюстрирует рисунок 4.

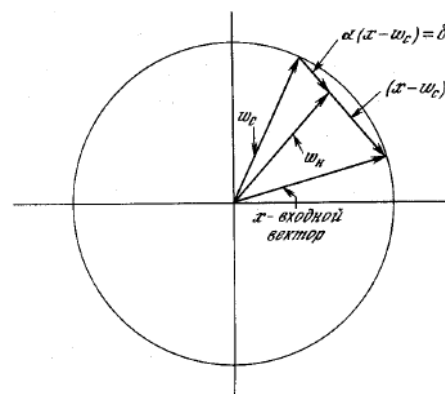


Рисунок 3 – Коррекция весов нейрона Кохонена

Так как входной вектор x нормирован, т.е. расположен на гиперсфере единичного радиуса в пространстве весов, то при коррекции весов по этому правилу происходит поворот вектора весов в сторону входного сигнала. Постепенное уменьшение скорости поворота позволяет произвести статистическое усреднение входных векторов, на которые реагирует данный нейрон.

Однако имеется несколько проблем. Первая – выбор начальных значений весов. Так как в конце обучения вектора весов будут располагаться на единичной окружности, то в начале их также желательно нормировать на 1.0. В моей модели вектора весов выбираются случайным образом на окружности единичного радиуса.

Вторая – если весовой вектор окажется далеко от области входных сигналов, он никогда не даст наилучшего соответствия, всегда будет иметь нулевой выход, следовательно, не будет корректироваться и окажется бесполезным. Оставшихся же нейронов может не хватить для разделения входного пространства сигналов на классы. Для решения этой проблемы предлагается много алгоритмов, в работе применяется правило «работать»: если какой-либо нейрон долго не находится в активном состоянии, он повышает веса связей до тех пор, пока не станет активным и не начнет подвергаться обучению. Этот метод позволяет также решить проблему тонкой классификации: если образуется группа входных сигналов, расположенных близко друг к другу, с этой группой ассоциируется и большое число нейронов Кохонена, которые разбивают её на классы.

Заключение

Итогом данного исследования стало модульное приложение, осуществляющее голосовую идентификацию пользователя, использующее модернизированный алгоритм вычисления нейронов в слое Кохонена. Программа состоит из трех основных частей. Первая выполняет добавление пользователей, вторая выполняет

идентификацию и третья хранение голосовых записей.

Как показало исследование, полученный алгоритм позволяет значительно ускорить работу программы голосовой идентификации. Данная модернизация позволяет использовать программу на предприятиях с большим потоком пользователей.

Так же программный комплекс очень гибок и имеет большое пространство для дальнейшего усовершенствования и добавления новых функций, что делает его не только выгодным программным продуктом, но и перспективным проектом для развития и получения прибыли.

Библиографический список

[Bosi M., 2003] Introduction to digital audio coding and standards / M. Bosi, R.E. Goldberg - Springer Science+Business, Media USA. - 2003. - 434 p.

[You Y., 2010] AudioCoding: Theory and Applications / Y. You - NY: Springer, 2010 - 349 p.

[Загуменнов, А. П., 1999] Компьютерная обработка звука./ А. П. Загуменнов - М. : ДМК, 1999. - 384 с

VOICE USER IDENTIFICATION IN ACCESS CONTROL SYSTEMS

Menshakov P.A., Murashko I.A.

*Department of Information Technology
Gomel State Technical University*

*named by P.O. Suhoi
Gomel, Belarus*

pmenshakov@gmail.com

iamurashko@tut.by

At the moment, the primary means of access control checkpoints are equipped with a variety of means of access control. But most of the access controls are expensive. And most of the costs fall on the allocation of personal identification means for each user. The solution to this problem is to voice recognition. The use of biometrics eliminates the chips and access cards, identification means to eliminate loss and theft. And the use of the voice will eliminate the expensive equipment to read the data.

Introduction

Currently, voice recognition, biometrics as a whole, already widespread. The simplest example - fingerprint scanners installed on almost every laptop. Biometrics recognition system involves people on one or more physical or behavioral traits. In the field of information technology, biometric data is used as a form of identity management and access control access. Also, biometric analysis is used to identify people who are under the supervision of. Just provide biometrics and behavioral analysis of the object. These include walking, gestures, etc. Authorization process, the use of biometrics, is quite simple. Using an apparatus for varying the characteristics of an identified current data scanned and compared with previous data. Biometric systems have a

number of important advantages. Biometrics uses the properties of the human body and its behavior, which make them unique. Unlike paper identifiers from a password or personal identification number (PIN), biometric characteristics can not be subjected to theft, can not be lost or forgotten. Quite a long time fingerprints are used to identify criminals and prevent theft or fraud. Some people are able to mimic the voice, but it requires special skills that are not often met in everyday life.

Main Part

To implement voice recognition is necessary to make a specific course of action. With a microphone turns voice recording identified and sent to the computer. The optimal reception is WAV file, since handling ease. The resulting voice recording should be divided into frames.

The next step is to eliminate the undesirable effects and noises. It is necessary to record obtained at different time correspond to each other, regardless of external factors. There are many ways in which to reduce the effects of noise.

To date, the most successful are the voice recognition system, using the knowledge of the hearing aid device. They are based on the fact that the ear interprets sounds not linearly but in a logarithmic scale. In view of these features is necessary to bring the frequency response for each frame of mels.

This is the last step required for the subsequent conversion to vector features, which, compared to the base of voice recordings. The vector will comprise mel-cepstral coefficients.

The resulting feature vector is added to the database for later comparison. But a more accurate alternative is to use multiple entries of the same voice. A predetermined number of voice samples may be used to train the neural network. We used learning without a teacher, because it is much more plausible model of learning in the biological system. Kohonen developed and many others, it does not need to output the target vector and therefore.

Conclusion

The result of this study is a modular application that voice recognition by using an upgraded algorithm for computing the neurons in the Kohonen layer. The program consists of three main parts. The first performs the addition of users, the second and third authenticates voice recordings.

The study showed that the resulting algorithm can significantly speed up the program of voice identification. This upgrade allows you to use the program at the enterprises with the influx of users.

The same software system is very flexible and has plenty of room for further improvements and adding new features, making it not only profitable software product, but promising project for development and profit.



УДК 681

ИНФОРМАЦИОННАЯ МОДЕЛЬ ЛИЧНОСТИ

Харламов А.А.

*Институт высшей нервной деятельности и нейрофизиологии РАН,
г. Москва*

kharlamov@analyst.ru

В работе рассмотрены вопросы автоматической обработки текстов, описывающих личность, а также сопутствующих текстов, эффективного представления этой информации, а также – разработки удобных интерфейсов упомянутой информации с пользователем.

Ключевые слова: информационная модель личности, семантические представления, лингвистическая обработка текстов, диалог с пользователем

Введение

Если не принимать во внимание психологических особенностей личности, за основу информационной модели личности можно взять языковую модель личности [Караулов, 1987], в которой структура личности рассматривается как совокупность трех уровней, а именно: 1) вербально-семантического, где лексикон личности, понимаемый в широком смысле, включает также фонд грамматических знаний личности; 2) лингвокогнитивного, представляющего тезаурус личности, в котором запечатлен "образ мира", или система знаний о мире; и 3) мотивационного уровня - уровня деятельностно-коммуникативных потребностей, отражающего прагматику личности: систему ее целей, мотивов, установок и интенциональностей.

Расширим это представление в содержательную сторону – добавим в модель совокупность информационных источников, характеризующих моделируемую личность. Имея в виду существующие на настоящий момент технологии хранения и обработки информации, совокупность текстов, принадлежащих моделируемой личности, или описывающих ее, в наибольшей степени содержит информацию для моделирования.

Существующие медийные технологии лишь в небольшой степени содержат информацию для моделирования, причем, в основном, внешнего к содержанию характера. Поэтому, в работе речь пойдет исключительно об использовании текстовой информации для моделирования личности.

В работе рассмотрены вопросы создания информационного автомата, который реализует автоматическую обработку текстовой информации, формируя те самые три уровня языковой личности, и их расширение в сторону использования базы знаний [Харламов, 2006].

1. Модель мира человека

С информационной точки зрения модель личности в наибольшей степени определяется моделью мира упомянутой личности. Поэтому сначала поговорим о модели мира человека. Модель мира человека состоит из трех компонентов, в том числе, индивидуального многомодального образного правополушарного (у правшей), социализированного многомодального схематического левополушарного, и языкового, который также локализован в левом (у правшей) полушарии. Многомодальные компоненты не просты по структуре, по обработке информации, и потому, плохо моделируются. А, главное, они в малой степени архивируются в процессе деятельности человека. Языковой компонент, напротив, как правило, хорошо заархивирован, хорошо обрабатывается и визуализируется. А, главное, языковой компонент, будучи, запараллелен в процессе формирования с двумя многомодальными, поэтому, изоморфен по представляемой информации упомянутым многомодальным компонентам, и потому, может быть полноправным их представителем.

Языковой компонент модели мира человека представляет собой, с одной стороны, иерархию лингвистических представлений, а с другой – надлингвистические описания семантики и

прагматики модели мира человека. Языковой компонент моделирует, в своей нижней части, формальные основы языка – графематику, морфологию, синтаксис и семантику отдельного предложения. Надлингвистическая часть языкового описания картины мира включает статику семантических представлений и динамику прагматического описания модели мира – ее описательную и алгоритмическую части.

1.1. Модель языка

Модель языка в составе модели мира человека является иерархией представлений, включающих в себя словари, учитывающие формальную сочетаемость языковых единиц разных уровней. В том числе, словарь первого уровня – $\{B_i\}_1$ – морфологический словарь [Харламов с соавт., 2013a], второго уровня – $\{B_k\}_2$ – лексикон, словарь третьего уровня – $\{B_l\}_3$ – словарь синтаксических структур [Харламов с соавт., 2013b], и словарь четвертого уровня – $\{B_m\}_4$ – словарь попарной сочетаемости слов в тексте [Рахилина, 2000].

1.2. Семантическая сеть

Словарь попарной сочетаемости корневых основ $\{B_m\}_4$ позволяет построить сеть, характеризующую смысловую структуру текста. Получается так называемая ассоциативная (однородная семантическая) сеть N как совокупность несимметричных пар понятий (корневых основ) $\langle c_i, c_j \rangle$, где c_i и c_j – понятия (корневые основы), связанные между собой отношением ассоциативности (совместной встречаемости в некотором фрагменте текста, например, в предложении) [Харламов с соавт., 2008]: $\langle c_i, c_j \rangle \Rightarrow B_m \in \{B_m\}_4$:

$$N = \cup_{ij} \langle c_i, c_j \rangle. \quad (1)$$

Эта сеть получается также, если предварительно пары слов объединить в звездочки: все пары слов с одинаковым первым словом это одна звездочка – $d = \langle c_i, \langle c_j \rangle \rangle = \cup_j \langle c_i, c_j \rangle$, где $\langle c_j \rangle$ – множество семантических признаков главного слова c_i .

$$N = \cup_i \langle c_i, \langle c_j \rangle \rangle. \quad (2)$$

Семантическая сеть, характеризующая сочетание ключевых понятий в рамках целого текста, является первым надлингвистическим уровнем языковой модели мира. В ней учитываются значимость отдельных ключевых понятий и их взаимосвязи в целом тексте. Если рассматривать модель мира как множество текстов, описывающих мир человека (можно говорить также о множестве моделей предметных областей, которые также описываются текстами), то взаимосвязи ключевых понятий и их

ранги и составляют основное содержание семантического представления модели мира.

Для более тонкого анализа семантики можно заменить ассоциативные связи между ключевыми словами на весь спектр связей, используемых в языке. Для этого на морфологическом уровне выявляется вся морфологическая информация о словах $\{B_i\}_1 = \{m_i\}$, а на синтаксическом – информация о связях слов в группах и между группами $\{B_k\}_3 = \{r_k\}$, где r_k – предикативная связь субъекта с главным объектом, а $r_k | k > 1$ – все остальные типы связей. При этом структуры синтаксического уровня укладываются в рамки словаря шаблонов минимальных структурных схем предложения, и словаря валентностей глаголов.

Тогда для каждого простого предложения можно построить расширенную предикатную структуру, которая после небольших преобразований сводится тоже к звездочке $d = \cup_j \langle c_i, r_k, c_j \rangle$. Правда, в отличие от звездочки с простыми ассоциативными связями, в звездочке, построенной из расширенной предикатной структуры, вместо пар понятий используются тройки $\langle c_i, r_k, c_j \rangle$, где между парой понятий имеется связь, размеченная одним из k типов отношений [Kharlamov et al., 2008].

При этом ассоциативная сеть N может быть построена и из таких звездочек тоже:

$$N = \cup_i d_i = \cup_i \langle c_i, \langle r_{ij}, c_j \rangle \rangle. \quad (3)$$

1.3. Прагматическое описание

Необходимо заметить, что семантическая сеть текста одновременно включает в себя все понятия текста. Но если спроецировать предложения текста на эту сеть, то мы получим последовательность понятий сети, которые следуют друг за другом последовательно во времени. Эти понятия включены в фрагменты текста, которые либо являются описаниями чего-либо, либо описывают алгоритм реализации чего-либо.

Отдельные предложения этих фрагментов описывают отдельные фрагменты ситуации. Расширенной предикатной структуре отдельного предложения соответствует, после описанных выше преобразований, звездочка $d = \cup_j \langle c_i, r_{ij}, c_j \rangle$. Тогда цепочка расширенных предикатных структур содержит смысл этих фрагментов – описаний, или алгоритмов:

$$D = (d_i | i = \overline{1, N}). \quad (4)$$

1.4. Модель предметной области

Подберем корпус текстов таким образом, чтобы он описывал некоторую предметную область. Смысл последовательностей предложений этого корпуса текстов может быть представлен

последовательностями расширенных предикатных структур этих предложений. То есть последовательность расширенных предикатных структур (и цепочек соответствующих звездочек), соответствующих предложениям корпуса текстов и является моделью предметной области:

$$M = \cup_i D_i \quad (5)$$

1.5. Гипертекстовая структура

Семантическая сеть текста (или всего корпуса текстов) вместе с самим текстом представляют собой гипертекстовую структуру, которая позволяет ассоциативно навигировать по тексту. При этом каждое ключевое понятие семантической сети соотносится с множеством предложений текста, в которых оно содержится, ну и с позициями этих предложений в тексте.

2. Модель личности

Модель личности условно мы представили множеством текстов, которые, с одной стороны, были сгенерированы этой личностью («сто томов партийных книжек» В.И. Ленина), с другой стороны – это тексты, в которых эта личность описывается. Необходимо добавить, что модель личности обычно включает фоновые знания, характерные для ее эпохи, то есть множество текстов, описывающих текущее состояние общества (или последовательность таких состояний для разных промежутков времени). Первые два множества текстов (первое подмножество – в большей степени, чем второе) описывают языковой эквивалент индивидуального многомодального компонента модели мира человека. Второе множество соответствует социализированному компоненту модели мира человека, так как оно в большей степени, чем первые два, характерно для представлений всего общества.

2.1. Индивидуальная модель

Индивидуальная часть модели личности, таким образом, строится на основе персональных текстов, сгенерированных личностью. Это письма, дневники, художественные, и не очень произведения (вплоть до технических отчетов). Помимо персональных текстов для формирования индивидуальной части модели личности используются тексты, в которых эта личность описана – художественные и прочие произведения.

2.2. Социализированная модель

Социализированная часть модели личности также строится на основе текстов из СМИ, Интернета. Но для ее построения, в отличие от текстов, использованных для построения индивидуальной части модели, берутся тексты на темы, характерные для индивидуальной части модели, но не относящиеся к моделируемой

личности (дополняющие тексты предыдущего раздела).

Так для запроса пользователя: «Что нам делать с Украиной?» из Интернета скачиваются тексты по запросу «Украина». Для этого из индивидуальной модели мира личности – семантической сети – извлекается минимальный древовидный подграф – тематическое дерево, в котором отыскивается тема «Украина». Рассматриваются первые ее подтемы, затем – ниже. Из Интернета скачиваются тексты по запросам, соответствующим этим темам. Для каждой темы формируется своя модель предметной области.

2.3. Целеполагание

Для эффективного взаимодействия модели личности с пользователем необходимо уяснить, что нужно от нее пользователю. И вообще пользователю предпочтительны не ответы на вопросы, а активное поведение модели. Однако внесение потребностей в систему является отдельной большой проблемой. Поэтому, мы ее попытаемся избежать. Заменим потребности системы потребностями пользователя. Для этого нам необходимо выяснить исходную точку интересов пользователя на семантической сети и конечную точку. Далее, поведение модели определяется цепочкой вершин – ключевых понятий – на семантической сети от начальной точки до конечной точки. Выяснение намерений пользователя осуществляется в процессе диалога, о котором пойдет речь ниже.

3. Интерфейс

3.1. Диалог

Модель личности хороша сама по себе. Но наиболее эффективное (и эффектное ее применение), несомненно, возможно в диалоге с пользователем (оппонентом): «Владимир Ильич, ну и где Ваш коммунизм?» - «Э, батенька, условия изменились, да и люди теперь не те!».

Так для запроса пользователя: «Что нам делать с Украиной?» (см. Раздел 2.2) строится их общая семантическая сеть предметной области «Украина». Выявляются таким же образом темы и подтемы суммарной модели предметной области. Пользователю генерируется вопрос: «Что Вас интересует в первую очередь?», и ему перечисляются все ключевые слова предметной области, имеющие максимальный вес. Так делается до тех пор, пока не станут понятными исходная и конечная темы.

3.2. Синтез текста

Синтез корректного текста также является непростой задачей. В настоящий момент не существует систем, реализующих корректный синтез текста. Поэтому можно идти двумя путями. Более простой вариант – просто подбирать предложения из множества подходящих

предложений, собранных в гипертекстовой структуре соответствующей модели предметной области (см. Раздел 1.5), описывающие конкретную последовательность тем (от исходной темы к конечной). Можно синтезировать предложения по правилам синтаксиса языка и с использованием шаблонов синтаксических структур для той же самой последовательности тем.

3.3. Говорящая голова

Интерфейс в виде диалога оказывается удобным реализовать с помощью аватара – говорящей головы, клонирующей голос моделируемой личности. Для этого можно использовать говорящую голову и клонирующую голос программу, разработанную Б.М. Лобановым.

Заключение

Целью настоящей работы было показать возможность реализации информационной модели личности, общение с которой не только дает представление о логике мышления моделируемого, но и позволяет взаимодействовать с моделью в реальных ситуациях.

Информационная модель личности основана на семантических представлениях моделируемой личности в виде семантической сети множества текстов, описывающих моделируемую личность, или сгенерированных ею. Семантические представления расширяются лингвистическим анализом до описания прагматики, которая моделирует динамику на семантической сети.

Модель строится как совокупность индивидуальной и социализированной частей, первая из которых представляет семантику самой моделируемой личности, вторая – семантику моделей предметных областей, характеризующих конкретные ситуации в предметных областях за конкретный промежуток времени на основе текстов из любых доступных источников.

Взаимодействие пользователя с моделью личности осуществляется в форме диалога, синтез ответа в котором осуществляется либо подбором соответствующего предложения из сформированного для модели предметной области гипертекстового представления, либо на основе синтаксических шаблонов и правил.

Модель может быть оформлена в виде аватара – говорящей головы – имитирующей голос моделируемой личности.

Работа была выполнена в рамках НИР «Исследование механизма ассоциативных связей в речемыслительной деятельности человека методом нейросетевого моделирования при анализе текстовой информации» (при финансовой поддержке Российского фонда фундаментальных исследований, Грант 14-06-00363).

Библиографический список

- [Караулов, 1987] Караулов Ю.Н. Русский язык и языковая личность. /Отв. ред. член-кор. Д.Н. Шмелев. – М.: Наука, 1987.
- [Харламов, 2006] Харламов А.А. Нейросетевая технология представления и обработки информации (естественное представление знаний). – М.: Радиотехника, 2006.
- [Харламов с соавт., 2013а] Харламов А.А., Ермоленко Т.В., Дорохина Г.В., Журавлев А.О. Предсинтаксический анализ русско-английских текстов //Программная инженерия, № 10, 2013. – С. 37 – 47.
- [Харламов с соавт., 2013б] Харламов А. А., Ермоленко Т. В. Разработка компонента синтаксического анализа предложений русского языка для интеллектуальной системы обработки естественно-языкового текста //Программная инженерия № 7, 2013. – С. 37-47.
- [Рахилина, 2000] Рахилина Е.В. Когнитивный анализ предметных имен: семантика и сочетаемость. – М.: Русские словари, 2000.
- [Харламов с соавт., 2008] Харламов А.А., Раевский В.В. Перестройка модели мира, формируемой на материале анализа текстовой информации с использованием искусственных нейронных сетей, в условиях динамики внешней среды //Речевые технологии, N 3, 2008. – С. 27-35.
- Alexander A. Kharlamov, Tatyana V. Yermolenko, Andrey A. Zhonin Text Understanding as Interpretation of Predicative Structure Strings of Main Text's Sentences as Result of Pragmatic Analysis (Combination of Linguistic and Statistic Approaches) //M. Zelezny, I. Habernal, A.Ronzhin Eds., LNAI 8113 Speech and Computer, Proceedings of 15th Int. Conf., SPECOM 2013, Pilsen, Czech Republic, September 2013г. – Pp. 333-339.

PERSON PHYSIOGNOMY INFORMATIONAL MODEL

Kharlamov A.A.

*Institute of Higher Nervous Activity of RAS,
Moscow*

kharlamov@analyst.ru

Questions of automatical analysis of texts concerning person physiognomy description and also accompanied texts are represented in the paper. Questions of its effective representation and suitable user interface development are represented also.

Introduction

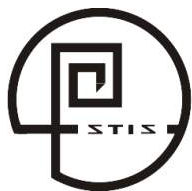
At the moment language person physiognomy model is well known now. For more exact representation of the person one need to extent the model in the direction of enlargement of the model specific content.

Main Part

Represented in the paper world model of person consists from three components: from individual one, socialized and language components. The language component includes the four-level language model. And the individual and the socialized components are constructed from semantic and pragmatic representations. The pragmatic representation is realized by full linguistic analysis.

Conclusion

In such a way we can consider the person physiognomy model in terms of semantic and pragmatic components which added by text corpuses of persons world model with talking head as an interface.



OSTIS-2015

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822 + 81'322.2

ОБ ОДНОМ МЕТОДЕ ПРИМЕНЕНИЯ ОБОБЩЕННОЙ ОНТОЛОГИИ ДЛЯ АНАЛИЗА ЕСТЕСТВЕННО-ЯЗЫКОВЫХ СООБЩЕНИЙ ПОЛЬЗОВАТЕЛЕЙ СЕТИ ИНТЕРНЕТ

Деева Н.В., Вишнеvский С.Я.

*Гродненский государственный университет имени Янки Купалы,
г. Гродно, Республика Беларусь*

nvdeeva@gmail.com

svishnevskij@gmail.com

В данной статье предложен один метод применения обобщенной онтологии для анализа открытых текстовых отзывов в сети Интернет для поддержки принятия решения пользователя по выбору наиболее предпочтительного варианта. Описан алгоритм определения списка критериев анализа отзывов «на лету», а также алгоритм получения эмоциональной оценки базовых слов по критерию и визуализация полученных результатов.

Ключевые слова: естественно-языковые сообщения, обобщенная онтология, эмоциональная оценка словосочетания

Введение

В современном мире все большую зависимость от знаний всемирного интернет сообщества приобретают различные сферы жизнедеятельности человека. Такую зависимость в целом определяет переход в начале 21 века подавляющего большинства web-ресурсов на концепцию формирования контента и взаимодействия с пользователем Web 2.0, а в последние 5 лет стремительный переход на Web 3.0. Web 2.0 характеризует пользователя не как пассивного потребителя информации, а как активного участника создания и верифицирования интернет-контента. В концепции Web 3.0 особый акцент сделан на семантику создаваемой и хранимой информации, извлечению знаний из массивов данных.

Все чаще пользователи высказывают свое мнение о покупке, услуге, сервисе и т.д. в виде комментариев, отзывов или статей, тем самым, расширяя не только количественную, но и географическую область обмена впечатлениями. Как правило, такого рода информацию размещают на официальных сайтах производителя в разделе комментариев и на форумах, в социальных сетях, а также на специализированных площадках для обмена отзывами по различным направлениям (туризм, шопинг, медицинские услуги и др.).

Очевидна актуальность извлечения данных из контента упомянутых выше ресурсов и их дальнейшая обработка с целью создания среды для поддержки принятия решения пользователя в части выбора услуги или товара. Задача обработки текстового контента открытого ресурса сводится к трем принципиально различным подзадам: формирование базы небольших текстовых сообщений, то есть разбор страницы ресурса и извлечение фрагментов текста из нее; подзадача обработки текстов на естественном языке и визуализация полученных результатов.

В данной статье главный акцент сделан на обработку уже готового набора текстовых сообщений на естественном языке, а также построения на основе полученных данных визуального представления проанализированного контента, технические вопросы по разбору страниц web-ресурса опущены, так как в контексте исследования не представляют научного интереса.

1. Постановка задачи

В данном исследовании в качестве источника информации используется ресурс по обмену отзывами о курортных отелях, размещенный по адресу <http://otzyv.ru/>. Ресурс позволяет оставлять отзывы и оценки по отелям даже незарегистрированным пользователям, тем самым предоставляя возможность оставлять анонимные

комментарии, упрощая интерфейс обмена информацией между желающими отдохнуть. Справедливости ради необходимо отметить, что именно эта открытость для пользователя не гарантирует объективность и достоверность предоставленной в комментариях информации, предлагая доступный механизм манипулирования информацией в конкурентной среде. Однако тема ложных или намеренно сгенерированных отзывов не является целью данного исследования, и мы будем исходить из предположения, что вся информация, приведенная в отзывах пользователей достоверна.

Итак, предложенный в качестве источника данных открытый ресурс позволяет пользователю определить независимую оценку отеля, и проанализировать пользовательские предпочтения, но только лишь в ручном режиме обработки данных.

Свои оценки пользователи оставляют в виде текстовых сообщений на естественном языке, а также выставляют баллы по нескольким категориям, например, «Питание», «Обслуживание» и т.д. На рисунке 1 приведен фрагмент пользовательского отзыва.

Оценки отелю:	
Расположение:	5
Территория:	5
Обслуживание:	3+
Питание:	4
Развлечения:	4
Для детей:	4

Отзыв:
Ну вот мы и вернулись. Сразу отпишусь, пока впечатления свежи. Завтраки. Несколько видов хлопьев, йогурты натуральные и фруктовые в упаковке (убы, желатина в них много). Рис с молоком, овсянка на воде, блины. Яйца вареные, скрембл, омлет или глазунью пожарят прямо для вас. Сосиски, фасоль, вареные бобы, пюре (плохое) или вареная обжаренная картошка (отличная). Несколько видов сладкой выпечки, много вкусного хлеба и СКРЕТ – теплые круассаны ищите в гастрюльке с крышечкой. Сыр, неплохая фета, местные колбасы, оливки и свежие овощи и фрукты.

Рисунок 1 – Фрагмент отзыва пользователя, размещенного на открытом ресурсе <http://www.otzyv.ru/>

Безусловно, в разделе «Оценки отелю» пользователь достаточно быстро может получить некоторые цифры – первичные критерии фильтрации, не отражающие по сути ни причин, ни конкретных фактов, повлиявших на автора при их выставлении. Глубинный анализ потребует от пользователя прочтения всех отзывов авторов, участвующих в оценке. А в современных реалиях, количество отзывов по одному направлению может достигать тысяч, что делает процесс обработки в ручном режиме зачастую невыполнимым в приемлемые сроки.

Необходимо предложить решение задачи ускорения анализа большого числа отзывов. В данной статье предлагается формировать некоторое визуальное представление для каждого отзыва по заданному критерию. Такое представление должно при беглом просмотре отзыва пользователем акцентировать его внимание на положительных и отрицательных фактах по заданному пользователем критерию, например, просмотр темы «Питание».

Целевой аудиторией данного решения в первую очередь являются рядовые пользователи, планирующие свой отдых, а также работники в сфере предоставления туристических услуг.

2. Алгоритм обработки пользовательских сообщений

2.1. Формирование списка критериев на базе обобщенной онтологии

С целью формирования списка критериев анализа необходимо определить наиболее популярные направления, раскрытые в отзывах пользователей. Для этого было принято решение получить набор наиболее частотных ссылок по тексту на концепты обобщенной онтологии. А в качестве обобщенной онтологии была выбрана классификация лексики Н.Ю. Шведовой, которая находится в открытом доступе на ресурсе <http://slovari.ru/> [slovari, 2014]. На рисунке 2 приведен фрагмент классификации лексики с частично раскрытым концептом «Еда, питье, кушанья: их компоненты».

- Продукты питания, еда, напитки
 - Общие обозначения
 - Еда, питье, кушанья: их компоненты
 - Общие обозначения
 - Хлеб, хлебные изделия, Мука, крупа: изделия из них
 - Молоко, молочные изделия, Яйца, яичные изделия, Жиры
 - Мясо, Рыба, Мясные и рыбные блюда, кушанья
 - Общие обозначения
 - Мясо млекопитающих, птиц: блюда из него
 - Само мясо млекопитающих (оола; озоа)
 - Части мясной туши (оола; озоа)

Рисунок 2 – Фрагмент классификации лексики Н.Ю. Шведовой

В качестве источника для анализа были выбраны около 20 000 уникальных отзывов, размещенных на открытом ресурсе www.otzyv.ru. Далее для формирования коллекции слов, претендующих на роль ссылок на концепты обобщенной онтологии, выполним морфологический анализ каждой найденной лексемы каждого сообщения, определим ее нормальную форму и добавим в ассоциативный массив-коллекцию слов, выполняя на данном этапе также подсчет частоты встречаемости данного слова во всем наборе сообщений. Кандидатами на роль ссылок будем считать слова, являющиеся именами существительными с признаком неодушевленности. На рисунке 3 представлен фрагмент xml-файла, содержащего полученную коллекцию слов.

```
<Word attr="1" count="77">САЛАТ</Word>
<Word attr="3" count="19">ОГУРЕЦ</Word>
<Word attr="3" count="28">ПОМИДОР</Word>
<Word attr="3" count="5">ПЕРЕЦ</Word>
<Word attr="1" count="10">КАПУСТА</Word>
<Word attr="1" count="8">ВИЛОК</Word>
<Word attr="3" count="3">БАКЛАЖАН</Word>
```

Рисунок 3 – Фрагмент xml-файла, содержащего коллекцию слов, претендующих на ссылки на концепты онтологии

Уровень морфологического анализа, на котором определяются морфологические характеристики

каждой лексемы, а также ее нормальная форма, реализован с использованием свободного СОМ-объекта, предоставленного группой разработчиков «Диалинг» [aot, 2014]. Данный программный инструментарий позволяет для заданной словоформы получить все возможные наборы морфологических характеристик, не решая тем самым проблему снятия омонимии и многозначности. В силу того, что на данном этапе строится только список критериев, то есть определяются основные темы массива сообщений, можно пренебречь заявленными выше проблемами и сформировать коллекцию для всех полученных слов, с условием их принадлежности к классу неодушевленных существительных.

Для определения ссылок на концепты обобщенной онтологии, построим xml-представление для основных концептов, доопределив их конкретными найденными экземплярами. На рисунке 4 представлен фрагмент xml-файла, содержащий конкретизацию концепта «Овощи. Бобовые. Бахчевые. Грибы». Название концептов соответствует классификации общепотребимой лексики [Шведова, 1998].

```
<TreeNode xsi:type="Sheet" word_count="22">
  <name>Овощи. Бобовые. Бахчевые. Грибы</name>
  <Scheme>0049</Scheme>
  <Position>0311</Position>
  <WordList>
    <Word count="77">САЛАТ</Word>
    <Word count="23">КАРТОШКА</Word>
    <Word count="4">ГОРОХ</Word>
    <Word count="19">ОГУРЕЦ</Word>
    <Word count="28">ПОМИДОР</Word>
    <Word count="20">ЗЕЛЕНЬ</Word>
    <Word count="13">КАРТОФЕЛЬ</Word>
    <Word count="5">ПЕРЕЦ</Word>
    <Word count="10">КАПУСТА</Word>
    <Word count="18">АРБУЗ</Word>
    <Word count="6">ГРУША</Word>
    <Word count="3">БАКЛАЖАН</Word>
    <Word count="2">ФАСОЛЬ</Word>
    <Word count="14">ДЫНЯ</Word>
    <Word count="5">ГРИБ</Word>
    <Word count="2">ЛУК</Word>
    <Word count="7">КАБАЧОК</Word>
    <Word count="1">МОРКОВЬ</Word>
    <Word count="2">РЕПА</Word>
    <Word count="1">ТОМАТ</Word>
    <Word count="1">ТЫКВА</Word>
    <Word count="3">ШПИНАТ</Word>
  </WordList>
```

Рисунок 4 – Фрагмент xml-файла, содержащего найденные экземпляры концептов

Каждый концепт в данном xml-представлении содержит информацию о количестве различных слов, конкретизирующих его, а каждое слово хранит информацию о его частотности в массиве текстовых сообщений. Используя данные числовые характеристики, а также выбирая пороговую величину вхождения слова в массив и величину мощности, конкретизирующих концептов, можно настраивать величину и степень детализации списка критериев анализа пользовательских отзывов.

В данном исследовании на базе более 20000 отзывов по 1000 отелей было получено более 4000

слов в коллекции, 1500 из них были выделены в качестве экземпляров концептов. Так, например, коллекция уникальных слов по разделу «Питание» составила порядка 300 единиц.

2.2. Формирование эмоциональных оценок словосочетаний

На предыдущем этапе были построены коллекции слов-экземпляров для каждого критерия из полученного списка. В результате выделения популярных тем была частично решена проблема омонимии и многозначности, исходя из предположения, что верное значение слова (на классе неодушевленных существительных) с большей вероятностью будет экземпляром наиболее важного концепта, с одной стороны, и неверное значение(я) слова с меньшей вероятностью войдут в список конкретизирующих важный концепт, с другой стороны.

Для дальнейшего анализа отдельных отзывов, например, описывающих 1 отель, выполним поиск всех присутствующих слов-конкретизаций для заданного критерия и определим эмоциональную оценку для каждого из них, если представляется такая возможность.

Будем считать, что оценка эмоционального окраса экземпляра концепта может быть определена интерпретацией, связанного с ним, прилагательного.

Наиболее распространенным типом связи существительного и прилагательного в словосочетании является согласование. При этом виде связи прилагательное согласовано с существительным в числе, роде и падеже. При наличии данных морфологических признаков для участников вероятных словосочетаний, задача нахождения согласованных существительных и прилагательных может быть решена в автоматическом режиме с достаточно высоким коэффициентом достоверности.

Таким образом, на этапе морфологического анализа будем получать наборы морфологических характеристик не только для неодушевленных существительных, но и для прилагательных, находящихся в непосредственной близости с ними. В случае, если прилагательное и существительное согласуются, они образуют словосочетание и в дальнейшем прилагательное при необходимости может составить для существительного эмоциональную оценку.

Однако не все прилагательные могут дать эмоциональную окраску существительному, будем рассматривать только класс качественных прилагательных. Причем по всему массиву отзывов найдем все потенциальные для оценивания прилагательные и составим из них список уникальных прилагательных. Затем в ручном режиме каждое прилагательное оценим по шкале от -1 до +1, с шагом 0.1, где 0 дает нейтральную окраску, в случае, если затруднительно оценить прилагательное вне контекста. Все «негативные»,

т.е. вызывающие негативные эмоции прилагательные оцениваются отрицательными значениями, абсолютная величина которых характеризует силу негативной оценки. Аналогично несущие позитивный смысл прилагательные оцениваются положительными числами. Так, например, в словосочетании «ужасное питание», прилагательное имеет оценку -1, а существительное «питание» получает максимально негативную эмоциональную оценку. На рисунке 5 представлен фрагмент xml-файла, содержащего оцененные прилагательные.

```
<adjective score="0.7" id="432">шустренский</adjective>
<adjective score="0.8" id="433">уважаемый</adjective>
<adjective score="-1" id="434">ужасный</adjective>
<adjective score="-0.8" id="435">громкий</adjective>
<adjective score="0.1" id="436">вечный</adjective>
<adjective score="-1" id="437">вонючий</adjective>
```

Рисунок 5 – Фрагмент xml-файла, содержащего прилагательные с эмоциональными оценками (где «score» - оценка)

По всему массиву отзывов было получено около 4000 словосочетаний, на базе которых был построен список из порядка 1000 уникальных прилагательных, для каждого из которых была назначена оценка. 453 прилагательных получили нейтральную оценку, что означает что более 50 % уникальных прилагательных получили положительную или отрицательную оценку по предложенной шкале.

2.3. Визуализация анализа текста отзыва

Последний этап в решении поставленной задачи была определена подзадача визуализации проведенного, на предыдущих шагах, анализа. То есть необходимо предоставить пользователю визуальное представление отобранных по отелю отзывов с проведенным на этих отзывах анализом – поиском экземпляров концептов с их эмоциональной оценкой. Такое представление будем формировать с помощью цветовой схемы: зеленым выделяются положительные оценки, красным – отрицательные, желтым – нейтральные; причем интенсивность цвета показывает близость к максимальному по модулю значению. На рисунке 6 приведена демонстрация анализа и визуализации его результатов на одном фрагменте текста отзыва по критерию «Продукты питания, еда, напитки».

Рисунок 6 – Пример анализа отзыва (желтым выделена нейтральная оценка, зеленым - положительная, красным - отрицательная)

Предложенная визуализация анализа позволяет пользователю моментально обратить внимание на негативные оценки в отзыве, а цветовая схема позволит определить суммарную семантику оценки по заданному критерию, кроме того такой унифицированный способ визуализации позволяет

по единой шкале проводить визуальную оценку разных отелей упрощая и ускоряя задачу выбора, что и было основной целью данного исследования.

Стоит тем не менее отметить, что в данном исследовании поиск словосочетаний ограничивался биграмами, нахождение триграмм и более сложных словосочетаний является темой другого исследования.

Заключение

В данной статье предложен метод использования обобщенной онтологии для анализа естественно-языковых текстов сети интернет, на примере обработки текстов туристических отзывов одного ресурса. Полученное в результате решение позволяет пользователю оперативно получать визуальное представление результатов анализа по сформированному «на лету» списку критериев, исключая из рассмотрения недостаточно представленные в описании. А также предложенная модель может быть применена на различных предметных областях, так как в основу метода положена обобщенная онтология, описывающая общеупотребительную лексику.

Библиографический список

- [Шведова, 1998] Шведова, Н.Ю. 1. Толковый словарь, систематизированный по классам слов и значений / Российская академия наук. Ин-т рус. яз. им. В. В. Виноградова; Под общей ред. Н.Ю. Шведовой // М.: Азбуковник, 1998.
- [slovari, 2014] Классификация лексики [Электронный ресурс]. – 2014. – Режим доступа: <http://slovari.ru/default.aspx?s=0&p=2672> – Дата доступа: 13.04.2014.
- [aot, 2014] Автоматическая обработка текстов [Электронный ресурс]. – 2014. – Режим доступа: <http://aot.ru/>. – Дата доступа: 1.02.2014.

ABOUT ONE METHOD OF USE OF GENERALIZED ONTOLOGY FOR THE ANALYSIS OF THE NATURAL LANGUAGE MESSAGES OF INTERNET USERS

Deeva N.V., Vishneuski S.Y.

The Yanka Kupala Grodno State University,
Grodno, Republic of Belarus

nvdeeva@gmail.com

svishnevskij@gmail.com

In this article we propose one method of using a generalized ontology for the analysis of shared text reviews on the Internet for the user decision support in the selection of the preferred option. There is also the description of an algorithm for determining the list of criteria for analyzing the feedback in real time, as well as an algorithm of obtaining emotional evaluation of basic words and visualization of the results.

Keywords: natural language messages; generalized ontology; emotional evaluation of collocation.



OSTIS-2015

(Open Semantic Technologies for Intelligent Systems)

УДК 004.891.2

ЧАСТОТНЫЙ МЕТОД КЛАССИФИКАЦИИ ТЕКСТОВ С ЛЕКСИЧЕСКИМ РАЗБОРОМ СЛОВА

Третьяков Ф.И., Серебряная Л.В.

*Белорусский государственный университет информатики и радиоэлектроники,
г. Минск, Республика Беларусь*

Fiodor.Tretyakov@gmail.com

l_silver@mail.ru

В работе рассмотрен алгоритм выделения стема из слова на русском языке. Приведен новый частотный метод классификации. Предложена функциональная модель программного средства универсальной десятичной классификации. Данные методы и средства позволяют сделать присвоение УДК автоматическим с высокой точностью.

Ключевые слова: классификация; стем; удк; эвристика

Введение

На сегодняшний день существует большое количество неупорядоченной текстовой информации. Поэтому поиск и классификация необходимой информации по ключевому слову является одной из важнейших задач. Особенно остро проблема стоит в сфере науки, потому как исследователю часто приходится изучить множество научных работ, прежде чем найти что-то важное для себя. Иногда можно, только взглянув на работу, определить ее тематику, а бывает, что приходится прочитать большую часть текста, чтобы понять его смысл.

Чтобы сразу было известно, к какой области знаний относится научная работа, была придумана универсальная десятичная классификация (УДК). Она является обязательным атрибутом любой печатной или выложенной в электронном виде научной работы. С помощью УДК выполняется классификация информации, необходимая во всем мире для систематизации произведений науки, литературы и искусства, периодической печати, различных видов документов и организации картотек [1].

В настоящее время УДК назначается вручную на основе специальных справочников библиотекарями или специально обученным персоналом. Данная работа посвящена методам и средствам, позволяющим автоматически присваивать работе УДК, не привлекая к этому человека. Поэтому цель

работы можно определить как автоматизация универсальной десятичной классификации.

Поставленная задача сводится к тому, что для каждого текста, входящего в множество из n текстов, определить категорию m из УДК.

Предмет исследования работы – универсальная десятичная классификация текстов.

1. Подходы к решению задачи

Существует множество способов решить названную задачу. Прежде всего выбор подхода зависит от количества исходных данных. Если имеется набор текстов-образцов и категорий, то речь идет о контролируемом обучении и классификации. Затем необходимо определить решающее правило и разделяющую функцию, с помощью которых будет выполняться классификация «незнакомых» текстов. Сами тексты обрабатываются и из них выделяются метрики, которые подставляются в качестве параметра в разделяющую функцию, в результате чего определяется принадлежность текстов к одному из классов.

Одним из существенных факторов, влияющих на выбор класса, является язык, на котором написан текст [2]. Настоящая работа посвящена русскоязычным текстам. Поэтому для построения алгоритма классификации будут использоваться особенности русского языка.

Одним из самых популярных способов классификации является поиск по полному

совпадения. Однако он может давать коллизии. К примеру, пользователь задает слово для поиска «кошка». Поэтому в тексте, либо совокупности текстов будут выделяться слова, которые состоят из «кошка..» и любого продолжения этого слова. Такой поиск самый примитивный и совершенно очевидно, что он отсекает такие результаты, которые могли бы быть полезны для пользователя. Например, «кошачий», «кот». Данный способ является самым быстрым, но при этом максимально неточным. Поскольку созданное программное средство сначала не будет работать в режиме реального времени, и современные компьютеры имеют аппаратное обеспечение, способное реализовывать с приемлемым быстродействием «тяжелые» алгоритмы, то необходимо модифицировать алгоритм поиска по полному совпадению с целью повышения его точности [3].

В русском языке лексемы имеют сложные и разнообразные структуры, что существенно затрудняет процедуру классификации текстов. Однокоренные слова могут иметь различные окончания, суффиксы и приставки, которые не должны влиять на результат классификации. Однако при проверке формального совпадения однокоренных лексем и получении отрицательного результата сравнения классификация текстов, построенная на основе неточных результатов сравнения, оказывается неверной. Поэтому для анализа русского языка в качестве разделяющей необходимо выбрать функцию, оперирующую только частью слова и выдающую ответ на его основе. Такой частью может быть корень слова, но он игнорирует специфику русского языка, например, беглые гласные. Поэтому для частотной классификации с максимально возможной точностью его использовать нельзя [2]. Часть слова, которая будет служить его смыслом для классификации, является основа слова [1].

2. Стемминг как часть классификатора

Стемминг – один из способов выделения определенной части слова. Это процесс нахождения основы слова заданной лексемы. Основа не всегда совпадает с морфологическим корнем слова [1]. Задача нахождения основы слова представляет собой давнюю проблему в области компьютерных наук. Первая публикация на заданную тему датируется 1968 годом. Стемминг применяется в поисковых системах для расширения поискового запроса пользователя и является частью процесса нормализации текста. На сегодняшний день созданы различные реализации алгоритмов стемминга. Они применяются для решения различных задач интеллектуальной обработки текстовой информации.

Для решения задачи классификации используется специальный алгоритм стемминга под названием стеммер [3]. Он может выделять значимую часть слова (стем). Однако стеммер

может допускать ошибки, которые классифицируются следующим образом.

Ошибки стемминга 1-го рода. Стем дает слишком большое обобщение и поэтому сопоставляется с грамматическими формами более чем одной словарной статьи. Это самая многочисленная группа ошибок стемминга. К примеру, если при стемминге вам даст вам, то в дальнейшем поиск текста даст совпадение с вампир. В русском языке может быть весьма трудно полностью устранить данные ошибки. Например, глагол пасть при спряжении дает формы пади и пал. В результате стемминг дает па, и это очень большое расширение при поиске. Впрочем, ошибки такого типа могут рассматриваться и как способ включить в поиск однокоренные слова. В примере с кошкой это могут быть формы прилагательного кошачий. Компенсация ошибок первого рода успешно выполняется либо введением списка стоп-слов, либо более качественно – лемматизатором или флексером.

Ошибки стемминга 2-го рода. Усечение формы дает слишком длинный стем, которые не сопоставляется с некоторыми грамматическими формами этого же слова. К таким ошибкам приводит стремление разработчика стеммера найти компромисс с ошибками 1-го рода в случае, когда при словоизменении меняется основа слова. Такие слова есть даже в крайне регулярном в плане словоизменения английском языке. Например, группа неправильных глаголов. В русском языке случаи изменения основы даже не являются основанием для отнесения слова к группе неправильных, настолько часто это явление. В качестве примера, на котором обычно спотыкаются многие реализации стеммера, можно взять слова кошка и пачка, которые имеют формы кошек и пачек. Обычно стеммеры выполняют в этих случаях усечение до кошк и птичк, которые несопоставимы с формами родительного и винительного падежа множественного числа.

Ошибки стемминга 3-го рода. Стем построить невозможно из-за изменения в корне слова, которое оставляет единственную букву в стеме. Либо модель словоизменения подразумевает использование приставок. Пример для первого случая – глагол выпиться, имеющий форму вопьемся. Второй случай возникает в рамках грамматического словаря для сравнительной степени прилагательных и наречий в русском языке. Например, покрасивее как форма прилагательного красивый, или помедленнее как форма наречия медленно.

Для качественного выделения основы слова одного стеммера оказывается недостаточно. Для работы с русским языком можно использовать два дополнительных модуля грамматического словаря: лемматизатор и флексер (склонение и спряжение). С помощью лемматизатора слова приводят к базовой форме, что выполняется после обработки лексемы стемом. Флексер умеет выдавать все грамматические формы слова на основе базовой.

Это позволяет улучшить результат, проверяя найденные фрагменты по набору форм ключевого слова.

Среди всех реализаций стеммеров можно выделить два типа:

1. Использующие словарь для выделения части слова;
2. Использующие эвристическую модель [2].

3. Алгоритм выделения стема

Для выделения корня слова был разработан программный модуль, включающий в себя стеммер, флексер и лемматизатор. Стеммер использует эвристическую модель.

Для создания эвристического стеммера необходимы словари окончаний, формы причастий и деепричастий, суффиксов и приставок. По данным словарей и будет эвристически определяться часть речи. Суть алгоритма сводится к определению части речи для слова по его окончанию, используя словари окончаний. Порядок определения задается уникальностью окончания данной части речи. К примеру, окончания причастий невозможно спутать ни с чем другим, поэтому, стемминг начинается именно с них.

Для создания эвристического стеммера необходимы словари окончаний, формы причастий и деепричастий, суффиксов и приставок. По данным словарям и будет эвристически определяться часть речи. Суть алгоритма сводится к определению части речи для слова по его окончанию используя словари окончаний. Порядок определения задается уникальностью окончания данной части речи. К примеру, окончания причастий невозможно спутать ни с чем другим, поэтому, стемминг начинается именно с них.

Далее приведены словари окончаний:

Причастия и деепричастия:

Группа 1: в, вши, вшись.

Группа 2: ив, ивши, ившись, вы, бвши, бвшись.

Окончания группы 1 должны следовать после: а или я.

Прилагательные:

ее, ие, ые, ое, ими, ыми, ей, ий, ый, ой, ем, им, ым, ом, его, ого, ему, ому, их, ых, ую, юю, ая, яя, ою, ею.

Частицы:

Группа 1: ем, нн, вш, ющ, щ.

Группа 2: ивш, бвш, ующ.

Окончания группы 1 должны следовать после: а или я.

Возвратные местоимения:

ся, сь.

Глаголы:

Группа 1: ла, на, ете, йте, ли, й, л, ем, н, ло, но, ет, ют, ны, ть, ешь, нно.

Группа 2: ила, ыла, ена, ейте, уйте, ите, или, ыли, ей, уй, ил, ыл, им, ым, ен, ило, ыло, ено, ят, ует, уют, ит, ыт, ены, ить, ыть, ишь, ую, ю.

Окончания группы 1 должны следовать после: а или я.

Существительные:

а, ев, ов, ие, ье, е, иями, ями, ами, еи, ии, и, ией, ей, ой, ий, й, иям, ям, ием, ем, ам, ом, о, у, ах, иях, ях, ы, ь, ию, ью, ю, ия, ья, я.

Превосходная степень существительного:

ейш, ейше.

Словообразовательные:

ост, ость.

Приставки

Стемминг будет проходить по следующему алгоритму.

1. Происходит поиск окончаний причастий и деепричастий в слове. Если оно найдено, то удаляется и выполняется переход к шагу 3.

2. Осуществляется поиск окончаний прилагательных, глаголов или существительных. Если они найдены, то удаляются.

3. Если слово оканчивается на «и», оно удаляется.

4. С начала слова в нем ищется последовательность: гласная-согласная. Все буквы после этого сочетания будут блоком *n*. Если ее нет или блок *n* пустой, переход к шагу 7.

5. Ищется в блоке *n* блок *m*. Это блок, который следует после конструкции гласная-согласная. Если его нет, или он пустой, переход к шагу 7.

6. Ищутся в блоке *m* части слова «ост» и «ость». Если они найдены, то удаляются.

7. Если слово имеет окончание «ейш» или «ейше», то оно удаляется.

8. Если на конце слова найдено удвоенное «н», второе «н» удаляется.

9. Если на конце слова «ь», он удаляется.

10. Удаляется любое окончание, на которое оканчивается блок *n*.

11. Блок *n*+первая буква *m* и будет стемом.

Стеммер позволяет сделать поиск в тексте на русском языке более осмысленным и логичным. Минусами является сложность модуля и пониженная точность.

Примером данного алгоритма служит рисунок 1, где рассматривается слово «поигравший». На блоках сверху показаны шаги алгоритма от 0 (до алгоритма) и до 11 (после алгоритма). Перечеркнутым начертанием выделены буквы, которые будут удалены на текущей итерации. Жирным – найденное соответствие. Увеличенным шрифтом – искомый блок.

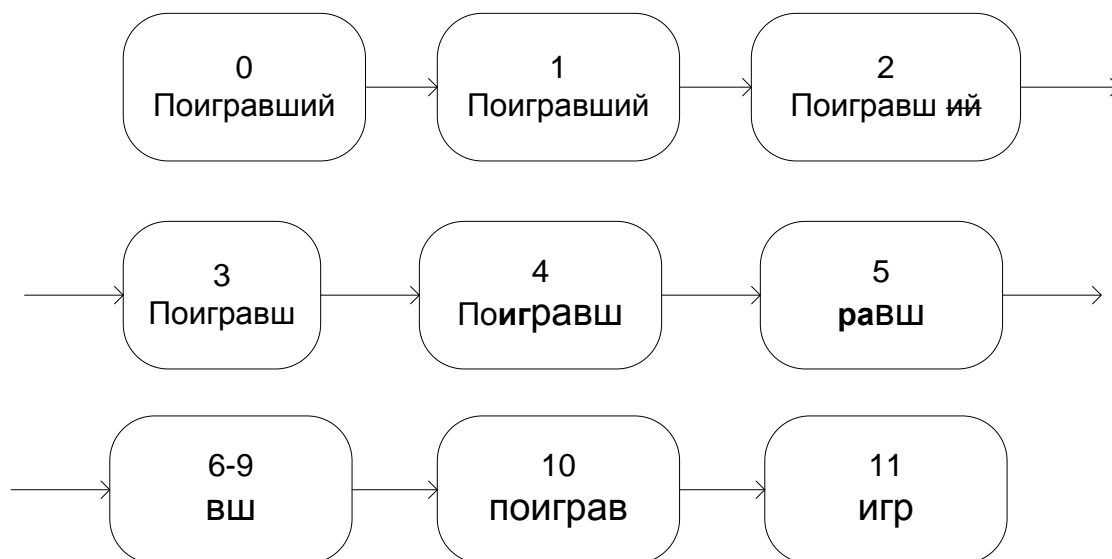


Рисунок 1 – Создание стема

4. Классификация текстов

Следующим шагом будет создание модуля классификации. Рассмотрим алгоритм классификации на основе созданного стеммера.

1. Происходит названий всех категорий с помощью модуля, выделив корни слов и поместив результаты в соответствующий словарь. Каждая строка в нем имеет ключ, является корень слова, а значение в строке – количество всех словоформ по ключу из названия категории.

2. Выполняется шаг 1 для всех, применив его не к названиям текстов, а к ним самим.

3. Для каждого текста находится подходящая категория. Ее номер определяется значения переменной T , вычисленной по следующей формуле:

$$T = \sum_{i=0, j=0}^{i<n, j<m} a_i \times b_j, \quad (1)$$

где n – размер словаря категории,

m – размер словаря текста,

a_i – слово из словаря категории,

b_k – слово из словаря текста.

4. Происходит выбор категории для текста, где T максимально.

Заключение

С помощью стеммера, флексера и лемматизатора можно классифицировать тексты с высокой точностью. Минусом является сложность архитектуры модуля.

Чтобы продемонстрировать работу данного алгоритма его необходимо встроить в программное средство, оформив в программный модуль. Модуль будет иметь следующую спецификацию:

1. Принимать на входе текст.

2. Выполнять классификацию.

3. Заносить данный текст в базу данных с соответствующим индексом, чтобы учесть данный результат в последующих классификациях.

4. Вывести результат пользователю на экран.

В итоге мы получается систему, которая позволяет присвоить УДК тексту с высокой скоростью, точностью и автоматизированно.

Библиографический список

[Толстых, 2012] Глубинный анализ текста. Из цикла лекций «Современные Internet-технологии» / Толстых В.Л.– М. : Вильямс, 2012.

[Браславский, 2005] Избранные прикладные задачи информатики / Браславский П.Ю.– М. : Вильямс, 2005.

[Duda, etc. 2005] Pattern classification / Duda R.O., Hart P.E., Stork D.G. N. Y. : John Wiley & Sons, 2001.

TEXT CLASSIFICATION FREQUENCY METHOD WITH WORD LEXICAL ANALYSIS

Tretyakov F.I. *, Serebryanaya L.V. *

*Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

Fiodor.Tretyakov@gmail.com

I_silver@mail.ru

In this article there was considered a stemming algorithm for Russian language. There was reviewed a new frequency classification method. This article proposes a functional model of a software for Universal Decimal Classification. These methods and software make it possible to do automatic Universal Decimal Classification with high accuracy.



OSTIS-2015

(Open Semantic Technologies for Intelligent Systems)

УДК 004.912

ПРОВЕРКА ИНФОРМАТИВНОСТИ КЛАССИФИКАЦИОННЫХ ПРИЗНАКОВ В ЗАДАЧЕ АВТОМАТИЧЕСКОЙ КЛАССИФИКАЦИИ ТЕКСТОВ НА ЕСТЕСТВЕННОМ ЯЗЫКЕ

Глазкова А.В.

*Тюменский государственный университет,
г. Тюмень, Россия
anya_kr@aol.com*

В статье рассматриваются базовые единицы автоматической обработки текстовой информации на примере автоматической классификации русскоязычной художественной литературы по ее возрастной аудитории. Экспериментально проверяется информативность ряда морфологических признаков, оцениваются лексические и некоторые количественные характеристики текстов различных категорий. Постановка данной задачи описывалась авторами в ходе выступления на конференции OSTIS-2014.

Ключевые слова: извлечение знаний; классификация; обработка естественного языка; семантический анализ.

Введение

Работа посвящена выявлению базовых признаков для автоматической обработки неструктурированной текстовой информации на русском языке на примере автоматической классификации художественной литературы по ее возрастной аудитории [Глазкова, 2014]. Эта задача актуальна в первую очередь для решения проблемы оптимизации информационного поиска в сети Интернет и хранилищах электронных документов.

Визуально определить возраст потенциального адресата текста обычно несложно, трудность заключается в поиске автоматического алгоритма решения задачи. В настоящий момент универсального набора признаков, позволяющего провести классификацию по заданному основанию, не выявлено, однако в работе ряда российских и зарубежных исследователей предлагаются отдельные признаки, которые могли бы быть положены в основание такой классификации. Часть признаков предложена авторами исследования. В работе проводится экспериментальная проверка информативности выявленных признаков.

В ходе исследования тексты делились на две категории: детские и взрослые. Это обусловлено наличием только этих двух категорий в выборке текстов, используемой для эксперимента. В дальнейшем планируется увеличить число классификационных категорий.

1. Корпус текстов

Для проведения эксперимента использовалась база Национального корпуса русского языка [НКРЯ, 2014]. Корпус состоит из заведомо качественных и максимально разнообразных текстов на русском языке с известным жанром. Авторы работали с двумя выборками – художественными текстами различных жанров (историческая проза, приключения, документальная проза и т.д., кроме детской литературы, – всего 5 902 документа, 9 332 659 предложений, 94 538 056 слов) и детской литературой (всего 632 документа, 547 735 предложений, 4 742 627 слов).

Также в работе использовалась [«База данных метатекстовой разметки Национального корпуса русского языка» (коллекция детской литературы)], состоящая из 510 текстов, ориентированных на читателей детского возраста.

2. Уточнение границ категорий

Поскольку в своем исследовании мы делим тексты на взрослые и детские, необходимо уточнить, что включают в себя данные категории.

В [ФЭБ, 2014] говорится, что детская литература включает в себя художественные, научно-художественные и научно-популярные тексты, написанные специально для детей, однако далее уточняется, что «обычно в понятие «детская литература» включается также широкий круг произведений для взрослых, прочно вошедших в

обиход детского чтения, — прежде всего, произв. народного творчества и классиков». Исходя из этого, можно сделать вывод о том, что границы категорий в задаче классификации текстов по возрасту их адресатов являются весьма размытыми. В данной работе под текстами, относящимися к детским, понимались те из них, которые:

- являются понятными детям с точки зрения лексики;
- соответствуют уровню их коммуникативного развития, являются информативными и представляют интерес для детской аудитории.

3. Краткий обзор литературы и выбор классификационных признаков

В ряде работ [Алексеева, 2009; Здир, 2009] в качестве отличительной характеристики текстов, предназначенных для младшей возрастной аудитории, указывается ограниченное использование автором функционально-смыслового типа речи описание. Под типом речи [Культура письменной речи, 2014] понимается способ изложения, избираемый автором и ориентированный на одну из задач: статически изобразить действительность, описать ее; динамически отразить действительность, рассказать о ней; отразить причинно-следственные связи явлений действительности. В соответствии с этими целями коммуникации выделяются три основных типа речи: описание, повествование, рассуждение.

В данном исследовании мы оценили возможность использования морфологических характеристик описания в качестве основания для классификации текстов по их возрастной аудитории.

Функционально-смысловой тип речи описание [Культура письменной речи, 2014] отличается от других типов речи тем, что он дает представление о каком-либо предмете или явлении перечислением их признаков и свойств. Ведущую роль в описании играют прилагательные, обеспечивающие выразительность и наглядность изображения. Описание как способ изложения противоположно повествованию и направлено на статическое отображение действительности, что выражается в меньшей в сравнении с другими типами речи частоте использования глаголов и глагольных форм.

В связи с этим в выборках проводился поиск следующих характеристик и их последующее сравнение:

- число глаголов;
- число особых глагольных форм (деепричастия, причастия);
- число прилагательных.

В соответствии с развернутым определением [Культура письменной речи, 2014] было выдвинуто предположение о том, что в текстах выборки $V_1 = \{T_1^{V_1}, T_2^{V_1}, \dots, T_n^{V_1}\}$, где $T_k^{V_1}, k = \overline{1, n}$ — тексты

художественной литературы, кроме детской, доля глаголов и особых глагольных форм должна быть выше, что в текстах выборки $V_2 = \{T_1^{V_2}, T_2^{V_2}, \dots, T_l^{V_2}\}$, где $T_k^{V_2}, k = \overline{1, l}$ — тексты, относящиеся к детской литературе. И напротив — доля прилагательных в текстах выборки V_2 будет превышать соответствующий показатель в выборке V_1 .

Также в качестве классификационных признаков были рассмотрены количественные признаки, основой для выбора которых послужили работы, посвященные оценке удобочитаемости (readability) текстов [Flesch, 1948; Оборнева, 2005; Шпаковский, 2012]. В качестве примера в данной работе оценены:

- среднее количество слов в предложении;
- средняя длина слов текста;
- процент многосложных слов в тексте (более трех слогов).

Кроме того, множества слов текстов из выборки V_1 и выборки V_2 были представлены в качестве моделей bag-of-words [EECS, 2014], далее было проведено сравнение полученных множеств.

4. Эксперимент

4.1. Морфологические характеристики

Значения итоговых показателей представлены в таблице 1, где для каждой выборки и морфологической характеристики приводится частота рассматриваемых частей речи:

$$F_{V_i} = \frac{N'_{V_i}}{N_{V_i}}, \quad (1)$$

где N_{V_i} — общее число слов в выборке, N'_{V_i} — число вхождений части речи в соответствии с результатами поиска.

Таблица 1 – Морфологические характеристики, полученные на основе выборок

Выборка	Частота (F_{V_i})		
	Глаголы	Особые формы глагола	Прилагательные
V_1	0,2	0,03	0,11
V_2	0,21	0,025	0,1

4.2. Количественные признаки

В таблице 2 отражены значения количественных признаков текстов двух выборок.

Таблица 2 – Количественные характеристики, полученные на основе выборок

Признак	Выборка	
	V_1	V_2
Среднее количество слов в предложении	11	6
Средняя длина слов текста (для слов, состоящих больше, чем из двух букв)	7	6
Процент многосложных слов в тексте (более трех слогов)	22,95	13,9

4.3. Лексические признаки

Для оценки лексического состава слова текстов каждой из рассматриваемых выборок были представлены в виде множества лексем, объединяющих в себе словоформы каждого встречающегося в тексте слова, и соответствующих им частотностей:

$$T_k^{V_i} = \{L_j, c_j\}, \quad (2)$$

где L_j – лексема, c_j – частотность данной лексемы.

Таким образом, на основе выборок и (2) были организованы модели bag-of-words:

$$M^{V_i} = \{L_j^{T_k^{V_i}}, c_j^{T_k^{V_i}}\}. \quad (3)$$

В дальнейшем были получены разности множеств, построенных на основе V_1 и V_2 , включающие в себя только те лексемы, которые входят в одну из выборок:

$$\begin{aligned} M^{V_1} \setminus M^{V_2}, \\ M^{V_2} \setminus M^{V_1}. \end{aligned} \quad (3)$$

Стоит отметить, что стоп-слова при построении множеств не исключались, поскольку они по определению оказались бы исключенными при вычитании множеств.

Пример выборки случайных лексем, входящих в разности множеств, представлен в таблице 3.

Таблица 3 – Примеры лексем, входящих в разности множеств

$M^{V_1} \setminus M^{V_2}$ (лексемы, отсутствующие в текстах детской литературы)	$M^{V_2} \setminus M^{V_1}$ (лексемы, присутствующие только в текстах детской литературы)
siemens	буквочка
гуманный	булочный
директива	бульканье
иконопись	булькин
исхлестать	капустница
навытяжку	колоннада
ноющий	накостылять
оборать	нисколечко
обустройство	остроухий
окопаться	пернатое
синергетик	пилка
стольпинский	примерещиться
сторублевка	тарашиться
татуировать	трусоватый
третьекурсник	уголек

5. Обсуждение результатов

Значения, полученные для морфологических признаков текстов (таблица 1), позволяют увидеть, что предположение, выдвинутое в пункте 3 и касающееся преобладающих типов речи, является верным. В соответствии с итоговыми показателями частота глаголов в выборке 1 составляет 0,2 (то есть 20%), особых глагольных форм – 0,03, прилагательных – 0,11. При этом частоты для выборки 2 – 0,21, 0,025 и 0,1 соответственно.

Несмотря на полученное подтверждение, различие в полученных значениях невелико, что не позволяет использовать данные характеристики в качестве классификационного признака.

Значения количественных признаков текстов (таблица 2) демонстрируют более существенные различия между текстами двух выборок. Предполагается, что полученный список информативных признаков можно расширить, добавив в него характеристики, связанные с синтаксической сложностью предложения (число грамматических основ, количество и вид придаточных предложений и т.д.).

Лексические признаки также могут послужить основанием для классификации. Безусловно, размер рассматриваемых выборок не является достаточным для формирования разностей множеств, исчерпывающе характеризующих словарный состав текстов, адресованный разным возрастным категориям читателей, однако даже на данном этапе работы полученные данные являются довольно информативными.

Заключение

Авторы благодарят НП "Национальный корпус русского языка" за предоставление базы текстов для проведения экспериментов, а также «Фонд Михаила Прохорова», финансировавший тревел-грант в рамках Открытого благотворительного конкурса «Академическая мобильность», программа «Образование как социальный институт», блок «Наука, образование, просвещение».

Библиографический список

- [Flesch, 1948] Flesch R. A new readability yardstick / R. Flesch // Journal of Applied Psychology, 1948, №32, С. 221-233.
- [EECS, 2014] Bag-of-words representation of text / EECS. [Электронный ресурс]. –2014. – Режим доступа: <https://inst.eecs.berkeley.edu>. – Дата доступа: 27.11.2014.
- [Алексеева, 2009] Редактирование детской литературы / Алексеева М.И.; – М.: Московский государственный университет, 2009.
- «База данных метатекстовой разметки Национального корпуса русского языка» (коллекция детской литературы), 2014] «База данных метатекстовой разметки Национального корпуса русского языка» (коллекция детской литературы). [Электронный ресурс]. –2014.
- [Глазкова, 2014] Глазкова А.В. Возможность автоматического определения адресата на основе семантико-синтаксических особенностей текста / А.В. Глазкова // Открытые семантические технологии проектирования интеллектуальных систем = Open Semantic Technologies for Intelligent Systems (OSTIS-2014): материалы конференции, 2014, С. 509-513.

[Здир, 2009] Здир В.В. Возраст читателя и специфика произведения для детей/ В.В. Здир // Практическая психология и логопедия, 2009, № 4, С. 44-47.

[Культура письменной речи, 2014] Культура письменной речи. [Электронный ресурс]. –2014. – Режим доступа: gramma.ru. – Дата доступа: 07.05.2014.

[НКРЯ, 2014] Национальный корпус русского языка. [Электронный ресурс]. –2014. – Режим доступа: ruscorgo.ru. – Дата доступа: 28.05.2014.

[Оборнева, 2005] Оборнева И.В. Автоматизация оценки качества восприятия текста/ И.В. Оборнева // Вестник Московского городского педагогического университета, 2005, №5, С. 86-91.

[ФЭБ, 2014] Фундаментальная электронная библиотека «Русская литература и фольклор». [Электронный ресурс]. –2014. – Режим доступа: <http://feb-web.ru>. – Дата доступа: 18.11.2014.

[Шпаковский, 2012] Шпаковский Ю.Ф. Оценка трудности восприятия текста / Ю.Ф. Шпаковский // Труды БГТУ. Издательское дело и полиграфия, 2012, №9, С. 72-75.

CLASSIFICATION FEATURES INFORMATIONAL CONTENT TESTING FOR AUTOMATIC NATURAL TEXTS CLASSIFICATION TASK

Glazkova A.V.

Tyumen State University, Tyumen, Russia

anya_kr@aol.com

The article deals with the basic units of automatic text processing on the example of automatic classification of the Russian fiction for its age audience. Experimentally the author verify informational content of several morphological characteristics, estimate lexical and some quantitative characteristics of texts of different categories. The statement of this problem is described by the authors in a report at the conference OSTIS-2014.

Introduction

The work is devoted to identifying the basic features for automatic processing of unstructured text information in Russian on the example of the automatic classification of fiction for its age audience. This problem is important for solving the optimization problem of information retrieval on the Internet and storages of electronic documents.

Visual determination of the age of a potential recipient of the text is not difficult, but the difficulty is in finding an automatic algorithm for solving the problem.

Currently, universal set of features allowing classification under the specified base, is not revealed, but in several Russian and foreign researchers offered some features that could be used as a basis of this classification. Some features proposed by the authors of the study. The paper deals with experimental verification of the informational content of the identified features.

In the research all texts were divided into two categories: children and adults. It's planning to increase the count of categories in the future.

Main Part

This research estimated three types of features. We call these types are: morphological, quantitative and lexical.

Morphological features include:

- number of verbs;
- number of special verbs forms (transgressives, participles);

- number of adjectives.

Quantitative features are:

- average count of words in a sentence;
- average length of words in a text;
- percent of polysyllable words in a text (more than three).

For estimation of lexical features of the text the author suggests creating bag-of-words model for both children and adults texts. Then by the subtraction sets of words the author offer to make a list of words that are present in the same category and absent in the other.

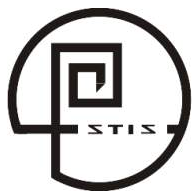
It should be noted that the stop words in the construction of the sets were not excluded, since they are by definition would have been excluded by subtracting sets.

Conclusion

Values obtained for the morphological features allow confirming the differences for different age categories. Despite the received acknowledgment, the difference in the values obtained is small, which makes use of characteristic data as a classification feature.

The values of quantitative traits texts show a significant difference between the texts of the two samples. It is assumed that the list of informative features can be expanded by adding the characteristics associated with the syntactic complexity of sentences (the number of grammatical foundations, the number and type of subordinate clauses, etc.).

Lexical features may also serve as a basis for classification. Of course, the dimensions of the samples is not sufficient to form the set difference exhaustively characterize the vocabulary of the text addressed to different age groups of readers, but even at this stage of the findings are quite informative.



OSTIS-2015

(Open Semantic Technologies for Intelligent Systems)

УДК 007:519.816

ВЕРИФИКАЦИЯ МОДЕЛЕЙ ПРОЦЕССОВ В ДИНАМИЧЕСКИХ СИСТЕМАХ ПО МЕТОДУ MODEL CHECKING

Королев Ю.И.

*Национальный исследовательский университет «МЭИ»,
г. Москва, Россия*

KorolevYu@gmail.com

Рассматриваются средства моделирования процессов в сложных динамических системах в плане их использования в интеллектуальных системах поддержки принятия решений реального времени. В качестве формального аппарата предлагается использовать сети Петри специального типа – раскрашенные (colored) сети Петри реального времени с поддержкой темпоральной логики Аллена. Обуславливается необходимость верификации моделей, разработанных с помощью данного формализма, предлагается использование метода верификации Model Checking. Работа выполнена при финансовой поддержке РФФИ и Фонда содействия инновациям.

Ключевые слова: интеллектуальная система поддержки принятия решений, моделирование процессов, сети Петри, темпоральная логика, верификация систем.

ВВЕДЕНИЕ

Интеллектуальные системы поддержки принятия решений реального времени (ИСППР РВ), предназначенные для помощи оперативно-диспетчерскому персоналу (лицу или группе лиц, принимающих решения, ЛПР) при управлении сложными динамическими объектами или процессами [Вагин и др., 2001], относятся к классу динамических интеллектуальных систем (ДИС) [Осипов, 2008]. Понятие динамической системы применяется в ситуациях, когда исследуется вопрос, как система (объект) развивается во времени посредством установления взаимосвязи между значениями параметров системы в различные моменты времени. Такой подход необходим при изучении и моделировании поведения многих сложных технических, организационно-технических, социальных, экологических систем, в частности, энергетических и транспортных систем [Еремеев и др., 2010]. Возникающие в таких системах зависимости и связи, зачастую в условиях различного типа неопределенности (так называемых НЕ-факторов или «зашумленных» данных) могут быть очень сложными и плохо формализуемыми, описываться качественными параметрами, а законы функционирования задаваться с использованием эмпирических или экспертных знаний. Все это не допускает традиционного аналитического представления и моделирования поведения

системы. Таким образом, в ДИС типа ИСППР РВ необходимо наличие специальных средств моделирования, которые могут быть использованы и для моделирования процессов и в самой ДИС. В качестве эффективного средства моделирования и анализа процессов динамических систем предлагается использовать аппарат цветных сетей Петри реального времени с поддержкой темпоральной логики Аллена (РСП РВ ТЛА) [Еремеев и др., 2013]. Надежность и предсказуемость поведения подобных систем зачастую являются более важными свойствами, чем производительность, модифицируемость и т.п. Кроме того, использование сетей Петри подразумевает достаточно высокий уровень параллелизма, что, как и учет темпоральных зависимостей, требует наличия средств анализа и верификации моделей, созданных на его основе данного аппарата.

1. Анализ модели

Многопоточные программы, характерные для многих систем управления, в том числе, реального времени типа ИСППР РВ, крайне подвержены ошибкам. Хорошо известно, что даже в тех случаях, когда функционирование каждой из параллельных взаимодействующих компонент системы абсолютно ясно, человеку трудно понять работу всей системы в целом. Такие системы годами могут сохранять «тонкие» ошибки, проявляющиеся в исключительных ситуациях. Рассматриваемый

подкласс РСП РВ ТЛА представляет собой визуальный язык программирования [Еремеев и др., 2013] с формально определенным синтаксисом. Модели, разработанные с помощью этого аппарата, кажутся полностью формализованными. Однако с точки зрения семантики это не так: из самой модели не следует непосредственно полное формальное описание ее поведения. Параллелизм, присущий сетям Петри в целом, и учет темпоральных зависимостей, введенный для упрощения разработки, зачастую делают целостное восприятие процесса более сложным. Поэтому для анализа поведения и верификации моделей, построенных с помощью РСП РВ ТЛА, необходимо использовать дополнительные инструменты.

Известны три основных группы методов анализа сетей Петри: основанные на построении графов изменения состояний; матричные методы, использующие уравнения сети и инварианты; методы редукции. При работе с раскрашенными сетями Петри последние две группы методов используются редко из-за высокой сложности формальных определений подобных подклассов. В [Еремеев и др., 2014] в качестве основного инструмента анализа РСП РВ ТЛА предлагается использовать графы покрытия (ГП). При этом для состояний сети (пары (M, S) , где M – маркировка сети, а S – вектор временных меток мест) вводится бинарное отношение покрытия. Два состояния покрывают друг друга, если их маркировки совпадают, а временные метки либо совпадают, либо не превышают максимального возраста доступа места, то есть такого значения темпоральной метки, когда фишки-токены становятся доступными для всех выходных переходов места. При этом удовлетворяются условия рефлексивности, симметричности и транзитивности, следовательно отношение покрытия есть отношение эквивалентности (\sim) на множестве $R(M_0, S_0)$ – множестве всех состояний, достижимых из начального состояния сети (M_0, S_0) . При построении ГП после определения нового состояния сети необходимо проверить, есть ли в графе вершина, которая отображает состояние, покрывающее новое. Если есть, то необходимо добавить только новую дугу, которая идет к найденной вершине. В противном случае вершина нового состояния добавляется в ГП вместе с соответствующей дугой. Таким образом, каждая вершина ГП помечена элементом фактор-множества $R(M_0, S_0)/\sim$, причем не существует двух или более вершин, помеченных одним и тем же элементом, а количество вершин ГП совпадает с числом элементов $R(M_0, S_0)/\sim$.

ГП для РСП РВ ТЛА всегда конечен, поскольку конечно множества $R(M_0, S_0)/\sim$:

- множество всех возможных маркировок M на конечном множестве мест сети конечно;
- максимальный возраст доступа каждого места задается в общем случае вещественным числом.

Анализ свойств сети может осуществляться с помощью маркировки узлов ГП и меток дуг. Каждая метка дуги представляет собой тройку, состоящую из перехода, его подстановки и значения временного промежутка перед его срабатыванием. Последний параметр позволяет определить время, затраченное на переход от одного состояния к другому. Используя алгоритмы поиска минимального и максимального пути между двумя узлами мультиграфа, можно найти минимальное и максимальное время перехода из одного состояния в другое. ГП позволяет увидеть все состояния сети с точностью до временных меток. Анализируя его, разработчик оценивает корректность выполнения поставленной задачи. Отметим, что ГП и для сравнительно малых РСП РВ ТЛА может достигать довольно больших размеров. Поэтому прямые исследования сетей путем их компьютерного моделирования могут упростить задачу разработчика.

2. Проверка правильности модели

2.1. Понятие верификации

Наиболее очевидным и широко распространенным методом проверки правильности программных систем является тестирование – проверка работы построенной системы в различных ситуациях, при различных исходных данных. Однако в случае с параллельными системами обычно невозможно заранее определить все возможные траектории функционирования. Поэтому в качестве основного метода повышения качества разработки применяется верификация – формальная проверка того, что система (модель) удовлетворяет сформулированным заранее требованиям [Карпов, 2010]. Методы верификации различаются в зависимости от того, какой аппарат лежит в основе проверяемой системы. Для верификации сложных систем свойства их поведения должны быть выражены формально логическими утверждениями, истинность которых зависит от времени, например, «Посланный запрос когда-нибудь позже будет обработан». Обычная логика высказываний является плохо пригодной для формулировки утверждений о поведении сложных динамических систем при изменении их состояний во времени. Формализация даже простейшего примера «Любой посланный запрос когда-нибудь позже будет обслужен» с помощью, например, логики предикатов первого порядка приводит к довольно громоздкому утверждению [Карпов, 2010]:

$$(\forall t \geq 0) (\text{Послан}(\text{Запрос}, t) \rightarrow \\ \rightarrow (\exists t' > t) (\text{Обслужен}(\text{Запрос}, t'))))$$

Поэтому при верификации темпоральных конструкций используются выражения темпоральных логик. Если нет необходимости подробно описывать закономерности поведения системы и взаимодействие ее объектов, целесообразнее применять не сложные интервальные логики, а простые расширения

обычной логики высказывания. Традиционно при верификации используются темпоральные логики линейного времени (*Linear Temporal Logic – LTL*) [Pnueli, 1977] и ветвящегося времени (*Computational Tree Logic – CTL*) [Clarce et al., 1986].

2.2. Метод верификации Model Checking

Перспективным методом верификации РСП РВ ТЛА является *Model Checking* (проверка модели) *MC* [Clarce et al., 1981]. Другие методы – дедуктивная верификация [Floyd, 1967] и проверка эквивалентности [Milner, 1980] – в общем случае не могут быть полностью автоматизированы, что негативно сказывается на возможности их применения при создании систем управления и ИСППР РВ. С другой стороны, исследования в области *MC* привели в последнее время к разработке очень эффективных алгоритмов верификации, позволяющих проверять реальные, разрабатываемые промышленностью программно-аппаратные системы. В частности, в работе [Карпов, 2010] приводятся эффективные алгоритмы *MC*, позволяющие проверить, что формула темпоральной логики *LTL* или *CTL*, выражающая некоторое свойство поведения динамической системы во времени, выполняется (является истинной) на модели системы с конечным числом состояний. В качестве модели при этом используется структура Крипке [Kripke, 1963], с помощью которой можно адекватно представить поведение реагирующих систем: дискретных систем управления, параллельных и распределенных алгоритмов, протоколов и т. п. Формально структура Крипке задается как пятерка

$$K = (W, W_0, H, AP, L), \text{ где:}$$

- W – конечное непустое множество состояний;
- $W_0 \subseteq W$ – непустое множество начальных состояний;
- $H \subseteq W \times W$ – множество переходов, удовлетворяющее требованию:

$$(\forall w \in W)(\exists w' \in W)((w, w') \in H);$$

- AP – конечное множество атомарных предикатов;
- $L: W \rightarrow 2^{AP}$ – функция пометок, сопоставляющая каждому состоянию множество истинных в нем атомарных предикатов.

Стандартными шагами доказательства того, что поведение реагирующей системы обладает некоторым свойством, являются следующие:

1) Для верифицируемой системы строится адекватная модель Крипке K , т. е. система переходов с конечным числом состояний. Поведения реальной системы представляются разверткой (деревом вычислений) построенной структуры Крипке.

2) С помощью переменных и параметров верифицируемой системы выражаются

интересующие разработчика атомарные предикаты структуры Крипке – логические выражения, которые могут принимать значения «истина» или «ложь» в каждом состоянии системы.

3) Проверяемое свойство выражается формулой ϕ темпоральной логики *LTL* или *CTL* с использованием атомарных утверждений, темпоральных операторов и кванторов пути.

4) Проверяется истинность утверждения $K \models \phi$ (т. е. утверждения, что структура Крипке является моделью формулы ϕ) с помощью полностью автоматизированной процедуры.

2.3. Верификация моделей РСП РВ ТЛА

Основную трудность при верификации по методу *MC* представляет необходимость построения структуры Крипке. Дальнейшие действия по проверке истинности темпоральных формул *LTL* или *CTL*, выражающих свойства системы, поддаются автоматизации, как показано в [Карпов, 2010]. В случае разработки модели процессов в парадигме формализма РСП РВ ТЛА нетрудно убедиться, что рассмотренный выше инструмент анализа ГП уже является структурой Крипке:

- W соответствует множество вершин ГП;
- W_0 соответствует вершина начального состояния (M_0, S_0) ;
- H соответствует множество переходов между вершинами ГП, требование существования перехода из любой вершины обеспечивается правилом построения ГП;
- AP соответствует конечное фактор-множество $R(M_0, S_0)/\sim$, содержащее по одному элементу для каждого класса эквивалентности по отношению покрытия состояний РСП РВ ТЛА;
- L соответствует функция пометок вершин ГП элементами множества $R(M_0, S_0)/\sim$

Таким образом, существует возможность автоматической верификации систем, разработанных на базе предложенного формализма; верификация РСП РВ ТЛА с помощью метода *MC* является естественным расширением начального анализа сетей с помощью графов состояний. Обобщенная схема процесса верификации приведена на рисунке 1.

Заключение

Общепризнано, что как тестирование, так и верификация по отдельности не могут гарантировать достаточного уровня правильности разрабатываемых систем. Существует большое число примеров, когда в тщательно проверенных и оттестированных реализациях с помощью верификации впоследствии обнаруживались тонкие ошибки. С другой стороны, нельзя надеяться только на верификацию. Часто причина ошибок кроется в том, что и при разработке, и при доказательстве алгоритмов неявно выдвигаются неправильные предположения о характере работы, т. е. используются неадекватные формальные модели.

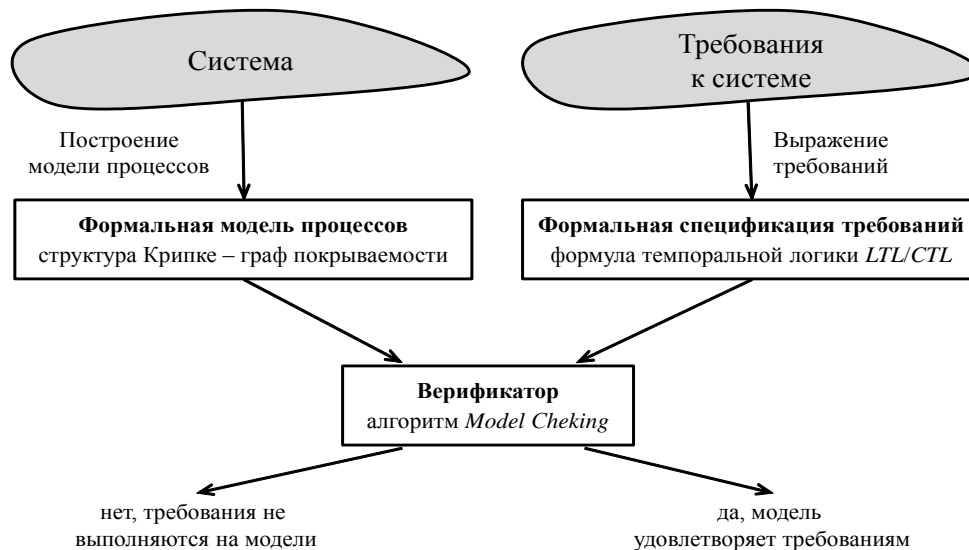


Рисунок 1 – Схема процесса верификации на основе алгоритма *Model Checking*

Таким образом, как тестирование, так и верификация обладают своими преимуществами и недостатками, поэтому эти подходы можно считать взаимодополняющими. Для повышения надежности реализаций при разработке систем управления, программных и аппаратных систем должны применяться оба подхода. В настоящий момент ведется разработка соответствующего программного обеспечения на языках высокого уровня, в которое планируется включить также возможность автоматической верификации моделей.

Библиографический список

- [Вагин и др., 2001] Вагин В.Н., Еремеев А.П. Некоторые базовые принципы построения интеллектуальных систем поддержки принятия решений реального времени // Изв. РАН. Теория и системы управления. 2001. № 6. - С. 114-123.
- [Осипов, 2008] Осипов Г.С. Динамические интеллектуальные системы // Искусственный интеллект и принятие решений. 2008. № 1. - С. 47-54.
- [Еремеев и др., 2010] Еремеев А.П., Куриленко И.Е. Средства темпорального вывода для интеллектуальных систем реального времени // В кн.: Интеллектуальные системы. Коллективная монография. Выпуск 4./ Под. ред. В.М. Курейчика. - М.: Физматлит, 2010. - С. 222-252.
- [Еремеев и др., 2013] Еремеев А.П., Королев Ю.И. Реализация интеллектуальных систем реального времени на основе сетей Петри с поддержкой темпоральных зависимостей // Программные продукты и системы. – 2013. – №3. – С. 88-94
- [Еремеев и др., 2014] Еремеев А.П., Королев Ю.И. Анализ и верификация раскрашенных сетей Петри реального времени с поддержкой логики Аллена // Открытые семантические технологии проектирования интеллектуальных систем: материалы IV Междунар. научн.-техн. конф. (Минск, февраля 2014г.)/ редкол.: В.В. Голенков (отв. ред.) [и др.] – Минск: БГУИР, 2014. - С. 461-464.
- [Карпов, 2010] Карпов Ю.Г. Model Checking. Верификация параллельных и распределенных программных систем. – СПб.: БХВ-Петербург, 2010. – 560 с.
- [Pnueli, 1977] Pnueli A. The temporal logic of program // Proc. of the 18th Anny. Symp. on Foundation of Computer Science. 1977. – P. 46-57.
- [Clarce et al., 1986] Clarke E.M., Emerson E.A., and Sistla A.P. Automatic verification of finite-state concurrent systems using temporal logic specifications // ACM Trans. Program. Lang. Syst. 1986. Vol. 8. No. 2. – P. 244-263.
- [Clarce et al., 1981] Clarke E.M., Emerson E.A. Design and synthesis of synchronization skeletons using branching-time temporal logic // Logic of Programs, 1981. - P. 52-71.

- [Floyd, 1967] Floyd R.W. Assigning meaning to programs // Proc. Symposium on Applied Mathematics. 1967. Vol. 9. - P. 19-32.
- [Milner, 1980] Milner R. A Calculus of Communicating Systems. Lecture Notes in Computer Science, vol. 92. - Springer-Verlag, 1980.
- [Kripke, 1963] Kripke S.A. Semantical consideration on modal logic // Acta Philosophica Fennica. 1963. Vol. 16. - P. 83-94.

VERIFICATION OF MODELS OF PROCESSES IN DYNAMIC SYSTEMS USING MODEL CHECKING METHOD

Eremeev A.P., Korolev Y.I.

National Research University «Moscow Power Engineering Institute», Moscow, Russia

Eremeev@appmat.ru

KorolevYu@gmail.com

In the paper the tools of modeling processes in complex dynamic systems are considered in terms of their use in intelligent decision support systems real time. A special type of Petri nets - real-time colored Petri net with support of Allen temporal logic is proposed to use as a basic formalism. The need to validate the models developed using this formalism is emphasized, it is proposed to use the Model Checking method of verification. This work was financially supported by RFBR and the Foundation for the promotion of innovation.

Keywords: intelligent decision support system, process modeling, Petri nets, temporal logic, system verification.



OSTIS-2015

(Open Semantic Technologies for Intelligent Systems)

УДК 004.832.3

ИССЛЕДОВАНИЕ И РАЗРАБОТКА СИСТЕМЫ АБСТРАКТНОЙ АРГУМЕНТАЦИИ С ВЕРОЯТНОСТНЫМИ СТЕПЕНЯМИ ОБОСНОВАНИЯ

Дервянко А. В., Моросин О. Л.

*Национальный исследовательский университет «МЭИ» (Московский энергетический институт)
г. Москва, Россия*

777alterego777@gmail.com

omorsik@gmail.com

В данной работе рассматривается возможность назначения вероятностей аргументам, являющихся элементами аргументационной системы. Вероятности аргументам присваиваются тогда, когда невозможно определить, является ли аргумент истинным или ложным, но имеются некоторые предположения относительно этого. Традиционная теория аргументации не позволяет учитывать дополнительную информацию о вероятностях аргументов при принятии решений. Теория аргументации с вероятностными степенями обоснованиями успешно решает эту проблему.

Ключевые слова: абстрактная аргументация; степени обоснования; эпистемический подход; групповой подход

ВВЕДЕНИЕ

Любая современная интеллектуальная система поддержки принятия решений содержит базу знаний. Данные базы знаний формируются, как правило, на основе экспертных знаний. Однако, такие базы знаний часто содержат противоречивую информацию. Причины возникновения противоречий могут быть различными: ошибки формализации знаний, неверная интерпретация, противоречивость мнений нескольких экспертов. Проблема обработки противоречий является актуальной, так как классические методы логического вывода не применимы для баз знаний, содержащих противоречия. Одним из способов обнаружения и разрешения внутренних противоречий является применение аргументации. Под аргументацией обычно понимают процесс построения предположений относительно некоторой анализируемой проблемы. Как правило этот процесс включает в себя обнаружение конфликтов и поиск путей их решения. В отличие от классической логики, аргументация предполагает, что могут быть доводы как “за”, так и “против” некоего предположения.

Недостатком классических систем аргументации является оперирование качественными статусами аргументов – “поражен” и “не поражен”. Введение степеней обоснования позволяет оперировать не только терминами “за” и “против”, но и давать числовую оценку аргументам и контраргументам.

Для подтверждения некоторого предположения, необходимо доказать, что существует больше доводов “за” и степень их обоснования выше, чем у доводов “против”.

Существуют несколько формализаций теории аргументации. Например, системы абстрактной аргументации, предложенные Дангом (Dung P.M.) [Dung, 1997], аргументационная система Лина и Шоэма (Lin F., Shoham Y.) [Lin, 1989], система Вресвийка (G.A.W. Vreeswijk [Vreeswijk, 1997], система аргументации Поллока (John L. Pollock) [Pollock, 1992] и некоторые другие.

В данной работе будет рассмотрена абстрактная система аргументации с вероятностными степенями обоснования. Абстрактная аргументация удобна для формального рассмотрения противоречивых данных, анализа аргументов и контраргументов. Абстрактная аргументация с вероятностными степенями обоснования позволяет учитывать дополнительную информацию о вероятностях аргументов в логическом выводе.

Под вероятностью аргумента, можно подразумевать вероятность принадлежности аргумента графу аргументов (иначе говоря, степень обоснованности появления аргумента на графе аргументов). Данное обоснование вероятности аргумента используется в групповом подходе. С другой стороны, под вероятностью аргумента можно рассматривать степень убежденности в истинности аргумента [Hunter, 2012].

Чем больше информации о характере происхождения аргументов известно, тем более точным будет представление об их вероятностях. Например, если аргумент включает в себя некоторые посылки (поддержка аргумента), а из посылок проистекает некоторое заключение (утверждение аргумента), то вероятность истинности аргумента является функцией вероятности, которая определяется истинностью поддержки аргумента и вероятностью того, что утверждение аргумента следует из этих посылок. Таким образом, предположение об истинности аргумента основывается на неопределённости, присущей используемой информации и неопределённости, присущей процессу рассуждения. Неопределённости рассуждений, а именно способы получения степеней обоснования для систем аргументации, обладающих механизмом получения новых аргументов с помощью пересматриваемых рассуждений рассматривались, например, в [Вагин, 2013] и [Pollock, 2001]. В данной работе усилия направлены на изучение неопределённости, присущей информации, а не процессу рассуждения, поэтому будем считать, что вероятности заданы априорно и анализировать уже имеющиеся графы аргументов с заданными вероятностями.

Для достижения поставленной цели были решены следующие задачи:

1. Произведён обзор различных подходов к построению аргументационных систем с вероятностными степенями обоснования;
2. Приведены способы работы с графами вероятностей аргументов;
3. Разработана система абстрактной аргументации с применением группового подхода;
4. Произведена реализация, отладка и тестирование системы на примерах.

1. Абстрактная аргументация

Рассмотрим теоретические основы абстрактной аргументации, предложенной Дангом [Dung, 1995]. В системах абстрактной аргументации набор аргументов представляется в виде направленного бинарного графа. Вершины данного графа обозначают аргументы, а его рёбра — отображают отношения атак аргументов.

Определение 1. *Графом аргументов* называется пара (A, R) , где $A = \{A_1, A_2, \dots, A_n\}$ — множество аргументов и R — бинарное отношение над A (т. е., $R \subseteq A \times A$).

Каждый элемент $A_i \in A$, $1 \leq i \leq n$ называется *аргументом* и запись $(A_i, A_j) \in R$ означает, что A_i атакует аргумент A_j .

Определение 2. Пусть $\Gamma \subseteq A$ — множество аргументов. Γ атакует $A_j \in A$ если существует аргумент $A_i \in \Gamma$, такой, что A_i атакует A_j . Γ защищает от атак $A_j \in A$, если для каждого аргумента $A_j \in A$, если A_j атакует A_i то Γ атакует A_j .

Определение 3. Говорят, что множество аргументов $\Gamma \subseteq A$ не содержит противоречий, если

не существует таких $A_i, A_j \in \Gamma$, что A_i атакует A_j .

Определение 4. Множество аргументов $\Gamma \subseteq A$ — *допустимое (admissible) множество*, если оно не содержит противоречий и защищает от атак все свои элементы.

Для того, чтобы множество аргументов было приемлемо, необходимо, чтобы для каждого аргумента, который атакуется контраргументом (не содержащимся в этом множестве), данное множество содержало основания для оспаривания этих контраргументов. Всегда существует по крайней мере одно допустимое множество: пустое множество всегда допустимо. Очевидно, что допустимость множества является минимальным требованием для его приемлемости. В этой работе рассматриваются следующие классы приемлемых множеств аргументов, предложенные Дангом [Dung, 1995]:

Определение 5. Пусть Γ — не содержащее противоречий множество аргументов и пусть имеется функция $Defended(\Gamma)$ такая, что $Defended(\Gamma) = \{A_i \mid \Gamma \text{ защищает } A_i\}$.

1. Γ — *полное (complete) расширение*, если $\Gamma = Defended(\Gamma)$.

2. Γ — *фундаментальное (grounded) расширение*, если оно является минимальным (относительно включения по множеству) полным расширением.

3. Γ — *предпочтительное (preferred) расширение*, если оно является максимальным (относительно включения по множеству) полным расширением.

4. Γ — *устойчивое (stable) расширение* если Γ предпочтительное расширение, атакующее все аргументы в $A \setminus \Gamma$.

Пример 2. Рассмотрим граф аргументов, изображённый на рисунке 1:



Рисунок 1 - Граф аргументов

Множества, которые не содержат противоречий: $\{\}, \{A\}, \{B\}, \{C\}$, и $\{A, C\}$; Допустимые множества: $\{\}, \{A\}, \{C\}$, и $\{A, C\}$; единственное полное расширение: $\{A, C\}$, оно же фундаментальное, предпочтительное и устойчивое.

2. Способы задания распределения вероятностей в аргументационных системах

2.1 Графы вероятностей аргументов

Для каждого аргумента на графе аргументов можно задать вероятность, показывающую нашу степень уверенности в достоверности аргумента. Таким образом, каждый аргумент A_i получает значение $p(A_i)$ из единичного интервала.

Определение 6. Графом вероятностей аргументов называется кортеж (A, R, P) , где (A, R) это граф аргументов и $P: A \rightarrow [0, 1]$.

В общем случае, нет никаких ограничений на значения вероятности, присваиваемые аргументам. Так, например, можно назначить каждому аргументу графа вероятность равную 1. В таком случае, рассматриваемая система будет эквивалентна системе абстрактной аргументации без степеней обоснования. Аналогичным образом можно присвоить каждому аргументу нулевую вероятность, в этом случае получаем граф вероятностей аргументов с вершинами нулевых вероятностей, в котором не будет приемлемых множеств аргументов. Далее рассмотрим, некоторые варианты, которые имеются для выбора и использования вероятностных оценок.

Для того, чтобы уточнить понятие определения вероятностей на графе потребуем некоторые условия и введём обозначения, которые будем использовать далее. Обозначим $G = (A, R, P)$ — граф вероятностей аргументов, и $A' \subseteq A$ — некоторое подмножество аргументов. *Маргинализация* R с A' , обозначаемая $R \otimes A'$, это подмножество R , включающее в себя только отношения между аргументами из A' (т.е. $R \otimes A' = \{(A_i, A_j) \in R \mid A_i, A_j \in A'\}$). Если $G = (A, R, P)$ и $G' = (A', R', P)$ — графы вероятностей аргументов, то G' является полным подграфом G , обозначаемым $G' \subseteq G$, если $A' \subseteq A$ и R' такое, что $R = R' \otimes A'$. На рис. 2 приведено 7 полных подграфов, для графа (а). При этом граф (а) также является полным подграфом для самого себя.

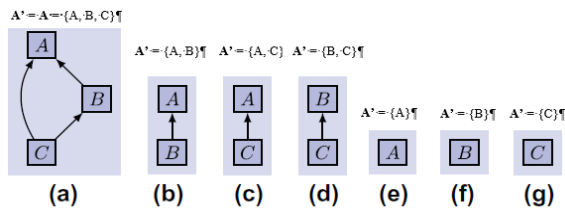


Рисунок 2- Семь полных подграфов графа (а).

Т. к. множество аргументов полного подграфа является любым подмножеством аргументов исходного графа, то для того, чтобы составить множества аргументов всех полных подграфов некоторого графа вероятностей с множеством аргументов $A = \{A_1, A_2, \dots, A_n\}$ необходимо найти все размещения без повторов для множества A .

Рассмотрим два способа присваивания значений вероятностей аргументам. Первый способ — *эпистемический подход* [Pollock, 1970], который интересен нам «рациональными» распределениями вероятностей. Это такие распределения, в которых атакующему присваивается высокий уровень доверия, в то время как атакуемому аргументу присваивается низкий уровень доверия и наоборот. Второй способ — *групповой подход*, предложенный в [Li, 2011], в котором аргументам присваиваются вероятности для создания распределения вероятностей над полными подграфами. Таким образом, можно рассматривать каждое $G' \subseteq G$, как «интерпретацию» G . Такое распределение может быть использовано для создания распределения вероятностей над расширениями.

2.2 Эпистемический подход

Данный подход рассматривает такие распределения, в которых атакующим аргументам присваивается высокий уровень доверия, в то время как атакуемому аргументам присваивается низкий уровень доверия и наоборот. Эпистемический подход был предложен в [Pollock, 1970]. В данном подходе вероятностное распределение над аргументами используется чтобы напрямую указать, какие аргументы наиболее достоверны. То есть чем выше вероятность аргумента, тем более обоснованным он считается. Это удобно, т. к. кроме структуры графа, то есть информации о том, какие аргументы рассматриваются, можно определить «рациональные» степени представления об аргументах. Обычно, при эпистемическом подходе используется следующая интерпретация вероятностей [Dung, 1995]:

- $P(A) = 0$ показывает, что существует уверенность, в том что A ложно;
- $P(A) < 0.5$ показывает, что A скорее всего ложно;
- $P(A) = 0.5$ показывает, что нет уверенности в том, что A ложно или истинно;
- $P(A) > 0.5$ показывает, что A скорее всего истинно;
- $P(A) = 1$ показывает, что существует уверенность, в том что A истинно.

2.3 Групповой подход

Данный подход предлагается в [Li, 2011]. В групповом подходе используется распределение вероятностей над подграфами аргументов исходных графов. С помощью полных подграфов можно проводить исследование более допустимых множеств и расширений (см. определения 4, 5).

Для аргумента A в графе G , с функцией вероятности P , $P(A)$ это вероятность того, что A содержится в произвольном полном подграфе G , и $1 - P(A)$ вероятность того, что A не содержится в произвольном полном подграфе G .

Пусть $G = (A, R, P)$ и $G' = (A', R', P')$ — графы вероятностей аргументов, такие, что $G' \subseteq G$. Вероятностью подграфа G' , обозначаемой $p(G')$ называется величина [Hunter, 2007]:

$$\left(\prod_{A \in A'} p(A) \right) \times \left(\prod_{A \in A \setminus A'} (1 - p(A)) \right) \quad (1)$$

Если $G = (A, R, P)$ граф вероятностей, то $\sum_{G' \subseteq G} p(G') = 1$, т. е. сумма вероятностей всех полных подграфов графа вероятностей равна 1.

Пусть $G = (A, R, P)$ — граф вероятностей аргументов, $\Gamma \subseteq A$, и $G' = \{G_1, G_2, \dots, G_k\}$ — множество тех подграфов, для которых Γ является допустимым множеством. Вероятность того, что Γ является расширением равна сумме вероятностей всех подграфов множества G' :

$$\sum_{G_i \in G'} p(G_i) \quad (2)$$

Приведем пример использования группового подхода.

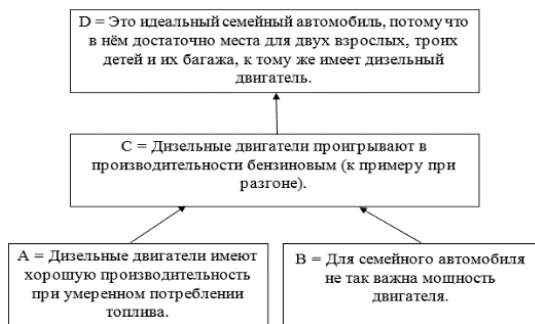


Рисунок 3 - Пример графа аргументов

Пример 3. Рассмотрим пример графа аргументов, приведённый на рис. 3:

Предположим, что аргументы A и B истинны, но это ничего не говорит об истинности или ложности аргументов C и D . Например, предположим, что имеются вероятности: $p(A) = 1$, $p(B) = 1$, $p(C) = 0.5$, и $p(D) = 0.5$. Теперь неопределённость представлена в виде 4 полных подграфов G^1 , G^2 , G^3 , и G^4 ненулевой вероятности:

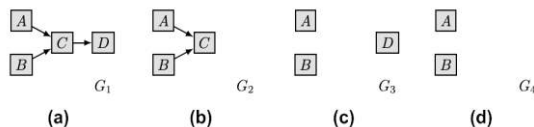


Рисунок 4 - Подграфы ненулевой вероятности.

Таблица 1 - Допустимых множеств для полных подграфов

Полный подграф	Допустимые множества
G^1	{A, B, D}, {A, B}, {A, D}, {B, D}, {A}, {B}, {}
G^2	{A, B}, {A}, {B}, {}
G^3	{A, B, D}, {A, B}, {A, D}, {B, D}, {A}, {B}, {D}, {}
G^4	{A, B}, {A}, {B}, {}

Каждый подграф имеет вероятность $1/4$ (вероятность подграфа вычисляется по формуле (1)). Вероятность каждого допустимого подмножества вычисляется по формуле (2). В результате получено 8 допустимых множеств ненулевой вероятности: $p(\emptyset^{ad}) = 1/4 + 1/4 + 1/4 + 1/4 = 1$, $p(\{A\}^{ad}) = 1/4 + 1/4 + 1/4 \cdot 1/4 = 1$, $p(\{B\}^{ad}) = 1/4 + 1/4 + 1/4 \cdot 1/4 = 1$, $p(\{D\}^{ad}) = 1/4$, $p(\{A, B\}^{ad}) = 1/4 + 1/4 + 1/4 \cdot 1/4 = 1$, $p(\{A, D\}^{ad}) = 1/4 + 1/4 = 1/2$, $p(\{B, D\}^{ad}) = 1/4 + 1/4 = 1/2$, и $p(\{A, B, D\}^{ad}) = 1/4 + 1/4 = 1/2$, где ad — допустимое множество (см. определение 4). Следовательно, любое множество, содержащее C имеет нулевую вероятность и любое допустимое множество, содержащее D вместе с A и B имеет вероятность $1/2$.

После рассмотрения графа аргументов, C опровергается, а D не получает определенного статуса, т.е. не опровергается, но и не подтверждается.

В результате получено два расширения ненулевой вероятности: $p(\{A, B\}^X) = 1/2$ и $p(\{A, B, D\}^X) = 1/2$, где X обозначает, что расширение является полным, предпочтительным, фундаментальным и устойчивым (см. опред. 5).

3. Разработка программной реализации системы абстрактной аргументации с применением группового подхода

3.1 Общая структура системы

Система реализована в интегрированной среде разработки JetBrains IntelliJ IDEA на языке объектно-ориентированного программирования Java. При реализации системы были созданы следующие модули.

- Граф аргументов (основан на используемой библиотеке для прорисовки графов).
- Граф вероятностей аргументов (расширяет возможности модуля графа аргументов, позволяя назначать аргументам вероятности).
- Модуль редактирования графа вероятностей аргументов, включающий в себя граф вероятностей аргументов и инструменты для его редактирования, сохранения в файл и загрузки из файла.
- Модуль полных подграфов графа аргументов, с возможностью отображения всех полных подграфов аргументов и таблиц их характеристик.
- Модуль группового подхода, включающий в себя модуль полных подграфов, который отображает информацию, полученную в ходе применения группового подхода (вероятности допустимых множеств, опровергаемые и подтверждаемые аргументы, полные расширения).

- Вспомогательные инструменты.

Каждый модуль это несколько связанных классов, с реализацией всех методов, необходимых для работы с объектами этого класса на абстрактном уровне.

3.2 Алгоритмы

В данном разделе представлены основные алгоритмы программной системы.

Алгоритм построения и вычисления характеристик полных подграфов.

Входные данные: вершины графа вероятностей аргументов, значения вероятностей, отношения атак аргументов.

Выходные данные: Таблица полных подграфов, каждая строка которой содержит номер подграфа, функцию вероятности, допустимые множества подграфа, значение вероятности подграфа.

Шаг 1. На основе множества графа вероятностей аргументов находятся все размещения аргументов без повторений, к примеру, для графа аргументов $\{A, B, C\}$ полными подграфами будут графы с вершинами $\{A, B, C\}$, $\{A, B\}$, $\{A, C\}$, $\{B, C\}$, $\{A\}$, $\{B\}$, $\{C\}$, $\{\}$.

Шаг 2. Для каждого полного подграфа вычисляются характеристики:

Шаг 2.1. Вычисление функции вероятности. Функция вычисления вероятности подграфа представляет собой произведение вероятностей аргументов, причём, если аргумент графа

вероятностей аргументов имеется в данном полном подграфе, то в функцию вероятности входит $p(Arg)$, если же аргумент имеется в графе вероятностей аргументов, но отсутствует в данном полном подграфе, то в функцию вероятности входит $1 - p(Arg)$. Например, если имеется граф вероятностей аргументов с аргументами $\{A, B, C\}$, то вероятность полного подграфа $\{A, B\}$ определяется функцией:

$$P\{A, B\} = p(A) * P(B) * (1 - p(C))$$

Шаг 2.2. Нахождение допустимых подмножеств (см. определение 4).

Сначала, по аналогии с получением подграфов, строятся размещения аргументов без повторений из аргументов подмножества. Затем, производится проверка противоречивости (см. определение 3) и защищённости множества (см. определение 2).

Шаг 2.3. Вычисляется вероятность подграфа. Для вычисления используется функция, полученная на шаге 2.2 и значения вероятностей, назначенные аргументам на графе вероятностей.

Алгоритм исследования графа вероятностей аргументов с помощью группового подхода.

Входные данные: вершины графа вероятностей аргументов, значения вероятностей, отношения атак аргументов, список характеристик полных подграфов.

Выходные данные: Допустимые множества аргументов с их вероятностями, списки опровергаемых, поддерживаемых и не опровергаемых и не поддерживаемых аргументов, полные расширения подграфов, расширения ненулевой вероятности.

Шаг 1. Вычисление вероятностей допустимых множеств по формуле (2).

Шаг 2. Поиск опровергаемых и подтверждаемых аргументов среди аргументов с вероятностью

$p \in (0, 1)$ (см. определение 4).

Шаг 3. Поиск полных расширений подграфов (см. определение 5). Программа также выделяет те расширения, которые соответствуют подграфам ненулевой вероятности.

Шаг 4. Поиск вероятностей полных расширений. Для полных расширений, полученных на предыдущем шаге определяется вероятность, аналогично вероятностям допустимых подмножеств, т. е. находится сумма вероятностей тех подграфов, в которые входит данное расширение.

В заключение приведём результат применения разработанной системы для решения задачи, описанной в примере 3.

Строим граф и назначаем вероятности $p(A) = 1$, $p(B) = 1$, $p(C) = 0,5$ и $p(D) = 0,5$ (см. рис. 5). На рис. 6 показаны первые 4 полных подграфа, показанные программой.

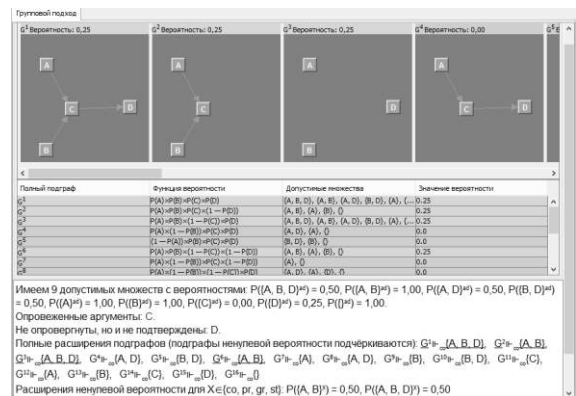


Рисунок 7 - Выводы, сделанные в ходе применения группового подхода.

Выводы, сделанные в ходе рассмотрения данного примера в предыдущей главе, совпали с выводами, сделанными программой (см. рис. 7).



Рисунок 5 - Граф вероятностей аргументов, построенный в программе.



Рисунок 6 - Первые четыре полных подграфа графа.

ЗАКЛЮЧЕНИЕ

В данной работе были рассмотрены основные понятия абстрактной аргументации, используемые при рассмотрении аргументов с вероятностными степенями обоснования такие как: «граф аргументов», «отношение атаки», «расширение», а также различные возможности назначения вероятностей аргументам и способы работы с данной неопределённостью. Изучены два способа работы с вероятностями аргументов: эпистемический и групповой подходы. Эпистемический подход, используется для нахождения рационального относительно аргументов графа распределения вероятностей. Групповой подход, позволяет строить распределения вероятностей над полными подграфами данного графа аргументов, с помощью которых можно проводить исследование более допустимых множеств, расширений и выводов.

Также реализована программная среда — система абстрактной аргументации с применением группового подхода. Программный комплекс был реализован в виде нескольких модулей, взаимодействие которых предоставляет возможность создания, редактирования графов вероятностей аргументов и работы с ними. Был реализован модуль группового подхода, позволяющий находить допустимые множества, полные расширения и их вероятности, а также подтверждать или опровергать аргументы, истинность которых находится под сомнением.

Библиографический список

- [Dung, 1997] Dung P.M., Bondarenko A., Kowalski R.A., Toni F. An Abstract Argumentation-Theoretic Framework for Defeasible Reasoning // Artificial Intelligence. – 1997. – Vol. 93, № 1-2. – с. 63-101.
- [Hunter, 2012] A. Hunter, Some foundations for probabilistic argumentation, in: Proceedings of the International Conference on Computational Models of Argument (COMMA'12), 2012.
- [Вагин, 2013] В.Н. Вагин, О.Л. Моросин. Применение механизма степеней обоснования в системах аргументации. // Вестник Ростовского гос. ун-та путей сообщения. ISSN 0201-727X. № 3. 2013. Стр. 43-50.
- [Pollock, 2001] Pollock J.L. Defeasible reasoning with variable degrees of justification // Artificial Intelligence. Artificial Intelligence. – Vol. 133, 2001, p. 233–282.
- [Lin, 1989] Lin F., Shoham Y. Argument Systems. A Uniform Basis for Nonmonotonic Reasoning // Proceedings of the First International Conference on Principles of Knowledge Representation and Reasoning. – San Mateo CA: Morgan Kaufmann Publishers Inc, 1989. – с. 245-355.
- [Vreeswijk, 1997] Vreeswijk G.A.W. Abstract Argumentation Systems // Artificial Intelligence. – 1997. – Vol. 90. – с. 225-279.
- [Pollock, 1992] Pollock J.L. How to Reason Defeasibly // Artificial Intelligence. – 1992. – Vol. 57.
- [Pollock, 1970] Pollock, J. (1970). The structure of epistemic justification. American Philosophical Quarterly, 4, 62–78.
- [Dung, 1995] On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games, Artificial Intelligence 77 (2) (1995) 321–357.
- [Li, 2011] Li J., Oren N., Norman T., Probabilistic argumentation frameworks, in: Proceedings of the First International Workshop on the Theory and Applications of Formal Argumentation (TFAFA'11), 2011.

[Hunter, 2007] A. Hunter, Real arguments are approximate arguments, in: Proceedings of the 22nd AAAI Conference on Artificial Intelligence (AAAI'07), 2007.

[Hunter, 2013] A. Hunter, A probabilistic approach to modelling uncertain logical arguments// International Journal of Approximate Reasoning, Vol. 54, Iss. 1, 2013, с.47–81.

RESEARCH AND DEVELOPMENT OF THE ABSTRACT ARGUMENTATION SYSTEM WITH PROBABILISTIC DEGREE OF JUSTIFICATION

Derevyanko Andrey, Morosin Oleg

Moscow Power Engineering Institute(Technical University)

777alterego777@gmail.com

omorsik@gmail.com

In this paper, the methods of using probability degrees of justification in abstract argumentation system are considered. Probabilistic justification degrees allow us to solve various argumentation problems that need numerical estimation of an answer.

INTRODUCTION

The probability degrees are assigned to the arguments when it is impossible to determine whether the argument is true or false, but there are some assumptions about it. The traditional theory of argumentation cannot take into account additional information about the probabilities of arguments when making decisions. The theory of argumentation with probabilistic degree of justification successfully solves this problem.

MAIN PART

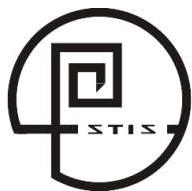
There are several formalizations of the argumentation theory. In this paper, we will consider the development of abstract argumentation systems(AAS). In AAS it is assumed that the argumentation problem is formalized in terms of orientated graph, where arguments are nodes and edges are binary relation “attack”. Such systems do not have mechanism of inference, but they are useful for analyzing complicated argumentation problems.

The problem of such system is that, there are no mechanism for obtaining numerical estimation of argument plausibility. It is proposed to use justification degrees to cope with this problem.

Two methods of calculating justification degrees were considered – epistemic and constellations approach.

CONCLUSION

Abstract argumentation could be used in many intelligent systems. Nevertheless, the implementation of different mechanisms of calculating justification degrees, including methods of fuzzy logic, is a topic of a future work.



УДК 004.912

МЕТОДЫ ТЕМАТИЧЕСКОЙ КЛАСТЕРИЗАЦИИ ПРИМЕНИТЕЛЬНО К АНАЛИЗУ НОВОСТНЫХ СТАТЕЙ

Солошенко А.Н. *, Орлова Ю.А. *, Заболеева-Зотова А.В. **

* *Волгоградский государственный технический университет,
г. Волгоград, Россия*

nastyasolan@gmail.com

yulia.orlova@gmail.com

** *Российский фонд фундаментальных исследований, г. Москва, Россия*

zabzot@gmail.com

Данная работа посвящена проблеме выделения сюжетов и тем из потока новостных сообщений. Кратко рассмотрены алгоритмы кластеризации, такие как алгоритмы k-средних, минимальное покрывающее дерево и др. Проанализированы результаты их работы на новостных текстах. В работе представлена методика комплексного анализа новостного текста, основанная на комбинации статистических алгоритмов извлечения ключевых слов и алгоритмов формирования семантической связности блоков текста. Особое внимание уделено особенностям структуры новостного текста.

Ключевые слова: тематическая кластеризация; алгоритмы кластеризации; новостные статьи; представление документов.

Введение

В наши дни заметно усилилась проблема информационной перегрузки. В начале XXI века американская исследовательская служба Cyveillance сообщила о том, что количество страниц в Internet превысило 4 млрд, и с каждым днем увеличивается на 7 млн. В частности, темпы роста аудитории онлайн-новостных ресурсов практически вдвое превышают темпы роста общей численности пользователей интернета, составляя к сегодняшнему дню 43,2% российских интернет-пользователей (исследование сотрудников Nielsen//NetRatings).

«Сырые» неструктурированные данные составляют большую часть информации, с которой имеют дело пользователи, поэтому многие организации и частные лица заинтересованы в эффективных технологиях автоматизированного анализа информации, представленной на естественном языке. При этом автоматическая кластеризация, т.е. выявление групп семантически похожих текстов, является одной из приоритетных задач, решаемых информационными системами.

1. Обзор существующих систем анализа текстов, обеспечивающих возможность кластеризации документов

На международном рынке представлено множество программных продуктов, предоставляющих функцию кластеризации текстовых документов.

Среди отечественных стоит выделить системы TextAnalyst, Galaktika-ZOOM, из зарубежных – мощный инструмент анализа текстов IBM Text Miner. В возможности TextAnalyst входит создание семантической сети большого текста, подготовка аннотации, автоматическая классификация и кластеризация текстов. IBM Text Miner содержит утилиты классификации, кластеризации, поиска ключевых слов и составления аннотации текстов. Однако на обработку новостных статей программы не направлены.

Российская система Яндекс Новости позволяет автоматически группировать данные в новостные сюжеты и составлять аннотации статей на основе кластера новостных документов. Сервис InfoStream, обеспечивает доступ к оперативной информации в поисковом режиме с учетом семантической близости документов. Мобильный агрегатор

новостей Summly, купленный в марте 2013 компанией года Yahoo!, также осуществляет группировку новостных статей по темам. Однако приложение абсолютно неприменимо для обработки текстов на русском языке.

Таким образом, существующие программные системы полностью не решают поставленную проблему. Основная идея заключается в разработке программного продукта для анализа новостных текстов, сочетающего кластеризацию новостных статей с комплексным анализом текста.

2. Особенности структуры новостного текста в решении задачи кластеризации

Проанализировав ряд статей, представленных на новостных сайтах, относящихся к топ-30 новостных порталов Рунета), можно построить обобщенную структуру текста новости (рисунок 1).

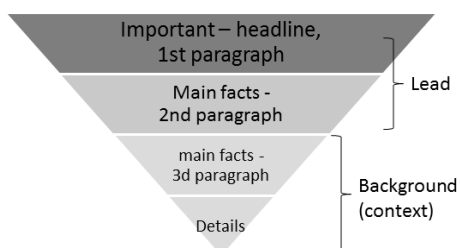


Рисунок 1 – Структура новостного текста

В ее основу заложен принцип «перевернутой пирамиды», который требует размещение основной информации в самом начале материала и последующее ее раскрытие далее по тексту в деталях.

- Заголовок новости отражает ее тему и содержит не более 10 слов (около 80 символов). Так, для примера, в Яндексе отображается не более 15-ти слов в тайтле, Google показывает до 70 слов;
- Основные факты, касающиеся события, отражены в 1-2 абзацах, и составляют так называемый лид текста (освещает главную тему);
- 3-й и последующие абзацы составляют бэкграунд новости (контекст). Как правило, здесь раскрываются детали происходящего, дается информация, напрямую касающаяся новости.

Таким образом, для содержимого новости справедлива формула: (Who? + What? + Where? + Why? + When? + How?) [Добров, 2010]. Это так называемый закон «пять W и одно H», приписываемый Р. Киплингу. Если бы все новостные сообщения строились по единой структуре, то решение задачи кластеризации могло бы значительно упроститься.

3. Кластеризация новостного потока

Прежде чем перейти к описанию алгоритмов кластеризации, определим основные понятия исследуемой области.

Кластеризация - разбиение множества документов на кластеры - подмножества, параметры которых заранее неизвестны. Количество кластеров может быть произвольным или фиксированным. Основные группы алгоритмов кластеризации – это иерархические и плоские, четкие и нечеткие алгоритмы.

Иерархические алгоритмы строят не одно разбиение выборки на непересекающиеся кластеры, а систему вложенных разбиений. То есть на выходе мы получаем дерево кластеров, корнем которого является вся выборка, а листьями — наиболее мелкие кластера. Плоские алгоритмы строят одно разбиение объектов на кластеры.

Четкие (непересекающиеся) алгоритмы каждому объекту выборки ставят в соответствие номер кластера, т.е. каждый объект принадлежит только одному кластеру. Нечеткие (пересекающиеся) алгоритмы каждому объекту ставят в соответствие набор вещественных значений, показывающих степень (вероятность) отношения объекта к кластерам.

3.1. Формализация задачи кластеризации документов

Рассмотрим задачу кластеризации документов более формально.

Пусть X – множество объектов, Y – множество номеров кластеров. Задана функция расстояния между объектами $\rho(x, x')$. Имеется конечная обучающая выборка объектов $X^m = \{x_1, \dots, x_m\} \subset X$. Требуется разбить выборку на непересекающиеся подмножества, называемые кластерами, так, чтобы каждый кластер состоял из объектов, близких по метрике ρ , а объекты разных кластеров отличались. При этом каждому объекту $x_i \in X^m$ приписывается номер кластера y_i [Большакова и др., 2011].

Алгоритм кластеризации – это функция $a: X \rightarrow Y$, которая любому объекту $x \in X$ ставит в соответствие номер кластера $y \in Y$.

3.2. Методы кластеризации

Далее рассмотрим лишь несколько алгоритмов кластеризации, оптимальных, с нашей точки зрения, для обработки новостного потока.

3.2.1. Метод агломеративной кластеризации

Иерархические алгоритмы разделяются на два вида: агломеративные (восходящие) и дивизимные (нисходящие). Первые строят кластеры снизу вверх, начиная с множества кластеров, содержащих по одному одиночному документу коллекции, затем последовательно объединяют пары кластеров, пока не получат один кластер, содержащий все документы коллекции. Вторые разбивают кластеры сверху вниз, начиная с одного кластера, которому принадлежат все документы коллекции, затем этот кластер делится на два и так рекурсивно до тех пор,

пока каждый документ не окажется в своём отдельном кластере.

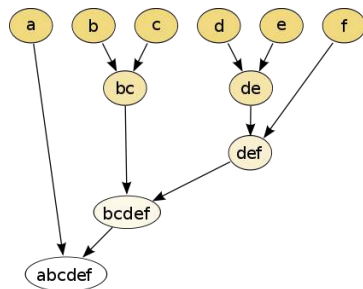


Рисунок 2 – Дендрограмма (граф без циклов, построенный по матрице мер близости)

Основное их различие заключается в выборе критерия, используемого для принятия решения о том, какие кластеры следует объединить на текущем шаге алгоритма. Большое распространение получили следующие критерии:

- одиночная связь (минимальное расстояние, или максимальное сходство): сходство двух кластеров - сходство между их наиболее похожими документами;
- полная связь (максимальное расстояние, или минимальное сходство): сходство двух кластеров есть сходство между их наиболее непохожими документами;
- групповое усреднение (усреднение всех показателей сходства): сходство двух кластеров есть среднее сходство всех пар документов, включая пары документов из одного кластера, исключая близость документа самому себе;
- центроидный метод;
- метод Уорда [Vandyopadhyay et al., 2013].

3.2.2. Алгоритм k-means (k-средних)

При заранее известном числе кластеров k алгоритм начинает с некоторого начального разбиения документов и уточняет его, оптимизируя целевую функцию – среднеквадратичную ошибку кластеризации как среднеквадратичное расстояние между документами и центрами их кластеров:

$$e(D, C) = \sum_{j=1}^k \sum_{i: d_i \in c_j} \| \vec{d}_i - \vec{\mu}_j \|^2 \quad (1)$$

где μ_j - центроид кластера C_j .

Обычно исходные центры кластеров выбираются случайным образом. Затем каждый документ присваивается тому кластеру, чей центр является наиболее близким документу, и выполняется повторное вычисление центра каждого кластера как центроида, или среднего своих членов. Такое перемещение документов и повторное вычисление центроидов кластеров продолжается до тех пор, пока не будет достигнуто условие остановки. Условием остановки может служить: (а) достигнуто пороговое число итераций, (б) центроиды кластеров больше не изменяются и (в) достигнуто пороговое значение ошибки кластеризации.

3.2.3. Нечеткие алгоритмы классификации – FCM

Был предложен как решение проблемы мягкой кластеризации, то есть присвоения каждого документа более чем одному кластеру. Как и его чёткий вариант, k-means, данный алгоритм, начиная с некоторого начального разбиения данных, итеративно минимизирует целевую функцию, которой является следующее выражение:

$$e_m(D, C) = \sum_{i=1}^{|D|} \sum_{j=1}^{|C|} u_{ij}^m \| \vec{d}_i - \vec{\mu}_j \|^2 \quad (2)$$

где m – степень нечеткости, $1 < m < \infty$, u_{ij} - степень принадлежности i-го документа j-му кластеру.

3.2.4. Алгоритм минимального покрывающего дерева

Алгоритм минимального покрывающего дерева сначала строит на графе минимальное покрывающее дерево, а затем последовательно удаляет ребра с наибольшим весом. На рисунке изображено минимальное покрывающее дерево, полученное для девяти объектов.

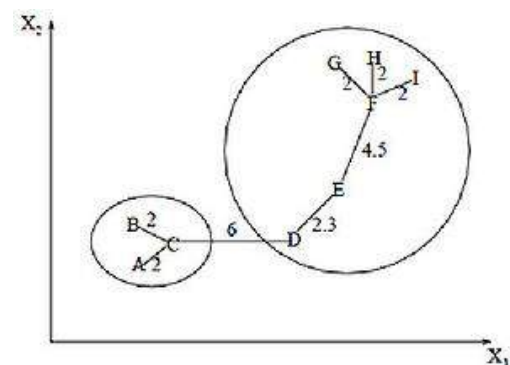


Рисунок 3 – Иллюстрация алгоритма MST

Путём удаления связи, помеченной CD, с длиной равной 6 единицам (ребро с максимальным расстоянием), получаем два кластера: {A, B, C} и {D, E, F, G, H, I}. Второй кластер в дальнейшем может быть разделён ещё на два кластера путём удаления ребра EF, которое имеет длину, равную 4,5 единицам [Pera et al., 2012].

3.2.5. Применение нейронных сетей

Алгоритм самоорганизующихся карт (SOM, Self Organizing Maps) был предложен как решение проблемы визуализации и кластеризации данных. Визуализация данных осуществляется путём проецирования многомерного пространства данных в двумерное пространство – карту данных. Такая карта, построенная для массива полнотекстовых документов, может служить как поисковый механизм, альтернативный поиску по запросу, предлагающий обзор/навигацию по коллекции документов [Kiryakov, 2004].

Идея алгоритма заключается в том, чтобы обучить нейронную сеть без учителя. Сеть состоит из некоторого числа нейронов, упорядоченных по узлам двумерной сетки. Каждый нейрон имеет

координаты в исходном $|\tau|$ - мерном пространстве документов и двумерном пространстве карты.

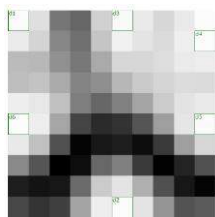


Рисунок 4 – Самоорганизующаяся карта из 6 документов

В процессе обучения нейроны упорядочиваются в пространстве документов так, чтобы наилучшим образом описать входной массив документов. Этот процесс является итерационным, на каждой итерации t :

- случайным образом выбирают из входного массива $d_i \in D$;
- находят нейрон-победитель $m_c \in M$, ближайший к документу d_i ;
- корректируют веса нейрона-победителя, его соседей: $m_i(t+1) = m_i(t) + h_{ci}(t)[d_i - m_i(t)]$.

3.3. Сравнение методов кластеризации применительно к обработке новостных текстов

При выборе оптимального алгоритма кластеризации новостного потока необходимо учитывать его следующие особенности: постоянно растущая коллекция документов, одна и та же статья может отражать несколько сюжетов, новости имеют определенную структуру текста, разные части документа должны иметь различный вес при нахождении близости, сюжеты и документы могут иметь перекрестные ссылки друг на друга.

В соответствии с вышесказанным, проведем сравнение рассмотренных нами алгоритмов кластеризации новостного потока.

- Агломеративный: иерархический, четкий, не требующий задания числа кластеров;
- K-means: плоский, четкий, требующий задания числа кластеров;
- FCM: плоский, нечеткий, требующий задания числа кластеров;
- MST: плоский, четкий, не требующий задания числа кластеров.

Для работы с новостными текстами желательно, чтобы алгоритм был плоским, нечетким, инкрементальным. Поэтому более перспективными методами для данной задачи видится применение алгоритма FCM или нейронных сетей.

Далее рассмотрим аспекты семантического анализа новостного текста.

4. Анализа новостного текста после кластеризации новостного потока

Новостные агрегаторы представляют собой сложные программно-аппаратные комплексы,

решающие широкий круг задач: кластеризацию, ранжирование документов внутри кластера, обзорное реферирование, выявление ключевых лиц, тематическую классификацию и поиск по новостям и т.д. Семантический анализ является основной составляющей вышеперечисленных задач, его можно разбить на несколько этапов, рассмотренных далее.

4.1. Предварительная обработка текста новости

Графематический анализ представляет собой начальный этап обработки текста, в ходе которого вырабатывается информация, необходимая для дальнейшей обработки морфологическим и синтаксическим анализаторами. В задаче графематического анализа входит внутреннее представление структуры новости: $T = \langle P, S, W \rangle$, где P – абзацы, S – предложения, W – слова. При этом необходимо корректно выделить заголовки и первое предложение абзаца, содержащее основные факты статьи.

Следующим этапом является морфологический анализ, цель которого – построение морфологической интерпретации слов входного текста. Все методы можно разделить на словарные и вероятностно-статистические (без использования словаря). Недостатками вторых являются большой объем лексиконов, плохая работа на малой выборке, отсутствие точных лингвистических методов. Словарный же метод основан на подключении словаря, тезауруса, дает максимально полный анализ словоформы.

Поэтому для данного блока целесообразно использовать морфологические библиотеки, например, Lemmatizer, FreeLing, NLTK, MCR, tokenizer [Михайлов и др., 2009].

4.2. Задача синтаксического анализа

Синтаксический анализ рассматривается как задача построения дерева зависимостей предложения. В ходе его проведения, происходит выделение синтаксических конструкций, определение связности и подчинения фрагментов [Grune, 2012].

Приведем обзор основных инструментов синтаксического анализа, которые возможно использовать в своем проекте:

Таблица 1 – Модули синтаксического анализа

Название	Методы	Языки
AOT	грамматика HPSG	русский, английский, немецкий
MaltParser	машинное обучение	русский, английский
Link Grammar Parser	грамматика связей	русский, английский
NLTK	машинное обучение	английский
Solarix	правила	русский, английский

4.3. Поиск ключевых слов, построение аннотации

Для новостных текстов уже существуют программные модули для извлечения ключевых сущностей. К таковым относится PullEnti, полностью написанный на C#.NET.

Поэтому был разработан алгоритм поиска ключевых слов [Солошенко и др., 2014a], сочетающий выделение именованных сущностей из текста новости (на основе результатов морфоанализа и подключаемого модуля PullEnti), подсчет веса слова с учетом частоты его встречаемости (рисунок 5).



Рисунок 5 – Алгоритм поиска ключевых фраз

Пороговое значение для признания слова ключевым – это значение относительной частоты встречаемости слова-кандидата в ключевые слова, с индексом, равным $(0,2 \times \text{количество сущностей})$. Такое значение вычислено экспериментально на выборке из 100 текстов.

В итоге структуру совокупности знаний S текста новости можно определить следующим образом: $S = \{M, F\}$, где M – множество всех понятий данной совокупности знаний, F – отношение «смысловая связь». В качестве формальной модели структуры знаний можно использовать семантическую сеть, определяемую в виде ориентированного графа $G = (E, V)$, где E – множество вершин, поставленное во взаимно однозначное соответствие с множеством понятий; V – множество ориентированных дуг; дуга выходит из вершины, соответствующей основному понятию A , и входит в вершину, соответствующую понятию, которое сочетается по смыслу с понятием A . [Дмитриев и др., 2013; Машечкин и др., 2011]. Таким образом, содержание новости можно представить наглядно в виде ключевых понятий и связей между ними, либо в виде так называемой mind map или интеллектуальной карты (рисунок 6).



Рисунок 6 – Интеллектуальная карта текста новости

Подсчет веса предложений при построении аннотации осуществляется в зависимости от его нахождения в тексте новости и рассчитывается по формуле:

$$W_s = N(kw) \cdot Rf(kw) \cdot \text{Вес параграфа} \cdot k \quad (3)$$

W_s здесь – вес предложения; $N(kw)$ – количество вхождений ключевого слова в предложение; $Rf(kw)$ – относительная частота ключевого слова; вес параграфа – относительный вес параграфа в тексте, равен 0.35 для первого параграфа (лид), 0.2 для второго, 0.1 для остальных (контекст); k – коэффициент значимости предложения внутри параграфа. Для первого предложения в абзаце равен 1, для остальных – 0.8 [Солошенко и др., 2014].

В итоговую аннотацию включаются предложения с наибольшим весом, в зависимости от заданного коэффициента сжатия.

5. Основные результаты

Для проверки теоретических рассуждений была реализована система, основанная на принципе многокомпонентности программного обеспечения [Заболеева-Зотова и др., 2013].

Был также проведен эксперимент [Солошенко и др., 2014b], цель которого – доказать, что за счет автоматизации обработки статей новостного потока повысилась эффективность обработки новостных интернет-статей, то есть снизилось время на обработку и повысилось качество получаемого результата.

Были получены следующие результаты:

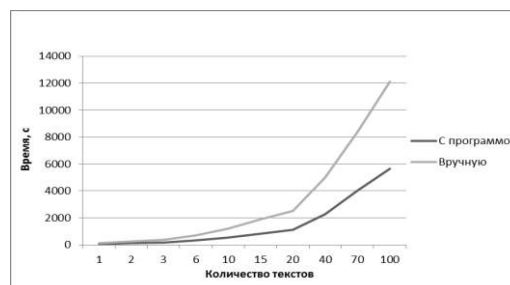


Рисунок 7 – Зависимость времени обработки от количества текстов

Время обработки уменьшилось как минимум в два раза. При этом время с помощью программы учитывает не только непосредственно время

составления аннотации, но и время, необходимое для окончательной корректировки текстов.

Качество аннотации можно оценить по следующим критериям: сохранение ключевых фактов, связность новостной статьи, сохранение синтаксической структуры текста после удаления незначащих частей. Каждый из названных критериев оценивался экспертами по шкале от 0 до 10 баллов, затем для оценки качества полученной аннотации находилось среднее арифметическое трех показателей для каждого текста.

В результате качество обработанных новостей остается на том же уровне, как и при анализе текста человеком.

Заключение

Итак, анализ новостных текстов включает в себя задачу кластеризации и последующую комплексную обработку статей. Был проделан анализ новостного потока, выделены особенности документов. Проанализированы методы кластеризации для новостных статей, предложено наиболее подходящее для нашей задачи решение - алгоритм FCM или нейронные сети. Однако необходимо заметить, что оно не является единственно верным. Реализована часть системы онлайн агрегации новостей из интернет-источников и проведены исследования эффективности ее работы.

Работа частично поддержана Российским фондом фундаментальных исследований (проекты 13-07-00351, 13-07-97042, 14-07-97017, 15-07-07519).

Библиографический список

- [Большакова и др., 2011] Автоматическая обработка текстов на естественном языке и компьютерная лингвистика : учеб. пособие / Е.И. Большакова [и др.]. – М.: МИЭМ, 2011. – 272 с.
- [Добров, 2010] Добров Б. В. Исследование качества базовых методов кластеризации новостного потока в суточном временном окне // Труды 12й Всероссийской научной конференции «Электронные библиотеки: перспективные методы и технологии, электронные коллекции» RCDL'2010, Казань, Россия, 2010. – С. 287–295.
- [Дмитриев и др., 2013] Automatic identification of time and space categories in the natural language text / Дмитриев А.С., Заболеева-Зотова А.В., Орлова Ю.А., Розалиев В.Л. // Applied Computing 2013 : proceedings of the IADIS International Conference (Fort Worth, Texas, USA, October 23-25, 2013) / IADIS (International Association for Development of the Information Society), UNT (University of North Texas). – [Fort Worth (Texas, USA)], 2013. – P. 187-190.
- [Заболеева-Зотова и др., 2013] Formalization of initial stage of designing multi-component software / Заболеева-Зотова А.В., Орлова Ю.А., Розалиев В.Л., Фоменков С.А., Петровский А.Б. // Multi Conference on Computer Science and Information Systems 2013 (Prague, Czech Republic, July 23-26, 2013) : Proceedings of the IADIS International Conference Intelligent Systems and Agents 2013 / IADIS (International Association for Development of the Information Society). – [Prague], 2013. – P. 107-111.
- [Машечкин и др., 2011] Машечкин И. В. Латентно-семантический анализ в задаче автоматического аннотирования / И. В. Машечкин, М. И. Петровский // Программирование. – 2011. – Т. 37, № 6. – 67-77.
- [Михайлов и др., 2009] Михайлов Д. В. Морфология и синтаксис в задаче семантической кластеризации / Д. В.

Михайлов, Г. М. Емельянов // Математические методы распознавания образов (ММРО-14), Владимирская область, Суздаль, 21-26 сентября 2009 г. – С. 1-4.

[Солошенко и др., 2014a] Thematic Clustering Methods Applied to News Texts Analysis / Солошенко А.Н., Орлова Ю.А., Розалиев В.Л., Заболеева-Зотова А.В. // Knowledge-Based Software Engineering : Proceedings of 11th Joint Conference, JCKBSE 2014 (Volgograd, Russia, September 17-20, 2014) / ed. by A. Kravets, M. Shcherbakov, M. Kultsova, Tadashi Iijima ; Volgograd State Technical University [etc.]. – [Б/М] : Springer International Publishing, 2014. – P. 294-310. – (Series: Communications in Computer and Information Science ; Vol. 466).

[Солошенко и др., 2014b] Автоматизация семантического анализа новостных Интернет-текстов. / Солошенко А.Н., Розалиев В.Л., Орлова, Ю.А. // Открытые семантические технологии проектирования интеллектуальных систем OSTIS-2014 : матер. IV междунар. науч.-техн. конф. (Минск, 20-22 февр. 2014 г.). Белорус. гос. ун-т информатики и радиоэлектроники, Администрация Парка высоких технологий. Минск, с. 435-438.

[Bandyopadhyay et al., 2013] Bandyopadhyay, S., Saha, S.: Unsupervised Classification. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013.

[Grune, 2012] Grune D. Tokens to Syntax Tree – Syntax Analysis. New York, NY: Springer New York. – 2012

[Kiryakov, 2004] Kiryakov A. Semantic annotation, indexing, and retrieval // Web Semantics: Science, Services and Agents on the World Wide Web. – 2004. Т. 2. № 1. – С. 49–79.

[Pera et al., 2012] Pera, M.S., Ng, Y.-K.D: Using maximal spanning trees and word similarity to generate hierarchical clusters of non-redundant RSS news articles. In: J. Intell. Inf. Syst, vol 39, pp. 513–534

THEMATIC CLUSTERING METHODS APPLIED TO NEWS ARTICLES ANALYSIS

Soloshenko A.N., Orlova Yu.A. *,

Zaboleeva-Zotova A.V. **

*Volgograd State Technical University,
Volgograd, Russia

nastyasolan@gmail.com

yulia.orlova@gmail.com

** Russian Foundation for Basic Research,
Moscow, Russian Federation

zabzot@gmail.com

This paper is devoted to a problem of partition documents from the news flow into groups, where each group contains documents that are similar to each other. The existing clustering algorithms such as k-means, minimum spanning tree and etc. are considered and analyzed. It is shown which of these algorithms give the best results working with news texts. This paper also presents a methodic of comprehensive news texts analysis based on a combination of statistical algorithms for keywords extracting and algorithms forming the semantic coherence of text blocks. Particular attention is paid to the structural features of the news texts.

Keywords: thematic clustering, clustering algorithms, news articles, document representation.



УДК 004.588

РАЗРАБОТКА ИНТЕЛЛЕКТУАЛЬНОГО ОБУЧАЮЩЕГО ИНТЕРФЕЙСА ДЛЯ ДЕТЕЙ С ИСПОЛЬЗОВАНИЕМ СЕНСОРА LEAP MOTION

Розалиев В.Л. *, Вяхирев А.А. *, Заболеева-Зотова А.В.**

* *Волгоградский государственный технический университет, г. Волгоград, Россия*

vladimir.rozaliiev@gmail.com

** *Российский фонд фундаментальных исследований, г. Москва, Россия*

zabzot@gmail.com

Данная статья посвящена разработке игрового приложения, использующего сенсор Leap Motion. Программа позволяет проводить обучение детей младшего возраста выбору цвета, размера объектов и оценке формы. Обучение строится на основе оценки эмоционального поведения ребенка и эффективно адаптируется под его состояние. В статье рассматриваются первые шаги разработки приложения, выполненные задачи и результаты.

Ключевые слова: Leap Motion; Unity3D; распознавание движений рук человека; разработка игровых приложений; эмоциональное состояние.

Введение

Обычная мышка на сегодняшний день остается самым популярным средством ввода информации, однако все большей популярностью пользуются сенсорные и жестовые средства управления. Многие специалисты продолжают разработки для внедрения альтернативных средств взаимодействия человека с компьютерной техникой. Таким образом, на сегодняшний день наиболее актуальна проблема усовершенствования алгоритмов и методов человеко-компьютерного взаимодействия, а так же поиска области приложения данной технологии [Бобков и др., 2011].

Общей идеей проекта является создание приложений способных дистанционно отслеживать жесты человека, интерпретировать их, отслеживать эмоциональное состояние и адаптироваться под него [Розалиев и др., 2013; Розалиев и др., 2014]. Первым этапом стало создание игрового приложения для детей младшего возраста, в котором ребенку предлагается обучиться выбору цвета, размера и формы некоторых объектов. Таким образом, целью текущего этапа работы являлась разработка игрового приложения с возможностью управления жестами, определяемыми с помощью сенсора Leap Motion.

1 Выбор контроллера для отслеживания движений

Leap Motion – сенсор, представляет собой небольшое устройство, по размерам не сильно отличающиеся от стандартного USB-флеш-накопителя. Его габариты всего 79 x 30 x 11 мм, что позволяет удобно его позиционировать.

Leap Motion может обнаружить и отследить движения рук, пальцев и различных инструментов (например, карандаш, ручку, находящийся в руках). Рабочая зона прибора представляет собой перевернутую пирамиду, вершина которой располагается в центре устройства. Эффективный диапазон работы сенсора колеблется от 25 до 600 миллиметров над устройством.

Данный сенсор использует Декартову правую систему координат «Рисунок 1».

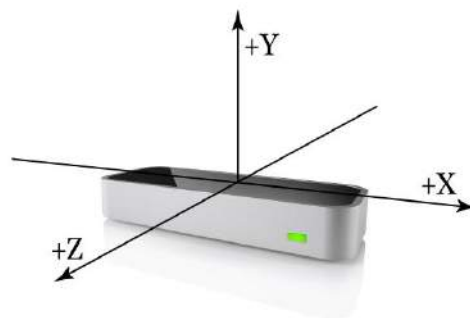


Рисунок 1 – Система координат сенсора Leap Motion

Отслеживаемые данные, Leap Motion предоставляет в виде набора кадров. Каждый кадр содержит списки основных данных рук, пальцев, инструментов, а также опознанных жестов и информации о движении по сцене. При обнаружении, каких либо элементов руки, ему присваивается уникальный идентификатор, который будет существовать пока рука находится в поле зрения сенсора.

Модель руки представляется сенсором в виде её координат в пространстве, характеристиках и движении, а также списками пальцев и инструментов связанных с ней.

На основе обзора аналогов контроллеров была составлена таблица 1 с критериями их возможностей (1-детектирование конечностей, 2-выделение кисти, 3-оценка расстояния до объекта, 4-детектирование угла наклона пальцев).

Таблица 1 – Сравнение аналогов контроллеров

	1	2	3	4
Leap Motion	+	+	+	+
Kinect	+	+	+	-
Mycestro	-	-	+	-
WAVI Xtion	-	+	+	-
MYO	-	+	-	-
Wii Remote	+	+	+	-
Creative Senz3D	-	+	+	-

В результате анализа рассмотренных контроллеров был выявлен их общий недостаток, а именно, невозможность детектирования угла наклона пальцев. Сенсор Leap Motion единственный обладает технической возможностью определения угла наклона пальцев. Поэтому именно его мы выбрали для разработки игрового приложения.

Затем был произведён анализ официальных существующих приложений для Leap Motion. Было установлено, что почти все приложения, имеют один общий недостаток: быстрое накопление усталости рук при сравнительно непродолжительном времени работы. К сожалению, этот недостаток пока не преодолен в нашей работе.

2 Выбор Unity3D и разработка объемных моделей

Был произведён анализ игровых движков для создания приложений. Контроллер Leap Motion не располагает собственной платформой для создания 3D приложений. Однако в программном обеспечении сенсора присутствует дополнительная библиотека для использования данных отслеживания сенсора с инструментом Unity3D. Данный критерий оказался ключевым при выборе платформы разработки.

Стоит отметить, Unity3D – это мультиплатформенный инструмент для разработки двух- и трёхмерных приложений и игр. Обладает обширными возможностями в создании игр и постоянно развивается. [Blackman, 2011] Несмотря на закрытую структуру, имеет массу преимуществ, в сравнении с другими средствами разработки игр.

В игре нам были необходимы объемные модели. Для создания объектов и требуемых для них форм была использована система Autodesk 3ds Max - полнофункциональная профессиональная программная система для создания и редактирования трёхмерной графики и анимации [Мэрдок, 2009]. Созданные модели объектов экспортировались в платформу Unity3D.

3 Разработка математического и программного обеспечения

На следующем шаге были разработаны алгоритмы для работы Leap Motion в среде Unity3D, а именно:

- 1) алгоритм распознавания вращения руки;
- 2) алгоритм распознавания перемещения руки;
- 3) алгоритм распознавания захвата объекта рукой;
- 4) алгоритм распознавания вращения объекта при помощи 2-х рук;
- 5) алгоритм перемещения объектов по 3D сцене.

Языком разработки был выбран C# [Api Overview, 2015]. C# – чрезвычайно мощный язык, содержащий средства создания эффективных программ практически любого назначения, от низкоуровневых утилит и драйверов до сложных программных комплексов самого различного назначения. Leap Motion имеет библиотеку для работы с ним, используя данный язык, как и в свою очередь C# является одним из поддерживаемых языков программирования на платформе Unity3D [Ламот, 2006].

Далее была спроектирована и реализована многокомпонентная программа [Заболеева-Зотова и др., 2013], использующая Leap Motion и Unity3D. Архитектура приложения приведена на Рисунке 2.

Представим краткое описание модулей программы:

- 1) Контроллер Leap Motion. Вход: движения рук человека. Выход: наборы кадров.

Является главным источником входных данных. Принимает на вход различные движения, жесты рук пользователя и передаёт их в виде набора кадров для модуля обработки информации.

- 2) Модуль обработки данных. Вход: наборы кадров и начальные параметры приложения. Выход: данные о каждом кадре.

Принимает наборы кадров от сенсора и получает из них требуемую информацию для обработки. Также принимает начальные параметры программы

от пользователя. Это необходимо для начала анализа движений рук.

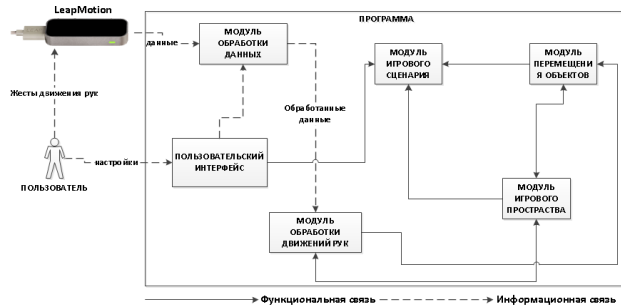


Рисунок 2 – Архитектура игрового приложения

3) Пользовательский интерфейс. Вход: задание начальных параметров приложения. Выход: начало исполнения сценария и обработки информации от сенсора.

С помощью интерфейса пользователь настраивает начальные параметры приложения (индивидуальные настройки разрешения, выбор режима работы приложения и т.д.) и запускает программу на исполнение сценария.

4) Модуль обработки движений рук. Вход: данные о кадрах от сенсора. Выход: передача информации сцене с объектами.

Данные о кадрах сенсора используются для представления рук пользователя на сцене и взаимодействия с объектами сцены.

5) Модуль игрового сценария. Вход: информация об изменениях в прогрессе сценария. Выход: результат выполнения сценария.

Данный модуль отвечает за все изменения сценария приложения. Он принимает любые результаты действий на сцене, перемещения объектов, которые могут привести к выполнению сценария. От данного модуля будет зависеть, как завершится сценарий.

6) Модуль перемещения объектов. Вход: изменения координат положения объектов. Выход: изменение положения объектов.

Данный модуль принимает данные от движений рук и текущее положение объектов на сцене. В зависимости от результатов действий на сцене могут измениться положения объектов.

7) Модуль игрового пространства. Вход: информация о движениях рук и перемещения объектов. Выход: изменение состояния объектов.

Модуль игрового пространства отвечает за сцену. Любые перемещения объектов фиксируются модулем. Также одна из его подзадач – просчёт физики объектов их столкновение, взаимодействие.

Пользователь является источником данных для сенсора, как и для системы в целом.

4 Настройка сцены

Далее была проведена настройка сцены, включающая в себя:

- 1) Добавление объектов и форм на сцену.
- 2) Добавление окружения.
- 3) Добавление физики объектов.
- 4) Добавление цветов и раскраска объектов.

5) Добавление скриптов взаимодействия Leap Motion и Unity3D.

Одним из наиболее важных этапов настройки сцены является добавление физики объектам. После создания рабочей сцены и вставки формы для объектов требуется создание физики объектов. Для примитивов, не имеющих вогнутых поверхностей или рельефа, создание физики происходит добавлением элементарного свойства объекту, в настройках Unity3D.

Форма, в которую вставляется объект, по определению, уже имеет вогнутость. Для этого создаётся набор, состоящий из группы примитивов. Непосредственно, на примере формы для куба, создаётся серия прямоугольников, заполняющих пространство вокруг формы.

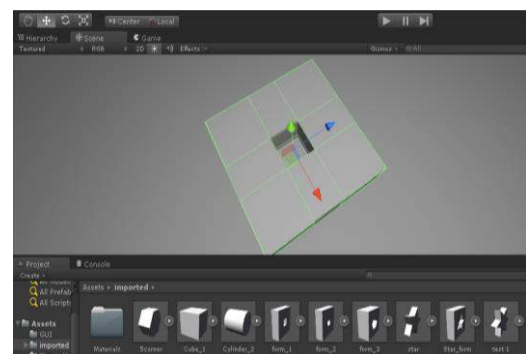


Рисунок 3 – Создание физики для формы куба

5 Сценарий игрового приложения

Следующим шагом был разработан сценарий для приложения. Общая цель игры заключается в последовательном прохождении уровней игры, используя различные движения рук и жесты.

Сценарий приложения будет направлен в сторону развивающих игр для детей. Его можно разбить на две части: первая – распознавание базовых форм объектов, вторая – распознавание цветов.

В первом случае задача ребёнка, перемещая различные объекты, вставить их в основание специальных форм с вырезами, практически совпадающими с размерами объектов. Уровень считается пройденным, если объект зафиксировался в подходящей для него форме. Игра будет завершена только в том случае, если все уровни приложения будут пройдены.

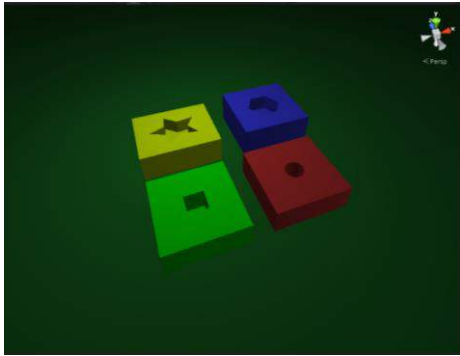


Рисунок 4 – Иллюстрация форм объектов

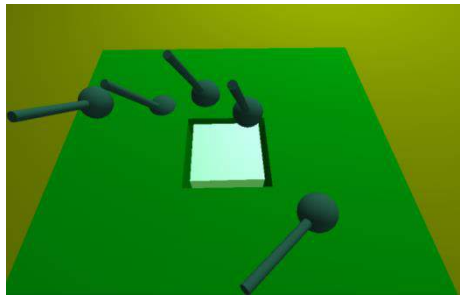


Рисунок 5 – Иллюстрация первой части сценария

Вторая часть тесно связана с первой. Требуется вставить объект в форму, учитывая его цвет.

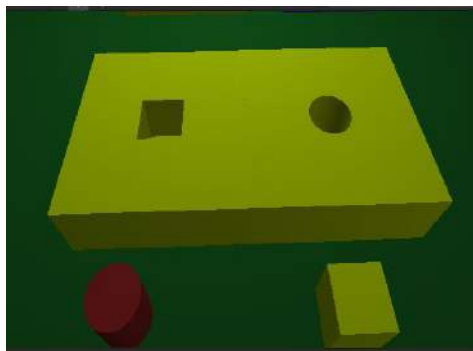


Рисунок 6 – Иллюстрация второй части сценария

Заключение

В заключении, стоит отметить основные результаты текущего этапа работы и выводы по полученным результатам, а также перспективы развития. Произведен анализ контроллеров и выделен сенсор Leap Motion для применения в игровом приложении. Разработаны объемные модели (объекты в игре). Разработан сценарий игры. Спроектировано и разработано игровое приложение для жестового взаимодействия с детьми. На текущем этапе не происходит отслеживание эмоционального состояния, что является перспективой работы.

Работа частично поддержана Российским фондом фундаментальных исследований (проекты 13-07-00459, 13-07-97042, 15-07-06322).

Библиографический список

[Бобков и др., 2011] Развитие системы автоматизированного определения эмоций и возможные сферы применения / Бобков

А.С., Заболеева-Зотова А.В., Орлова Ю.А., Розалиев В.Л. // Открытое образование. - 2011. - № 2. - С. 59-62.

[Мэрдок, 2009] 3ds Max 2009. Библия пользователя / К. Мэрдок. – М. : Вильямс, 2009. – 1328 с.

[Ламот, 2006] Программирование трехмерных игр для Windows. Советы профессионала по трехмерной графике и рендеризации / А. Ламот. – М.: Вильямс, 2006. – 1424 с.

[Заболеева-Зотова и др., 2013] Formalization of initial stage of designing multi-component software / Заболеева-Зотова А.В., Орлова Ю.А., Розалиев В.Л., Фоменков С.А., Петровский А.Б. // Multi Conference on Computer Science and Information Systems 2013 (Prague, Czech Republic, July 23-26, 2013) : Proceedings of the IADIS International Conference Intelligent Systems and Agents 2013 / IADIS (International Association for Development of the Information Society). – [Prague], 2013. – P. 107-111.

[Розалиев и др., 2013] Розалиев, В.Л. Methods and Models for Identifying Human Emotions by Recognition Gestures and Motion / Розалиев В.Л., Заболеева-Зотова А.В. // The 2013 2nd International Symposium on Computer, Communication, Control and Automation 3CA 2013, December 1-2, 2013, Singapore : Papers. – [Amsterdam – Beijing – Paris] : Atlantis Press, 2013. – P. 67-71.

[Розалиев и др., 2014] Розалиев, В.Л. Applying the automated system for the determination of emotions in the education's tasks for people with disabilities / Розалиев В.Л., Орлова Ю.А. // Innovation Information Technologie : mater. of the 3rd Int. scien.-pract. conf. (Prague, April 21-25, 2014). Part 2 / МИЭМ ВШЭ, Рос. центр науки и культуры в Праге. - М., 2014. - С. 446-452.

[Api Overview, 2015] Api Overview [Электронный ресурс]. – Режим доступа: https://developer.leapmotion.com/documentation/csharp/devguide/Leap_Overview.html

[Blackman, 2011] Beginning 3D Game Development with Unity: All-in-one, multi-platform game engine / Blackman S. – USA: Apress, 2011. – 970 с

THE DEVELOPMENT OF INTELLIGENT LEARNING INTERFACE FOR CHILDREN USING SENSOR LEAP MOTION

Rozaliev V.L. *, Vyakhirev A.A. *,

Zaboleeva-Zotova A.V. **

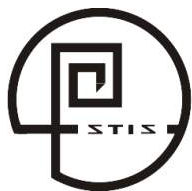
*Volgograd State Technical University,
Volgograd, Russia

vladimir.rozaliev@gmail.com

** Russian Foundation for Basic Research,
Moscow, Russian Federation

zabzot@gmail.com

This article is devoted to the development of gaming applications that use sensor Leap Motion. The program allows you to train young children to color selection, size of objects and the evaluation form. The training is based on the evaluation of the emotional behavior of the child and effectively adapts to his condition. The article discusses the first steps of the development application, tasks performed and the results.



УДК 004.853

АВТОМАТИЗИРОВАННОЕ ВЫДЕЛЕНИЕ КИСТЕЙ РУК ЧЕЛОВЕКА ДЛЯ РАСПОЗНАВАНИЯ ЖЕСТОВОЙ РЕЧИ

Розалиев В.Л., Агафонов Г.В., Кириченко М.И.

Волгоградский государственный технический университет, г. Волгоград, Россия

vladimir.rozaliev@gmail.com

Данная статья посвящена проблеме детектирования и анализа кисти с помощью камеры глубины. Особое внимание уделено анализу и трекингу кисти. В работе скомбинированы алгоритмы поиска, трекинга и анализа кистей рук человека. Рассмотрены существующие системы распознавания жестов.

Ключевые слова: язык жестов; детектирование; Kinect; depth изображение.

Введение

Важной проблемой является создание современных информационных технологий, в том числе методов, с помощью которых можно создавать новые компьютерные системы обучения и коммуникации для людей с нарушениями слуха. В мире насчитывается более 1 процента людей с заболеваниями, проявляющимися нарушением звуковосприятия, по России же их количество превышает 1000000 человек. При создании таких систем необходимо разрабатывать методы распознавания жестов, их интерпретации и перевода в текст на естественном языке.

1. Жестовая речь

Речь жестовая (от франц. *gestes* – деяния) – способ межличностного общения людей, лишенных слуха, при помощи программы жестов, характеризующейся своеобразными лексическими и грамматическими закономерностями. Закономерности жестовой речи обусловлены выраженным своеобразием ее основной семантической единицы – жеста, а также его функциональным назначением (использованием в сфере непринужденного общения). Жестовая речь используется как вспомогательное средство (наряду с основным – словесной речью) в процессе обучения и воспитания детей с недостатками слуха. В сфере официального общения (собраний, перевод лекций) применяется калькирующая жестовая речь, когда жесты последовательно используются для воспроизведения слов. В современном обществе полезно знание интернациональных языков, а жестовый язык является одним из самых понятных во всем мире.

Жесты различают статические и динамические жестов. Под статическим жестом подразумевается положение руки в пространстве в отсутствие каких-либо движений. Напротив, динамический жест характеризуется последовательным движением руки в пространстве из начальной точки в конечную за фиксированный промежуток времени.

В наши дни существует специальная наука – семиотика, занимающаяся изучением знаков и знаковых систем (главным образом в приложении к естественным и искусственным языкам). Семиотика разделяет знаки на три группы: иконические, индексные и символические. К первой группе относят знаки, похожие на обозначаемый предмет (фотографии, географические карты). Во вторую, индексную, группу включают знаки, либо обозначающие часть предмета, либо такие, где изображение выступает как знак предмета в целом. Наконец, к третьей, символической, группе относят изображения, не имеющие ничего или почти ничего общего с обозначаемым предметом.

Приведем примеры некоторых распространенных жестов и определим их положение в соответствующих семиотических группах. Слово «дом» изображают соприкасающимися концами пальцев обеих рук, представляющих собой как бы двускатную крышу. Такое изображение следует отнести ко второй, индексной, группе. Жест статический. Также статическими являются жесты «женщина» — вертикальная ладонь у одной из щек и «мужчина» — согнутая кисть руки у лба. Первый из этих жестов чистый символ (третья группа), второй иногда относят к индексной группе, если понимать его, как прикосновение пальцев к козырьку фуражки или кепки. Соответственно жестам «женщина» и

«мужчина» динамические жесты «мама» и «папа» производят в первом случае движением вертикальной ладони от одной щеки к другой, а во втором — аналогичным перемещением горизонтальной ладони от лба к подбородку (оба жеста, разумеется, чистые символы). Интернациональный жест «мы вас (тебя) любим» представляется условным символом: поднятые вверх большой, указательный пальцы и мизинец одной из рук. Здесь одно движение заменяет три слова.

2. Существующие системы распознавания жестов

Fingual - это система, способная преобразовывать жесты языка глухонемых в символы на компьютере. Она состоит из компьютера со специализированным программным обеспечением и особой информационной перчатки с постоянными магнитами, закрепленными на кончике каждого пальца. Изменения магнитного поля, возникающие при жестикуляции, регистрируются магнитными датчиками и отправляются на обработку в компьютер. Полученная информация затем анализируется и преобразовывается в символы. Данные о характеристиках магнитного поля отдельных жестов Fingual использует в качестве шаблонов. Это позволяет ускорить процесс распознавания. Основным минус данной системы в том, что набор распознаваемых жестов сильно ограничен, так как перчатка воспринимает только статические жесты. То есть данная разработка не позволяет распознавать язык жестов в целом, а только отдельные символы – дактильный алфавит. Также, несмотря на то, что компоненты перчатки достаточно дешёвы – вряд ли придется ожидать, что уже в ближайшее время она поступит в производство и станет общедоступной.

Исследователи из китайского отделения Microsoft Research применили технологию распознавания жестов Kinect для перевода жестового языка в текст и живую речь. Их разработка поддерживает, как американский, так и китайский варианты жестового языка для глухих и может переводить и жесты в слова, и слова в жесты, которые демонстрирует на экране виртуальный персонаж. С февраля 2012 года Microsoft Research работает над этим проектом совместно с Академией наук Китая и Пекинским объединенным университетом. Сейчас система умеет распознавать 370 наиболее часто употребляемых слов китайского и американского жестовых языков.

3. Перевод русского жестового языка в текст

На основе рассмотренных систем и методов распознавания жестов предлагается реализовать систему способную переводить русский жестовый язык в текст. Для отслеживания движений (skeleton

tracking) был выбран Microsoft Kinect [Соболев, 2012]. Kinect – бесконтактный сенсорный игровой контроллер, первоначально созданный для консоли Xbox 360, и значительно позднее для персональных компьютеров под управлением ОС Windows. Одним из важных аспектов камеры – диапазон рабочих расстояний, на которых сенсор способен различать объекты.

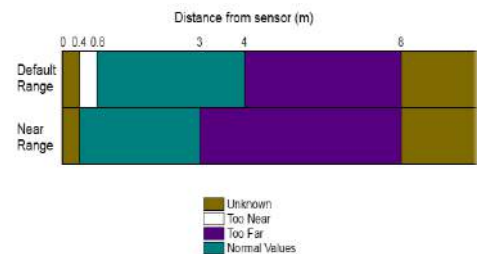


Рисунок 1 – Диапазон рабочих расстояний Kinect

На рисунке 1 представлены два режима работы. Лучшие результаты будут получены при нахождении объектов в рамках второго варианта (Near Range). По skeleton-данным можно получить информацию о координатах (положении) руки человека.

Проектируемая система должна включать в себя следующие подсистемы и предусматривать принцип многомодульности [Заболеева-Зотова и др., 2013а]:

- подсистема работы с пользователем;
- подсистема обработки изображения и вычисления параметров векторной модели;
- подсистема распознавания статического жеста.

Назначение подсистемы работы с пользователем – предоставление пользователю удобного интерфейса работы с системой. Это включает в себя: вывод обработанного изображения на экран, возможность изменять настройки программы, формирование текста из распознанных жестов, вывод сформированного текста на экран, доступ к справочной информации. Также подсистема должна предоставлять возможность работы двум видам пользователей.

Назначение подсистемы обработки изображения и вычисления параметров векторной модели – сегментирование карты глубины с камеры Kinect, обработка сегментированного изображения и вычисление параметров векторной модели по обработанному изображению. 30 раз в секунду на вход подсистеме поступает информация с камеры Kinect. С такой же частотой подсистема обрабатывает входные данные и вычисляет параметры векторной модели кисти руки.

Назначение подсистемы распознавания жеста – отнести данные, получаемые из подсистемы обработки изображения и вычисления параметров векторной модели к одной из дактильных букв или знаку препинания, заложенных в системе. Далее она передаёт значение распознанного жеста в модуль формирования текста подсистемы работы с пользователем (рисунок 2).

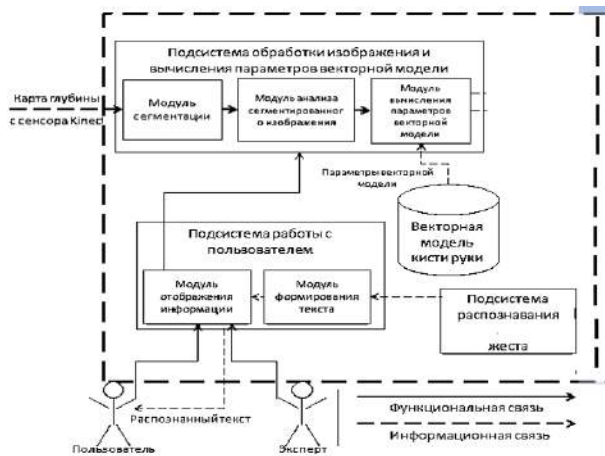


Рисунок 2 – Архитектура системы распознавания жестов

Для реализации распознавания жестов в системе используются визуальные методы распознавания [William, 2008]. Визуальные методы распознавания жестов являются наиболее активной и перспективной областью исследований. По сути, отслеживание голых рук основано на методах, работающих исключительно с визуальной информацией и не предъявляющих специальных требований к оснащению пользователя дополнительным оборудованием. Выделения контуров и силуэтов являются наиболее распространенными характеристиками, используемыми для идентификации позы руки. Пока эти сигналы являются общими и устойчивы к различным условиям освещения и, исходя из этого, требуют больших вычислительно сложных определяющих алгоритмов, которые ищут многомерные позиции пределов руки. В свою очередь визуальные методы распознавания жестов можно разделить на три большие категории. К первой относятся методы, которые основаны на восстановлении полной модели кисти с 27 степенями свободы по входному изображению. В данном подходе, рука представляется как соединение составляющих частей, и каждое состояние называется "степенью свободы". Степени свободы определяют позиции и ориентации реального объекта. Моделирование руки с учетом всех ее степеней свободы представляет собой сложную задачу [Christopher, 2013]. Поэтому используется упрощенная модель руки с 27 степенями свободы, ставшая неофициальным стандартом (рисунок 3).

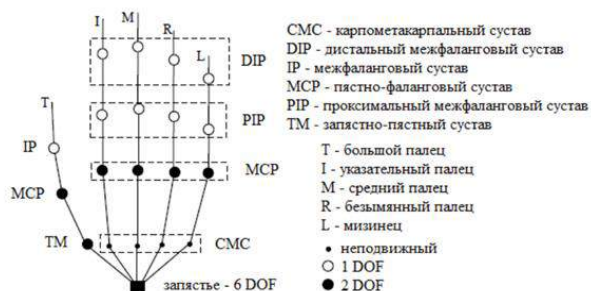


Рисунок 3 – Модель руки с 27 степенями свободы

Ко второй категории относятся статистические методы, которые вместо восстановления полной модели руки предлагают построение признакового описания входного изображения и дальнейшую классификацию жестов именно по этому описанию. Решающую роль в методах подобного рода играет выбор подходящих характеристических признаков. Так для статических поз руки, можно использовать геометрические признаки, такие как кончики пальцев, направление пальцев, контур руки, а также негеометрические признаки (цвет кожи, форма, текстура и др.). Однако эти признаки не всегда доступны или адекватны для распознавания из-за условий освещения. Важным моментом в этих подходах является классификатор, который предоставляет возможность разделить (по категориям) различные признаки или информацию, выделенные из "грубых данных", например, изображения или видеопотока.

К третьей категории относятся метрические методы распознавания жестов. Подобные методы предполагают построение некоторой метрики на множестве входных изображений и выполнение классификации за счет сравнения входного изображения с набором эталонов. Так, например, предлагается метрика, характеризующая степень сходства скелетов силуэтов ладони, и выполняется классификация жестов с помощью метода ближайшего соседа [Куракин, 2012].

4. Распознавание кисти

Чтобы реализовать распознавание кисти необходимо решить три подзадачи: детектирование, трекинг и анализ кисти.

На первом этапе детектирования рук на изображении необходимо сегментировать изображение, чтобы найти область интереса - кисть. Для этого используется алгоритм Виолы-Джонса. Обучение классификаторов идет медленно, но сам процесс поиска кисти проходит быстро, именно поэтому был выбран данный метод распознавания кисти рук на изображении. Виола-Джонс является одним из лучших по соотношению показателей эффективность распознавания/скорость работы. Алгоритм даже хорошо работает и распознает кисти под небольшим углом, примерно до 30 градусов. При угле наклона больше 30 градусов процент обнаружений резко падает.

В общем виде, задача обнаружения кисти человека на цифровом изображении выглядит так, имеется изображение, на котором есть искомые объекты [Гонсалес, 2005]. Оно представлено двумерной матрицей пикселей размером $w \times h$, в которой каждый пиксель имеет значение:

- от 0 до 255, если это черно-белое изображение;
- от 0 до 255^3 , если это цветное изображение (компоненты R, G, B).

В результате своей работы, алгоритм должен определить кисть и пометить ее – поиск осуществляется в активной области изображения прямоугольными признаками, с помощью которых и описывается найденная кисть: $rectangle_i = \{x, y, w, h, a\}$, где x, y – координаты центра i -го прямоугольника, w – ширина, h – высота, a – угол наклона прямоугольника к вертикальной оси изображения.

Иными словами, применительно к рисункам и фотографиям используется подход на основе сканирующего окна (scanning window): сканируется изображение окном поиска (так называемое, окно сканирования), а затем применяется классификатор к каждому положению. Система обучения и выбора наиболее значимых признаков полностью автоматизирована и не требует вмешательства человека, поэтому данный подход работает быстро.

Для решения проблемы проведения сложного обучения существует технология бустинга. Бустинг – комплекс методов, способствующих повышению точности аналитических моделей. Бустинг представляет собой жадный алгоритм построения композиции алгоритмов (greedy algorithm) – это алгоритм, который на каждом шагу делает локально наилучший выбор в надежде, что итоговое решение будет оптимальным. Т.е. при бустинге наряду с множествами X и Y вводится вспомогательное множество R , называемое пространством оценок. Далее рассматриваются алгоритмы, имеющие вид суперпозиции $a(x) = C(b(x))$, где функция $b: X \rightarrow R$ называется алгоритмическим оператором, функция $C: R \rightarrow Y$ – решающим правилом. В алгоритме Виолы-Джонса используется пороговое решающее правило, где, как правило, сначала строится оператор при нулевом значении, а затем подбирается значение оптимальное. Процесс последовательного обучения базовых алгоритмов применяется, пожалуй, чаще всего при построении композиций. Для критерия остановки используется критерий точности на обучающей выборке.

Алгоритм бустинга для детектирования кисти:

- 1) определение слабых классификаторов по прямоугольным признакам;
- 2) для каждого перемещения сканирующего окна вычисляется прямоугольный признак на каждом примере;
- 3) выбирается наиболее подходящий порог для каждого признака;
- 4) отбираются лучшие признаки и лучший подходящий порог;
- 5) перевзвешивается выборка.

Каскадная модель сильных классификаторов – это по сути то же дерево принятия решений, где каждый узел дерева построен таким образом, чтобы детектировать почти все интересующие образы и отклонять регионы, не являющиеся образами. Помимо этого, узлы дерева размещены таким образом, что чем ближе узел находится к корню дерева, тем из меньшего количества примитивов он

состоит и тем самым требует меньше времени на принятие решения. Данный вид каскадной модели хорошо подходит для обработки изображений, на которых общее количество детектируемых образов мало.

В этом случае метод может быстрее принять решение о том, что данный регион не содержит образ, и перейти к следующему. Пример каскадной модели сильных классификаторов:

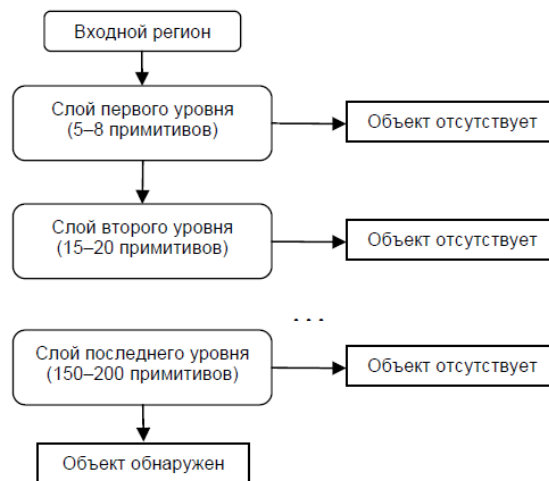


Рисунок 4 – Алгоритм работы каскада классификаторов

Сложность обучения таких каскадов равна $O(xyz)$, где применяется x этапов, y примеров и z признаков.

Далее, каскад применяется к изображению. Работа с «простыми» классификаторами – при этом отбрасывается часть «отрицательных» окон. Положительное значение первого классификатора запускает второй, более приспособленный и так далее. Отрицательное значение классификатора на любом этапе приводит к немедленному переходу к следующему сканирующему окну, старое окно отбрасывается. Цепочка классификаторов становится более сложной, поэтому ошибок становится намного меньше.

Алгоритм трекинга CAMShift был создан Гарри Брадски в 1998 г. и способен отслеживать лица [Местецкий 2002]. Он комбинирует алгоритм отслеживания объекта Mean Shift, основанный на карте вероятности цвета кожи, с адаптивным шагом изменения размера области отслеживания. Вероятность цвета кожи каждого пикселя изображения определяется методом Histogram Backprojection, основанным на цвете, представленном в виде цветового тона (Hue) модели HSV. Так как алгоритм CAMShift способен отслеживать лица на основе вероятности цвета кожи, то он может применяться для отслеживания руки.

Преимуществами данного алгоритма являются: низкие требования к вычислительным ресурсам, гибкие настройки точности позиционирования, возможность работы в различных условиях освещенности. Также дополнительным

преимуществом алгоритма является возможность работы в условиях частичного перекрытия отслеживаемого объекта.

Рассмотрим множество точек (распределение пикселей на гистограммы обратного проецирования). Имеется небольшое окно, которое должно двигаться в сторону области с максимальной вероятностью совпадения пикселей (рисунок 5).

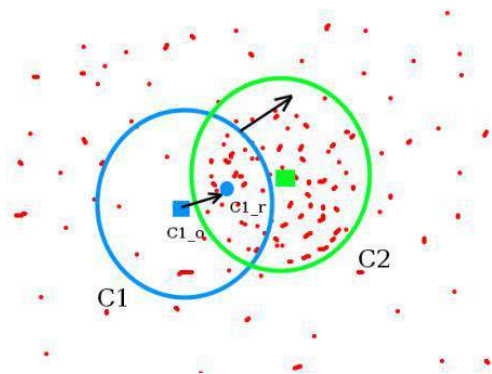


Рисунок 5 – Пример работы алгоритма CAMShift

Алгоритм CAMShift состоит из следующих шагов:

1) Построить гистограмму для канала «Тон» для области изображения с найденной кистью.

2) Вычислить «обратную проекцию» гистограммы. Обратная проекция – это новая гистограмма, в которой показаны вероятности принадлежности пикселя к кисти.

3) Найти новое положение отслеживаемых объектов. Данное действие производится с помощью вероятностей из обратной проекции. Центр смещается к точке с наибольшей концентрацией высоких вероятностей принадлежности пикселя к кисти.

Наибольшая концентрация высоких вероятностей пикселей кисти находится с помощью моментов, которые вычисляются по следующим формулам.

Нулевой момент:

$$Z_{00} = \sum_x \sum_y I(x, y) \quad (1)$$

Моменты первого порядка:

$$\begin{aligned} Z_{10} &= \sum_x \sum_y xI(x, y) \\ Z_{01} &= \sum_x \sum_y yI(x, y) \end{aligned} \quad (2)$$

где $I(x, y)$ - значение вероятности принадлежности пикселя к кисти.

По вычисленным моментам можно найти центр и размер отслеживаемой области:

$$\begin{aligned} x_c &= Z_{10} / Z_{00} \\ y_c &= Z_{01} / Z_{00} \end{aligned} \quad (3)$$

$$s = 2 \sqrt{\frac{Z_{00}}{256}} \quad (4)$$

После нахождения области интереса (кисти), чтобы определить основные характеристики кисти необходимо проанализировать полученные данные. Для анализа кисти необходимо определить контур с помощью детектора границ Кенни и выделить где находятся пальцы с помощью алгоритма к-изгиба.

Детектор границ Кенни состоит из пяти отдельных шагов [Грин, 2014]:

1) Сглаживание. Фильтр Гаусса. Размытие изображения для удаления шума.

2) Поиск градиентов. Границы отмечаются там, где градиент изображения приобретает максимальное значение.

3) Подавление не-максимумов. Только локальные максимумы отмечаются как границы.

4) Двойная пороговая фильтрация. Потенциальные границы определяются порогами.

5) Трассировка области неоднозначности. Итоговые границы определяются путём подавления всех краёв, несвязанных с определенными (сильными) границами.

Перед применением детектора, преобразуем изображение в оттенки серого, чтобы уменьшить вычислительные затраты, иначе быстрого преобразования не получится.

После определения границ необходимо найти пальцы на изображении. Контур кисти является достаточно изменчивой характеристикой, поэтому необходимо выделить более постоянные характеристики, например центр ладони, а также концы пальцев кисти.

Центр ладони находится как центр окружности с максимальным радиусом, вписанной в ладонь. Далее от центра до каждой точки границы ладони считается евклидово расстояние. Такое представление кисти хорошо описывает ее топологические свойства. Для нахождения концов пальцев кисти применим алгоритм к-изгиба [Grauman, 2004].

На первом шаге алгоритма вычисляется минимально выпуклая оболочка (МВО) контура руки. Задача поиска МВО решается с помощью алгоритма Грэхема [Graham, 1972]. Алгоритм решает данную задачу с помощью стека, сформированного из точек кандидатов. Все точки входного множества заносятся в стек, а затем точки, которые не являются вершинами постепенно удаляются из него. По завершении работы алгоритма в стеке остаются только вершины.

В результате определяется кисть человека и ее характеристики (ключевые точки, углы между

векторами). На основе этих данных можно распознавать движения и жесты.

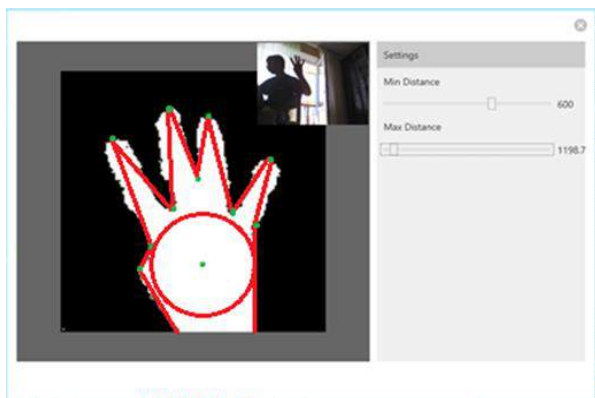


Рисунок 6 – Распознавание четырех пальцев

Заключение

Автоматизированный перевод русского жестового языка в текст играет важную роль в жизни людей с ограниченными возможностями. Создание подобных приложений имеет важное социальное значение и должно помочь адаптации таких людей. Кроме того, разрабатываемые подходы имеют и практическое значение, поскольку жестовые интерфейсы получают все большее распространение и развитие.

В виде отдельного программного приложения реализованы алгоритмы детектирования, трекинга и анализа кисти. Для обучения при детектировании руки используется технология бустинга. На данный момент улучшается производительность и скорость работы разработанных алгоритмов детектирования движений рук [Розалиев и др., 2013] и строится система определения по движениям смысла жеста [Розалиев и др., 2014] и его эмоциональной оценки [Заболеева-Зотова и др., 2013b].

Работа частично поддержана Российским фондом фундаментальных исследований (проекты 13-07-00459, 13-07-97042, 14-07-97016, 15-07-06322).

Библиографический список

- [Гонсалес, 2005] Гонсалес, Р. Цифровая обработка изображений: пер. с англ. / Р. Гонсалес, Р. Вудс.-Москва: 2005.-1072 с.
- [Грин, 2014] Грин, Б. Алгоритм выделения контуров Кенни [Электронный ресурс] / Б. Грин. - Режим доступа: http://www.pages.drexel.edu/~weg22/can_tut.html.
- [Заболеева-Зотова и др., 2011] Заболеева-Зотова А. В., Орлова Ю. А., Розалиев В. Л., Бобков А. С. Применение нечетких темпоральных высказываний для описания движений при эмоциональных реакциях. – Открытое образование, 2011.
- [Заболеева-Зотова и др., 2013а] Formalization of initial stage of designing multi-component software / Заболеева-Зотова А.В., Орлова Ю.А., Розалиев В.Л., Фоменков С.А., Петровский А.Б. // Multi Conference on Computer Science and Information Systems 2013 (Prague, Czech Republic, July 23-26, 2013) : Proceedings of the IADIS International Conference Intelligent Systems and Agents 2013 / IADIS (International Association for Development of the Information Society). – [Prague], 2013. – P. 107-111.
- [Заболеева-Зотова и др., 2013б] Automated identification of human emotions based on analysis of body movements / Заболеева-Зотова А.В., Бобков А.С., Орлова Ю.А., Розалиев В.Л., Половинкин А.И. // Multi Conference on Computer Science and

Information Systems 2013 (Prague, Czech Republic, July 23-26, 2013) : Proceedings of the IADIS International Conferences Interfaces and Human Computer Interaction and Game and Entertainment Technologies 2013 / IADIS (International Association for Development of the Information Society). – [Prague], 2013. – P. 299-304.

[Куракин, 2013] Куракин А.В. Распознавание динамических жестов в системе компьютерного зрения на основе медиального представления формы изображений // Московский государственный университет имени М. В. Ломоносова. – 2012.

[Местецкий 2002] Местецкий, Л. М. Математические методы распознавания образов / Л. М. Местецкий. -Москва: МГУ, 2002–2004.- С. 42 – 44

[Розалиев и др., 2013] Автоматизация построения векторной модели тела человека / Розалиев В.Л., Орлова Ю.А., Шпирко А.А., Дорофеев Н.С. // Электротехнические и информационные комплексы и системы. - 2013. - № 2, т. 9. - С. 102-106.

[Розалиев и др., 2014] Розалиев, В.Л. Applying the automated system for the determination of emotions in the education's tasks for people with disabilities / Розалиев В.Л., Орлова Ю.А. // Innovation Information Technologie : mater. of the 3rd Int. scien.-pract. conf. (Prague, April 21-25, 2014). Part 2 / МИЭМ ВШЭ, Рос. центр науки и культуры в Праге. - М., 2014. - С. 446-452.

[Соболев, 2012] Соболев Е.Г. Средство интерактивного взаимодействия реального и виртуального пространств на основе трехмерного сканирования объектов с использованием платформы Microsoft Kinect. – Москва, 2012.

[Christopher, 2013] Christopher Kawatsu. Development of a Fall Detection System with Microsoft Kinect / Christopher Kawatsu, Jiaying Li, C. J. Chung // Robot Intelligence Technology and Applications 2012, 2013.

[Grauman, 2004] K. Grauman and T. Darrell. Fast contour matching using approximate earth mover's distance. In CVRP, 2004.

[Graham, 1972] Graham, R. L. An efficient algorithm for determining the convex hull of a finite planar set / R. L. Graham.- Information Processing Letters, 1972.- 7:175-180.

[William, 2008] William L. Johnson, Michael Feuerstein. An Interpreter's Interpretation: Sign Language Interpreters' View of Musculoskeletal Disorders / William L. Johnson, Michael Feuerstein // Journal of Occupational Rehabilitation, September 2008.

AUTOMATED ALLOCATION OF THE HANDS OF THE PERSON TO RECOGNIZE SIGN LANGUAGE

Rozaliev V.L., Agafonov G.V., Kirichenko M.I.

*Volgograd State Technical University,
Volgograd, Russia*

vladimir.rozaliev@gmail.com

This article is devoted to the problem of detecting and analyzing brushes with the camera depth. Special attention is given to the analysis and tracking of the brush. In the work of the combined algorithms for searching, tracking and analysis of the hands of man. Reviewed existing systems for gesture recognition. Also in the paper highlighted the advantages and disadvantages of visual methods.

The aim of the study was to finding and rapid analysis of the hands of man. To achieve this goal were as follows: conduct a review of existing approaches and software for the detection and analysis of the hands; develop a method for the detection of hands on the depth image; develop a method for analyzing the brush; implement the methods and algorithms in software.



УДК 004.823

АВТОМАТИЗАЦИЯ РАСПОЗНАВАНИЯ ВНЕШНЕГО ВИДА ЧЕЛОВЕКА ПО ТЕКСТУ НА ЕСТЕСТВЕННОМ ЯЗЫКЕ И СИНТЕЗ ПОРТРЕТНОГО ИЗОБРАЖЕНИЯ

Орлова Ю.А., Долбин А.В., Кипаева Е.В.

Волгоградский государственный технический университет, г. Волгоград, Россия

yulia.orlova@gmail.com

Данная статья посвящена разработке приложения по выделению элементов внешнего описания человека на основе текста. В статье описана модель внешнего вида человека, а также основные методы поиска внешности. Кратко описываются принципы построения фотороботов по текстовому описанию.

Ключевые слова: обработка текста; информационный поиск; латентно-семантический анализ; оценка релевантности; портретная идентификация; технология построения фоторобота.

Введение

Анализ текстовой информации для последующего синтеза некоторого изображения относится к сложным задачам и в большинстве случаев требует индивидуального решения. В данном проекте мы решили объединить две ведущие нами разработки: выявление внешнего вида человека из текста и синтез фоторобота по текстовому описанию.

На вход модуля выявления внешнего вида человека по тексту подается текст на русском языке в произвольной форме. Выходом является фрейм-экземпляр внешнего вида человека. Так как, объем входных данных может быть очень большим, то необходимо обеспечить максимальную производительность программы. Для обеспечения должного уровня быстродействия необходимо формировать инвертированной индекс входных данных [Дмитриев и др., 2013].

На вход модуля построения фоторобота подается полученный фрейм, выходом является синтезированное изображение.

1. Алгоритм выявления описания внешнего вида человека

Общий алгоритм выявления описания следующий:

- 1) графематический анализ текста;
- 2) морфологический анализ;
- 3) поиск упоминаний о личности в тексте;
- 4) применение латентно-семантического анализа

для установления зависимостей между ключевыми параметрами внешнего вида человека и множеством предложений;

5) выделение в предложении элементов внешнего вида человека.

Выделение предложений из сплошного текста – процедура необходимая для дальнейшего анализа текста в любой системе анализа естественных языков. Для решения данной задачи необходимо провести графематический анализ текста. [Солошенко и др., 2014a] Графематический анализ проводится в два этапа:

1) Выделение предложений из текста на основе поиска стоп-слов. В ходе разработки были выделены следующие стоп-символы: «.», «!», «;», «?».

2) Выделение отдельных слов в полученных предложениях.

Задачей морфологического разбора является описание морфологической формы слова и её синтаксической роли в предложении [Солошенко и др., 2014a; Солошенко и др., 2014b]. Как правило, анализу подвергаются слова в контексте предложения, но в данном проекте был принят ряд допущений:

1) Входной текст должен быть только на русском языке.

2) Входной текст не должен содержать каких-либо грамматических или синтаксических ошибок.

3) Букву «ё» не следует заменять буквой «е», так как это приведет к неправильному морфологическому анализу слова.

4) Упрощенный разбор «двусмысленных предложений».

2. Модель внешнего вида человека

Был произведен анализ важных элементов описания внешнего вида человека для разных профессий: художник, писатель, криминалист, скульптор. В итоге была составлена информационная модель в виде фрейма-прототипа на языке frame representation language:

```
(frame Внешний_Вид
(Рост (value(M)) (IF_NEEDED(ph_s))
(Телосложение (value(M)) (IF_NEEDED(ph_s))
(Голова (value(M)) (IF_NEEDED(ph_s))
(Волосы (value(M)) (IF_NEEDED(ph_s))
(Лицо (value(M)) (IF_NEEDED(ph_s))
(Лоб (value(M)) (IF_NEEDED(ph_s))
(Брови (value(M)) (IF_NEEDED(ph_s))
(Глаза (value(M)) (IF_NEEDED(ph_s))
(Ресницы (value(M)) (IF_NEEDED(ph_s))
(Нос (value(M)) (IF_NEEDED(ph_s))
(Уши (value(M)) (IF_NEEDED(ph_s))
(Губы (value(M)) (IF_NEEDED(ph_s))
(Подбородок (value(M)) (IF_NEEDED(ph_s))
(Зубы (value(M)) (IF_NEEDED(ph_s))
(Шея (value(M)) (IF_NEEDED(ph_s))
(Плечи (value(M)) (IF_NEEDED(ph_s)))
```

Здесь M – множество внешних характеристик, а ph_s – демон поиска заранее заданных фразеологизмов. Изначально планировалось задать фиксированный набор элементов описания на каждый параметр внешнего вида. Но в ходе разработке от этой идее пришлось отказаться, так как каждый параметр может быть описан огромным множеством различных речевых оборотов. Поиск описания внешнего вида человека производится по морфологическим признакам [Баргесян, 2004].

В случае, когда один или несколько слотов фрейма остались незаполненными, то обрабатывается процедура поиска фразеологизмов в тексте. Набор фразеологизмов для каждого слота задается в программе в графическом интерфейсе пользователя и сохраняется на жёстком диске в файле структуры xml. Фразеологизмы представляют собой продукционную модель знаний: если <условие>, то <действие>. Применительно к рассматриваемой теме условие можно сформулировать следующим образом: если <в данном предложении упоминается описание личности и встречается один из указанных фразеологизмов>, то <заполнить слот фрейма соответствующим значением>. Механизм работы с фразеологизмами был добавлен для случаев, когда разработанного функционала программы недостаточно для выполнения тех или иных операций.

3. Обзор корпусов русского языка и систем информационного поиска

Был проведен обзор корпусов и словарей русского языка. Список рассмотренных корпусов:

- Национальный корпус русского языка.
- Открытый корпус русского языка.
- Тюбингенский корпус.
- Хельсинский аннотированный корпус.

В результате анализа был выбран открытый корпус русского языка, так как он удовлетворял ряду необходимых требований:

- Большое число словоупотреблений с морфологической разметкой.
- Наличие подкорпуса со снятой омонимией.
- Наличие частотных списков.
- Возможность использования на локальной машине.

Не смотря на то, что национальный корпус русского языка имеет больший объем, емкость открытого корпуса (более 200 млн. словоупотреблений) является вполне достаточным для решаемых задач.

Следующим шагом являлась разработка алгоритма по нахождению упоминаний личности в тексте. Был проведен анализ систем семантического анализа текста по критериям (таблица 1): не требует участия человека для формирования выборки (1), возможность выделения фактов из текста (2), построение семантической сети текста (3), выделения внешнего описания человека в тексте (4), учет сложных синтаксических конструкций текста (5).

В результате сравнительного обзора были выявлены недостатки существующих на данный момент систем:

- 1) Требуется предварительная настройка или наличие обучающей выборки.
- 2) Представление результатов в неудобном и неструктурированном для пользователя виде.
- 3) Отсутствие сформированной информационной модели внешнего вида человека.

Чтобы решить недостатки 1, 2 были сделаны следующие допущения и ограничения на работу нашего алгоритма по нахождению упоминаний личности в тексте:

- 1) Отсутствие чётких правил для разрешения анафоры местоимений третьего лица.
- 2) Во входном тексте должна упоминаться только одна личность.

Таблица 1 – Сравнительный обзор систем семантического анализа текста

	1	2	3	4	5
Томита-парсер (Yandex)	-	+	+	+	+
AOT (www.aot.ru)	+	-	+	-	-
Fact extractor desktop (RCO)	-	+	+	+	-
Поисковые системы (google.com)	+	+	+	+	-
Semantix (Синергетические системы)	-	-	+	-	-

Индексация предложений с описанием человека проходит в два этапа. На первом этапе проводится лемматизация слов и нахождение ФИО человека. На втором этапе проводится поиск местоимений, которые согласуются по морфологическим признакам. Простота такого подхода и принятые допущения безусловно снижают универсальность и качество получаемых результатов, однако мы планируем исправить и усовершенствовать алгоритм.

4. Латентно-семантический анализ

В качестве индексирования текста на естественном языке применялся метод латентно-семантического анализа [Стенин, 2013].

Минусы латентно-семантического анализа:

- Скорость работы данного метода значительно ухудшается при увеличении объема входных документов.
- Латентно-семантический анализ не учитывается контекст документа.

Положительные стороны применения латентно-семантического анализа:

- Частичное снятие омонимии с индексируемых слов.
- В работе метода используются матрицы, которые являются результатом сингулярного разложения частотной матрицы.

Реализованный алгоритм работы латентно-семантического анализа, выводом которого является векторное пространство текста:

1) По входному тексту посчитать количество вхождений каждого слова в каждое предложение. Составить частотную матрицу.

2) Применение оценки релевантности полученных результатов методом TF-IDF над частотной матрицей. TF-IDF является методом

взвешивания на основе комбинации частоты и обратной документной частоты [Маннинг, 2011].

3) Использование сингулярного разложения над матрицей, полученной на втором этапе, для формирования матриц U , S , Vt , где U — пространство документов, Vt — пространство слов, S — диагональная матрица.

4) Для матрицы Vt в дальнейшей работе метода не учитываются строки с индексом больше 2, а для матрицы U — столбцы с индексом больше 2.

5) Определить индекс слота фрейма относительно остальных слов.

6) По матрицам Vt , U определить координаты ключевого параметра внешнего вида человека и всех документов.

Особенность сингулярного разложения матрицы в том, что оно выделяет ключевые составляющие матрицы, позволяя игнорировать шумы.

Алгоритм сингулярного разложения матрицы:

1) Пусть дана частотная матрица A . Тогда необходимо вычислить матрицу по формуле:

$$B = A^t * A. \quad (1)$$

2) Построить уравнение:

$$\det(B - \lambda * E) = 0. \quad (2)$$

3) Если n – размерность матрицы, то получаем степенное уравнение вида:

$$a_0 \lambda^n + a_1 \lambda^{n-1} + \dots = 0. \quad (3)$$

4) Если V — искомый собственный вектор, то найти собственные вектора можно следующим образом:

$$\det(B - \lambda * E)V = 0. \quad (4)$$

5) Точно также найти все собственные векторы, из которых сформировать матрицу U .

6) Сформировать матрицу S из корней собственных чисел, расположенных по диагонали.

7) Рассчитать заключительную матрицу V по формуле:

$$V^t = S^{-1}U^t A. \quad (5)$$

5. Определение элементов внешнего вида

Основная идея заключается в том, чтобы по морфологическим признакам определить качества, которые относятся к тем или иным слотам фрейма. В каждом индексированном предложении с найденным упоминанием о внешнем виде человека определяется позиция самого объекта и внешней характеристики. Если два слова расположены близко и совпадают по морфологическим признакам, то заполнить слот фрейма. На данный момент отбор происходит исключительно по прилагательным, характеризующими качество объекта.

6. Фоторобот

Наука об описании человеческого лица и построении фоторобота называется габитоскопией. Портретная криминалистическая экспертиза является одной из форм портретной идентификации. Принцип фоторобота кардинально отличен от принципов живописного портрета или фотографии. Фоторобот составляется из фрагментов множества иных, неидентичных искомому лицу, через сведение отдельных типических черт и черточек (элементарных частиц человеческого облика).

В практике криминалистов фоторобот призван дать не точное изображение лица прототипа, но максимальное приближение к тому впечатлению от лица, которое сложилось в памяти очевидца. Таким образом в отличие от фотографии (дающей мгновенный объективный слепок лица) и от портрета (где художник как будто нарочно умерщвляет лицо своей "модели", превращая само изображение в "модель восприятия") - фоторобот есть не что иное как неотрывное от воспринимающего субъекта-автора запечатление живого лица, лица из жизни, лица, проплывающего мимо нас в потоке времени. Это изменчивое лицо живого существа - открытое ко времени и ко множеству взглядов.

7. Системы построения фоторобота

Было проведен обзор и сравнение систем для построения фоторобота.

Программа Faces содержит более 9000 особенностей строения человеческого лица. Все отобранные части лица соединяются в единое целое, в результате чего получается фотография человека. Также в редакторе имеются возможности коррекции тона волос и внесение различных дополнений, таких как пирсинг, тату и т.п. Конечный результат - составленный со слов очевидца фоторобот - мало чем отличается от фотографии. В процессе составления фоторобота генерируется его уникальный код, по которому на любом рабочем месте может быть сгенерирован требуемый фоторобот (рисунок 1).

«Каскад-Фоторобот» предназначен для автоматизации процесса составления портрета лиц мужского и женского пола, монголоидной и европеоидной расы, в профиль и анфас. В данном программном продукте присутствуют такие дополнительные возможности, как: коррекция элементов внешности (масштабирование, поворот, свободное трансформирование); дорисовка индивидуальных особенностей в ручном режиме (инструменты «кисть», «карандаш», «ластик», «аэрограф», «палец»); подготовка нескольких вариантов изображения одного человека (рисунок 2).



Рисунок 1 – Пример работы программы Faces 4.0



Рисунок 2 – Интерфейс программы «Каскад-Фоторобот»

3D Фоторобот (3DHead) – это система предназначена для оперативной портретной идентификации (т.е. для построения портрета человека по памяти с целью идентификации), однако может быть использована не только правоохранительными органами для проведения розыскных мероприятий, но и в любых других областях, где требуется создание трехмерных моделей головы человека.

Традиционно программы этого класса представлены в виде библиотек готовых элементов лица, перебором которых нужно собрать образ, максимально приближенный к оригиналу. Эффективность их не высокая, так как для создания качественного образа нужны большие библиотеки, в которых трудно найти подходящий рисунок (рисунок 3).

В этой системе использован другой подход – создана математическая модель головы человека, изменяя параметры которой можно получить практически любой тип лица и головы человека. То есть не нужно выискивать в списках похожий элемент лица – нужно только изменить связанные с этим элементом параметры модели, причем пересчет модели производится в реальном времени, по мере изменения любого параметра одновременно изменяется и образ на экране.

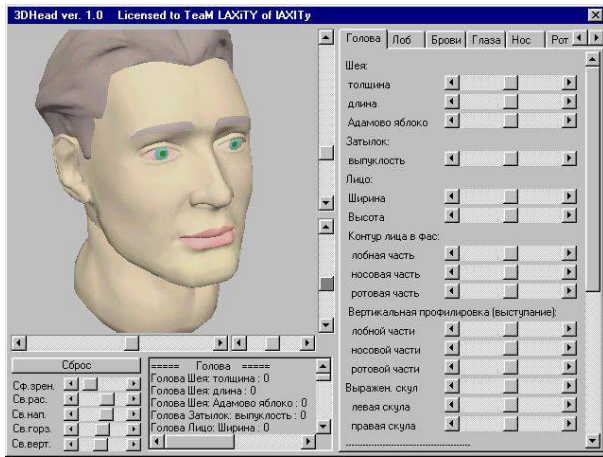


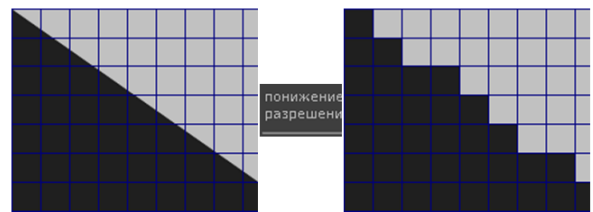
Рисунок 3 – Пример работы программы 3D Head

Однако, у программы существует несколько серьезных недостатков. Крайне сложный интерфейс, внушительная ресурсоемкость и самое длительное время построения фоторобота среди всех рассмотренных аналогов.

8. Синтез портретного изображения

В компьютерной графике давно известна и используется технология морфинга, позволяющая получать "среднее арифметическое" между несколькими изображениями. Если использовать её для объединения нескольких фотороботов, узнаваемость итогового изображения заметно повышается. Морфинг может применяться и непосредственно при создании фотороботов. Корректируя фоторобот с его помощью, можно добиться большего соответствия итогового изображения, чем при "сборке" лица из готовых частей, применяемой сейчас [Michael, 2008].

Для избавления от случайных шумов на изображении будет использоваться метод сглаживания. Сглаживание или анти-алиасинг является процессом, который пытается минимизировать появление ступенчатых или зубчатых диагональных границ, которые придают тексту или изображениям грубый цифровой вид [Норвиг, 2006]. Сглаживание удаляет эти ступеньки и создаёт впечатление более мягких границ и высокого разрешения. Оно принимает во внимание, насколько идеальная граница перекрывает смежные пиксели. Ступенчатая граница просто округлена вверх или вниз без промежуточного значения, тогда как сглаженная граница выдаёт значение, пропорциональное тому, насколько много от границы попало в каждый пиксель (рисунок 4).



Идеальная граница в мелком масштабе

Ступенчатая сглаженная граница в крупном масштабе

Рисунок 4 – Границы изображения

Для реализации программы была собрана база данных изображений с частями лица. Каждое такое изображение было проиндексировано и отмечено к какой части головы оно относится.

Общий принцип работы программы, построенной по принципу многокомпонентного программного обеспечения [Заболеева-Зотова и др., 2013], заключается в следующем: пользователь вводит текстовое описание человека или подругает его. Текст анализируется, формируется фрейм-экземпляр. По нему максимально близко выбирается изображение. После чего набор изображений компоуется в одно изображение (рисунок 5).

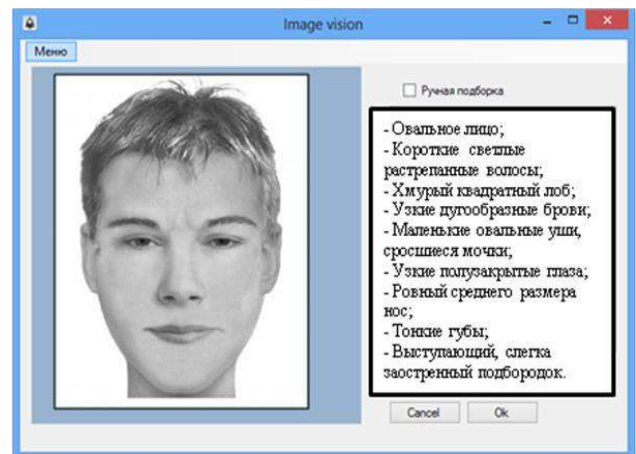


Рисунок 5 – Пример работы программы разрабатываемой системы

Также в данном программном продукте имеется возможность подобрать описание частей лица и вручную настроить их размеры (рисунок 6).

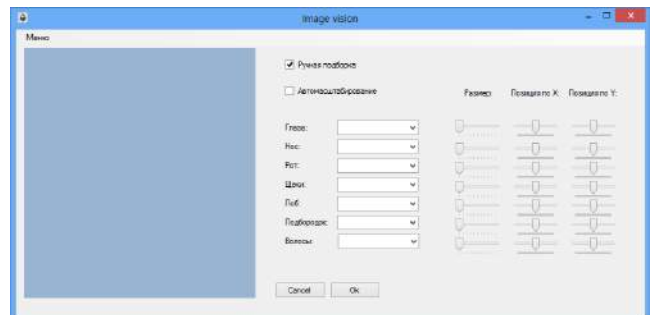


Рисунок 6 – Интерфейс программного продукта при ручном подборе описания человека

Заключение

В результате была разработана программа на языке Python 3.3 с использованием библиотеки морфологического анализа rymorphy2, выделяющая из текста внешнее описание человека по разработанным в ходе работы над проектом алгоритмами. Результаты показывают высокую точность распознавания. В ходе дальнейшей работы планируется решить следующие проблемы: разрешение омонимии на основе контекстных правил, использование семантического и синтаксического анализа языковых конструкций, улучшение алгоритма поиска личности, улучшение алгоритма разрешения анафоры. Синтез изображения и его коррекция может быть проведена с учетом работ [Розалиев и др., 2010; Бобков и др., 2011]. Создание фоторобота человека играет важную роль не только в криминалистике, но и в других сферах жизни. Благодаря новой предлагаемой системе пользователю не надо будет делить лицо на несколько частей, чтобы составить фоторобот, ему следует просто представить описание человека, которого он видел, что достаточно упростит задачу.

Работа частично поддержана Российским фондом фундаментальных исследований (проекты 13-07-00351, 13-07-97042, 14-07-97016, 14-07-97017, 15-07-06322, 15-07-05440).

Библиографический список

- [Баргесян, 2004] Методы и модели анализа данных OLAP и Data Mining / А. А. Баргесян [и др.]. — СПб.: БХВ-Петербург, 2004 — 336 с.
- [Бобков и др., 2011] Развитие системы автоматизированного определения эмоций и возможные сферы применения / Бобков А.С., Заболева-Зотова А.В., Орлова Ю.А., Розалиев В.Л. // Открытое образование. - 2011. - № 2. - С. 59-62.
- [Дмитриев и др., 2013] Automatic identification of time and space categories in the natural language text / Дмитриев А.С., Заболева-Зотова А.В., Орлова Ю.А., Розалиев В.Л. // Applied Computing 2013 : proceedings of the IADIS International Conference (Fort Worth, Texas, USA, October 23-25, 2013) / IADIS (International Association for Development of the Information Society), UNT (University of North Texas). – [Fort Worth (Texas, USA)], 2013. – P. 187-190.
- [Заболева-Зотова и др., 2013] Formalization of initial stage of designing multi-component software / Заболева-Зотова А.В., Орлова Ю.А., Розалиев В.Л., Фоменков С.А., Петровский А.Б. // Multi Conference on Computer Science and Information Systems 2013 (Prague, Czech Republic, July 23-26, 2013) : Proceedings of the IADIS International Conference Intelligent Systems and Agents 2013 / IADIS (International Association for Development of the Information Society). – [Prague], 2013. – P. 107-111.
- [Маннинг, 2011] Маннинг, К. Д. Введение в информационный поиск / К. Д. Маннинг, П. Рагхаван, Х. Шютце; пер. с англ. под ред. П. И. Браславского, Д. А. Ключина, И. В. Сегаловича. - Москва.: И. Д. Вильямс, 2011. — 528 с.
- [Морозов и др., 2010] Морозов, Д.А. Обработка растровых изображений с помощью Грид-технологии / Морозов Д.А., Стуров Д.А. // Алгоритмы, методы и системы обработки данных : сб. науч. тр. Вып. 15 / Муромский ин-т (филиал) ГОУ ВПО "Владимирский гос. ун-т". - Муром, 2010.
- [Норвиг, 2006] Норвиг П., Искусственный интеллект. Современный подход. – М.: ВИЛЬЯМС, 2006.
- [Розалиев и др., 2010] В.Л. Розалиев, А.С. Бобков, О.С. Федоров Применение нейронных сетей и грануляции при построении автоматизированной системы определения эмоциональных реакций человека / Изв. ВолгГТУ. Серия «Актуальные проблемы управления, вычислительной техники и

информатики в технических системах»: межвуз. сб. науч. ст. / ВолгГТУ. - Волгоград, 2010. - Вып. 9, № 11. - С. 63-68.

[Стенин, 2013] Латентно-семантический метод извлечения информации из интернет ресурсов / А. А. Стенин [и др.] // Восточно-Европейский журнал передовых технологий. Автоматика. Вычислительная техника. — 2013. — Вып. 9, Том 4. — С. 19-22.

[Солошенко и др., 2014a] Thematic Clustering Methods Applied to News Texts Analysis / Солошенко А.Н., Орлова Ю.А., Розалиев В.Л., Заболева-Зотова А.В. // Knowledge-Based Software Engineering : Proceedings of 11th Joint Conference, JCKBSE 2014 (Volograd, Russia, September 17-20, 2014) / ed. by A. Kravets, M. Shcherbakov, M. Kultsova, Tadashi Iijima ; Volograd State Technical University [et.]. – [Б/М] : Springer International Publishing, 2014. – P. 294-310. – (Series: Communications in Computer and Information Science ; Vol. 466).

[Солошенко и др., 2014b] Автоматизация семантического анализа новостных Интернет-текстов. / Солошенко А.Н., Розалиев В.Л., Орлова, Ю.А. // Открытые семантические технологии проектирования интеллектуальных систем OSTIS-2014 : матер. IV междунар. науч.-техн. конф. (Минск, 20-22 февр. 2014 г.). Белорус. гос. ун-т информатики и радиоэлектроники, Администрация Парка высоких технологий. Минск, с. 435-438.

[Ужва, 2013] Ужва А.Ю. Автоматизированная разработка онтологической модели предметной области для поиска образовательных ресурсов с использованием анализа текстов рабочих программ // Современные проблемы науки и образования. – Волгоград, 2013.

[Michael, 2008] Michael L. Matthews: Discrimination of Identikit constructions of faces: Evidence for a dual processing strategy / Michael L. // Perception & Psychophysics, March 2008, Volume 23, Issue 2, pp 153-161.

AUTOMATED RECOGNITION OF THE APPEARANCE OF MAN ON THE NATURAL LANGUAGE AND SYNTHESIS OF PORTRAIT IMAGES

Orlova Yu.A., Doldin A.V., Kipaeva E.V.

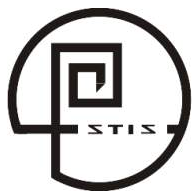
*Volograd State Technical University,
Volograd, Russia
yulia.orlova@gmail.com*

This article is devoted to the development application for the allocation of items to the external description of a person on the basis of the text.

The article describes the model of the appearance of a person, as well as the main methods of searching appearance. Briefly describe the principles of composite on a text description.

This paper describes the concept and principle of the identikit. Also examined existing systems and methods for constructing composite picture. On the basis of the study proposes the creation of a new system for the identification of human portraiture text description. Creating a composite sketch of a man playing an important role not only in criminology, but also in other areas of life. With the new proposed system, the user will not have to divide the face into several pieces to make an identikit, he should simply provide a description of the man whom he had seen enough to make it easier.

Keywords: text processing; information retrieval; latent semantic analysis; assessment of relevance; portrait identification; technology building sketch.



УДК 004.932.2

КОНТЕКСТНО-ЗАВИСИМЫЙ АНАЛИЗ ПОРТРЕТНЫХ ФОТОГРАФИЙ

Алексеев А.В., Орлова Ю.А.

Волгоградский государственный технический университет, г. Волгоград, Россия

Alekseev.yeskela@gmail.com

В данной работе представлена методика контекстно-зависимого анализа портретных фотографий. Весь анализ автоматический, вмешательство пользователя не требуется. Разработанная методика позволяет локализовать лицо и найти контуры рта и глаз. В заключении показаны примеры работы на тестовых фотографиях.

Ключевые слова: обработка изображений; контекстно-зависимый анализ; анализ портретных изображений; лицо человека; автоматический поиск.

Введение

В настоящее время в связи со стремительным развитием цифровой фотографии и цифрового видео очень перспективной является задача распознавания образов на цифровых изображениях и лиц в частности.

Последние десять лет в области распознавания лиц и их элементов ведутся активные разработки и предложены различные методы распознавания, использующие такие подходы, как метод сверточных и обычных нейронных сетей [Розалиев и др., 2010], скрытый марковских моделей [Nefian, 1999], активные модели формы [Prabhu et al], активные модели внешнего вида [Mitchell et al, 2002], методы с использованием гистограмм, статистические методы и т. д.

Распознавание лиц и элементов лиц может применяться в следующих областях: системы автоматического учета числа посетителей; системы пропускного контроля в учреждениях, аэропортах и метро; автоматические системы предотвращения несчастных случаев; интеллектуальные интерфейсы "человек-компьютер" и др. Нахождение различных элементов лица позволяет вести учет дополнительной статистики и получать больше информации о человеке. Задача обнаружения лица человека (а также элементов лица) является сложной ввиду нескольких основных причин: высокая вариативность лиц человека, обусловленная анатомическими и фенотипическими особенностями индивидов; различные условия освещенности, определяющиеся типом, количеством и направлением источников света.

Целью данной работы является автоматический поиск лиц на изображении и выделение контура для каждого лица контуров следующих элементов: рот, глаза.

Предлагаемая методика

Мы предлагаем следующую методику контекстно-зависимого анализа лиц:

1. Детектирование лиц на основе метода Виолы-Джонса, хорошо зарекомендовавший себя метод с высоким качеством и скоростью работы.

2. Все искомые элементы будут находиться в зависимости от найденного прямоугольника лица, исходя из физиологии человека. Глаза симметричны и находятся в верхней части лица (на симметрию накладывается ограничения в случае незначительных наклонов головы, тогда глаза не будут симметричны относительно найденного прямоугольника). Рот находится посередине в нижней части лица.

3. Локализовать рот мы будем на основе эвристически подобранного соотношения относительно прямоугольника лица. Уточнение границ рта мы будем делать методом сегментации GrabCut, метод трактует все изображение как граф, и по заданным правилам в этом графе ищется минимальный разрез, в результате сегментации мы получаем две области: объект и не объект.

4. Локализовать глаза мы будем также с помощью метода Виолы-Джонса, только в этом раз мы будем искать глаза в пределах верхней части лица, также дополнительно будет учитываться то, что глаза должны быть не цвета кожи.

Детектирование лица

Для детектирования лиц мы используем метод Виолы-Джонса [Viola et al, 2001]. Метод был разработан в 2001 году Полом Виолой и Майклом Джонсом, на данный момент является одним из лучших по соотношению качества и скорости для детектирования объектов на изображении (лиц в частности). Основные принципы, на которых основан метод, таковы:

1) Используются изображения в интегральном представлении, что позволяет быстро вычислять необходимые объекты. Интегральное представление изображения – это матрица, размерность которой совпадает с размерностью исходного изображения и каждый элемент интегрального изображения содержит в себе сумму пикселей изображения в прямоугольнике от пиксела с координатами (0,0) до него самого (x,y).

2) Используются признаки Хаара, с помощью которых происходит поиск нужного объекта (лица) (Рисунок 1). Признаки Хаара – признаки цифрового изображения, используемые в распознавании образов. Своим названием они обязаны интуитивным сходством с ветвями Хаара. Признак Хаара состоит из смежных прямоугольных областей. Они позиционируются на изображении, далее суммируются интенсивности пикселей в областях, после чего вычисляется разность между суммами. Эта разность и будет значением признака, заданного размера, определенным образом спозиционированного на изображении.

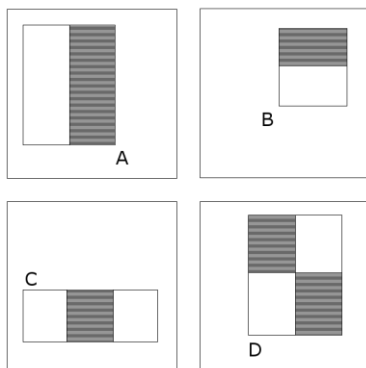


Рисунок 1 – Признаки Хаара

3) Используется бустинг для выбора наиболее подходящих признаков для искомого объекта на данной части изображения. Бустинг – комплекс методов, способствующих повышению точности аналитических моделей. Эффективная модель, допускающая мало ошибок классификации, называется «сильной». «Слабая» же, напротив, не позволяет надежно разделять классы или давать точные предсказания, делает в работе большое количество ошибок. Поэтому бустинг означает дословно «усиление» «слабых» моделей [Jan et al, 2010] – это процедура последовательного построения композиции алгоритмов машинного обучения, когда каждый следующий алгоритм стремится компенсировать недостатки композиции всех предыдущих алгоритмов.

4) Все признаки поступают на вход классификатора, который даёт результат «истина» - объект есть, либо «ложь» - объекта нет.

5) Используются каскады классификаторов для быстрого отбрасывания окон, где не найдено лицо. Каскад представляет собой цепочку классификаторов, каждый из которых обучается на ошибках предыдущего. Объект считается найденным только тогда, когда все классификаторы каскада вернули «истину».

Детектирование рта

1.1. Локализация рта

Детектирование рта мы проводим в 2 этапа, первый – получение ограничивающего прямоугольника, это делается на основе физиологии лица, в результате экспериментов были получены границы, которые зависят от полученного ранее прямоугольника лица. Прямоугольник рта относительно лица определяется следующими соотношениями:

- Ширина рта = 50% от ширины лица;
- Высота рта = 30% от высоты лица;
- Относительно верхнего левого угла прямоугольника лица:
 - Координата x (ширина) рта = $0,25 * \text{ширину лица}$;
 - Координата y (высота) рта = $0,7 * \text{высоту лица}$;

На рисунке 2 представлен локализирующий рот прямоугольник (зеленый), красный прямоугольник – найденное лицо. Здесь и далее все изображения лиц взяты с базы MICT [Nicolls et al, 2010].



Рисунок 2 – Локализация рта

1.2. Нахождение контура рта

Границы рта уточняются на основе алгоритма сегментации GrabCut. Данный алгоритм основан на приближенном построении минимального разреза специального графа, который строится по пикселям изображения. Найденный контур рта представлен на рисунке 3.

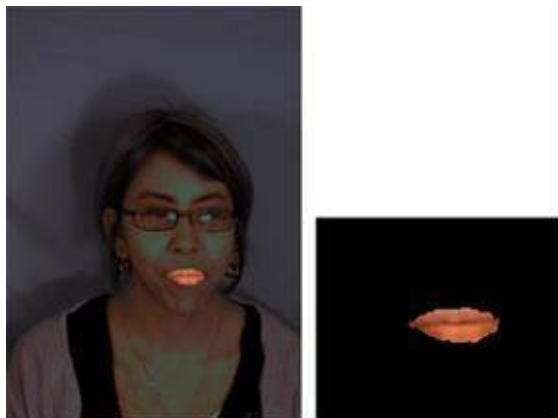


Рисунок 3 – Контур рта

Детектирование глаз

1.3. Локализация глаз

Для детектирования глаз используется также метод Виолы-Джонса, специальный обученный каскад, как и каскад для детектирования лиц были взяты из открытой библиотеки компьютерного зрения OpenCV [OpenCV, 2014]. Кроме того, используется предположение о том, что глаза находятся в верхней части лица (остальные детектирования отбрасываются как ложные). На рисунке 4 представлен пример детектирования глаз с помощью каскада: красный прямоугольник – лицо, синие – примерная область нахождения глаз, зеленые – найденные глаза.

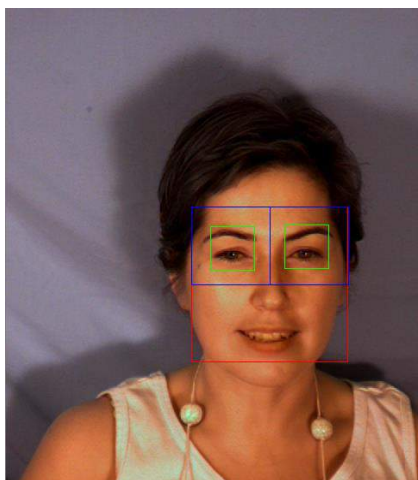


Рисунок 4 – Локализация глаз

1.4. Нахождения контура глаз

Контур глаз мы будем находить, ориентируясь на то, что глаза (непосредственно белки и зрачки) не телесного цвета. Цвет тела будем определять согласно [Vezhnevets et al, 2003] по формуле (1):

$$R > 95 \text{ and } G > 40 \text{ and } B > 20 \text{ and } R > B \\ \text{and } R - G > 15 \text{ and } \max(R, G, B) - \min(R, G, B) > 15 \text{ and } (H < 25 \text{ or } H > 230) \quad (1)$$

где: R, G, B – соответственно значения красного, зеленого и синего каналов в цветовой модели RGB;

H – значение канала цветового тона в цветовой модели HSV.

Цвет пиксела считается цветом кожи тогда, когда значения соответствующих каналов удовлетворяют формуле (1).

Полученная маска кожи и выделенные контуры глаз с учетом локализации представлены на рисунке 5.



Рисунок 5 – Выделение контуров глаз

Результаты

В данной работе представлен метод контекстно-зависимого анализа портретных фотографий, который заключается в поиске лиц и локализации для каждого из найденных лиц глаз и рта, а также нахождения их контуров. Данная методика представляет собой комбинацию других методов, поэтому на нее накладываются ограничения, которые накладываются на каждый используемый метод в отдельности [Заболеева-Зотова и др., 2010]. А именно, метод детектирования Виолы-Джонса чувствителен к сильным поворотам лица, каскады для глаз чувствительны к очкам, эта проблема решается путем использования каскадов, специально обученных для детектирования глаз в очках или же лиц в профиль. Так же детектирование рта может работать некорректно в случае наличия обильной растительности на лице, GrabCut может посчитать за искомый объект именно усы, а не рот. Результаты работы предложенной методики изображены на рисунке 6, найденные рот и глаза выделены цветом.



Рисунок 6 – Детектирование рта и глаз

Заключение

В данной работе была представлена методика автоматического поиска лиц и выделения на них таких элементов как рот и глаза. Представлены отдельные этапы данной методики, указаны ограничения на применение, как отдельных этапов, так и всей методики в целом.

Работа частично поддержана Российским фондом фундаментальных исследований (проекты 13-07-00459, 13-07-97042, 14-07-97016, 15-07-06322).

Библиографический список

[Алексеев и др., 2012] Алексеев, А.В. Автоматизация определения шрифтов по изображению / Алексеев А.В., Розалиев В.Л. // Труды конгресса по интеллектуальным системам и информационным технологиям "IS&IT'12" (Дивноморское, Краснодарский край, 2-9 сент. 2012 г.). В 4 т. Т. 1 : тр. конф. "Интеллектуальные системы '12" и "Интеллектуальные САПР – 2012" / ЮФУ [и др.]. - М., 2012. - С. 292-293.

[Заболоева-Зотова и др., 2010] Автоматизация начальных этапов проектирования программного обеспечения / Заболоева-Зотова А.В., Орлова Ю.А. // Изв. ВолгГТУ. Серия «Актуальные проблемы управления, вычислительной техники и информатики в технических системах»: межвуз. сб. науч. ст. - Волгоград, ВолгГТУ. 2010. - Вып. 8, № 6. - С. 121-124.

[Розалиев и др., 2010] В.Л. Розалиев, А.С. Бобков, О.С. Федоров Применение нейронных сетей и грануляции при построении автоматизированной системы определения эмоциональных реакций человека / Изв. ВолгГТУ. Серия «Актуальные проблемы управления, вычислительной техники и информатики в технических системах»: межвуз. сб. науч. ст. / ВолгГТУ. - Волгоград, 2010. - Вып. 9, № 11. - С. 63-68.

[Jan et al, 2010] Jan Šochman, Jiří Matas, «AdaBoost», Center for Machine Perception, Czech Technical University, Prague, 2010

[Mitchell et al] S. C. Mitchell, J. G. Bosch, B. P. F. Lelieveldt, R. J. van der Geest, J. H. C. Reiber, and M. Sonka. 3-d active appearance models: Segmentation of cardiac MR and ultrasound images. IEEE Trans. Med. Imaging, 21(9):1167–1178, 2002

[Nefian, 1999] Nefian, A.V. Hidden Markov Model-Based Approach for Face Detection and Recognition / Ara Nefian. 1999.

[Nicolls et al, 2010] Nicolls S.M. and J.M. and F. The MUCT Landmarked Face Database // Pattern Recognit. Assoc. South Africa. 2010.

[OpenCV, 2014] OpenCV – Open computer vision library, Itseez, Retrieved from <http://opencv.org>, 2014

[Prabhu et al] Prabhu U., Seshadri K. Facial Recognition Using Active Shape Models, Local Patches and Support Vector Machines // contrib.andrew.cmu.edu. P. 1–8.

[Vezhnevets et al, 2003] Vezhnevets V., Sazonov V., Andreeva A. A survey on pixel-based skin color detection techniques // Proc. Graph. Moscow, Russia, 2003. - P. 85–92.

[Viola et al, 2001] P. Viola and M.J. Jones, «Rapid Object Detection using a Boosted Cascade of Simple Features», proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR 2001), 2001

CONTEXT-SENSITIVE ANALYSIS OF PORTRAIT IMAGES

Alekseev A. V., Orlova Yu. A.

Volgograd State Technical University

Alekseev.yeskela@gmail.com

This paper presents a method of context-sensitive analysis of the portrait photos. The entire analysis automatically, no user interaction is required. The presented method allows to locate the person and find the contours of the mouth and eyes. Finally, examples of work on the test photos.

Introduction

Currently, due to the rapid progressing of digital photography and digital video pattern recognition in digital images and faces in particular is very perspective task.

The aim of this work is the automatic search of faces on the image and contour selection for each person outlines the following elements: mouth, eyes.

Main Part

The proposed method consists in the following:

1. Detection of persons on the basis of the viola-Jones;
2. All required elements will be depending on the found rectangle face, based on human physiology;
3. To localize the mouth we will be on the basis of heuristically selected ratio relative to the rectangle face. The delimitation of the mouth are we going to do with the method of the GrabCut segmentation;
4. To localize the eyes we will also use the methods of the viola-Jones, we clarify the boundaries on the basis of skin color

Conclusion

This work was presented a technique to automatically search faces and allocation of items such as mouth and eyes. This paper presents the individual steps of this method, constraint specified on the application as the individual steps and the overall methodology.



OSTIS-2015

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822:514

МЕТОД ХРАНЕНИЯ ИЕРАРХИЧЕСКИХ СЕМАНТИЧЕСКИХ СЕТЕЙ В РЕЛЯЦИОННОЙ БАЗЕ ДАННЫХ

Ванясин Н.В., Сидоркина И.Г.

*Поволжский государственный технологический университет,
г. Йошкар-Ола, Республика Марий Эл, Россия*

nikita.vanyasin@gmail.com

igs592000@mail.ru

В данной статье предложено решение задачи хранения множества семантических иерархических сетей в реляционной базе данных. Решение базируется на иерархических семантических сетях, определенных в программной системе iSpring Cloud. Особое внимание уделено оптимизации существующего метода хранения, проведенной в целях уменьшения дублирования данных. В работе скомбинированы алгоритмы операций над семантическими сетями и алгоритмы хранения иерархических данных в реляционной базе данных.

Ключевые слова: семантические сети; метаданные; хранение иерархических данных.

Введение

При разработке программной системы iSpring Cloud (компания iSpringSolutions) [iSpringCloud, 2014] возникла задача хранения структур папок и файлов которые загружаются пользователями системы. Каждый пользователь в системе обладает своей автономной структурой папок и файлов и может управлять ею (создавать папки, загружать файлы, копировать и перемещать файлы и папки и т.д.). Кроме того пользователи могут делиться с другими пользователями своими папками и файлами для просмотра и редактирования. Программная система уже использует реляционную базу данных, т.о. предложено организовать хранение иерархических структур папок и файлов пользователей так же в реляционной базе данных, что упростило бы разработку.

Структура папок и файлов пользователя фактически является семантической сетью с иерархическими отношениями «папка (предок) – файл (потомок)». Так как планируемое количество будущих пользователей программной системы может быть большим (от 20000 и более), а скорость отображения иерархии для конкретного пользователя должна быть достаточно высокой, и пользователь не должен ощущать задержек при работе с системой (то есть в пределах 1-2 секунд), возникает задача хранения семантических иерархических сетей для большого количества пользователей в реляционной БД (причем в таких структурах данных, которые были бы пригодны для

быстрого получения иерархии при помощи операций реляционной алгебры).

Основными проблемами, стоящими на пути к решению данной задачи являются:

- У каждого пользователя есть возможность открыть доступ к своим папкам и файлам любому количеству других пользователей в системе. Соответственно при «наивной» реализации этой возможности через копирование части иерархии от одного пользователя к другому появится дополнительное количество избыточных данных, что замедлит в целом работу с семантической сетью.

- Отсутствие возможности использования рекурсивных операций в СУБД (MySQL 5.6), которая используется для работы с базой данных в программной системе iSpring Cloud.

1. Выбор метода хранения иерархических данных в реляционной БД

В настоящее время существует множество подходов к управлению иерархическими структурами данных. Наиболее исследованными, обладающими высокой эффективностью хранения и обработки данных в реляционных системах, являются методы управления деревьями [Маликов, 2008]. Основными методами хранения иерархических данных в реляционной системе являются:

- Adjacency List (список смежных вершин);
- Materialized Path (материализованный путь);
- Nested Sets (вложенные множества);
- Closure Table (таблица замыканий).

Рассмотрим подробнее каждый из методов.

1.1. Adjacency List (список смежных вершин)

Отношение «предок-потомок» хранится вместе с данными как простая ссылка на прямого предка.

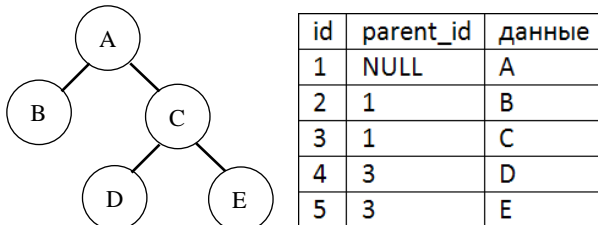


Рисунок 1 – Adjacency List

На рисунке 1 представлен пример иерархической структуры данных и её хранение с использованием этого метода. Основной проблемой при использовании такого способа хранения – невозможность быстро получить всю иерархию полностью, так как получение каждого следующего уровня требует выполнения операции соединения, что накладывает ограничение на максимально возможную глубину структуры (дерева) [Смусенко и др., 2013].

1.2. Materialized Path (материализованный путь)

Для каждого узла в структуре иерархических данных хранится ещё и перечисление всех его предков.

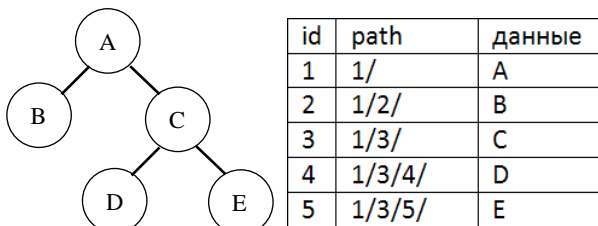


Рисунок 2 – Materialized Path

На рисунке 2 представлен пример иерархической структуры данных и её хранение с использованием этого метода. Так как перечисление предков хранится в одном элементе кортежа, основным недостатком данного метода являются дополнительные накладные расходы на работу с этим элементом: проверки на отсутствие циклов, конкатенация строк и т.д. [Celko, 2012]

1.3. Nested Sets (вложенные множества)

Для каждого узла в структуре иерархических данных добавляется два дополнительных поля: left (левая граница множества, некоторое число, которое меньше чем все числа которые используют потомки

текущего узла) и right (правая граница множества, число, которое меньше чем все числа которые используют потомки текущего узла). То есть отношение «предок-потомок» определяется вхождением потомка в промежуток (left; right) предка.

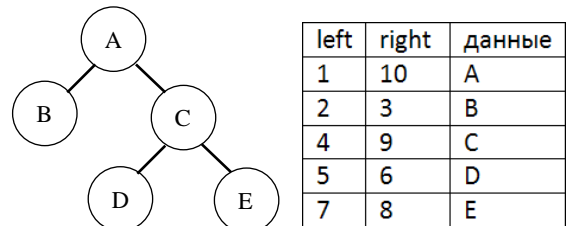
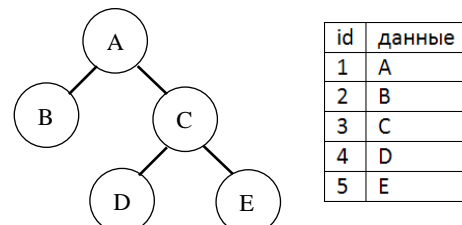


Рисунок 3 – Nested Sets

На Рисунке 3 представлен пример иерархической структуры данных и её хранение с использованием этого метода. Использование nested sets позволяет избавиться от ограничения на максимальную возможную глубину дерева, а совместное использования этого метода с методом Adjacency List позволит извлекать всю структуру иерархических данных при помощи одной операции реляционной алгебры. Однако затраты на вставку новых узлов и перенос требуют большого количества дополнительных операций для пересчета значений left и right, что особенно сильно проявляется на иерархических структурах с большой глубиной вложенности [Karwin, 2010].

1.4. Closure Table (таблица замыканий)

Используется два реляционных отношения – одно перечисляет узлы семантической сети и содержит данные, а другое перечисляет все связи (замыкания) между узлами (в том числе косвенные).



parent_id	child_id	depth
1	1	0
1	2	1
1	3	1
1	4	2
1	5	2
2	2	0
3	3	0
3	4	1
3	5	1
4	4	0
5	5	0

Рисунок 4 – Closure Table

На Рисунке 4 представлен пример иерархической структуры данных и её хранение с использованием этого метода. Кроме самих данных хранятся метаданные семантической сети – информация для каждого узла, «родителем» каких

«потомков» он является и на каком уровне вложенности. Извлечение иерархической структуры данных при таком методе включает в себя операцию соединения этих отношений по предикатам `child_id` и `parent_id`. Извлечение структуры из реляционной системы при использовании данного метода не требует вызова рекурсивных операций, что устраняет проблему отсутствия возможности выполнения рекурсивных операций при помощи используемой в системе СУБД [Karwin, 2010]. Предикат `depth` может использоваться для выборки одного уровня иерархии, либо в целях оптимизации запросов в СУБД.

Данный метод хранения использует разные реляционные отношения для хранения данных

- об узлах семантической сети;
- о метаданных – связях между узлами в сети,

что позволяет использовать одни и те же узлы в семантических сетях разных пользователей, упрощая реализацию возможности совместного использования одной сети разными пользователями. По этой причине для решения поставленной задачи хранения иерархической семантической сети в реляционной базе данных был выбран метод Closure Table (таблица замыканий).

2. Описание метода хранения иерархической семантической сети

Для решения задачи хранения иерархической семантической сети предлагается добавить еще один предикат в таблицу замыканий для связи иерархической семантической сети с конкретным пользователем программной системы, что позволит хранить иерархии всех пользователей системы и совершать операции над множеством иерархий сразу. Таким образом реляционное отношение будет выглядеть следующим образом: (`user_id`, `parent_id`, `child_id`, `depth`). Данный метод хранения позволяет производить все основные операции над семантической сетью без задержек, заметных для пользователя системы.

Рассмотрим основные операции.

Выборка всей иерархии папок конкретного пользователя: `SELECT * FROM node INNER JOIN closure ON closure.child_id = node.id WHERE closure.user_id = $currUserId`

Добавление нового узла в иерархию папок конкретного пользователя: `INSERT INTO closure (user_id, parent_id, child_id, depth) VALUES ($userId, $nodeId, $nodeId, 0); INSERT INTO closure (user_id, parent_id, child_id, depth) SELECT $userId, p.parent_id, c.child_id, (p.depth+c.depth+1) FROM closure p, closure c WHERE p.user_id=$userId AND c.user_id=$userId AND p.child_id=$parentId AND c.parent_id = $nodeId;`

Первый запрос вставит замыкание узла на самого себя, а второй – добавит необходимые замыкания для всех «предков» нового узла.

Удаление поддерева: `DELETE FROM closure WHERE user_id = $userId AND child_id = $nodeId;`

Перемещение поддерева представляет собой сначала «отсоединение» поддерева от основной иерархии и последующее «подсоединение» путем добавления новых связей «предок-потомок».

Одним из недостатков описанного метода хранения иерархической семантической сети в реляционной системе является увеличение объема избыточных данных при использовании возможности открытия доступа к редактированию иерархии другим пользователям программной системы. При «наивной» реализации данной возможности через копирование части иерархии от одного пользователя к другому объем избыточных данных может возрасти многократно.

Чтобы исключить данный недостаток предлагается провести денормализацию таблицы замыканий и добавить дополнительный предикат – идентификатор пользователя, который владеет узлом иерархии, с целью введения связей между иерархиями семантических сетей различных пользователей. Таким образом реляционное отношение будет выглядеть следующим образом: (`user_id`, `parent_id`, `child_id`, `depth`, `owner_id`). Соответственно при использовании возможности открытия доступа к части иерархии другим пользователям системы, пользователи будут редактировать часть иерархии, которой владеет первый пользователь.

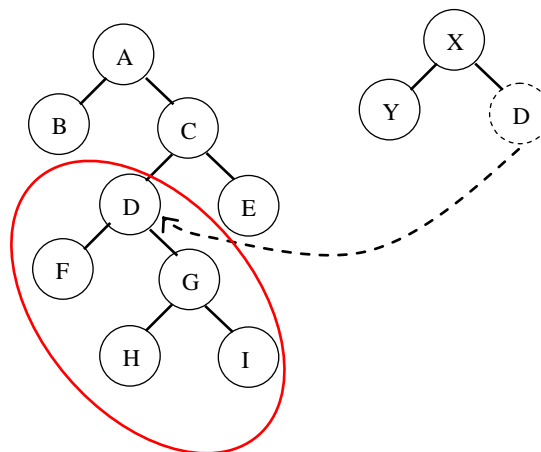


Рисунок 5 – Связь двух семантических сетей

На рисунке 5 в левой части показана иерархия первого пользователя, который открыл доступ к поддереву узла D второму пользователю (его семантическая сеть показана в правой части рисунка). Таким образом в дереве второго пользователя при открытии ему доступа добавляется только один дополнительный узел, который фактически является ссылкой на узел в другой семантической сети, что обеспечивает

построение связей между иерархическими сетями разных пользователей и уменьшение объема дублируемых данных.

Предложенное решение позволяет извлечь иерархическую структуру данных пользователя не более чем за две операции выборки:

- выборка первичных ключей всех пользователей, которые открыли доступ к своей иерархии текущему пользователю. `SELECT DISTINCT owner_id FROM closure WHERE user_id = $currUserId;`
- выборка всей иерархической структуры, которая доступна текущему пользователю. `SELECT * FROM node INNER JOIN closure ON closure.child_id = node.id WHERE closure.user_id = $currUserId OR closure.user_id IN ($foreignTreeOwnerIds)`

Заключение

В работе предложено и обосновано решение о хранении иерархической семантической сети в реляционной базе данных. Разработан метод хранения, который позволяет выстраивать связи между однотипными сетями. Представленный метод доведен до практической реализации в программной системе iSpring Cloud.

Библиографический список

[Маликов, 2008] Маликов, А.В. Ориентированные графы в реляционных базах данных / А. В. Маликов // Доклады ТУСУРа, No 2 (18), часть 2, декабрь 2008 С. 100-104

[Смусенок и др., 2013] Анализ способов представления иерархических структур в реляционных базах данных с использованием стресс-тестов / Смусенок С.А. [и др.]; // Открытые информационные и компьютерные интегрированные технологии №62, 2013 С. 107-111

[Karwin, 2010] Karwin, Bill, SQL Antipatterns: Avoiding the Pitfalls of Database Programming / Bill Karwin. – Pragmatic bookshelf. – 2010. - P. 34-53

[Celko, 2012] Celko, Joe, Trees and Hierarchies in SQL for Smarties, Second Edition / Joe Celko. – Morgan Kaufmann. – 2012. – P. 296.

[iSpringCloud, 2014] iSpring Cloud Home Page <http://www.ispringcloud.com/> (дата обращения: 27.11.2014).

HIERARCHICAL SEMANTIC NETWORK STORAGE METHOD FOR RELATIONAL DATABASES

Vanyasin N.V., Sidorkina I.G.

*Volga State University of Technology, Yoshkar-
Ola, Republic of Mari El, Russia*

nikita.vanyasin@gmail.com

igs592000@mail.ru

This article proposes a solution to the problem storing a plurality of hierarchical semantic networks in a relational database. Particular attention is paid to optimizing existing storage method, carried out in order to reduce duplication of data. In this paper algorithms combined operations on semantic networks and

algorithms for storing hierarchical data in a relational database.

Keywords: semantic networks, metadata, hierarchical data storage.

Introduction

During designing a software system iSpring Cloud (iSpring Solutions Inc.) there was a problem of storage structures of folders and files that are uploaded by users of the system.

Structure of folders and files the user is actually the semantic network with hierarchical relationships "folder (ancestor) - file (descendant)". Thus, here arises a problem of semantic hierarchical networks storage method for a large number of users in a relational database (and in such data structures that would be suitable for quick hierarchy extraction using relational algebra operations)

Main Part

The most researched, highly effective storage methods for hierarchy in relational systems are tree management method. The basic methods for storing trees in relational systems are adjacency list, materialized path, nested sets and closure table. Extracting hierarchy structures from a relational system with closure table method does not require recursive operations that eliminates the problem of inability to perform recursive operations using the database management system used in the software product.

This storage method uses different relations for storing nodes of a semantic network separate from metadata (the connections between nodes in the network), that allows the use of the same nodes in the semantic networks of different users. This storage method allows you to perform all basic operations on a semantic network without noticeable delays. Also there are proposed improvements to storage method, that are targeted to use one relation for storing semantic networks for all users. The proposed solution allows to extract a hierarchical structure of user data is not more than two sampling operations.

Conclusion

In this whitepaper we propose and justify a decision to keep a hierarchical semantic network in a relational database. Proposed method allows you to build connections between the networks of same type. The presented method is brought to the practical implementation of a software system iSpring Cloud.



OSTIS-2015

(Open Semantic Technologies for Intelligent Systems)

УДК 005.8:615.478

NP-ЗАДАЧА БАЛАНСИРОВКИ НАГРУЗОЧНОЙ СПОСОБНОСТИ СЕТИ В САПР

Сидоркина И.Г. Сорокин О.Л.

*Поволжский государственный технологический университет,
г. Йошкар-Ола, Россия*

oleg-ussr2@yandex.ru

igs592000@mail.ru

В работе представлены основные актуальные сетевые задачи, требующие решения в современных САПР. Рассмотрено понятие семантики сети, а также нагрузочной способности, необходимых для балансировки нагрузки распределенных САПР. Применяя байесовские вероятностные характеристики к данной задаче решаем ее комплексно, с учетом априорных вероятностей.

Ключевые слова: распределенная САПР; семантика сети; балансировка; нагрузочная способность; теорема Байеса.

Введение

Одной из наиболее актуальных задач систем автоматизированного проектирования (САПР), требующих решения в настоящее время является проблема распределения сетевого трафика [Суханов,2012]. Суть этой проблемы заключается в ограниченной нагрузочной способности канала сети, что приводит к значительным задержкам всего процесса разработки конечного продукта и сводит к минимуму эффективность вычислений различных узлов конечной системы. Сложность такой задачи, прежде всего, заключается в необходимости комплексного подхода к решению проблемы баланса нагрузки и в том числе использование уже существующих методов, основанных на использовании лингвистики [Стецко,2006]. Однако наилучшего распределения, применимо к данной проблеме можно достичь, путем исследования общей семантики сети и исследования возможности баланса нагрузки в различных узлах системы. При балансировке в данном случае используются как методы обработки информации передаваемой в сети, так и методы, реализуемые непосредственно в ходе функционирования уже реализованной сетевой семантики.

1. Особенности структуры распределенной САПР

Под *семантикой* сети понимается смысл и назначение составляющих сети (узлов) и их взаимодействие. Процесс взаимодействия включает

в себя обмен информацией между узлами, а также неопределенность в выборе наилучшего маршрута для связи узлов сети и балансировки нагрузки. Наличие семантики, а также присутствие неопределенности при выполнении балансировки нагрузочной способности позволяет отнести данную задачу к разряду NP-полных, а смысловое значение можно определить, опираясь на классическую задачу коммивояжера [Суханов, 2012].

Нагрузочная способность – один из важнейших параметров, например для узла сети, показывающий возможность передачи определенного количества данных в единицу времени по этому узлу сети без потерь.

Рассмотрим задачу балансировки нагрузки в сети современной САПР, например предназначенной для проектирования схем, трассировки, компоновки и анализа печатных плат. Общая структура распределенной САПР представлена на рисунке 1.

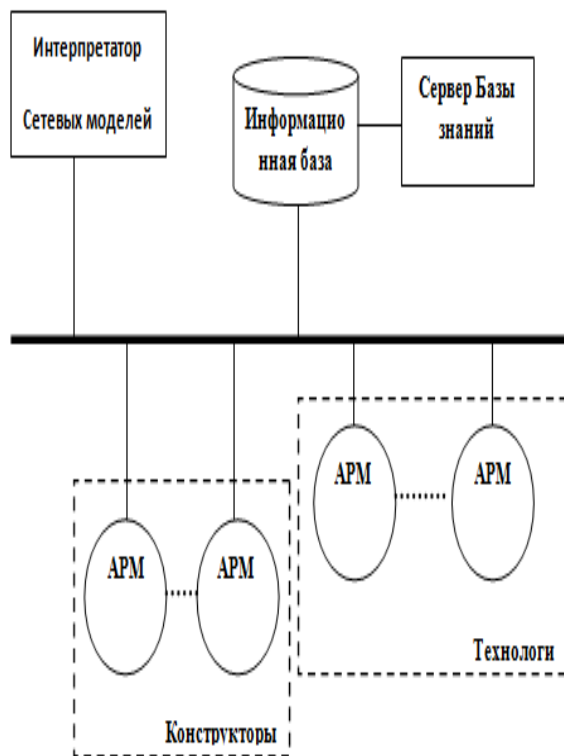


Рисунок 1 - Структура распределенной САПР

Выделим некоторые узлы рассматриваемой САПР, это: автоматизированные рабочие места технологов и конструкторов (АРМ) сервер баз знаний, файловый сервер, интерпретаторы сетевых моделей (ИСМ). Интерпретация сетевых моделей представляет собой инструмент для отображения процесса моделирования в среде САПР часто с использованием графического изображения модели.

Типовая технологическая цепочка проектирования в САПР включает следующие шаги:

- 1) подготовку описания детали (АРМ);
- 2) графическое моделирование;
- 3) генерация возможных вариантов решений;
- 4) вывод результата, сохранение на файловом сервере.

Установлено [Вашкевич, 2004], что формирование последовательности технологических переходов в системах с базой знаний осуществляется подсистемой логического вывода без участия технолога на основе использования базы технических знаний и результатов анализа текстового описания детали. Кроме того, также без участия технолога производится расчет технических требований. На основе информации с сервера баз знаний строится информационная среда проектирования [Сидоркина, 2014], а также реализуется логический вывод. ИСМ используются для организации сложных параллельных и конвейерных вычислений на основе принципа потока данных [Вашкевич, 2004]. С технической стороны узлы распределенной САПР представлены РС – совместимыми ПЭВМ в

конфигурации, соответствующей функциональному назначению узла. Так, например, сервер базы знаний по отношению к другим узлам должен иметь большой объем внешней памяти для хранения интенциональных и экстенциональных знаний и более высокое быстродействие.

2. Вычисление нагрузочной способности сети с использованием Байесовского подхода

Выше обосновано, что вычисление нагрузочной способности сети является одной из таких плохо формализуемых задач. Вычисление нагрузочной способности сети предполагает использование базы знаний и учета различных апостериорных вероятностей для определения имеющихся неопределенностей, влияющих на пропускную способность сети. Использование теоремы Байеса [Соловьев, 2012] дает возможность использовать вероятность гипотез с учетом наблюдавшегося результата опыта.

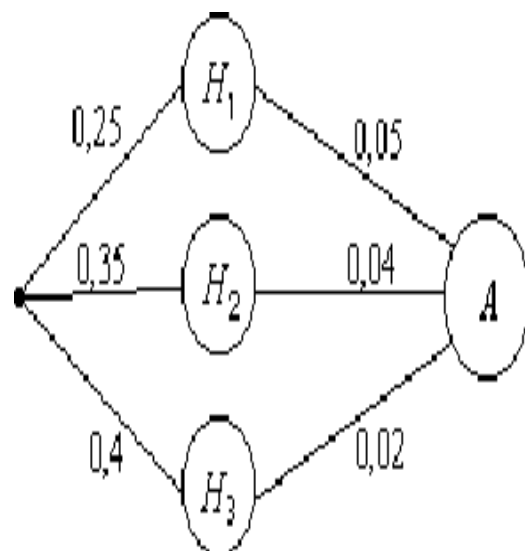


Рисунок 2 - Граф фрагмента исследуемой сети

Условная вероятность может находиться как отношение веса участка сети, проходящего через вершину, с соответствующей гипотезой, к весу всего вероятностного графа. Рассмотрим пример, когда известна пропускная способность трех участков сети. Обозначим их через отношение к общей суммарной пропускной способности всех трех участков, так что:

$P(H_1)$ – пропускная способность первого участка сети,

$P(H_2)$ – Пропускная способность второго участка сети,

$P(H_3)$ – Пропускная способность третьего участка сети.

$$P(H_1) = 0.25,$$

$$P(H_2) = 0.35,$$

$$P(H_3) = 0.4.$$

а также известны вероятности прохождения сигнала в сети равные соответственно 5%, 4% и 2% соответственно. Граф исследуемой сети представлен на рисунке 2. Выдвигаем три гипотезы:

- $H_1 = \{\text{сигнал прошел через первый участок сети}\}$,
- $H_2 = \{\text{сигнал прошел через второй участок сети}\}$,
- $H_3 = \{\text{сигнал прошел через третий участок сети}\}$,

тогда постаприорные гипотезы вероятности примут вид:

$P_1(A/H_1)$ – Сигнал успешно прошел первый участок сети, передача через участок H_1A .

$P_2(A/H_2)$ – Сигнал успешно прошел второй участок сети, передача через участок H_2A .

$P_3(A/H_3)$ – Сигнал успешно прошел третий участок сети, передача через участок H_3A .

$$P_1(A/H_1) = 0.05,$$

$$P_2(A/H_2) = 0.04,$$

$$P_3(A/H_3) = 0.02.$$

соответственно. Тогда легко можно определить вероятность того, что переданный сигнал по одному из трех каналов и выбранный случайным образом, будет передан:

$$P(A) = P(H_1) \times P(A/H_1) + P(H_2) \times P(A/H_2) + P(H_3) \times P(A/H_3),$$

$$P(A) = 0.25 \times 0.05 + 0.35 \times 0.04 + 0.4 \times 0.02 = 0.0345.$$

А также вероятность того что случайные данные были переданы по одному из трех каналов:

$P(H_1/A)$ – случайный сигнал передан по первому каналу

$P(H_2/A)$ – случайный сигнал передан по второму каналу

$P(H_3/A)$ – случайный сигнал передан по третьему каналу

$$P(H_1/A) = \frac{P(H_1) \times P(A/H_1)}{P(A)}$$

$$P(H_2/A) = \frac{P(H_2) \times P(A/H_2)}{P(A)}$$

$$P(H_3/A) = \frac{P(H_3) \times P(A/H_3)}{P(A)}$$

Рассчитаем численные значения вероятностей:

$$P(H_1/A) = \frac{0.25 \times 0.05}{0.0345} = \frac{25}{69},$$

$$P(H_2/A) = \frac{0.35 \times 0.04}{0.0345} = \frac{28}{69},$$

$$P(H_3/A) = \frac{0.4 \times 0.02}{0.0345} = \frac{16}{69}.$$

$$P(H_1/A) = 0,36231,$$

$$P(H_2/A) = 0,36231,$$

$$P(H_3/A) = 0,23188.$$

При большой связности графа такой подход обеспечит расчет наилучшего пути с наиболее подходящей пропускной способностью, что позволит решить задачу балансировки при реализации распределенной САПР.

Заключение

Таким образом, в области распределенных САПР задача балансировки нагрузки в сети осуществляется посредством анализа особенностей структуры современных САПР на основе рассмотрения семантики сети с использованием формулы Байеса. Данная задача, определяет неопределенность, которую можно вычислить путем введения гипотез и расчетов вероятностей для различных участков сети. В представленном примере в условиях неопределенности, выдвигая несколько гипотез, в результате обеспечивается более высокая эффективность работы подсистем САПР, за счет уменьшения времени передачи информации по сети. Использование постаприорных вероятностей позволяет решать плохоформализуемые задачи балансировки в сети посредством расчета вероятностей с большей точностью, за счет чего нагрузочная способность сети в распределенных САПР системах даже при сильной связности и условиях неопределенности будет определена наиболее точно.

Библиографический список

[Суханов,2012] Суханов В.И. Минимизация трафика в облачной инфраструктуре //В.И. Суханов// КубГАУ.-2012.- №78(04)- С. 1–10

[Стецко, 2006] Стецко А.А. Система моделирования и проектирования трафика телекоммуникационных сетей в условиях неопределенности // Информатика, системы искусственного интеллекта и моделирование технических систем: Труды Международной конференции «Континуальные

алгебраические логики, исчисления и нейроинформатика в науке и технике КЛИН-2006», Ульяновск, 2006.- Т. 2.- С.109-111.

[Вашкевич ,2004] Вашкевич Н.П.,Дубинин В.Н.,Зверев С.Л. Структура и функциональные возможности САПР ТП токарной обработки // Материалы Междунар. науч.-техн. конф. "Новые информационные техно-логии и системы" (НИТС'94). - Пенза, 1994. - С.143-144

[Сидоркина ,2014] Сидоркина И.Г. "САПР и интеллектуальные обучающие технологии", "NB: Кибернетика и программирование", № 1, 2014,с.23-47

[Соловьев,2012] Соловьев В. И. Методы оптимальных решений/ В. И. Соловьев//М.: Финансовый университет, 2012. - 364 с.

NP-BALANCING TASK RATED ABILITY OF NETWORK IN CAD

Sidorkina I.G. Sorokin O.L

Volga State University of Technology, Yoshkar-Ola, Russia

oleg-ussr2@yandex.ru

igs592000@mail.ru

The paper presents the main current network problems to be solved in modern CAD. Considers the concept semantic networks and the load capacity required for load balancing allocated CAD. By using Bayesian probabilistic characteristics to solve this problem, it is complex, with the a priori probabilities.

Introduction

One of the most actual tasks computer-aided design (CAD) to be resolved at the present time is the problem of the distribution of network traffic. The essence of the problem is the limited load capacity of the channel network, which leads to significant delays in the development process of the final product and to minimize the computational efficiency of different nodes of a finite system. The complexity of this problem, first of all, is the need for an integrated approach to solving the problem of load balancing and including the use of existing methods based on the use of linguistics.

Main Part

The presence of semantics and the presence of uncertainty when performing balancing the load capacity can be attributed to this task to the category of NP-complete, and the meaning can be determined based on the classic traveling salesman problem .Stress ability - one of the most important parameters, such as network node, showing the possibility of transferring certain amount of data per unit time on the network node without loss.

The conditional probability as the ratio may be weight network region, passing through a vertex corresponding to a hypothesis to the total weight of the probability graph. Then one can easily determine the probability that the signal transmitted by one of the three channels and randomly selected to be transmitted.

At high connectivity of the graph, this approach will provide the best path calculation with the most appropriate capacity, which will solve the problem of balancing the implementation of a distributed CAD.

Conclusion

So in distributed CAD load balancing problem in the network is carried out by analyzing the structure of modern CAD based network considering the semantics using Bayes' formulas. This task determines the uncertainty of which can be calculated by introducing a hypothesis and probability calculations for different sections of the network. In the present example in conditions of uncertainty, pushing several hypotheses provided as a result of the higher efficiency of the CAD subsystem, by reducing the information transmission time across the network.



OSTIS-2015

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822:514

АППАРАТНЫЙ ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ ДЛЯ СЕТЕВОЙ ИНФРАСТРУКТУРЫ САПР И СИСТЕМ УПРАВЛЕНИЯ

Шелеметьев А.М., Шелеметьева Я.В., Сидоркина И.Г.

*Поволжский государственный технологический университет,
г. Йошкар-Ола, Республика Марий Эл, Россия*

yanabaikova@yandex.ru

forandreika@yandex.ru

igs592000@ir.ail.ru

В работе обосновано применение аппаратного искусственного интеллекта для сетевой инфраструктуры САПР и систем управления. Описано применение интеллектуальных технологий в аппаратуре окончания канала данных (АОКД), раскрыты основные особенности выполнения интеллектуальных функций АОКД, представлена структурная схема интеллектуальной сетевой карты. Кроме этого дан пример реализации аппаратных интеллектуальных компонентов, используемых при создании системы управления транспортным средством, основанной на ПИД-регуляторе с применением таких методов настройки и подстройки коэффициентов ПИД-регулятора, как нечёткая логика, нейронные сети и генетические алгоритмы.

Ключевые слова: интеллектуальная сетевая карта; семантика протоколов передачи данных; Fuzzy-ПИД; нейро-ПИД; нейро-Fuzzy-ПИД; регуляторы с генетическими алгоритмами.

Введение

В настоящее время к вычислительным устройствам, отдельным его модулям и элементам, компьютерным сетевым системам предъявляются следующие требования: устройства и оборудование должны быть отказоустойчивыми, иметь компактный размер, решать вычислительные задачи быстро и эффективно. Вычислительные сетевые системы должны обладать высокой скоростью передачи данных, низкими задержками, пропускной способностью большого объема. Для удовлетворения указанных требований при разработке вычислительного оборудования и в работе вычислительных сетевых систем целесообразно применять интеллектуальные технологии.

1. Аппаратные методы искусственного интеллекта в системах автоматизированного проектирования

В силу комплексной компьютеризации сфер промышленного производства, интеллектуальные технологии нашли свое применение в системах оперативной обработки данных, в системах автоматизированного проектирования (САПР). Рабочими местами проектировщиков являются

вычислительные узлы, которые объединены между собой в вычислительные сети (основная структура технического обеспечения САПР) [Норенков, 2002]. Группы разработчиков, решающие общую задачу проектирования и отвечающие за реализацию разных функциональных блоков системы, объединенные вычислительной сетью для обмена данными, могут территориально находиться в разных зданиях и даже городах. Организация эффективного взаимодействия между удаленными участниками проектных групп – центральная проблема, возникающая при внедрении и дальнейшем функционировании распределенной САПР.

Для обеспечения эффективной работы открытой распределенной САПР: для своевременной, оперативной, высокоскоростной, высоконадежной с низкими задержками передачи данных, предлагается внедрить в оконечное оборудование данных (ООД) аппаратуру окончания канала данных (АОКД), осуществляющую интеллектуальную обработку входящей и исходящей сетевой информации. Здесь под АОКД будем понимать сетевую интерфейсную карту (сетевую плату), в которую перенесены функции обработки сетевых пакетов данных стека протоколов TCP/IP (для Ethernet сетей), которая реализует алгоритмы этого стека протоколов и выполняет следующие

интеллектуальные функции [Friedman, 2001], [InfinitiBand, 2007]:

Адаптация передаваемого пакета по правилам промежуточных локальных сетей. Для взаимодействия двух удаленных вычислительных узлов по сети необходимо, чтобы семантика формата заголовков пакета данных, отправляемого одним узлом, было понятна узлу-получателю пакета. Для этого интеллектуальная сетевая карта формирует сетевой пакет на основе имеющихся у него данных о правилах взаимодействия вычислительных узлов в конкретной сети. Узел-получатель, анализируя заголовки сетевого пакета на основе этих же правил, решает, что далее делать с сетевым пакетом: какому приложению или программе передать, в какую область памяти поместить и т.д.

Управление потоком данных с целью предотвращения блокировки памяти. Интеллектуальная сетевая карта за единицу времени успевает обрабатывать меньшее число пакетов, чем то, которое поступает из сети и ООД. Для хранения еще не обработанных пакетов в сетевой карте предусмотрены буферы памяти. Для предотвращения блокировки памяти, которая может возникнуть вследствие интенсивного сетевого трафика, сетевая карта на основе данных о длине пакета, содержащихся в заголовке каждого сетевого пакета, прогнозирует заполнение памяти и запрещает прием определенных пакетов.

Управление срочностью, надежностью, пропускной способностью данных. Для обеспечения требуемых параметров вычислительной сети семантикой сетевых протоколов предусмотрена возможность управления срочностью, надежностью и пропускной способностью данных с помощью установки определенных бит заголовков сетевых пакетов в нужное значение. Анализируя входящие и исходящие пакеты данных все сетевое оборудование вычислительной сети и интеллектуальные сетевые платы в том числе, принимают решение, на каком уровне качества обслуживания (QoS) передавать сообщение.

Принятие решения о способе передачи данных в ООД на основе анализа входящих и исходящих сетевых данных на уровне заголовков сетевого пакета. Интеллектуальная сетевая карта может обрабатывать сетевые пакеты данных разного типа. Например, в случае если в сети возможна передача данных и с помощью технологии удаленного прямого доступа к памяти RDMA, и с помощью стандартных механизмов передачи данных (стек протоколов TCP/IP), то на основе анализа заголовков сетевого сообщения, сетевая карта принимает решение о том, в какую область памяти поместить входящие данные: в оперативное запоминающее устройство (ОЗУ) ООД (для дальнейшей обработки данных процессором ООД) или напрямую в память приложения.

Если не внедрять перечисленные функции в сетевые платы, большую часть операций по обработке сетевых данных будет выполнять процессор [Шелеметьева, 2013]. При интенсивном сетевом трафике, который сопровождает распределенное решение задачи САПР, процессор вынужден часто прерываться на обработку сетевых операций, «отвлекаясь» при этом от своей непосредственной задачи – задачи проектирования. Если сетевой трафик очень интенсивен, то возможно не только увеличение времени решения задачи проектирования, но и полная загрузка и сбой в работе процессора, который ведет к простоему вычислительного узла, а значит, и всей сетевой системы в целом. Освобождая процессор от обработки части сетевых данных, мы не только сокращаем время решения задачи САПР, но и повышаем надежность системы.

Для реализации интеллектуальных функций в устройстве сетевой платы вводят дополнительные функциональные блоки. Если интеллектуальные сетевые платы поддерживают технологию удаленного прямого доступа к памяти RDMA, то в устройство добавляется механизм преобразования виртуального адреса (которым оперируют приложения) в физический. За реализацию механизма отвечают блоки: «блок защиты памяти» и «блок преобразования виртуального адреса в физический». Помимо непосредственного преобразования адреса, блоки отвечают за права доступа к памяти приложения и работают с ключами доступа, которые используются для организации защиты памяти от несанкционированного доступа.

В устройстве сетевой платы выделяют следующие блоки: блоки сопряжения со средой и системной шиной отвечают за преобразование данных, необходимое из-за разницы представления данных в сетевом адаптере, среде передачи и системной шине. Блок анализа заголовков пакетов определяет тип входящего пакета от ЭВМ. Блоки формирования кадров IEEE 802.2 и 802.3ae, IP и TCP пакетов анализируют заголовки соответствующих кадров и пакетов. Блоки формирования пакетов: RDMA, TCP, IP, IEEE 802.2 и 802.3ae добавляют сетевые заголовки в зависимости от типа пакета. Блок определения RDMA пакетов – определяет, является ли данный пакет RDMA. Блок расчета физического адреса в ОЗУ – для стандартных команд передачи (без RDMA) определяет буфер памяти, куда следует поместить пакет. Блок формирования выходного пакета формирует конечный пакет для приложения. Блок преобразования виртуального адреса в физический транслирует адреса памяти приложения в физические адреса ОЗУ. Блоки формирования адреса в ОЗУ выходных пакетов и в ОЗУ входных пакетов отвечают за определение адреса, куда будут записываться данные в ОЗУ.

Применение интеллектуальных сетевых плат в ООД вычислительных сетей, спроектированных и

разработанных для обеспечения эффективной работы распределенных САПР, позволит улучшить два основных показателя (пропускная способность и задержка передачи данных), которые характеризуют работу вычислительной сети. За единицу времени интеллектуальные сетевые адаптеры передают с низкими задержками данные большого объема, по сравнению с адаптерами, в которых интеллектуальные функции не реализованы.

2. Аппаратные методы искусственного интеллекта в системах автоматизированного управления

Другой пример реализации аппаратных интеллектуальных компонентов применяется при разработке колесного транспортного средства, где в качестве двигателя используются мотор-колеса, в основе которых находится вентильный электродвигатель, требуется создание системы управления данным транспортным средством. В её состав входят контроллер управления, система датчиков, коммутационный блок управления двигателями, соединенные между собой универсальным интерфейсом, в качестве которого может использоваться, широко применяемая CAN – шина. В подобной системе, учитывающей многократные изменения скорости в единицу времени вращения каждого из электродвигателей, требуется высокая надежность не только интерфейсов взаимодействия электронных компонентов, но и алгоритма по которому данные изменения происходят. Такую систему можно назвать системой оперативной обработки данных.

В статье [Смирнов А.В., 2013] описана методика настройки пропорционально-интегрально-дифференциального (ПИД) регулятора для управления электродвигателем Gekko MR12-100. ПИД-регулятор - устройство в управляющем контуре с обратной связью. Используется в системах автоматического управления для формирования управляющего сигнала с целью получения необходимых точности и качества переходного процесса. Суть методики заключается в том, чтобы в зависимости от подобранных коэффициентов ПИД регулятора, менять напряжение, подаваемое на каждую из обмоток электродвигателя, с целью обеспечения наибольшей точности регулирования скоростью вращения электродвигателем.

В первую очередь для правильного подбора коэффициентов ПИД – регулятора требуется найти следующие характеристики двигателя: $T_{\text{Э}}$ – электрическая постоянная времени, $T_{\text{м}}$ – электрохимическая постоянная времени, $T_{\text{ж}}$ – желаемое время работы системы, $K_{\text{дв}}$ – коэффициент двигателя. В результате подставив полученные значения в формулы, получим необходимые нам коэффициенты ПИД-регулятора.

$$W = \frac{T_{\text{м}} - T_{\text{Э}}}{T_{\text{ж}} \times K_{\text{дв}}} + \frac{1}{T_{\text{ж}} \times K_{\text{дв}}} + \frac{T_{\text{м}} \times T_{\text{Э}}}{T_{\text{ж}} \times K_{\text{дв}}} \quad (1)$$

где $K_{\text{п}} = \frac{T_{\text{м}} - T_{\text{Э}}}{T_{\text{ж}} \times K_{\text{дв}}}$ – коэффициент

пропорционального звена $K_{\text{и}} = \frac{1}{T_{\text{ж}} \times K_{\text{дв}}}$ –

коэффициент интегрального звена $K_{\text{д}} = \frac{T_{\text{м}} \times T_{\text{Э}}}{T_{\text{ж}} \times K_{\text{дв}}}$

– коэффициент дифференциального звена.

Расчет этих постоянных по времени параметров электродвигателя ($T_{\text{Э}}$, $T_{\text{ж}}$) рассчитывается на основе времени установления скорости вращения при подаче номинального напряжения.

Однако ПИД-регулятор, полученный данным методом, имеет неточные показатели качества при управлении нелинейными и сложными системами какой является система оперативной обработки данных, а также при недостаточной информации об объекте управления (характеристики электродвигателя, такие как крутящий момент, сопротивления обмоток и др.). Характеристики регуляторов в этих случаях можно улучшить с помощью методов нечёткой логики, нейронных сетей и генетических алгоритмов. В одном контроллере могут применяться комбинации перечисленных методов (Fuzzy-ПИД, нейро-ПИД, нейро-Fuzzy-ПИД регуляторы с генетическими алгоритмами) [Денисенко, 2007].

Основным недостатком нечётких и нейросетевых контроллеров является сложность их настройки (составления базы правил и обучения нейронной сети).

Настройка ПИД-регулятора по формулам обычно не является оптимальной и может быть улучшена с помощью дальнейшей подстройки. Подстройка может быть выполнена оператором на основании эвристических правил или автоматически, с помощью блока нечёткой логики. Блок нечёткой логики (Fuzzy - блок) использует базу правил и методы нечёткого вывода. Fuzzy - подстройка позволяет уменьшить перерегулирование, снизить время установления и повысить робастность ПИД-регулятора [Воронов, 1976].

Нейронные сети, как и нечёткая логика, подразумевают использование в ПИД-регуляторах двумя способами: для построения самого регулятора и для построения блока настройки его коэффициентов. Особенностью нейронной сети является способность к «обучению», что позволяет передать нейронной сети опыт эксперта. В отличие от нечёткого регулятора, где эксперт должен сформулировать правила настройки в лингвистических переменных, при использовании нейронной сети от эксперта не требуется формулировки правил, достаточно, чтобы он

несколько раз сам настроил регулятор в процессе «обучения» нейронной сети.

Еще один метод, которым можно воспользоваться при поиске наиболее верных значений ПИД регулятора являются генетические алгоритмы. Такие алгоритмы являются мощным методом оптимизации, позволяющим найти глобальный оптимум быстрее, чем другие методы случайного поиска. Существенным их достоинством является отсутствие проблем со сходимостью и устойчивостью. Эти методы используются для идентификации моделей объектов управления, для поиска оптимальных параметров регулятора, для поиска оптимальных положений функций принадлежности в Fuzzy-регуляторах и для «обучения» нейронных сетей. Чаще всего генетические алгоритмы используются совместно с нейронными сетями и регуляторами с нечёткой логикой [Денисенко, 2007].

Заключение

Применение аппаратных методов искусственного интеллекта в сетевой инфраструктуре САПР позволит повысить пропускную способность вычислительной сети, в которой САПР реализована, снизить задержки передачи данных между распределенными узлами пользователей. Применение интеллектуальных технологий при настройке и подстройке коэффициентов ПИД-регулятора в автоматизированной системе управления обеспечит оптимальность его работы, что повысит надежность работы регулятора, а значит и всей системы в целом.

Библиографический список

- [Смирнов А.В., 2013] Смирнов А.В. Методика настройки ПИД-регулятора скорости коллекторного двигателя GEKKOMR 12-100 / А.В. Смирнов, А.М. Шелеметьев // Человек, общество, природа в эпоху глобальных трансформаций. Шестнадцатые Вавиловские чтения: материалы постоянно действующей международной междисциплинарной научной конференции в 2 частях. Ч. 2. Йошкар-Ола: Поволжский государственный технологический университет, 2013, С. 403-405
- [Денисенко, 2007] Денисенко В.В. ПИД – регуляторы: принципы построения и модификации / В.В. Денисенко // Современные технологии автоматизации, 2007, №1, С. 78-88.
- [Воронов, 1976] Воронов А.А. Устойчивость управляемости, наблюдаемости / А.А. Воронов // – М.: Наука, 1979, 336 с.
- [Норенков, 2002] Норенков И.П. Основы автоматизированного проектирования: Учеб. Для вузов. / И.П. Норенков // М.: Изд-во МГТУ им. Н.Э. Баумана, 2002, 336 с.
- [Friedman, 2001] Friedman D. Building firewalls with intelligent network interface adapter card / D. Friedman, D. Nagle // School of Computer Science, Carnegie Mellon University - 2001- pp. 19 p.
- [Culley, 2007] Recio R., Metzler B., Culley P. A Remote Direct Memory Access Protocol Specification / P. Culley, B. Metzler, R. Recio // P.Fremont, CA: RFC, 2007 - pp. 66 p.
- [Шелеметьева, 2013] Шелеметьева Я.В. Исследование технологии удаленного прямого доступа к памяти в архитектурах высокопроизводительных систем / Я.В. Шелеметьева // Программные системы и вычислительные методы, 2013, № 3, С. 250-256.

HARDWARE ARTIFICIAL INTELLIGENCE FOR CADS NETWORK INFRASTRUCTURE AND FOR CONTROL SYSTEMS

Shelemeteva Y.V., Shelemetev.A.M.,
Sidororkina I.G.

**Volga State University of Technology, Yoshkar-Ola, Mari El Republic, Russia*
yanabaikova@yandex.ru
forandreika@yandex.ru
igs592000@ir.ail.ru

The application of hardware Artificial Intelligence in the network CAD infrastructure and in the management systems are justified in work. The example of hardware Intelligence components realisation which are used in the vehicle control system creating based on PID controller with fuzzy logic, neural networks and genetic algorithms methods is given in the article.

Introduction

Currently, the computing devices and its modules and components must provide required level of reliability, have compact size have, solve computational tasks quickly and efficiently. Computer network systems must provide high data rating, low latency, high-volume throughput. To meet above requirements it is advisable to apply intelligent technologies in the computing network systems functioning and during the computer equipment development.

Main Part

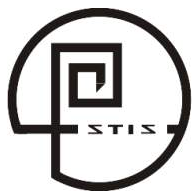
The intelligent network adapter card (iNIC) integrated in nodes where CAD is installed performs following processing s: adaptation of the transmitted packets according to the rules of intermediate LANs, memory lock prevention using data flow control, urgency, reliability, data bandwidth management.

The realization of RDMA technology in the iNIC allows to release node's CPU from network data processing and to increase the efficiency of CAD functioning

It is possible to integrate fuzzy logic, neural networks and genetic algorithms methods in the vehicle control system based on PID controller. Above mentioned technologies determine optimal PID odds.

Conclusion

In the given article it is shown how hardware artificial intelligence allows to increase the efficiency of CAD and automatic management system functioning.



OSTIS-2015

(Open Semantic Technologies for Intelligent Systems)

УДК 004.432.4

РЕШЕНИЕ ЗАДАЧИ ПРОГНОЗИРОВАНИЯ ВОЗРАСТАНИЯ НАГРУЗКИ В СИСТЕМЕ ДИСТАНЦИОННОГО ОБУЧЕНИЯ «ISPRING LEARN»

Сокольников А.М., Сидоркина И.Г.

*Поволжский государственный технологический университет,
г. Йошкар-Ола, Республика Марий Эл, Россия*

sokolnikov.alexey@gmail.com

igs592000@mail.ru

В статье исследуются способы прогнозирования высокой нагрузки для web-сервисов. Обосновывается необходимость их использования в системе дистанционного обучения iSpring Learn [iSpring Learn, 2014]. Раскрываются проблемы, которые возможно решить с помощью методов прогнозирования возрастания нагрузки и преимуществ их использования в связке с балансировщиком нагрузки.

Ключевые слова: Высоконагруженные системы; Система дистанционного обучения iSpring Learn; Балансирование нагрузки, Прогнозирование возрастания нагрузки.

Введение

Для разработчиков программного обеспечения сегодня все острее встает проблема обработки и хранения больших объемов данных. Количество просмотров страниц в социальной сети Facebook превышает 200 миллиардов в месяц [Блинков, 2010]. Для ускорения работы сервис использует несколько серверов, а обращения клиентов между ними распределяет особый класс программ, называемый «Балансировщик нагрузки».

Балансировка нагрузки — метод распределения заданий между несколькими сетевыми устройствами с целью оптимизации использования ресурсов, сокращения времени обслуживания запросов, горизонтального масштабирования кластера, а также обеспечения отказоустойчивости.

Большинство крупных сервисов используют данный класс программного обеспечения, но проблема заключается в том, что сервис не будет готов к резкому возрастанию нагрузки. Такая ситуация приведет к задержкам в выполнении запросов и возрастанию нагрузки на отдельно взятом сервере. Как один из вариантов решения этой проблемы предложено использование средства превентивного анализа высоких нагрузок.

1. Средство превентивного анализа нагрузки

Методы прогнозирования возрастания нагрузки

анализируют и обрабатывают запросы к серверу и сохраняют их метаданные. Кроме этого, они опрашивают состояние загруженности серверов. На основе этих данных производится поиск закономерностей выявления высоких нагрузок и вынесение решения о необходимости разворачивания дополнительных экземпляров сервиса.

Преимущество данного подхода заключается в том, что при использовании методов прогнозирования возрастания нагрузки в связке с балансировщиком, количество пиков запросов на один сервер значительно сократится, поскольку об авральной ситуации будет известно заранее до ее фактического возникновения.

Такие методы назовем средством превентивного анализа нагрузки, то есть средой, опережающей действия противной стороны [Комлев, 2006]. Превентивный анализ осуществляет обработку данных в запросах и пытается выявить закономерности в скачках нагрузок. Выделим четыре основных типа зависимостей, влияющих на нагрузку: календарные, временные, геолокационные и зависимости от источников трафика.

1.1. Метод выявления календарных закономерностей

Одной из закономерностей скачков нагрузки может служить зависимость сервиса от конкретных дат. Наиболее очевидный пример сервисов, нагрузка на которые зависит от календарных дат -

Автоматическая Телефонная Станция. В большинство будничных дней она справляется с возлагаемыми на нее нагрузками, однако в дни праздников у клиентов возникают проблемы с соединением.

Подобные проблемы можно наблюдать у таких сервисов, как социальные сети. Однако не стоит отрицать, что велика вероятность того, что проблема заключается не только на стороне сервиса, но и на стороне провайдера, предоставляющего доступ к сети в интернет.

Рассмотрим влияние календарных зависимостей на систему дистанционного обучения (СДО) iSpring Learn. Данную СДО используют более чем 2 000 ВУЗов, школ и компаний по всему миру. Минимальная численность организации – 50 пользователей. Самая крупная организация – более 23 000 пользователей. Годовой график нагрузки представлен на рисунке 1:

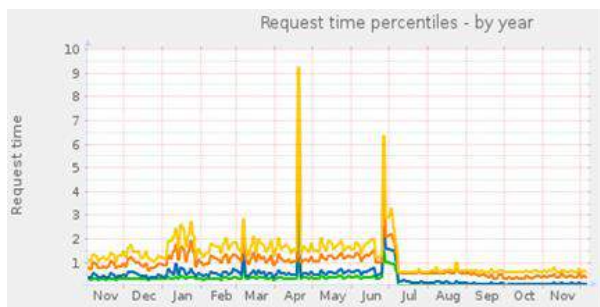


Рисунок 1 - Распределение нагрузки iSpring Learn за 2014 год

Помимо социальных сетей, зависимость от даты имеют периодические новостные издания. Примером может служить сайт журнала или блог, записи в котором публикуются с некой периодичностью. К примеру, каждую пятницу. Система превентивного анализа выявит эту закономерность и примет решение о необходимости дополнительных экземпляров сервиса.

Для выявления календарных закономерностей предложено использовать накопленные сервисом данные о нагрузке в разрезе дня. Для получения более точных прогнозов рекомендовано использовать синхронизацию с календарем событий. Примерами событий могут являться дни выпуска новой версии продукта, праздники, либо дни старта рекламной компании продукта.

1.2. Метод выявления временных закономерностей

Помимо календарных закономерностей необходимо брать в учет зависимости от времени суток. Время работы клиента с web-сервисом зависит от специфики сервиса. Таким образом, нагрузка распределяется неравномерно все 24 часов. В случае системы дистанционного обучения iSpring Learn график зависимости среднего времени выполнения скрипта от времени суток представлен на рисунке 2:

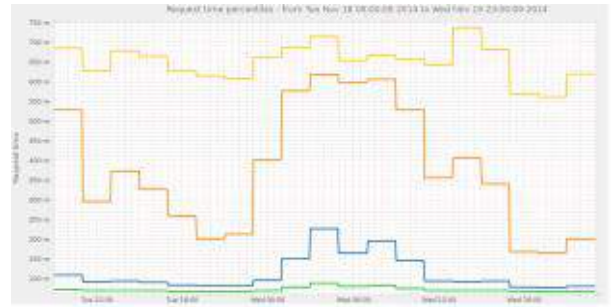


Рисунок 2 - Зависимость времени работы скрипта iSpring Learn от времени суток

На рисунке 2 отчетливо наблюдается скачок нагрузки в середине графика. Это связано с тем, что такие сервисы, как системы дистанционного обучения преимущественно используются клиентами в рабочее время. Время на сервере отличается от времени у клиента, и по графику заметно, что нагрузка нарастает с 00:00 и убывает к 12:00. Такой эффект возникает из-за разницы часовых поясов клиентов. В 3:00 проявляется ярко выраженный скачок – в этот момент наступает пересечение «рабочего времени» у клиентов с разными часовыми поясами.

Ссылаясь на описанные выше факты, можно утверждать, что нагрузка на сервере может отличаться не только в зависимости от дней, но и в зависимости от часов. По этой причине предварительный анализ трафика будет полезен и благодаря использованию системы прогнозирования появится возможность избежать случаев, когда нагрузка на сервере резко возрастает. Система выявит закономерности, проявляющиеся за день или неделю, и даст необходимые указания балансировщику нагрузки.

1.3. Метод выявления закономерностей источников трафика

При отслеживании календарных зависимостей сервиса, необходимо учитывать тот факт, что нагрузка может возрастать не только с периодичностью, связанной с событиями на анализируемом сервисе. Существуют варианты, когда периодичность загрузки одного сервиса зависит от периодичности другого. Примером такого взаимодействия может служить ссылка на анализируемый сайт, размещенная на ресурсе со значительным числом пользователей. Сервис Google Analytics [GoogleAnalytics, 2014] позволяет отслеживать адрес, с которого пришел клиент. Предоставляется несколько вариантов отчета: название поисковой системы, адрес сайта-источника, название одной из новостей и посещения тек, кто ввел адрес непосредственно в адресной строке.

Возросший трафик от поисковой системы будет говорить о том, что данный запрос сейчас является популярным. В большинстве случаев, такая популярность длится не более недели. Инструментальное средство превентивного анализа при увеличении доли трафика с поисковых систем

обеспечивает принятие решения о подготовке к высоким нагрузкам. Для анализа количества запросов компанией Google был разработан инструмент Google Trends [GoogleTrends, 2014], он позволяет оценивать количество использования ключевых слов не только в зависимости от времени, но и в зависимости от географического положения пользователей.

В случае резкого возрастания трафика с постороннего ресурса сети интернет, существует большая вероятность, что на нем была размещена ссылка на ваш ресурс. Если большого количества переходов с него до этого момента не наблюдалось, средство будет считать этот ресурс новостным. Новостные ресурсы имеют ленту новостей, которая в среднем обновляется за 1-2 дня. В таком случае средство примет решение об увеличении количества экземпляров сервиса на данный период. Если приток клиентов с этого ресурса уже наблюдался, то за период повышенной нагрузки будет принят средний период для этого ресурса.

Необходимо учитывать тот факт, что зависимость от источников трафика тесно связана с календарной зависимостью.

Таким образом, для реализации метода выявления закономерностей источника трафика, необходимо отслеживать адрес ресурса, с которого началось резкое увеличение количества пользователей и в случае, если этот источник – поисковая система – выяснить ключевое слово и определить популярность этого запроса. Основываясь на этих данных, появится возможность прогнозировать период времени высокой нагрузки сервиса.

1.4. Метод выявления геолокационных закономерностей

С целью повышения значимости данных прогноза, необходимо отслеживать географическое положение пользователя. Автором предложено определение региона с помощью IP-адреса запроса. Обладая данными географии запросов пользователей, в рекомендации, отсылаемой средством прогнозирования балансировщику нагрузки, появится возможность уточнения региона, в котором желательно подготовиться к скачку нагрузки.

Современные высоконагруженные сервисы используют сети дата-центров, расположенные по всему миру. К примеру, компания Facebook пользуется услугами дата-центров Силиконовой Долины в США и Шведскими дата-центрами в Европе [GigomResearch, 2013]. Компания iSpring использует два дата-центра. Первый находится в городе Даллас, США. Второй – в Москве. Такой подход позволяет избавить сервер, используемые российскими клиентами от нагрузок, связанными с запросами пользователей США. Благодаря такой организации работы инструментального средства существует возможность в случае чрезмерного

возрастания нагрузки в одном регионе увеличить для него производительные мощности серверов.

Таким образом, средство помимо календарной зависимости должно учитывать еще и геолокационную зависимость и при комбинации этих факторов принять решение об увеличении количества серверов в дата-центре определенного региона, к примеру США или Европы.

1.5. Прогнозирование нагрузок на основе выявленных закономерностей

На рисунке 3 показано схематическое представление web-сервиса, использующего методы прогнозирования возрастания нагрузки совместно с балансировщиком:

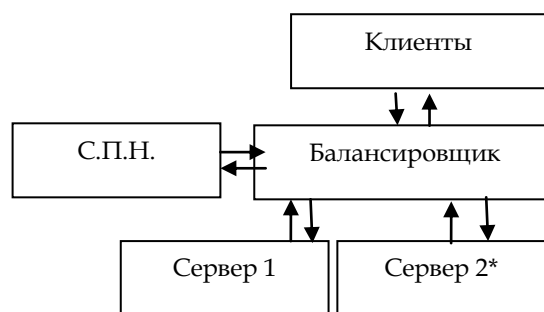


Рисунок 3 - Схематическое представление сервиса, использующего предложенную средство прогнозирования нагрузки.

Балансировщик нагрузки имеет статистику занятости серверов. Предположим, что в данный момент нагрузка на сервис не велика и с ней успешно справляется Сервер 1. Сервер 2* находится в режиме ожидания. Запросы на него не идут.

Введем обозначение С.П.Н. – средство, реализующее методы прогнозирования нагрузки. Если прогноз будет положительным и с потенциальной нагрузкой, рассчитанной анализатором сервис не справится, средство прогноза отправит запрос балансировщику на пробуждение Сервера 2*. Он выйдет из режима ожидания и начнет свою работу до того, как произойдет пик нагрузки.

При первом запуске средства прогнозирования ему задается максимальный приемлемый уровень загрузки сервера. Все запросы клиентов направляются на балансировщик. Он передает средству прогнозирования данные запросов.

Средство прогнозирования получает следующие данные: дата и время запроса, геолокация пользователя, количество и уровень загруженности серверов. Если доступны данные об источнике перехода, сохраняет и их. Сервис постоянно обучается, сохраняя новые данные. Критически важно отслеживать уровень загруженности серверов.

Для принятия решения системой используется метод экстраполяции для функции $L = f(t)$, где L –

уровень нагрузки сервера, а t – время. Если полученное в результате экстраполяции значение L больше, чем максимально заданное для сервера, будет принято решение об увеличении количества серверов.

При положительном решении для уточнения данных используются дополнительные параметры (источник перехода, географическое положение пользователя). В случае определения источника перехода будет определен его тип и в зависимости от него будет указана предположительная длительность увеличения нагрузки. Если скачок нагрузки происходит из конкретного региона, то средство прогнозирования нагрузки учтет этот факт.

Таким образом, в случае положительного прогноза, балансирующий получит данные в формате $\{T, G\}$. Где T – интервал времени, на который стоит увеличить количество серверов, G – регион, в котором необходимо это сделать.

С появлением средства, прогнозирующего появление высоких нагрузок, будет реализована возможность решения проблемы большого количества запросов заранее, до фактического столкновения с ней. Это положительно скажется на скорости и надежности работы высоконагруженного web-сервиса, вне зависимости от его предназначения.

Заключение

В настоящее время анализируется статистика нагрузки системы дистанционного обучения iSpring Learn для поиска закономерностей и выявления предпосылок высокой нагрузки.

Но уже сейчас можно говорить об острой практической необходимости в средстве прогнозирования, имеющем способность принимать решения, основываясь на данных от запросов клиентов СДО и данных о загруженности, полученных от серверов.

Библиографический список

- [iSpring Learn, 2014] Система дистанционного обучения iSpring Learn, 2014, <http://www.ispringsolutions.com/ispring-learn>
- [Блинков, 2010] Архитектура Facebook / Блинков И.; 2010 – <http://www.insight-it.ru/masshtabiruemost/arkhitektura-facebook/>
- [Комлев, 2006] Словарь иностранных слов / Комлев Н.Г.; 2006
- [GoogleAnalytics, 2014] Источники трафика/ Google.; 2014 - <https://support.google.com/analytics/answer/1033173?hl=ru>
- [GoogleTrends, 2014] Сервис анализа частоты запросов в поисковой системе Google/ Google.; 2014 - <https://www.google.ru/trends/>
- [GigamResearch, 2013] Facebook's first European data-center goes live in Sweden/David Meyer.; 2014 - <https://gigaom.com/2013/06/12/facebook-first-european-data-center-goes-live-in-sweden/>

SOLUTION TO THE PROBLEM OF PREDICTING LOAD INCREASES IN LEARNING MANAGEMENT SYSTEM "ISPRING LEARN"

Sokolnikov A.M., Sidorkina I.G.

*Volga State University of Technology, Mari El,
Russian Federation*

sokolnikov.alexey@gmail.com

igs592000@mail.ru

The article examines ways to predict high load web-services. Justify of their use in learning management system iSpring Learn. Reveals the problems that can be solved using the methods of forecasting load increases and benefits of their usage in conjunction with a load balancer.

Introduction

To solve the problems with high load web-services most developers used load balancers. The problem with this approach is that the service is ready to be a sharp increase in load. This situation will lead to delays in query execution and increase the load on a particular server. As a solution to this problem is proposed use of preventive means of analysis of high loads..

Tool for preventive load analysis

Increase in load forecasting methods analyze and process requests to the server and store their metadata. In addition, they poll current state of the server load. Based on these data, it searches for patterns of occurrence of high loads and a decision on the need to deploy additional instances of the service.

The advantage of this approach is that the use of forecasting methods of increasing the load balancer in conjunction with, the number of peaks queries on a single server is significantly reduced because of an emergency situation will be known in advance of its actual occurrence. Such methods will be called preventive load analysis.

Proactive analysis processes the data in a query and tries to identify patterns in the surge loads. There are four main types of dependencies that affect the load: calendar, time, location-based and depending on the sources of traffic.

Conclusion

With the tool, predicts the emergence of high loads, will be realized the possibility of solving the problem of a large number of requests in advance of the actual collision with it. This has a positive impact on the speed and reliability of a heavily web-service, regardless of its destination.



OSTIS-2015

(Open Semantic Technologies for Intelligent Systems)

УДК 004.8 + 004.942

ОПЫТ РАЗРАБОТКИ СИСТЕМЫ ИМИТАЦИИ, ОСНОВАННОЙ НА АГЕНТАХ

Замятина Е.Б. *, Каримов Д.Ф. **, Митраков А.А. **

* *Национальный исследовательский университет «Высшая школа экономики» (Пермский филиал)*

** *Пермский государственный национальный исследовательский университет
г. Пермь, Россия*

e_zamyatina@mail.ru,

googlicus@gmail.com,

mitrakov-artem@yandex.ru

В работе рассматривается экспериментальная система имитационного моделирования, основанная на агентах и особенности ее реализации. Обсуждаются вопросы целесообразности применения агентных систем имитации для решения ряда задач, выдвигаются требования, которые необходимо применять при их разработке, особенности организации систем имитации.

Ключевые слова: имитационное моделирование, онтология, адаптируемость, распределенное моделирование, агент, интеллектуальный агент

ВВЕДЕНИЕ

С развитием информационных технологий, ростом информационных ресурсов, совершенствованием сетевой инфраструктуры и увеличением производительности вычислительных машин появляются всё новые и новые задачи. Эти задачи являются более сложными и соответственно более затратными с точки зрения использования вычислительных ресурсов, кроме того, еще и менее формализуемыми. Новые задачи требуют поиска новых подходов и методов решения. Порою задачи бывают такими, что использование традиционных подходов и алгоритмов неприменимо в силу специфики области применения: недетерминированности, постоянной изменчивости, невозможности формализовать эти задачи и т.д.

Таким образом, возникает необходимость в более мощных и гибких интеллектуальных программных системах, способных непрерывно приобретать новые знания и изменять свою структуру и функции, развиваясь и адаптируясь к решаемым задачам и условиям внешней среды.

Одним из путей решения подобных задач можно назвать применение мультиагентных систем (МАС) – особой формы области искусственного интеллекта, которая базируется на знаниях и эвристических алгоритмах кооперативного поиска решения задачи.

Ключевым элементом этих систем становится программный агент, способный воспринимать ситуацию, принимать решения и взаимодействовать с другими агентами. В основу успешного коллективного решения задачи положено 3 фундаментальных принципа: кооперация, координация и коммуникация агентов.

В упрощённом виде можно представить агента как некоторую сущность, обладающую памятью и собственной базой знаний, умеющей находить решение некоторой узкой специфической задачи, взаимодействовать с другими агентами и менять правила поведения в динамике. Следует особо оговорить, что ни один агент в МАС не способен решить общую задачу самостоятельно, без взаимодействия с другими агентами.

Агента можно создать таким образом, что он будет иметь определенное отношение к принятию рискованных решений в условиях неопределенности. Агенты функционируют в едином виртуальном мире. В ходе переговоров агентов формируется текущее решение проблемы, которое гибко меняется в соответствии с динамичным изменением среды.

Тем не менее, успешному применению агентного моделирования на практике мешают различные проблемы, связанные в настоящее время в первую очередь с инструментарием для разработки таких моделей. Существующие системы либо достаточно

трудны для обучения с их последующим использованием. Зачастую они требуют навыков в программировании и не предназначены для использования пользователями, которые не являются специалистами в программировании. Кроме того, существуют проблемы, связанные с возможностью полного представления моделей. В этом случае требуется вести доработку системы, написания новых модулей и т.д. И необходимо отметить еще одну проблему – затраты вычислительных ресурсов для завершения имитационного эксперимента в приемлемое время. Для оптимизации имитационного эксперимента по времени необходимо распределить агенты имитационной модели (согласно определенной стратегии распределения) таким образом, чтобы они функционировали на различных компьютерах, или процессорах (или нескольких ядрах). При этом необходимо синхронизировать действия агентов (необходимо соблюдать каузальность событий).

Таким образом, при разработке системы имитации, основанной на агентах, необходимо придерживаться определенных правил и ограничений.

Дадим краткий обзор агентных систем имитации, попутно рассмотрим задачи, решение которых достигается с применением этих систем.

1. Обзор агентных систем имитации

В настоящее время существует достаточно большое количество агентных систем имитации. Одна из наиболее известных систем моделирования - *AnyLogic*[1]. Система поддерживает непрерывное моделирование, процессо-ориентированное и агентно-ориентированное. Кроме того, *Anylogic* располагает современным графическим интерфейсом и предоставляет пользователю набор стандартных библиотек, что упрощает процесс разработки модели. Имитационные модели, которые создаются средствами *Anylogic*, предназначены для решения достаточно широкого круга задач (на сайте www.xjtek.ru представлены примеры решения таких задач, как моделирование цепочек поставок, моделирование бизнес-процессов, моделирование поведения пассажиров в аэропорту, на вокзале и в торговом центре). Поведение агентов и их взаимодействие описывают с помощью графического языка. Если нет возможности описать модель средствами графического редактора, то пользователь может выполнить расширение модели с помощью языка Java. Интеграция компилятора Java в *AnyLogic* предоставляет более широкие возможности при создании моделей, а также создание Java-апплетов, которые могут быть открыты любым браузером. Среда моделирования *AnyLogic* поддерживает проектирование, разработку, документирование модели, выполнение компьютерных экспериментов с моделью, включая различные виды анализа — от анализа чувствительности до оптимизации параметров модели относительно некоторого критерия. Однако

в *AnyLogic* нет средств создания интеллектуальных агентов, кроме того, настоящая версия не поддерживает распределенное моделирование (но, судя по публикациям, работы такие велись и ведутся [Борщев А., 2002]).

Авторами из Екатеринбурга [Аксенов К.А., 2013] разработана агентно-ориентированная система моделирования *BPSim*. Система имитации *BPSim* изначально была предназначена для моделирования бизнес-процессов. В программной системе *BPSim* агенты управляют объектами процесса преобразования ресурсов. Агенты анализируют текущую ситуацию, обращаются к базе знаний, вырабатывают решение, контролируют достижение целей, обмениваются сообщениями. Программные средства *BPSIM* позволяют разработать концептуальную модель, динамическую модель, провести имитационный эксперимент и экспортировать результаты в EXCEL. В системе реализованы (а) реактивные агенты (их описывают с помощью диаграмм деятельности (конечный автомат), (б) реактивно-интеллектуальные агенты (описывают с помощью продукционной базы знаний, для описания моделей ЛПР (лицо принимающее решение), управляющих процессами), (в) интеллектуальные агенты (поведение их описывают планирующей системой, знания хранятся в фреймовой базе знаний, используют для построения сложных советующих ЭС и т.д.), (г) гибридные (построение сложных систем планирования). Однако вряд ли можно говорить о кроссплатформенности этих программных средств, поскольку в *BPSIM* генерируется код на Delphi, а также о поддержке параллельного (распределенного) моделирования.

REPAST [Repast, 2014] REcursive Porous Agent Simulation Toolkit (Repast)-это открытый и свободно распространяемый источник библиотек для крупномасштабного агентного моделирования. Repast поддерживает разработку гибких моделей из агентов и используется в моделировании социальных процессов, в маркетинге, логистике. Пользователь строит свою модель, включая в свои программы компоненты из библиотеки Repast или используя визуальный Repast для среды Scripting. Существует три версии Repast, названных Repast for Python (Repast Py) (Python), Repast for Java (Repast J) и Repast for the Microsoft.NET framework (Repast .NET)). Так Repast J включает (а) параллельный дискретный планировщик по времени (календарь событий); (б) среду для визуализации модели; (в) средства интеграции с географическими информационными системами с целью моделирования агентов на реальных картах; (г) средства описания поведения агентов (с применением компонентов из библиотек нейронных сетей, генетических алгоритмов, например). Модели Repast могут разрабатываться различными способами, включая использование ReLogo (диалект языка мультиагентного моделирования Logo), блок-схем, языка Groovy (динамического языка виртуальной машины Java) или самого языка

Java. Также все указанные способы редактирования модели можно чередовать без потери качества в ходе разработки модели. Кроме того, среда моделирования располагает средствами для обработки результатов моделирования (например, MatLab, SQL, Excel). Однако для того, чтобы создать модель, описать поведение агентов, требуется специальная подготовка, надо быть специалистом-программистом.

NetLogo [NetLogo, 2014] – среда моделирования, предназначенная для создания моделей, которые используют для описания естественных и социальных явлений. NetLogo позволяет пользователям оперировать моделями “на лету”, динамически меняя поведение системы под действием различных условий. Система достаточно проста, что позволяет пользователям без квалификации программиста открывать и запускать уже готовые модели, или строить их самим. Но также система достаточно “продвинута”, чтобы удовлетворить запросы исследователей из многих областей знаний. Среда моделирования NetLogo является кроссплатформенной и поставляется с библиотекой программных компонентов, которая представляет собой большой набор заранее написанных моделей. Эти модели могут быть использованы вновь или модифицированы.

Следующая агентно-ориентированная система моделирования MASON [MASON, 2014] представлена в виде набора библиотек на Java (кроссплатформенность). MASON включает мощные программные средства визуализации (2D,3D). В то же время, пользователь может не подключать эти средства. Следует отметить, что система MASON устойчива к взломам. Однако MASON не поддерживает распределенное моделирование (но в настоящее время ведутся работы по созданию распределенной версии) и требует серьезных знаний языка Java, т.е. не является удобным средством моделирования для малоопытного пользователя.

Еще одна кроссплатформенная агентная система моделирования *Ascape* [Ascape, 2014], написана на Java и является свободно распространяемой. Довольно удобная система моделирования для широкого круга пользователей.

Swarm [Swarm, 2014](стая, рой) был первой средой разработки АМ приложений, впервые запущен в 1994г. Разработчики Swarm стремились создать распределенную платформу для моделирования АМ. Пользователь создает модели путем включения компонентов из библиотек Swarm в свои программы.

2. Требования к построению агентно-ориентированных систем имитационного моделирования

Итак, обзор позволяет сделать вывод о том, что в основном инструментальные средства агентного

моделирования представляют собой библиотеки программных модулей, которые можно включить в программы пользователя. При разработке систем моделирования важно, чтобы модели могли разрабатывать не только пользователи с квалификацией программиста, но и обычные конечные пользователи [Власов С.А., 2013], [Замятина Е.Б., 2012]. Таким образом, агентно-ориентированные системы должны обладать визуальными программными средствами конструирования и редактирования моделей. Кроме того, для агентных систем моделирования характерны: (а) кроссплатформенность (чаще всего используют язык Java или C); (б) модульность, использование объектно-ориентированного подхода (агентов чаще всего представляют в виде объектов, которые обмениваются информацией друг с другом); (в) возможность динамического изменения моделей (изменение настроек, например) во время выполнения имитационного эксперимента. При разработке агентно-ориентированной системы моделирования авторы пытались учесть опыт перечисленных выше разработок и добиться выполнения требований, изложенных ниже:

(а) *операции над моделью* - при разработке агентной модели целесообразно предусмотреть возможность изменения набора агентов, совместное функционирование которых отображает протекающие в реальном мире процессы, возможность изменения связей между агентами [Замятина Е.Б., 2011];

(б) *иерархическое представление модели, возможность ее детализации* - имитационная модель должна быть иерархической и предоставлять возможность детализировать модель, заменяя конкретный агент (и его поведение) группой агентов или, наоборот, заменять группу агентов одним агентом, при этом новый агент должен реализовывать групповое поведение объединенных в группу агентов;

(с) *выделение структуры взаимодействия агентов* - имитационная модель отображает взаимосвязи агентов (структура моделируемой системы), которые обмениваются сообщениями. Целесообразно разработать программное обеспечение, которое в готовой имитационной модели выделяет ее структуру. Выделив структуру агентов, их взаимосвязи, можно провести дополнительный анализ имитационной модели, исследуя ее структурные характеристики (возможно, методами теории графов);

(д) *настройка на конкретную предметную область* - системе имитации приходится решать задачи, связанные с различными предметными областями, кроме того, с имитационной моделью можно работать в разных «терминах», так например, при моделировании компьютерной сети можно работать в «терминах» систем массового обслуживания, «в терминах» сетей Петри или теории графов. В этом случае целесообразно использовать онтологический подход, опыт применения онтологий при работе с

имитационными моделями приводится в [Mikov A., 2009], [Сухов А.О., 2013], [Замятина Е.Б., 2013-а].

(е) *оптимизация имитационного эксперимента по времени* – необходимость в этом возникает вследствие сложности задач, которые решаются методами имитационного моделирования. Имитационный эксперимент должен завершаться за приемлемое время. Решением в этом случае является использование нескольких вычислительных узлов (сети, кластера и т.д.), распределение агентов по вычислительным узлам и применение специализированных алгоритмов (оптимистического или консервативного, или их модификации) для сохранения каузальности событий и алгоритмов для сохранения равномерной загрузки вычислительных узлов [Zheng G., 2005], [Миков А.И., 2010]

(f) *оптимизация имитационного эксперимента по надежности* – целесообразно разработать специальное программное обеспечение, применяемое в условиях распределенного имитационного эксперимента для перемещения агентов на «живые» узлы в случае выхода из строя каких-либо вычислительных узлов;

(g) *удаленный доступ* – предполагает возможность использования системы имитации удаленно и реализацию совместной работы пользователей, которые географически находятся на расстоянии друг от друга [Замятина Е.Б., 2012-а].

(i) *интеллектуальная обработка результатов моделирования* - целесообразно использовать для получения структурированной информации с применением методов Data Mining [Kolevator G.A., 2012] .

(j) *интеллектуальные агенты* – необходимы для того, чтобы наиболее адекватно отображать моделируемые процессы в экономике, маркетинге, логистике очень важно реализовать интеллектуальных агентов. Большинство из рассмотренных выше систем предоставляет пользователю инструментальные средства для описания реактивного поведения, позволяют имитировать достаточно простое поведение. А необходимо сделать так, чтобы интеллектуальные агенты могли изменять свое поведение в зависимости от изменения обстановки (обучаться), преследовать цели, выбирать ту или иную стратегию в зависимости от роли, которую они выполняют.

3. Первый опыт создания агентной системы моделирования

Первая агентная система моделирования была разработана в 2007 году в виде библиотеки с набором основных классов (ядро), необходимых для реализации агентного моделирования [Замятина Е.Б., 2010].

Структура системы отображено на рис. 1 (диаграмме классов UML). Основой всей модели пользователя является класс модели (Model). Именно в нем описана основная семантика

поведения всей моделируемой системы в целом. Также методы данного класса (их переопределение) позволяют обеспечить вывод текущих результатов моделирования, а также запускать, останавливать, изменять основные параметры модели. Остановимся кратко на некоторых важных методах.

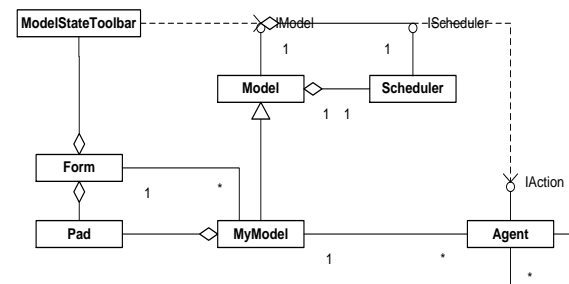


Рисунок 1. - Компоненты системы для построения агентных моделей

Метод инициализации отвечает за инициализацию всех основных внутренних рабочих параметров модели, его вызывают при первом запуске модели, а также после остановки модели. Модель выполняется по шагам (каждый шаг соответствует одной единице времени, то есть за один шаг в модели может произойти несколько событий, если они все были запланированы на одно и то же время). Планировщик (Scheduler) отвечает за продвижение времени в системе (реализуя соответствующий интерфейс). Его главной задачей является определение тех агентов, которые получают право хода, т.е. фактически возможность выполнять активные действия в модели, что собственно составляет суть всего процесса моделирования. Из очевидных преимуществ подхода с центральным планировщиком следует отметить его относительную простоту и эффективность реализации.

Графический интерфейс представлен несколькими реализованными компонентами: это планшет (Pad) для отображения различных графических объектов и привязки к ним агентов и собственно сами графические объекты, инспектор свойств для графических объектов и самой модели и панель инструментов для управления состоянием выполнения модели.

Планшет используется для отображения графических объектов с учетом их порядка по оси OZ. Поддерживаются следующие возможности: масштабирование, перемещение видимой области, перемещение графических объектов, их множественное и одиночное выделение, сжатие рабочей области планшета до реально занимаемой графическими объектами. В случае выхода графических объектов за границы рабочей области планшет автоматически увеличивается в размерах. Работает в многопоточном режиме.

Графические объекты (производные от Graphic) используются для графического представления агентов из модели, а также отображения различного рода связей между ними и обмена

информацией/ресурсами тоже между ними. Инспектор свойств используется для отображения свойств/параметров графических объектов (представляемых ими агентов) или модели. Поддерживается возможность обновления информации о выделенных объектах в динамике с продвижением имитационного времени модели и изменения состояния наблюдаемых агентов (выделенных графических объектах). Извлечение информации из модели для долгосрочного хранения и анализа результатов осуществлены с использованием инфраструктуры .NET.

Итак, первая агентная система моделирования, разработанная группой авторов, была *последовательной* и была основана на совместном функционировании *реактивных* агентов. Следует заметить, что опыт ее разработки был использован при создании системы моделирования «Рудопоток» [Чудинов Г.В., 2013].

Следующим шагом была разработка *распределенной* системы моделирования, основанной на совместном функционировании интеллектуальных агентов, основанных на *производственных правилах*.

4. Реализация распределенной агентной системы имитации

Для реализации агентной платформы распределенного моделирования был выбран язык программирования Scala.

Язык Scala представляет собой кросс-парадигменный объектно-функциональный язык программирования, совмещающий в себе ООП и функциональное программирование и специализирующийся на создании легко масштабируемого компонентного программного обеспечения (Scala была создана в 2004г. под руководством Мартина Одерски в Университете EPFL (Lausanne, Switzerland)) [Odersky M., 2013].

В настоящее время доступна для платформ Java и .NET Framework. Среди ключевых особенностей языка можно отметить: единую объектную модель, наличие примесей (traits), технику сопоставления с образцом, лямбда-исчисление, виды (type views bounds), линеаризацию типов (type linearization), параметрический и функциональный полиморфизм, вариантность типов, кейс-классы (case classes), вывод типов, поддержка хвостовой рекурсии, наличие многочисленных инструментов по созданию новых языковых конструкций и обработку списков. Обобщенная структурная схема симулятора изображена на рисунок. 2.

Таким образом, архитектура симулятора является не многокомпонентной, а многослойной. Достигается это за счёт того, что к модулю, реализующему некоторую базовую функциональность, «примешиваются» другие компоненты, расширяющие структуру, поведение и семантику.

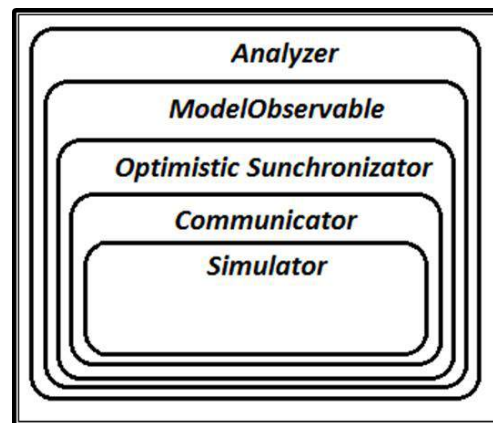


Рисунок. 2. - Многослойная архитектура симулятора

Трейт Simulator является базовым модулем, к которому подмешиваются другие элементы, уточняя его поведение. Фактически, Simulator, – это каркас для логического процесса. Трейт Communicator представляет собой модуль для работы с акторской системой. Он содержит в себе экземпляр актора, который занимается посылкой/приёмом сообщений. Трейт OptimisticSynchronizer является ключевым звеном для организации PDES – он реализует оптимистический алгоритм синхронизации Time Warp. Трейт ModelObservable является реализацией концепции информационных процедур, используемых для организации сбора статистики (информационные процедуры – программные компоненты, накладываемые на модель с целью сбора статистики в ходе имитационного прогона). Трейт Analyzer добавляет «интеллектуальность» алгоритмам синхронизации, используемых для реализации распределенного моделирования.

Особое внимание в разработанной системе имитации уделяется алгоритму синхронизации распределенной модели [Fujimoto R.M., 2003].

В системе реализован оптимистический алгоритм синхронизации, основанный на знаниях об имитационной модели. Существует ряд работ, в которых используются знания об имитационной модели для сокращения времени выполнения распределенного имитационного эксперимента. Сокращение времени выполнения распределенного алгоритма авторы достигают за счет реализации эффективного алгоритма синхронизации логических процессов, управляемого производственной экспертной системой, и за счет балансировки нагрузки на вычислительных узлах. Использование знаний об агентной имитационной модели, извлекаемые из онтологий, дали хорошие результаты при реализации алгоритмов синхронизации в агентной системе имитации. Эксперименты показали, что метрики, характеризующие скорость проведения распределенного эксперимента, управляемого модифицированным алгоритмом синхронизации (алгоритм KBASA, основанный на знаниях о модели) существенно сократились по сравнению с метриками классического оптимистического алгоритма Time Warp: количество откатов

сократилось с 71.2 до 3.8, количество отправленных антисообщений сократилось с 108.4 до 12.8.

5. Реализация интеллектуальных агентов

Известно, что создание интеллектуальных агентов является весьма сложной задачей, требующей теоретического фундамента для концептуального представления агентов. Таким фундаментом служат модели интеллектуальных агентов, по-разному описывающие знания, способы рассуждений, планирование поведения и непосредственные действия агентов.

Модели принято рассматривать в двух точках зрения: с точки зрения анализа свойств и поведения агентов в процессе функционирования системы в целом; с точки зрения изучения и конструирования свойств агента, определяющих его внутренние процессы (получение знаний, выработка целей, принятие решений и т.д.).

Выделено три вила архитектур: (а) делиберативные архитектуры и модели; (б) реактивные архитектуры и модели; (в) гибридные архитектуры и модели.

Реактивный подход позволяет использовать множество достаточно простых сценариев поведения агентов. Эти сценарии являются реакцией на появление того или иного события внешней среды. Недостатком является то, что достаточной полный ситуативный анализ всех возможных активностей агентов.

Делиберативные модели и архитектуры позволяют применять строгие формальные методы и хорошо отработанные технологии традиционного искусственного интеллекта, позволяющие относительно легко представлять знания в символической форме и переносить их в агентно-ориентированные системы.

Гибридный тип архитектуры совмещает преимущества упомянутых выше архитектур. Таким образом, интеллектуальный агент обладает высокоуровневым выводом и низкоуровневыми реактивными способностями.

Итак, разработка агентной системы моделирования предполагала разработку инструментальных средств набора базовых классов для представления интеллектуальных агентов. Было принято решение представить архитектуру агентов в виде гибридной и делиберативной схем. В качестве средств логического вывода авторы используют продукционные системы и нейронные сети. Известно, что нейронные сети обладают свойством самообучаться, что актуально для реализации интеллектуальных агентов, которые должны приспосабливаться к изменению внешней среды и менять свое поведение, принимая то или иное решение. На первом этапе исследование было использовать три типа нейронных сетей:

многослойный персептрон, сеть Хопфилда и сеть Хемминга. Для обучения нейронных сетей предложен генетический алгоритм.

Разрабатываемые инструментальные средства были протестированы. В качестве тестовых задач была выбрана задача поиска достопримечательностей в парке и «Искусственная жизнь».

Задача о поиске достопримечательностей имеет следующую формулировку: некая персона ищет достопримечательность в парке и пытается добыть информацию об ее месторасположении с помощью карты местности и информаторов. Получив информацию от информатора-человека, персона направится к остановке (трамвая), вместо того, чтобы исключительно пешком добираться до интересующей достопримечательности. Если в качестве источника информации используется карта, то персона добирается до цели пешком и т.д. Задача «Искусственная жизнь» достаточно хорошо известна.

Результаты тестовой задачи «Поиск достопримечательностей» приведены ниже в таблице.

Было проведено моделирование в среде размерами 400 на 600 точек, при условии наличия одной цели-достопримечательности, и всего было совершено 100 тестовых прогонов модели. В результате моделирования были получены результаты, показывающие возможность использования такого типа агентов для решения данной задачи (таблица 1).

Таблица 1. Результаты моделирования «поиск в парке» продукционными агентами

Кол-во агентов-информаторов	Среднее время поиска цели (в секундах)
1	29.6
3	27.9
5	19.5

Модель агента, основанная на нейронных сетях, была реализована в виде 2-слоеного персептрона, имеющего 6 нейронов входного уровня и 2 нейрона выходного слоя.

В качестве входных данных на входной слой сети от рецепторов передавалась следующая информация: (а) факт наличия информатора в поле видимости – info; (б) расстояние до информатора (если в поле видимости их несколько, то учитывается расстояние до ближайшего) - dinf.;(в) расстояние до цели - df.; и т.д.

Был проведен имитационный эксперимент в среде размерами 400 на 600 точек, при условии наличия одной цели-достопримечательности, и всего было совершено 100 тестовых прогонов модели. В результате моделирования были получены результаты, показывающие возможность использования такого типа агентов для решения данной задачи (таблица 2).

Таблица 2. Результаты моделирования “поиск в парке” нейронными агентами

Количество агентов-информаторов	Среднее время поиска цели (в секундах)
1	37.6
3	33.5
5	25.1

Для настройки системы имитации можно воспользоваться либо программными средствами, которые предоставляют языковые инструментарии (или DSM-платформы), предназначенные для создания предметно-ориентированных языков (DSL, Domain Specific Language), либо программными средствами на основе онтологий. Рассмотрим первый способ.

Заключение

Авторы разработали прототип агентной системы имитации. Агентная система соответствует требованиям, изложенным в статье (кроссплатформенность, возможность использования нескольких вычислительных узлов для оптимизации имитационного эксперимента по времени, использование онтологий и т.д., участие в имитационном эксперименте интеллектуальных агентов, в том числе, и основанных на нейронных сетях).

Разработанный алгоритм синхронизации KBASA, основанный на знаниях о модели, позволяет проводить эффективные имитационные эксперименты (метрики, указывающие на скорость проведения имитационного эксперимента, значительно улучшились). Кроме того, получены хорошие результаты при тестировании интеллектуальных агентов, основанных на нейронных сетях.

Благодарности

Работа выполнена при поддержке РФФИ и № 13-07-96506 р-юг-а «Разработка деонтических основ мультиагентных технологий формирования баз знаний для виртуальных лабораторий

Библиографический список

[Боршев А., 2013] Боршев А.В. Как строить простые, красивые и полезные модели сложных систем. Сборник докладов шестой всероссийской научно-практической конференции «Имитационное моделирование. Теория и практика» (ИММОД-2013). Том 1. // ISBN 978-5-9690-0221-0 // Издательство «ФЭН» Академии наук РТ, Казань, 2013, с. 21-34.

[Borshchev A., 2002] Borshchev A., Karpov Y., Kharitonov V. Distributed Simulation of Hybrid Systems with AnyLogic and HLA. // Parallel computing technologies. – 2002. № 18(6). – P.829-839.

[Аксенов К.А., 2013] Аксенов К.А., Ван Кай, Аксенова О.П. Разработка и применение метода анализа узких мест на основе мультиагентного имитационного моделирования // Сборник докладов шестой всероссийской научно-практической конференции «Имитационное моделирование. Теория и практика» (ИММОД-2013). Том 2. // ISBN 978-5-

9690-0221-0 // Издательство «ФЭН» Академии наук РТ, Казань, 2013, с. 19-23.

[Repast, 2014] Система моделирования “Repast” [Электронный ресурс] [Режим доступа: <http://repast.sourceforge.net/>] [Проверено:25.12.2014]

[NetLogo, 2014] Система моделирования “NetLogo” [Электронный ресурс] [Проверено:25.12.2014]

[MASON, 2014] Система моделирования “MASON” [Электронный ресурс] [Режим доступа: <http://cs.gmu.edu/~eclab/projects/mason/>] [Проверено:25.12.2014]

[Ascape, 2014] Система моделирования “Ascape” [Электронный ресурс] [Режим доступа: <http://ascap.sourceforge.net/index.html#Contact>] [Проверено:25.04.2014]

[SWARM, 2014] Система моделирования SWARM. [Электронный ресурс][режим доступа: www.swarm.org] [Проверено:25.04.2014]

[Власов С.А., 2013] Власов С.А., Девятков В.В., Назмеев М.М. Имитационная экспертиза: опыт применения и перспективы // Сборник докладов шестой всероссийской научно-практической конференции «Имитационное моделирование. Теория и практика» (ИММОД-2013). Том 1. // ISBN 978-5-9690-0221-0 // Издательство «ФЭН» Академии наук РТ, Казань, 2013, с. 54-63.

[Замятина Е.Б., 2012] Замятина Е.Б., Миков А.И. Программные средства системы имитации Triad.Net для обеспечения ее адаптируемости и открытости. Информатизация и связь. №5, 2012, АНО «Редакция журнала «Информатизация и связь», ISSN 2078-8320, С.130-133.

[Замятина Е.Б., 2011] Замятина Е.Б., Миков А.И., Михеев Р.А. Лингвистические и интеллектуальные инструментальные средства симулятора компьютерных сетей TRIADNS. International Journal “Information theories & Applications (IJ ITA). Vol 19, Number 4, 2012, pp.355-368. ITHEA, Sofia, 1000, P.O.B. 775, Bulgaria. ISSN 1310-0513 (printed)

[Mikov A., 2009] Mikov A., Zamyatina E., Kubrak E. An Ontology-based Approach to the Incomplete Simulation Model Analysis and its Automatic Completion. International Journal “Information Technologies & Knowledge”, 2009, Volume 3, Number 2, pp. 169-186.

[Сухов А.О., 2013] Сухов А.О. Трансформация визуальных моделей в системе MetaLanguage / Современные проблемы математики и ее прикладные аспекты – 2013. Сборник тезисов конференции. – Пермь: Пермский государственный национальный исследовательский университет, 2013. – С. 44.

[Замятина Е.Б., 2013-а] Замятина Е.Б., Лядова Л.Н., Сухов А.О. Мультиязыковое моделирование с использованием DSM платформы MetaLanguage. Информатизация и связь. №5, 2013, АНО «Редакция журнала «Информатизация и связь», ISSN 2078-8320, С.11-15.

[Чудинов Г.В., 2013] Чудинов Г.В. Архитектура и разработка инструментального средства с ориентацией на предметную область транспортировки руды в шахтах – ПБК «Рудопоток». Информатизация и связь. №5, 2013

[Fujimoto R.M., 2003] Fujimoto R.M. Distributed Simulation Systems. In Proceedings of the 2003 Winter Simulation Conference S. Chick, P. J. Sánchez, D. Ferrin, and D. J. Morrice, eds. The 2003 Winter Simulation Conference 7-10 December 2003. The Fairmont New Orleans, New Orleans, LA, pp. 124-134

[Zheng G., 2005] Zheng G. Achieving high performance on extremely large parallel machines: Performance prediction and load balancing: Ph.D. Thesis. Department Comput. Sci., Univ. of Illinois at Urbana-Champaign, 2005. 165 p. [Electron. resource]. <http://charm.cs.uiuc.edu/>.

[Миков А.И., 2010] Миков А.И., Замятина Е.Б., Козлов А.А. Мультиагентный подход к решению проблемы равномерного распределения вычислительной нагрузки. Natural and Artificial Intelligence, ITHEA, Sofia, Bulgaria, 2010, pp.173-180.

[Замятина Е.Б.,2012-а] Замятина Е.Б., Миков А.И. Применение онтологий и принципов организации сервис-ориентированно архитектуры при проектировании и реализации системы имитационного моделирования. Материалы 3-ей Международной научно-технической конференции «Технологии разработки информационных систем ТРИС-2012», Т.1., Таганрог, издательство Технологического института ЮФУ, Ростов –на-Дону, 2012, 9 сентября, стр. 61-65.

[Kolevator G.A.,2012] Kolevator G.A., Zamyatina E.B. Simulation Analysis Framework Based on Triad.Net. Proceedings of the 6-th Spring/Summer Young Reseachers' Colloquium on Software Engineering. SYRCoSE 2012, Perm, May 30-31, 2012-Perm, Russia,pp.160-163.

[Odersky M., 2013] Odersky M. The Scala Programming Language [Электронный ресурс]. URL: <http://www.scala-lang.org/node/25> (дата обращения: 06.06.2013)

[Замятина Е.Б., 2010] Замятина Е.Б., Чудинов Г.В. Разработка и использование программных средств для построения и исследования агентных имитационных моделей. / Е.Б. Замятина, Г.В. Чудинов // Вестник пермского университета. Математика, механика, информатика, 2010, №2(2), С. 80 – 84.

THE EXPIERANCE OF AGENT-BASED SIMULATION SYSTEM IMPLEMENTATION

Zamyatina E.B. *, Karimov D.F. *, Mitrakov A.A. *

**Perm State National Researching University,
Perm, Russian Federation*

e_zamyatina@mail.ru,

googlicus@gmail.com,

mitrakov-artem@yandex.ru

This paper discusses the problem of agent-based simulation system implementation. Authors consider more precisely the design and impementation of distributed simulation system (the design of special knowledge-based synchronization algorithm) and the design of agents based in neural networks

INTRODUCTION

Introduction considers the actuality of the problem. Indeed the agent-based simulation system are wide spread now, but some features and characteristics of these simulation system do not permit to use them for the solving of some problems. One of the problems is an optimization of simulation experiment in the respect to time. Next problem – the design and implementation of the intellectual agents.

MAIN PART

Main part includes the overview of agent-based simulation systems. The overview shows that it is necessary to make effective and flexible simulation system in order to solve complicate problems. The effectiveness of simulation system may be achieved by the using of special algorithm based on knowledge about simulation model. The flexibility of simulation system may be achieved by using of intellectual agents. Intellectual agents may change their behavior which depends from the characteristics of an environment (it is well known that agent is a program entity which communicate with the environment and other agents). Authors suggest to use agents based on neural network

CONCLUSION

The results of investigations and new directions in investigations are presented. The investigations shows that solutions of authors allow to design more flexible and efficient simulation systems.



OSTIS-2015

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822:514

РАЗРАБОТКА УЧЕБНОЙ ПРОГРАММНОЙ ПЛАТФОРМЫ ДЛЯ РОБОТА iROBOT CREATE

Бармина Е.И.* , Ланин В.В.* , Плетнёв А.О.**

* *Пермский филиал национального исследовательского университета
«Высшая школа экономики», г. Пермь, Россия*

enokokok@ya.ru

vlanin@live.com

** *Пермский государственный национальный исследовательский университет, г. Пермь, Россия*

artem.o.pletnev@gmail.com

Статья посвящена разработке учебного программного комплекса для исследования алгоритмов движения мобильного робота. В соответствии с предлагаемым подходом, алгоритмы основываются на концептуальном описании пространства, в котором передвигается робот. Прототип программный комплекс разрабатывается для мобильного робота iRobot Create на платформе Java.

Ключевые слова: мобильный робот; алгоритм движения; онтология пространства.

Введение

В рамках проекта планируется разработка программного комплекса для разработки алгоритмов движения мобильного робота на основе формальной концептуальной модели окружающего мира. Используемые на данный момент подходы к построению алгоритмов в большинстве случаев не допускают расширения и динамической настройки. В предлагаемом подходе алгоритм передвижения не закодирован жестко, он строится на базе вывода по базе знаний. Наличие такой интеллектуальной системы ориентации, позволит роботу планировать свои действия по перемещению в пространстве, просчитывать оптимальный маршрут движения, собирать, обобщать и использовать информацию об окружающем его пространстве и встречающихся препятствиях.

На данный момент существует несколько подходов к представлению окружающей местности на карте, используемой при движении робота: растровая карта, таблица связей, объектная карта и др. Растровая карта хранится как двумерный массив, показывающий доступные и закрытые для прохождения роботом зоны. При использовании таблицы связей предполагается, что в основу заложена карта связей, хранящая некие узлы на карте и связи между ними. Объектная карта – это попытка эмуляции способа мышления человека и

его подход к ориентации в пространстве [Bouten, 2012]. Данный подход предполагает, что робот, определив несколько ключевых точек карты, получает возможность самостоятельно прокладывать путь к нужной точке. Также необходимо уточнить, что алгоритмы зависят от аппаратных средств робота.

1. Платформа iRobot Create и поддерживающие ее инструментальные средства

1.1. Мобильный робот iRobot Create

Для реализации алгоритма отлично подойдет робот, выпущенный CES iRobot в 2007 году - iRobot Create, основанный на платформе Roomba. Он создан специально для разработчиков роботов и дает возможность собирать собственных роботов, писать для них программы, на существующей платформе разрабатывать свои собственные модели роботов [Isaacs, 2011]. Данный робот включает 4 контактных сенсора стены и всенаправленный инфракрасный приемник, расположенный в передней част робота.

Открытый интерфейс робота состоит из электронного и программного интерфейсов [iRobot, 2006]. Данные интерфейсы контролируют поведение и считывают информацию с сенсоров. Программный интерфейс позволяет манипулировать

поведением конструктора. Он считывает информацию с сенсоров при помощи серии команд, включает команды режима, команды привода, звуковые команды, демонстрационные команды и команды опроса сенсоров. Эти команды посылаются на последовательный порт конструктора с компьютера или микроконтроллера посредством Mini-DIN разъема или DB-25 разъема грузового отсека».



Рисунок 1 – Устройство робота

Следует заметить, что iRobot Create оснащен минимальным набором аппаратных средств, и чего следует, что алгоритмы разработанные для данной модели могут быть перенесены и на более совершенные устройства.

1.2. Инструментальные средства разработки для роботов

На данный момент наиболее популярными программными решениями для разработки, управления и программирования роботов являются ROS, MRDS, Player.

1.2.1. Система ROS

ROS (Robot Operating System) – операционная система с открытым исходным кодом для роботов (лицензия BSD). Проект разрабатывается исследовательской лабораторией Willow Garage в сотрудничестве с университетом Стэнфорда. ROS реализует системный уровень управления роботом, а на его основе развиваются прикладные пакеты: библиотека машинного зрения OpenCV, система планирования действий, сервер управления Player и другие технологии, используемые в десятках научных и прикладных проектов. С точки зрения архитектуры ROS – это распределённая система процессов (узлов), которые могут быть сгруппированы в так называемые пакеты и стеки, пригодные для распространения. ROS легко интегрируется с другими программными компонентами (на данный момент с OpenRAVE, OROCOS и Player). ROS не зависит от языка программирования, уже реализованы версии на C++, Python, LISP, Octave Java, Lua. ROS имеет встроенный пакет для тестирования `rotest`, облегчающий тестирование приложений. При создании программ на ROS строится «граф» – сеть

точка-точка (peer-to-peer network) из процессов, которые связываются друг с другом через инфраструктуру ROS.

1.2.2. Microsoft Robotics Developer Studio

Microsoft Robotics Developer Studio (Microsoft RDS, MRDS) – Windows-ориентированная среда разработки приложений для роботизированных платформ [Гай, 2012]. В состав Robotics Studio входят следующие компоненты: runtime environment, VPL, simulation environment. *Runtime environment* – окружение, в котором выполняется приложение для роботов, происходит отслеживание и взаимодействие с другими приложениями для роботов. В основе Runtime environment лежит CLR 2.0, что дает возможность разрабатывать приложения на языках программирования платформы Microsoft .NET. Runtime environment состоит из двух элементов: CCR (concurrency and coordination runtime) – библиотека для работы с параллельными и асинхронными потоками данных и DSS (Decentralized Software Services) – средство создания распределенных приложений на основе сервисов, базирующиеся на протоколе Decentralized System Services Protocol (DSSP). *VPL* (visual programming language) – визуальный предметно-ориентированный язык программирования для разработки приложений для роботов. *Simulation environment* – окружение для выполнения приложения для роботов в симулируемых условиях. Приложение в Robotics Studio – это композиция слабосвязанных параллельно выполняющихся компонентов. При этом все компоненты в Robotics Studio – это независимо исполняемые сервисы, т.е., например, для разработчика программы не существует физического мотора, а есть сервис с интерфейсом, к которому нужно обратиться, чтобы работать с мотором из написанной программы.

К существенным недостаткам RDS можно отнести зависимость от Windows и закрытый исходный код. Кроме того, Robotics Developer Studio не имеет встроенных систем компьютерного зрения, навигации и машинного обучения. И протокол SOAP, используемый для взаимодействия распределенных сервисов, плохо подходит для приложений, работающих в режиме реального времени.

1.2.3. Проект Player

Проект Player (ранее проект Player/Stage или проект Player/Stage/Gazebo) представляет собой инструментарий по созданию свободного программного обеспечения для исследования робототехнических и сенсорных систем. Проект состоит из 3 основных компонентов: сервера Player и платформ для симуляции роботов, двумерного симулятора Stage и трёхмерного симулятора Gazebo. Можно сказать, что Player наиболее часто используемый интерфейс в робототехнических исследованиях. Большинство ведущих журналов о робототехнике, регулярно публикуют статьи о применении Player (Stage и Gazebo) для управления

и моделирования робототехнических экспериментов (в академических, правительственных и промышленных лабораториях). Player обеспечивает сетевой интерфейс для различных роботов и сенсорного оборудования. Клиент-серверная модель, реализованная в Player, позволяет программам управления робота быть написанными на любом языке программирования и работать на любом компьютере, подключённого к сети вместе с роботом.

Компоненты проекта работают на POSIX-совместимых операционных системах, включая Linux, Mac OS X, Solaris и BSD; планируется портирование на Microsoft Windows. Проект был основан в 2000 году в Университете Южной Калифорнии в Лос-Анджелесе и широко используется в робототехнических исследованиях и обучении. Программное обеспечение распространяется под лицензией GNU General Public License с документацией под GNU Free Documentation License.

2. Алгоритм движения робота на основе онтологического подхода

В предлагаемом подходе карта для движения робота индексируется понятиями онтологии, в которой описаны возможные препятствия и способы их преодоления. Кроме возможных препятствий, в онтологии описываются такие объекты как само помещение, дверной проем, ступени, пороги, тип напольного покрытия и пр. Таким образом, для решения конкретной задачи перемещения, робот сначала строит или загружает построенную ранее карту, а затем прокладывает на ней маршрут, интерпретируя понятия представленных на карте объектов.

2.1. Онтология

Для того, чтобы робот мог ориентироваться в пространстве, ему нужно ввести некоторые понятия, такие как - что такое комната, какие объекты она может содержать, какие у неё есть свойства и т.д. Для описания всех этих параметров необходима онтология. Онтология – формальное явное описание понятий в рассматриваемой предметной области, свойств каждого понятия, описывающих различные свойства и атрибуты понятия, и ограничений, наложенных на свойства. Онтология вместе с набором индивидуальных экземпляров классов образует базу знаний.

Начнем построение онтологии с описания классов. Для робота, помещение рассматривается как некоторая среда, наполненная объектами, которые нужно распознать. Следовательно, онтология должна описывать отличительные черты этих объектов с точки зрения робота (форма объектов, их размер, материал и пр.). Начнем с того, что помещение – это не только расположенные в нем объекты, но и его геометрическая форма, так же оно имеет выходы. Объекты, расположенные в

помещении, прежде всего отличаются по геометрической форме и по размерам. Исходя из этих параметров можно сделать предположение об объекте. Так же необходимо описать отношения между классами (например, дизъюнктивные классы и т.д.). Далее осталось описать свойства объектов. Объекты в помещении (мебель и различные препятствия) тоже имеют свои параметры, такие как: периметр, координаты внутри помещения, диаметр, и др. Заключительный шаг в создании онтологии - создание конкретных экземпляров классов. Например: если объект имеет четыре опоры, а его периметр около 200 см, то, скорее всего, это стул. На данный момент прототип онтологии описан на языке OWL 2.0 с использованием системы Protégé (рис. 2).

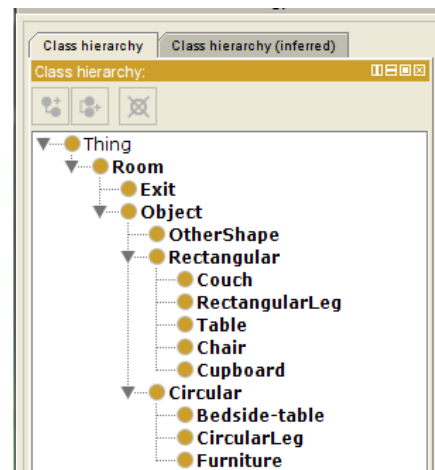


Рисунок 2 – Фрагмент иерархии классов онтологии

2.2. Алгоритм движения робота

Основные компоненты программно-технической системы представлены на рис. 3. Ключевым элементом системы является управляющий модуль, он обрабатывает информацию, полученную от сенсоров робота, и передает управляющие сигналы. Обрабатывая информацию, управляющий модуль делает запросы к онтологии для идентификации объектов, если же объект идентифицирован ранее, то информация извлекается из карты.

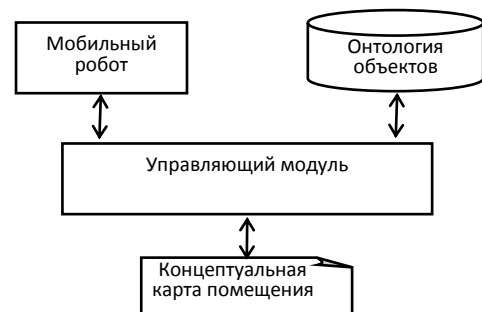


Рисунок 3 – Фрагмент иерархии классов онтологии

Перемещение робота может происходить в нескольких режимах: построение концептуальной карты помещения, поиск роботом целевого объекта и комбинированный режим (поиск с построением карты, см рис. 4).

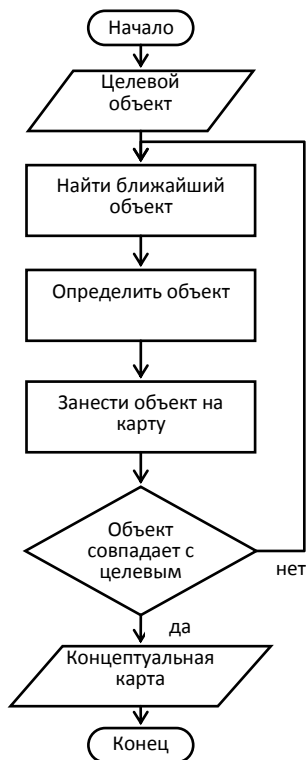


Рисунок 4 – Фрагмент иерархии классов онтологии

Заключение

Особенностью предлагаемого подхода является возможность расширения и динамической настройки алгоритма, за счет расширения онтологической базы знаний объектов пространства. На данный момент разработан прототип онтологии объектов и библиотека коммуникации с iRobot Create по протоколу Bluetooth. Платформой реализации является Java, что позволяет получить кроссплатформенное решение. В дальнейшем при развитии проекта планируется использовать систему ROS. Разработанный программный комплекс планируется использовать в учебном процессе в курсах, связанных с изучением интеллектуальных систем и онтологического инжиниринга.

Библиографический список

- [iRobot, 2006] iRobot Create Open Interface (OI) Specification, http://www.robotmart.ru/product_images/creator_Series_Manual.pdf
- [Гай, 2012] Гай В.Е. Microsoft Robotics Developer Studio. Программирование алгоритмов управления роботами.– М.: ЭКОМ Паблишерз, 2012. – 184 с.
- [Bouten, 2012] Ontology-Driven dynamic discovery and distributed coordination of a robot swarm.– Bouten N., Hristoskova A., Ongenaе F., Nelis J., Turck F.//Dependable Networks and Services Lecture Notes in Computer Science Volume 7279, 2012, pp 2-13.
- [Isaacs, 2011] A Guided Internship For High School Students Using iRobot Create Isaacs J., Klein D., Hespanha J.– 18th IFAC World Congress.– 2011.– V.16, pp. 12820- 12825.

EDUCATIONAL SOFTWARE PLATFORM DEVELOPMENT FOR IROBOT CREATE

Barmina E. *, Lanin V. *, Pletnev A. **

* Perm Branch of the National Research University Higher School of Economics, Perm, Russia

enokokok@ya.ru, vlanin@live.com

** Perm State National Research University, Perm, Russia

artem.o.pletnev@gmail.com

Article is about the development of an educational software for a mobile robot motion algorithms research. According to the proposed approach, the algorithms are based on the specially developed ontological resources that describe the robotic movement space. Software package is developed for the mobile robot iRobot Create.

Within the project the development of a software package for the development of a mobile robot motion algorithms, which are based on a formal conceptual model of the world planned. An already developed algorithms construction approaches can not be extended and configured dynamically. The proposed approach assumes that the movement algorithm is not strictly coded and is built on the basis of the ontological knowledge base output. Such intellectual orientation system will allow the robot to plan the steps to move in the space, to calculate the optimal route and to collect and use information about the surrounding obstacles.

At the moment, there are many different approaches to display the surrounding area on the map: raster map, table relationships, the object map and others. Raster map is stored as a two-dimensional array, which showing the areas, which are closed or available for the robot passage. If the linked table is used, it is assumed that the map of connections, which stores some nodes on the map and the links between them, is underlie. The object map is an attempt to emulate the human way of thinking and orientation in space, which assumes that the robot, having learned a few key points on the map, gets an opportunity to pave the way to the desired point. It is assumed that the algorithm is developed for a robot equipped with a minimum set of a hardware, that is the iRobot Create in the base set.

The proposed approach implies that the robot motion map is indexed by concepts of ontology which is developed on the OWL language and describes the possible obstacles and ways to overcome it. In addition, it describes such objects as the room itself, the doorway, the steps, the thresholds, the type of flooring, and so on. Thus, for a particular movement purpose, firstly the robot builds or loads the map, which is constructed before, and then, interpreting the concept of the map objects, paves the route.

Now, the prototype of an objects ontology and a library communication have been completed. The platform of implementation is Java. The developed software system will be used in the learning process in the courses related to the study of intelligent systems and ontological engineering.

АВТОРСКИЙ УКАЗАТЕЛЬ

А

Агафонов Г.В.	565
Азаров И.С.	513
Алексеев А.В.	577
Андреев И.А.	395
Аскарова С.А.	165
Афанасьев А.Н.	347, 385
Афанасьева Т.В.	385

Б

Бабамухамедова М.З.	369
Бармина Е.И.	605
Башаев В.А.	395
Бикмуллина И.И.	445
Билалов Р.Р.	439
Бойченко А.В.	225
Бойченко В.С.	455
Боргест Н.М.	417, 421
Борзов Д.А.	465
Борискин А.С.	93
Буданова Н.	169
Бурдо Г.Б.	461, 477

В

Вагин В.Н.	519
Ванясин Н.В.	581
Вереник Н.Л.	133
Виноградов Г.П.	465, 477
Вишневский С.Я.	533
Вишняков В.А.	173
Власов С.А.	205, 417
Войт Н.Н.	347
Воробьев В.В.	489
Воробьева Е.В.	461
Вяхирев А.А.	561

Г

Гаврилова Т.А.	205
Гайнуллин Р.Ф.	385
Галушка И.Н.	249
Ганишев В.А.	519
Гатауллин Р.Р.	451
Герасюто С.Л.	487
Гецэвіч Ю.С.	499, 507
Гильмуллин Р.А.	451
Гирель А.И.	133
Гладун А.Я.	255
Глазкова А.В.	541
Глоба Л.С.	231, 237, 241
Голенков В.В.	57
Головко В.А.	481
Гондаз Саз М.М.	173
Грибова В.В.	183
Гришин М.В.	381
Губаревич А.В.	93
Гулякина Н.А.	57
Гюнтар А.В.	499

Д

Давыденко И.Т.	79, 93, 429
Деева Н.В.	533
Деревянко А. В.	549
Джангазова К. А.	369
Дзенісюк Д.А.	499
Додонов А.Г.	225
Долбин А.В.	571
Доцанова М.Ю.	365, 369

Е

Ефименко И.В.	43
---------------	----

Ж

Жилякова Л.Ю.	303	Мальковский М.Г.	281
Жуков И.И.	79	Маргунов Е.А.	149
З		Массель А.Г.	199
Заболеева-Зотова А.В.	455, 555, 561	Массель Л.В.	199
Загорулько Г.Б.	177	Меньшаков П.А.	525
Загорулько Ю.А.	177	Мирзаев Д.А.	365
Замятина Е.Б.	597	Митраков А.А.	597
Захар'еў В.А.	499	Моздуоани Шираз М.Г.	173
Зуенко А.А.	297	Молчанов Ю.Н.	241
И		Моросин О. Л.	549
Иванов В.К.	471	Москаленко Ф.М.	193
Ивашенко В.П.	111, 133	Мошкин В.С.	395, 401
К		Муканова А.С.	165
Каешко А.И.	149	Мурашко И.А.	525
Кайгародава Л.І.	507	Мухамедшин Д.Р.	439
Каримов Д.Ф.	597	Н	
Кипаева Е.В.	571	Найденова К. А.	291
Кириченко М.И.	565	Наместников А.М.	407
Клейн В.В.	395	Нгуен Тхи Минь Ву	275
Коровин М.Д.	417, 421	Невзорова О.А.	439
Королев Ю.И.	545	Нишанов А.Х.	365
Корончик Д.Н.	79, 89, 93	Ниязова Р.С.	169
Коршиков Д.Н.	425	Нікалаенка К.А.	507
Костебелова В.К.	221	Новогрудская Р.Л.	231
Кривдюк Н.М.	331	Носкова А.И.	425
Крощенко А.А.	481	Нургазинова Г.Ш.	157
Крылов Д.А.	193	О	
Л		Оксанич И.Г.	249
Лабанаў Б.М.	499	Омарбекова А.С.	157
Ландэ Д.В.	225	Орлова Ю.А.	555, 571, 577
Ланин В.В.	605	Очинская А.А.	297
Ларин С.Н.	381	П	
Лахин О.И.	425	Палюх Б.В.	471
Лысы С.І.	499	Паркалов А.В.	79
М		Петровский А.А.	513

Петровский А.Б.	455	Трембач В.М.	325
Пивоварчик О.В.	141	Третьяков Ф.И.	537
Плесневич Г.С.	275	Ф	
Плетнёв А.О.	605	Федорищев Л.А.	183
Прокопович Г.А.	487	Федотова А.В.	429
Р		Федяев О.И.	357
Роберт И.В.	315	Филиппов А.А.	389
Рогушина Ю.В.	255, 265	Х	
Розалиев В.Л.	561, 565	Хала Е.А.	271
С		Харламов А.А.	529
Савин А.М.	215	Хорошевский В.Ф.	43
Сейткулов Е.Н.	133	Ч	
Серебряная Л.В.	537	Чан Ван Ан	285
Сидоркина И.Г.	581, 585, 589, 593	Чоракаев О.Э.	373
Смирнов С.В.	413	Ш	
Согоян А.Л.	309	Шалфеева Е.А.	187
Сокольников А.М.	593	Шарипбай А.А.	157, 161, 165
Соловьев В.И.	495	Шелеметьев А.М.	589
Соловьев С.Ю.	281	Шелеметьева Я.В.	589
Солошенко А.Н.	555	Шереметова Е.И.	209
Сорокин А. Ю.	477	Шеркунов В.В.	351
Сорокин О.Л.	585	Штогрин Е. С.	237
Соснин П.И.	373, 381	Шункевич Д.В.	79, 93
Спирина М.О.	421	Шурыгин Ю.А.	331
Страхович Э.В.	205	Шуть В.Н.	309
Субхангулов Р.А.	389, 407	Щ	
Сычѳв В.А.	487	Щербак С.С.	249
Т		Ю	
Таранчук В.Б.	339	Юрыгина Ю.С.	425
Тарасов В.Б.	25	Я	
Татур М.М.	133	Ямшанов А.В.	331
Тельнов Ю.Ф.	325	Янковская А.Е.	331
Терновой М. Ю.	237	Ярушкина Н.Г.	395, 401
Тимченко В.А.	193		

AUTHOR INDEX

A		Efimenko I.V.	43
Afanasjev A.N.	347, 385	Eremeev A.P.	545
Afanasjeva T.V.	385	F	
Agafonov G.V.	565	Fedorischev L.A.	183
Alekseev A.V.	577	Fedotova A.V.	429
Andreev I.A.	395	Fedyayev O.I.	357
Askarova S.A.	165	Filippov A.A.	389
Azarov E.	513	G	
B		Gainullin R.F.	385
Babamukhamedova M.Z.	369	Galushka I.M.	249
Barmina E.	605	Ganishev V.A.	519
Bashaev V.A.	395	Gataullin R.R.	451
Bikmullina I.I.	445	Gavrilova T.A.	205
Bilalov R.R.	439	Gerasuto S.L.	487
Borgest N.M.	417, 421	Gilmullin R.R.	451
Boriskin A.S.	93	Girel A.I.	133
Borzov D. A.	465	Gladun A.	255
Boychenko A.V.	225	Glazkova A.V.	541
Boychenko V.S.	455	Globa L.S.	231, 237, 241
Budanova N.	169	Golenkov V.V.	57
Burdo G.B.	461, 477	Golovko V.A.	481
C		Gongas Sas M.M.	173
Chorakaev O.E.	373	Gribova V.V.	183
D		Grishin M.V.	381
Davydenko I.T.	79, 93, 429	Guliakina N.A.	57
Deeva N.V.	533	H	
Denisyuk D.A.	499	Hetsevich Y.S.	499, 507
Derevyanko A.	549	Hiuntar E.V.	499
Djangazova K.A.	369	Hubarevich N.U.	93
Dodonov A.G.	225	I	
Doldin A.V.	571	Ivanov V.K.	471
Doshchanova M.Yu.	365, 369	Ivashenko V.P.	111, 133
E		K	

Kaigorodova L.I.	507	Moshkin V.S.	395, 401
Karimov D.F.	597	Mosdurani Shiras M.G.	173
Kayeshko A.I.	149	Moskalenko Ph.M.	193
Khala C.A.	271	Mukanova A.S.	165
Kharlamov A.A.	529	Mukhamedshin D.R.	439
Khoroshevsky V.F.	43	Murashko I.A.	525
Kipaeva E.V.	571	N	
Kirichenko M.I.	565	Naidenova X. A.	291
Klein V.V.	395	Namestnikov A.M.	407
Korolev Y.I.	545	Nevzorova O.A.	439
Koronchik D.N.	79, 89, 93	Nguyen Thi Minh Vu	275
Korovin M.D.	417, 421	Nikolaenko K.A.	507
Korovin M.D.	421	Nishanov A.H.	365
Korshikov D.	425	Niyazova R.S.	169
Kostebelova V.K.	221	Noskova A.	425
Krivdyuk N.M.	331	Novogrudskaya R.L.	231
Kroshchanka A.A.	481	Nurgazinova G. Sh.	157
Krylov D.A.	193	O	
L		Ochinskaya A.A.	297
Lakhin O.	425	Oksanich I.G.	249
Lande D.V.	225	Omarbekova A.S.	157
Lanin V.	605	Orlova Yu.A.	555, 571, 577
Larin S.N.	381	P	
Lobanov B.M.	499	Palyukh B.V.	471
Lysy S.I.	499	Parkalov A.V.	79
M		Petrovsky A.	513
Malkovsky M.G.	281	Petrovsky A.B.	455
Marhunou Y.A.	149	Pivovarchyk O.	141
Massel A.G.	199	Plesniewicz G.S.	275
Massel L.V.	199	Pletnev A.	605
Menshakov P.A.	525	Prakapovich R.A.	487
Mirzaev D.A.	365	R	
Mitrakov A.A.	597	Robert I.V.	315
Molchanov Y.N.	241	Rogushina J.	255, 265
Morosin O.	549	Rozaliev V.L.	561, 565

S		Telnov Yu.F.	325
Savin A.M.	215	Ternovoy M.Y.	237
Seitkulov Y.N.	133	Timchenko V.A.	193
Serebryanaya L.V.	537	Tran Van An	285
Shalfeeva E.	187	Trembach V.M.	325
Sharipbay A.A.	157, 161, 165	Tretyakov F.I.	537
Shcherbak S.S.	249	V	
Shelemetev.A.M.	589	Vagin V.N.	519
Shelemeteva Y.V.	589	Vanyasin N.V.	581
Sheremetova E.I.	209	Verenik N.L.	133
Sherkunov V.V.	351	Vinogradov G. P.	465, 477
Shtogrina O.S.	237	Vishneuski S.Y.	533
Shunkevich D.V.	79, 93	Vishniakou U.A.	173
Shurygin Y.A.	331	Vlasov S.A.	205, 417
Shuts V.N.	309	Voit N.N.	347
Sidorkina I.G.	581, 585, 583, 589	Vorobiev V.V.	489
Smirnov S.V.	413	Vorobyeva E.V.	461
Sogoyan A.L.	309	Vyakhirev A.A.	561
Sokolnikov A.M.	593	Y	
Soloshenko A.N.	555	Yamshanov A.V.	331
Soloviev S.Y.	281	Yankovskaya A.E.	331
Soloviev V. I.	495	Yarushkina N.G.	395, 401
Sorokin A. Y.	477	Yurygina Yu.	425
Sorokin O.L.	585	Z	
Sosnin P.I.	373, 381	Zaboleeva-Zotova A.V.	455, 555, 561
Spirina M.O.	421	Zagorulko G.B.	177
Strakhovich E.V.	205	Zagorulko Yu.A.	177
Subkhangulov R.A.	389, 407	Zakharyeu V.A.	499
Sychyou U.A.	487	Zamyatina E.B.	597
T		Zhilyakova L. Yu.	303
Taranchuk VB	339	Zuenko A.A.	297
Tarassov V.B.	25	Zukov I.I.	79
Tatur M.M.	133		

Научное издание

Открытые семантические технологии проектирования интеллектуальных систем

OSTIS-2015

Open Semantic Technologies for Intelligent Systems

**МАТЕРИАЛЫ
IV МЕЖДУНАРОДНОЙ
НАУЧНО-ТЕХНИЧЕСКОЙ КОНФЕРЕНЦИИ**

(Минск, 19–21 февраля 2015 года)

В авторской редакции
Ответственный за выпуск *В. В. Голенков*
Компьютерная верстка *Н. В. Гракова*

Подписано в печать **10.02.2014**. Формат 60×84 1/8. Бумага офсетная. Гарнитура «Таймс».
Отпечатано на ризографе. Усл. печ. л. 67,43. Уч.-изд. л. 78,5. Тираж 140 экз. Заказ 30.

Издатель и полиграфическое исполнение: учреждение образования
«Белорусский государственный университет информатики и радиоэлектроники»
ЛИ №02330/0494371 от 16.03.2009. ЛП №02330/0494175 от 03.04.2009.
220013, Минск, П. Бровка, 6



Проект OSTIS

(Open Semantic Technologies for Intelligent Systems)

Это открытый проект, направленный на создание массовой семантической технологии компонентного проектирования интеллектуальных систем различного назначения

Цели проекта OSTIS

- Создать массовую, комплексную и активно развивающуюся технологию проектирования интеллектуальных систем, включающую в себя теоретические и практические, программные и аппаратные аспекты
- Создать инфраструктуру, обеспечивающую сочетание научной и учебной, инженерной и коммерческой деятельности в области искусственного интеллекта

Особенности проекта OSTIS

- Является открытым комплексным проектом, состоит из большого числа частных проектов и предоставляет полный пакет документации по всем компонентам предлагаемой технологии (включая исходные тексты соответствующих программных средств)
- Ориентирован на широкий контингент разработчиков прикладных интеллектуальных систем (на массовое распространение предлагаемой технологии)
- Ориентирован на существенное сокращение сроков проектирования интеллектуальных систем

Возможными направлениями Вашего участия в развитии проекта OSTIS могут быть

- Разработка конкретных прикладных интеллектуальных систем в самых различных предметных областях. В каждой такой разработке Вы можете принять участие в качестве эксперта соответствующей предметной области, в качестве инженера знаний, в качестве разработчика операций обработки знаний и в качестве разработчика пользовательского интерфейса.
- Разработка различных ip-компонентов проектирования интеллектуальных систем:
 - ip-компонентов баз знаний
 - ip-компонентов машин обработки знаний
 - ip-компонентов пользовательских интерфейсов интеллектуальных систем
- Разработка различных фрагментов инструментальных средств проектирования интеллектуальных систем:
 - инструментальных средств проектирования баз знаний
 - инструментальных средств проектирования программ, ориентированных на обработку баз знаний
 - инструментальных средств проектирования машин обработки знаний (предметно независимых систем операций обработки знаний)
 - инструментальных средств проектирования пользовательских интерфейсов интеллектуальных систем
- Разработка интеллектуальных help-систем, предназначенных для информационного обслуживания и обучения разработчиков интеллектуальных систем:
 - интеллектуальных help-систем по проектированию баз знаний
 - интеллектуальных help-систем по проектированию программ, ориентированных на обработку баз знаний
 - интеллектуальных help-систем по проектированию машин обработки знаний
 - интеллектуальных help-систем по проектированию пользовательских интерфейсов интеллектуальных систем
- Комплексная разработка различных частных технологий проектирования различных классов прикладных интеллектуальных систем

Перечисленные направления Вашего участия в проекте OSTIS, а также и любые другие возможные направления Вы можете уточнить на основе документации проекта OSTIS (<http://ims.ostis.net/>).

Если Вы заинтересованы в наших продуктах или партнёрстве – свяжитесь с нами

Пишите нам по любым возникающим вопросам на следующие e-mail:

- golen@bsuir.by,
- ostisconf@gmail.com



VI международная научно-техническая конференция «Открытые семантические технологии проектирования интеллектуальных систем»

Open Semantic Technologies for Intelligent Systems

OSTIS-2016

18 – 20 февраля 2016 г. Минск. Республика Беларусь

И Н Ф О Р М А Ц И О Н Н О Е П И С Ь М О

Приглашаем принять участие в VI Международной научно-технической конференции «Открытые семантические технологии проектирования интеллектуальных систем» (OSTIS-2016).

Конференция пройдет в период с *18 по 20 февраля 2016* года в Белорусском государственном университете информатики и радиоэлектроники, г. Минск, Республика Беларусь.

Рабочие языки конференции: русский, белорусский, английский.

ОСНОВНЫЕ ОРГАНИЗАТОРЫ КОНФЕРЕНЦИИ

- Российская ассоциация искусственного интеллекта (РАИИ)
- Белорусский государственный университет информатики и радиоэлектроники (БГУИР)
- Государственное учреждение «Администрация Парка высоких технологий» (Республика Беларусь)

ПРОГРАММНЫЙ КОМИТЕТ КОНФЕРЕНЦИИ

Кузнецов О.П., д.т.н., проф., РФ
Боргест Н.М., к.т.н., доц., РФ
Борисов А.Н., д.т.н., проф., Латвия
Валькман Ю.Р., д.т.н., доц., Украина
Васильев С.Н., академик РАН, РФ
Гаврилова Т.А., д.т.н., проф., РФ
Глоба Л.С., д.т.н., проф., Украина
Голенков В.В., д.т.н., проф., РБ
Головко В.А., д.т.н., проф., РБ
Гордей А.Н., д.фил.н., проф., РБ
Грибова В.В., д.т.н., РФ
Гулякина Н.А., к.ф.-м.н., доц., РБ
Еремеев А.П., д.т.н., проф., РФ
Ефименко И.В., к. фил. н., РФ
Заболеева-Зотова А.В., д.т.н., РФ
Загорюлько Ю.А., к.т.н., доц., РФ
Клещев А.С., д.т.н., проф., РФ
Кобринский Б.А., д.мед.н., РФ
Козлов О.А., д.п.н., проф., РФ
Комарцова Л.Г., д.т.н., РФ
Курейчик В.М., д.т.н., проф., РФ
Ландэ Д.В., д.т.н., Украина
Лобанов Б.М., д.т.н., проф., РБ
Лукашевич Н.В., к.физ.-мат. н., РФ

Массель Л.В., д.т.н., проф., РФ
Найденова К.А., к.т.н., РФ
Невзорова О.А., к.т.н., доцент, РФ
Осипов Г.С., д.ф.-м.н., проф., РФ
Палюх Б.В., д.т.н., проф., РФ
Петровский А.А., д.т.н., проф., РБ
Петровский А.Б., к.ф.-м.н., д.т.н., проф., РФ
Плесневич Г.С., к.ф.-м.н., РФ
Роберт И.В., д.п.н., проф., РФ
Родченко В.Г., к.т.н., доц., РБ
Сидоркина И.Г., д.т.н., проф., РФ
Смирнов С.В., д.т.н., проф., РФ
Соловьёв С.Ю., д.ф.-м.н., проф., РФ
Соснин П.И., д.т.н., проф., РФ
Стефанюк В.Л., д.т.н., проф., РФ
Сулейманов Д.Ш., академик АН Республики Татарстан, РФ
Тарасов В.Б., к.т.н., доц., РФ
Тельнов Ю.Ф., д.э.н., проф., РФ
Тузиков А.В., д.ф.-м.н., проф., РБ
Харламов А.А., д.т.н., РФ
Хорошевский В.Ф., д.т.н., проф., РФ
Чернявский А.Ф., академик НАН Беларуси
Шарипбаев А.А., д.т.н., проф., Казахстан
Щербак С.С., к.т.н., доц., Украина

НАПРАВЛЕНИЯ РАБОТЫ КОНФЕРЕНЦИИ:

- *Принципы, лежащие в основе семантического представления знаний, и их унификация. Типология знаний и особенности семантического представления различного вида знаний и метазнаний. Связи между знаниями и отношения, заданные на множестве знаний. Семантическая структура глобальной базы знаний, интегрирующей различные накапливаемые знания*
- *Языки программирования, ориентированные на параллельную обработку семантического представления баз знаний*
- *Модели решения задач, в основе которых лежит обработка знаний, осуществляемая непосредственно на уровне семантического представления обрабатываемых знаний. Семантические модели информационного поиска, интеграции знаний, анализа корректности и качества баз знаний, сборки информационного мусора, оптимизации баз знаний, дедуктивного и индуктивного вывода в базах знаний, правдоподобных рассуждений, распознавания образов, интеллектуального управления. Интеграция различных моделей решения задач*
- *Семантические модели восприятия информации о внешней среде и отображения этой информации в базу знаний*
- *Семантические модели мультимодальных пользовательских интерфейсов интеллектуальных систем, в основе которых лежит семантическое представление используемых ими знаний, и унификация этих моделей*

- Семантические модели естественно-языковых пользовательских интерфейсов интеллектуальных систем. Структура семантического представления лингвистических баз знаний, описывающих естественные языки и обеспечивающих решение задач понимания естественно-языковых текстов и речевых сообщений, а также задач синтеза естественно-языковых текстов и речевых сообщений, семантически эквивалентных заданным фрагментам баз знаний
- Интегрированные комплексные логико-семантические модели интеллектуальных систем, основанные на семантическом представлении знаний, и их унификация
- Различные технические платформы и варианты реализации интерпретаторов унифицированных логико-семантических моделей интеллектуальных систем, основанных на семантическом представлении знаний
- Средства и методы, основанные на семантическом представлении знаний и ориентированные на проектирование различных типовых компонентов интеллектуальных систем (баз знаний, программ, решателей задач, интерфейсов)
- Средства и методы, основанные на семантическом представлении знаний и ориентированные на комплексное проектирование различных классов интеллектуальных систем (интеллектуальных справочных систем, интеллектуальных обучающих систем, интеллектуальных систем управления, интеллектуальных робототехнических систем, интеллектуальных систем поддержки проектирования и др.)
- Прикладные интеллектуальные системы, основанные на семантическом представлении используемых ими знаний

ЦЕЛЬ И ФОРМАТ ПРОВЕДЕНИЯ КОНФЕРЕНЦИИ

Целью конференции является обсуждение проблем создания **открытой комплексной семантической технологии компонентного проектирования интеллектуальных систем**. Этим определяется и формат её проведения, предполагающий (1) пленарные доклады, (2) секционные заседания; (3) круглые столы, посвященные обсуждению различных вопросов создания указанной технологии; (4) выставочные презентации докладов.

Выставочная презентация докладов даёт возможность каждому докладчику продемонстрировать результаты своей разработки на выставке. Формат проведения конференции предполагает точное время начала каждого доклада и точное время его выставочной презентации.

Важнейшей задачей конференции является привлечение к её работе не только учёных и аспирантов, но и студенческой молодежи, интересующейся проблемами искусственного интеллекта, а также коммерческих организаций, готовых сотрудничать с научными коллективами, работающими над интеллектуальными системами и созданием современных технологий и их проектированием.

УСЛОВИЯ УЧАСТИЯ В КОНФЕРЕНЦИИ

В конференции имеют право участвовать все те, кто интересуется проблемами искусственного интеллекта, а также коммерческие организации, готовые сотрудничать с научными коллективами, работающими над созданием современных технологий проектирования интеллектуальных систем.

Для участия в конференции OSTIS-2016 необходимо до 1 декабря 2015 года на электронную почту конференции ostisconf@gmail.com отправить:

- **статью** для публикации в Сборнике материалов конференции OSTIS-2016. Статья на конференцию должна быть оформлена в соответствии с правилами оформления публикуемых материалов;
- **заявку доклада** на конференцию OSTIS-2016. Каждое поле заявки обязательно для заполнения, в том числе указание того автора, кто будет представлять доклад. Заполняя регистрационную форму, Вы подтверждаете согласие на обработку Оргкомитетом конференции персональных данных, публикацию статей и информации об авторах в печатном и электронном виде. В заявке доклада должна содержаться информация по каждому автору. К заявке доклада должны быть прикреплены **цветные фотографии** всех авторов статьи (это необходимо для публикации Программы конференции).

Если доклад представляется на конкурс докладов молодых учёных или на конкурс программных продуктов молодых учёных, это должно быть явно указано в заявке доклада.

Отбор статей для публикации в Сборнике и участия в работе конференции осуществляется рецензентами из числа членов Программного комитета конференции.

Заявки и статьи, оформленные без соблюдения предъявляемых требований, не рассматриваются.

До 20 января 2016 года, авторам статей, включённых в Программу конференции, направляются приглашения для участия в конференции.

Участие в конференции не предполагает организационного взноса.

ПОРЯДОК ПРЕДСТАВЛЕНИЯ НАУЧНЫХ СТАТЕЙ

Статьи (только по перечисленным выше направлениям) представляются в готовом для публикации виде (правила оформления опубликованы на сайте конференции: http://www.conf.ostis.net/images/d/d3/Правила_оформления_публикуемых_материалов_OSTIS-2016.zip). Текст статьи должен быть логически законченным и содержать новые научные и практические результаты. От одного автора допускается не более двух статей.

Оргкомитет оставляет за собой право отказать в приеме статьи в случае, если статья не будет соответствовать требованиям оформления и тематике конференции, а также, если будет отсутствовать заявка доклада, соответствующая этой статье.

КОНКУРС ДОКЛАДОВ МОЛОДЫХ УЧЁНЫХ

Среди авторов доклада, представляемого на конкурс докладов молодых учёных, могут входить учёные со степенями и званиями, но непосредственно представлять доклад должны авторы, не имеющие степеней и званий в возрасте до 35 лет.

Для того, чтобы принять участие в конкурсе научных докладов молодых учёных необходимо:

- 1) заполнить заявку на участие в конференции (http://www.conf.ostis.net/images/7/7a/Заявка_на_участие_в_OSTIS-2016.zip), в которой чётко указать своё желание принять участие в данном конкурсе;
- 2) написать статью на конференцию и отправить её по электронному адресу ostisconf@gmail.com (правила оформления публикуемых материалов размещены на сайте конференции http://www.conf.ostis.net/images/d/d3/Правила_оформления_публикуемых_материалов_OSTIS-2016.zip);
- 3) лично представить доклад на конференции.

КОНКУРС ПРОЕКТОВ МОЛОДЫХ УЧЁНЫХ

Принимать участие в конкурсе проектов молодых учёных могут проекты прикладных интеллектуальных систем и систем ориентированных на поддержку проектирования интеллектуальных систем, при этом представлять проект на конкурсе должен молодой учёный в возрасте до 30 лет, не имеющие учёных степеней.

Для того, чтобы принять участие в конкурсе программных продуктов молодых учёных необходимо:

- 1) заполнить заявку на участие в конференции (http://www.conf.ostis.net/images/7/7a/Заявка_на_участие_в_OSTIS-2016.zip), в которой чётко указать своё желание принять участие в данном конкурсе;
- 2) написать статью на конференцию и отправить её по электронному адресу ostisconf@gmail.com (правила оформления публикуемых материалов размещены на сайте конференции http://www.conf.ostis.net/images/d/d3/Правила_оформления_публикуемых_материалов_OSTIS-2016.zip);
- 3) лично представить доклад на конференции;
- 4) провести выставочную презентацию, разработанного программного продукта.

КОНКУРС СТУДЕНЧЕСКИХ ПРОЕКТОВ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ

В конкурсе студенческих проектов могут принимать участие проекты, разработчиками которых являются студенты и магистранты высших учебных заведений, консультантами и руководителями проекта могут быть лица, имеющие научную степень и звание. Для того, чтобы принять участие в данном конкурсе необходимо:

- 1) ознакомиться с положением о конкурсе студенческих проектов (http://www.conf.ostis.net/images/9/94/Положение_о_конкурсе_студенческих_проектов.pdf http://conf.ostis.net/images/9/9a/Положение_о_конкурсе_студенческих_проектов_2016.pdf);
- 2) заполнить заявку на участие в конкурсе студенческих проектов (http://www.conf.ostis.net/images/4/44/Заявка_на_участие.zip);
- 3) подготовить описание проекта (http://www.conf.ostis.net/images/9/9f/Описание_проекта.zip).
- 4) выслать заявку на участие в конкурсе и описание проекта по электронному адресу конкурса студенческих проектов: ostis.stud@gmail.com.

ПУБЛИКАЦИЯ МАТЕРИАЛОВ КОНФЕРЕНЦИИ

Оргкомитет конференции предполагает публикацию статей, отобранных Программным комитетом по результатам их рецензирования, в Сборнике материалов конференции и на официальном сайте конференции <http://conf.ostis.net>. Неимущественные права принадлежат авторам статей, поэтому публикация и распространение материалов статей на иных информационных ресурсах допускается только с согласия авторов статей.

КЛЮЧЕВЫЕ ДАТЫ КОНФЕРЕНЦИИ

<i>1 октября 2015г.</i>	начало подачи материалов для участия в конференции
<i>1 декабря 2015г.</i>	срок получения материалов для участия в конференции Оргкомитетом
<i>15 января 2016г.</i>	срок принятия решения о публикации присланных материалов и рассылки приглашений для участия в конференции и сообщение о включении статьи в Сборник материалов конференции OSTIS
<i>1 февраля 2016г.</i>	размещение на сайте конференции http://conf.ostis.net/index.php?title=OSTIS-2016 проекта программы конференции
<i>9 февраля 2016г.</i>	размещение на сайте конференции http://conf.ostis.net/index.php?title=OSTIS-2016 Сборника материалов и Программы конференции OSTIS-2016
<i>18 февраля 2016г.</i>	регистрация участников и открытие конференции OSTIS-2016
<i>18-20 февраля 2016г.</i>	работа конференции OSTIS-2016
<i>25 февраля 2016г.</i>	публикация фоторепортажа и отчёта о проведённой конференции на сайте конференции: http://conf.ostis.net/index.php?title=OSTIS-2016

ФОРМИРОВАНИЕ ПРОГРАММЫ КОНФЕРЕНЦИИ

Программа конференции формируется Программным комитетом по результатам рецензирования, представленных статей, а также на основании подтверждения автора(-ов) статьи о прибытии на конференцию.

КОНТАКТНЫЕ ДАННЫЕ ОРГАНИЗАТОРОВ КОНФЕРЕНЦИИ OSTIS

Вся необходимая информация по предстоящей и предыдущих конференциях OSTIS находится на сайте конференции <http://conf.ostis.net>.

Материалы для участия в конференции представляются в Оргкомитет конференции по электронной почте ostisconf@gmail.com.

Методическая и консультативная помощь участникам конференции осуществляется только через электронную почту конференции.

Конференция проходит в Республике Беларусь, г. Минск, ул. Платонова, 39 (5-ый учебный корпус Белорусского государственного университета информатики и радиоэлектроники).