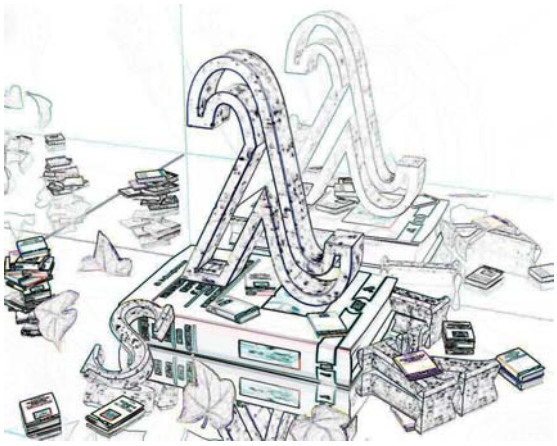
В.Э. Вольфенгаген (Ред.)

Аппликативные Вычислительные Системы

3-я международная конференция ABC 2012
Москва, 26-28 ноября 2012 г.
Труды
http://jurinfor.exponenta.ru/ACS2012



Институт «ЮрИнфоР-МГУ»

В.Э. Вольфенгаген (Ред.)

Аппликативные Вычислительные Системы

3-я международная конференция ABC 2012
Москва, 26-28 ноября 2012 г.
Труды
http://jurinfor.exponenta.ru/ACS2012

Институт «ЮрИнфоР-МГУ»

ББК 32.97я431(2Рос)
УДК 004.7(063)
А769

Аппликативные вычислительные системы: Труды 3-й международной конференции по аппликативным вычислительным системам (АВС'2012), Москва, 26-28 ноября 2012 г. / Под ред. В.Э. Вольфенгагена. – М.: НОУ Институт Актуального образования «ЮрИнфоР-МГУ», 2012. – IX+296 с.

Аппликативные вычислительные системы, или АВС, включают системы исчислений объектов, основанные на комбинаторной логике и лямбда-исчислении. Единственное, что существенно разрабатывается в этих системах – это представление об объекте. В комбинаторной логике единственный метаоператор – аппликация, или, по иной терминологии, приложение одного объекта к другому. В лямбда-исчислении два метаоператора – аппликация и функциональная абстракция, позволяющая связывать одну переменную в одном объекте. Возникающие в этих системах объекты ведут себя как функциональные сущности, имеющие следующие особенности: (1) число аргументных мест, или арность объекта, заранее не фиксируется, но проявляет себя постепенно, во взаимодействиях с другими объектами; (2) при конструировании составного объекта один из исходных объектов – функция, – применяется к другому – аргументу, – причем в других контекстах они могут поменяться ролями, то есть функции и аргументы рассматриваются как объекты на равных правах; (3) разрешается самоприменимость функций, то есть объект может применяться сам к себе. АВС дают основания аппликативному подходу к программированию.

Аппликативный компьютинг предполагает комбинационное построение вычисления как относительно самостоятельного блока, пользуясь уже имеющимися блоками вычислений, причем все переменные в каждом блоке вычисления связаны, а сам он – замкнут. Для осуществления аппликативного компьютинга применяют АВС.

ISSN 2304-7283

© Институт «ЮрИнфоР-МГУ», 2012
© АО «Центр ЮрИнфоР», 2012

Зарегистрированный патент № 50863 Государственным реестром промышленных образцов РФ 16 августа 2002 г.

Конференция осуществлена при поддержке РФФИ, проект № 12-07-06089-г
При сотрудничестве с МАБиУ

НИЯУ МИФИ • МФТИ • Институт «ЮрИнфоР-МГУ» • JurInfoR Ltd. • МАБиУ • РФФИ

Предисловие

АВС 2012 (Аппликативные Вычислительные Системы) представляла собой 3-х дневную конференцию, которая состоялась в г. Москве, 26-28 ноября 2012 г. Настоящая конференция была посвящена одному из наиболее интенсивно развиваемых направлений информатики и компьютерных наук – аппликативным вычислительным системам, включая модели вычислений, программные системы и вычислительные технологии, функциональное программирование, причем внимание было уделено также вопросам развития и использования идей, методов и представлений из области компьютерных и информационных наук. Указанное направление в настоящее время переживает период роста и становления, о чем свидетельствует все возрастающее число ежегодных крупных научных конференций по данной проблематике, научных журналов, в том числе, издаваемых в электронном виде, международных исследовательских проектов, в частности, поддерживаемых NSF США (например, проект национального масштаба GRID), Европейскими научными фондами и научными сообществами.

Подготовка и проведение конференции производились Институтом "ЮрИнфоР-МГУ", НИЯУ МИФИ, МФТИ, МАБиУ при поддержке Российского Фонда Фундаментальных Исследований (РФФИ). В целом проведение конференции преследовало исключительно научные и образовательные цели на некоммерческой основе. Особую роль в организации и проведении конференции занимало взаимодействие с международной научной общественностью в области аппликативных вычислительных технологий, а также в соответствующих отраслях компьютерных наук. Другой целью явилось развитие сотрудничества, установление личных контактов и развитие научной кооперации с научным сообществом, работающим в указанном направлении.

Конференция привлекла внимание свыше 190 участников, которыми было представлено 124 работ. Из них была отобрана 101 работа, представленная учеными из РФ и зарубежья. Было заслушано 3 обзорных приглашенных доклада, 24 секционных доклада и 18 докладов по выполненным исследованиям по грантам РФФИ. Всего было представлено и обсуждено 42 работы, организовано 2 круглых стола и дискуссии. Было отобрано и рекомендовано к опубликованию 45 статей, отражающих результаты выполненных исследований. Была организована специальная секция по случаю 70-летия Национального исследовательского ядерного университета «МИФИ» (НИЯУ МИФИ), на которой обсуждены связи компьютинга с наукой.

Декабрь 2012 г. В.Э. Вольфенгаген

Программный комитет:

Генеральный председатель:

акад. РАН А.С. Бугаев ИРЭ РАН, Москва, РФ

Председатель комитета:

проф. В.Э. Вольфенгаген НИЯУ МИФИ/МФТИ/ГК «ЮрИнфоР», Москва, РФ

Члены комитета:

Armin Cremers	Bonn University, Bonn, Germany
Roger Hindley	Swansea University, Swansea, UK
Uwe Konrad	HZDR, Dresden, Germany
Nikolai Mokhov	Fermilab, Batavia, USA
Jivka Ovtcharova	Karlsruhe Institute of Technology, Germany
Vladimir Tikhonchuk	Universite Bordeaux, Bordeaux, France
к.соц.н. О.А. Брежнев	Минский Университет, Минск, Беларусь
декан ф-та КиБ С.В. Дворянкин	НИЯУ МИФИ, Москва, РФ
к.ф.-м.н. В.Н. Домрачев	НБУ, Киев, Украина
проф. В.С. Жидков	Москва, РФ
к.ю.н. О.О. Журавлева	Институт «ЮрИнфоР-МГУ», Москва, РФ
к.т.н. Л.Ю. Исмаилова	НИЯУ МИФИ, Москва, РФ
проф. С.В. Клименко	МФТИ, Москва, РФ
С.В. Косиков	НИЯУ МИФИ, Москва, РФ
декан ФИВТ В.Е. Кривцов	МФТИ, Москва, РФ
проф. Г.С. Курганская	ИГУ, Иркутск, РФ
д.т.н. Г.С. Лебедев	ГУ ЦНИИОИЗ МЗ РФ, Москва, РФ
к.ф.-м.н. А.А. Лемперт	ИДСТУ СО РАН, Иркутск, РФ
к.х.н. Н.С. Лялюшко	РФФИ, Москва, РФ
И.В. Мажирин	ООО «КИФФ», Москва, РФ
к.т.н. С.Г. Маслов	ИжГТУ, Ижевск, РФ
к.т.н. С.А. Сергеев	МАБиУ, Москва, РФ
проф. В.В. Серов	МГУТУ им. К. Г. Разумовского, Москва, РФ
к.ф.-м.н. Е.А. Слободюк	МФТИ, Москва, РФ
проф. Ю.Г. Сметанин	ВЦ РАН/РФФИ, Москва, РФ
Р.В. Храпко	ЗАО «Интерлот», Москва, РФ
проф. Г.Ш. Цициашвили	ИПМ ДВО РАН, Владивосток, РФ

Организационный комитет:

к.т.н. С.А. Сергеев, (МАБиУ, Москва, РФ)
Э.Ю. Исмаилова (ГК «ЮрИнфоР», Москва, РФ)

Место проведения: МАБиУ
Москва, 5-й проезд Марьиной Рощи, д. 15-а

URL: http://mabiu.ru/contacts.php

Тел.: (495) 778-87-26, (495) 971-73-96; e-mail: info@jurinfor.ru.
URL: http://www.jurinfor.ru, http://jurinfor.exponenta.ru/acs2012

За последние двадцать лет одним из наиболее интенсивно разрабатываемым понятием было и остается представление о *вычислении* (computing), которое оказалось оснащенным комплексом дисциплин, состав и содержание которых непрерывно трансформируется. Только в самое последнее время специальной объединенной комиссией Ассоциации по Вычислительной Технике (Association for Computing Machinery, ACM) и Компьютерным сообществом Института Инженеров по Электротехнике и Электронике (IEEE Computer Society) был представлен отчет, содержащий рекомендации по преподаванию информатики и типовым учебным планам этой дисциплины. Вместе с тем и процесс научного становления этой дисциплины нельзя считать завершенным.

Книги серии “Компьютерные науки и информационные технологии” (http://www.jurinfor.ru/library/ser.php?SERID=CS)

Вольфенгаген В.Э. Конструкции языков программирования. Приемы описания. – М.: АО «Центр ЮрИнфоР», 2001. – 276 с. ISBN 5-89158-079-9.

Вольфенгаген В.Э. Категориальная абстрактная машина. Конспект лекций: введение в вычисления. – 2-е изд. – М.: АО «Центр ЮрИнфоР», 2002. – 96 с. ISBN 5-89158-102-7.

Вольфенгаген В.Э. Комбинаторная логика в программировании. Вычисления с объектами в примерах и задачах. – М.: МИФИ, 1994. – 204 с.; 2-е изд., М.: АО «Центр ЮрИнфоР», 2003. – 336 с. ISBN 5-89158-101-9; 3-е изд. дополн. и перераб. – М.: Институт «ЮрИнфоР-МГУ», 2008. – X+384 с. ISBN 978-5-91329-013-7.

Wolfengagen V.E. Combinatory logic in programming. Computations with objects through examples and exercises. – 2-nd ed. – M.: «Center JurInfoR» Ltd., 2003. – x+337 c. ISBN 5-89158-101-9.

Вольфенгаген В.Э. Методы и средства вычислений с объектами. Аппликативные вычислительные системы. – М.: JurInfoR Ltd., АО «Центр ЮрИнфоР», 2004. – xvi+789 с. ISBN 5-89158-100-0.

Вольфенгаген В.Э. Логика. Конспект лекций: техника рассуждений. 2-е изд., дополн. и перераб. – М.: АО «Центр ЮрИнфоР», 2004. – 229 с. ISBN 5-89158-135-3.

Косиков С.В. Информационные системы. Категорный подход. – Под ред. к.т.н. *Л.Ю. Исмаиловой.* – М.: АО «Центр ЮрИнфоР», 2005. – 92 с. ISBN 5-9587-0005-7.

Вольфенгаген В.Э. Аппликативные вычислительные технологии. Готовые решения для инженера, преподавателя, аспиранта, студента. / Под ред. к.т.н. *Л.Ю.Исмаиловой.* -- М.: ЗАО «ЮрИнфоР», 2009. 64 с. ISBN 978-5-903678-04-4.

Вольфенгаген В.Э. Парадигма функционального программмирования – Под ред. к.т.н. *Л.Ю. Исмаиловой.* – М.: АО «Центр ЮрИнфоР», 2012. – 96 с. ISBN 978-5-89158-178-4.

Вольфенгаген В.Э., Исмаилова Л.Ю., Косиков С.В. Парадигма функционального программирования (на CD) – Под ред. к.т.н. *Л.Ю. Исмаиловой.* – М.: АО «Центр ЮрИнфоР», 2012. – 96 с. ISBN 978-5-89158-182-5 Номер госрегистрации 0321200743

Книги серии “ Фундаментальные основы информационных технологий” (http://www.jurinfor.ru/library/ser.php?SERID=FBIT)

Wolfengagen V.E. Applicative computing. Its quarks, atoms and molecules. – Edited by Dr. L.Yu.Ismailova. M.: «Center JurInfoR» Ltd., 2010. – 62 p. ISBN 978-5-89158-177-7.

Содержание

Трудно формализуемые предметные области

Область между практическими навыками и фундаментальными принципами вычислений. 1-7
Вольфенгаген В.Э.

Pattern-Based Enterprise Systems: Models, Tools and Practices. 8-18
Zykov S.V.

Some approaches to formalization of subject domains of organizational systems on the basis of applied semiotics. 19-23
Kulyanitsa A.L., Fomitcheva O.E.

Информационная поддержка принятия решений при управлении академической мобильностью с использованием инструментария инженерии знаний и интеграции информации. 24-32
Юсупова Н.И., Сметанина О.Н., Ясинецкий С.П.

Использование кванторов в задаче оптимизации поиска в информационной системе по рукопашному бою. 33-39
Степанов А.Л., Балакирев А.А., Климаков А.В., Мамедов Н.В.

Проблемы повышения понятности знаний на семиотическом уровне в интеллектуальных системах. 40-44
Куляница А.Л., Фомичева О.Е.

Круглый стол: От науки к технологии и бизнесу

Клиентоориентированные информационные технологии в налоговом администрировании. 45-54
Журавлева О.О., Исмаилова Л.Ю.

Правовое регулирование налогообложения инновационного сектора КНР: сравнительно-правовой анализ. 55-57
Журавлева О.О., Севальнев В.В.

Оценка информационной надежности адаптивного аппаратного обеспечения, использующего RAM на основе элементов памяти с фазовыми переходами. 58-60
Егармин К.Н., Воронков Э.Н.

Гибридные информационные технологии для анализа и классификации сложноструктурируемых биомедицинских изображений. 61-64
Томакова Р.А., Брежнева А.Н.

Совершенствование технической подготовки студентов, обучающихся по направлению «Информационная безопасность» в Центре подготовки Институт Безопасности бизнеса МЭИ (ТУ). 65-69
Невский А.Ю.

Защита конфиденциальной информации при использовании «облачных» вычислений. 70-75
Невский А.Ю.

Моделирование объекта автоматизации в процессе проектирования информационных систем. 76-78
Ковтун И.И.

Проблемы обеспечения защиты информации при использовании информационных технологий. 79-81
Баронов О.Р.

Информационно-аналитическая платформа «Дедуктор» для решения задач бизнес аналитики. 82-87
Нестерович С.А.

Системы обучения и среда вычислений

Системный анализ проблемы управления качеством обучения студентов за рубежом и известных подходов к ее решению. 88-93
Ташани Н.А.

UML- модели акторов и их взаимодействие с интеллектуальной системой управления качеством обучения студентов за рубежом (ИС УКО). 94-100
Ташани Н.А.

Разработка электронного курса «Языки и методы программирования». 101-102
Шабунин Л.В., Сидорова Е.Б.

Перспективные исследования в моделях вычислений

Абстрактные алгебры различных классов программ. 103-128
Непейвода Н.Н.

Применение двойственных сетей для построения вычислений в сетевых моделях сложных систем. 129-130
Петров А.Е.

Асимптотика вероятности связности графа с низко надежными ребрами. 131-133
Цициашвили Г.Ш.

Аппликативная среда с ограничением для функций трехзначной логики. 134-135
Шабунин Л.В., Шабунин А.Л.

Перспективные исследования в информационных технологиях

Средства специализации концептуальных моделей на основе определённых дескрипций. 136-149
Исмаилова Л.Ю., Косиков С.В.

Специализация концептуальных моделей на основе интенсиональных операторов. 150-163
Исмаилова Л.Ю., Косиков С.В.

Коллективная информатика в решении инфраструктурных проблем. 164-166
Бельтюков А.П., Маслов С.Г.

Дескриптивно-конструктивные аспекты е-публикаций. 167-169
Маслов С.Г.

Инновационные системы

Описание процессов в инновационных проектах. 170-172
Степанова Е.Б., Тиренни А.

Модели и системы

Кластеризация на основе семантического анализа. 173-179
Бохуленков А.С.

Предметно-ориентированная разработка крупномасштабных программных систем с применением паттернов. 180-182
Зыков С.В.

Логический вывод в аппликативной вычислительной системе. 183-194
Мажирин И.В.

Система описания предметной области на основе семантических конструкций: интерактивная визуализация с использованием силового подхода. 195-206
Навроцкий В.В.

Применение искусственных нейронных сетей для распознавания рукописных символов. 207-212
Парубец В.В., Девятых Д.В., Берестнева О.Г.

Построение авторских ограничений на редактирование конструкций практикума в сервисе групповой разработки системы тестирования. 213-217
Парфёнова И.А.

Двухуровневая концептуализация предметных областей средствами дескрипционной логики. 218-223
Рословцев В.В., Шапкин П.А.

Аппликативные методы декомпозиции вычислительных процессов. 224-233
Рословцев В.В., Шумский Л.Д.

Конечно-элементное моделирование кварцекристаллических нановесов на основе микрокамертонных резонаторов. 234-236
Матисон Н.Л., Симонов В.Н.

Supporting problem-oriented conceptual dependencies
in the web-application. 237-243
Slobodskoy E.G.

Применение клеточных автоматов в мультиагентном моделировании
транспортных систем. 244-246
Барзиков К.В., Фомичева О.Е.

Формирование приводных скалярных полей при мультиагентном моделирова-
нии пассажиропотоков. 247-249
Барзиков К.В., Фомичева О.Е.

Математические модели автоматического реферирования текста. 250-257
Цветков А.И.

Разработка интеграционной шины для среды распределения вычислительных
процессов. 258-265
Рословцев В.В., Шумский Л.Д.

70-летие Национального исследовательского ядерного университета «МИФИ»

Temporal lambda calculus: Application of typed lambda calculus for constructing
interval temporal logic. 267-276
Shumsky L.D.

Level based information theory. 277-282
Shumsky L.D.

Компьютинг и его принципы:
направление, деятельность, научная дисциплина. 283-294
Вольфенгаген В.Э.

Указатель авторов 295-295

Приложение. Видеокурс “Комбинаторная логика” (13 лекций) 296-297

Область между практическими навыками и фундаментальными принципами вычислений

Вольфенгаген В.Э.

Кафедра ПКИ и ИТ, Институт ``ЮрИнфоР-МГУ",
Москва, 119435 РФ, М. Пироговская ул., д. 5, офис 37
vew@jmsuice.msk.ru,
WWW: http://www.wolfengagen.mephi.ru

Аннотация. В настоящее время происходит резкое усиление и расширение роли вычислений/компьютинга. Это наблюдается в каждом из проявлений общественной жизни но, вместе с тем, наблюдается и общее отсутствие признания обществом их основополагающих принципов. Значительный интерес, возникший к этому представлению, позволил сформировать подход, который обещает преодолеть сохраняющийся разрыв. В этом подходе в центре внимания область между практическими навыками и фундаментальными принципами вычислений, которая продолжает существовать с течением времени.

Введение

Время от времени появляются хорошие вопросы, касающийся роли той или иной научной дисциплины. Если говорить о компьютинге, то приходится признать, что такие вопросы, как: в чем состоит предмет компьютинга? в чем состоит метод компьютинга? – следует отнести именно к хорошим вопросам. Попытки ответа на каждый из них порождают определенные исследования и стимулируют развитие всей области информационных технологий как научной дисциплины.

Компьютинг так или иначе разворачивается в *инфраструктуре*, а изменения, происходящие в инфраструктуре, в полной мере должны на практике и в теории учитываться в компьютинге. Многими компьютинг понимается чисто прагматически: как практическая деятельность, направленная на порождение и поддержание информационного процесса или взаимодействующих информационных процессов, ведущих прямо или косвенно к получению результата -- вычислению значения выражения. Теория или даже сам факт ее наличия в обществе обычно отрицаются.

Если же сосредоточиться на паре: компьютинг-инфраструктура, -- то это может многое объяснить. Вычислительное мышление играет роль инструмента, дающего возможности анализа происходящих информационных процессов – вне зависимости от того, состоялись ли уже эти процессы, находятся ли они в стадии разворачивания или только еще предполагаются как возможные. Законы

компьютинга устанавливаются и понимаются как краткий и экономичный способ выразить то общее, что происходит в многообразии частностей для конкретных информационных процессов.

Есть разрыв между фундаментальными принципами компьютинга и практикой компьютинга. Этот разрыв во многом обусловлен сложившимися в обществе представлениями и, главным образом, доминирующим в сознании потребительским отношением к компьютингу. Насколько хорош будет мир, в котором признание получит “теоретический компьютинг”? А еще лучше спросить, что надо предпринять, чтобы теоретический компьютинг появился и получил общественное признание?

Вычислительное мышление

Представление о "вычислительном мышлении" (computational thinking), по видимому, было впервые сформулировано Ж. Винг [10] в 2006 году. В настоящее время, причем в каждом из проявлений общественной жизни, наблюдается резкое усиление и расширение роли вычислений/компьютинга. Вместе с тем, в обществе отсутствует признание их основополагающих принципов. Нарастающий интерес к представлению о компьютинге позволил сформировать подход, который обещает преодолеть сохраняющийся разрыв. Преодоление этого разрыва только часть происходящего процесса, поскольку имеется и обратное движение: необычайный успех компьютерных технологий показывает, что компромиссы, концептуальные инструменты, разработанные специалистами в компьютерных науках, находят широкое применение за рамками простого вычисления. В действительности, происходящая всеобщая информатизация выдвигает новый показатель квалификации специалиста. Этот показатель можно сформулировать как способность понимать и применять фундаментальные вычислительные принципы к широкому спектру человеческой деятельности. Вычислительное мышление, таким образом, обеспечивает “основу для непрерывного изучения все более новых и передовых вычислительных концепций и технологий” [6].

Инфраструктурные вопросы

Распространению интеллектуальных достижений в области компьютинга способствует распространение вычислительного мышления. При этом вне сферы вычислительного мышления, во многом, остается вычислительная инфраструктура, которая важна в равной степени как в социальном, так и в материальном отношении. Инфраструктура в первую очередь характеризуется чрезвычайно сложной системой материальных ресурсов, институциональной практики, экономических структур, стандартов и правил, которые обеспечивают саму возможность для предоставления вычислительных услуг. Следовательно, первостепенным является понимание *динамики инфраструктуры* как важнейшего

навыка, необходимого для мира, в котором эволюция бесчисленных профессиональных областей, начиная от ремонта автомобиля и до гуманитарных наук, стала тесно связанной с эволюцией самих вычислений. Действительно, для отстаивания своих профессиональных возможностей, студентам и специалистам-практикам различного профиля, приходится отвечать на не менее широкий спектр инфраструктурных вопросов. Среди этих вопросов следующие пять заслуживают наибольшего внимания:

- какие форматы данных и стратегии хранения предлагают лучшие гарантии для долгосрочного сохранения цифровых активов, с учетом рабочих процессов конкретной организации, существующих вычислительных ресурсов, бюджетных ограничений, а также нормативных требований;
- как экономии на масштабах предоставляемых облачных вычислений могут лучшим образом повлиять на данный проект? Какие сбалансированные издержки, эффективности и риски должны быть учтены;
- какой из имеющихся десятков стандартов метаданных обеспечивает -- для данного типа цифровых ресурсов, -- наилучшую отдачу инвестиции для их обнаружения и организации к ним доступа;
- какие новые и прибыльные информационные сервисы можно построить с использованием API Google Books? Какие преимущества могут принести эти сервисы при использовании достижений в интерфейсе и технологиях считывания, включая осязание, речь, слежение глазами;
- каковы последствия принятия программного обеспечения с открытым исходным кодом для организаций? Как можно различать типы управления и экономических структур, которые положены в основу различных проектов с открытым кодом.

Для ответа на эти вопросы требуется совершенно определенный набор навыков, который позволяет осуществлять анализ тех сложных сил, которые направляют развитие инфраструктуры.

При ответе на подобные вопросы ни приобретенные навыки работы с приложениями, ни знания фундаментальных принципов вычислений сами по себе не дают должных указаний. Они не проясняют смысла заголовков крупнейших газет и журналов, относящихся к технологии. При этом смысл достоинств и недостатков сетевого нейтралитета, интенсивности войны кодеков или определения инфраструктурных работ нашего времени, переход к облачным вычислениям остается невыясненным. Для ответа на такие вопросы нынешние специалисты и специалисты будущего должны иметь доступ к материально-технической основе вычислительной инфраструктуры и принципам ее социальной организации. Для выпускников профильных учебных заведений такое требование все чаще становится образовательным критерием, предъявляемым к рабочей силе в настоящее время [8]. От них требуется набор навыков для анализа сложных сил, которые направляют инфраструктурные эволюции. Эти навыки призваны обеспечить возможность и средства для прогнозирования кривой роста в постоянно развивающемся технологическом мире.

Тенденции развития инфраструктуры

По определению, все инфраструктуры стремятся к невидимости. На практике пользователи заинтересованы в приложениях, которые выполняют полезные для их жизни сервисы, а не в слоях системных абстракций или физических компонентов, которые материализуют эти сервисы. Тем не менее, проектирование, планирование и эксплуатация вычислительной инфраструктуры являются одним из основных направлений профессиональной деятельности ученых-компьютерщиков вне зависимости от того, происходят ли они на уровне операционной системы, центра обработки данных, коммуникационных протоколов, форматов файлов, стандартизации или политики работы. Примечательно, что волны социальных и технических инноваций, вызванные занятостью через Интернет, заставляют обращать внимание на те аспекты этой деятельности, в частности, протоколы TCP/IP и механизмы управления, которыми руководствуются при их разработке и реализации (см., например, [9]). Закрепление модульной структуры Интернет в качестве технической основы сетевой нейтральности еще больше подчеркивает связь между инфраструктурой, государственной политикой и экономическим развитием. Эти связи напоминают, что строение инфраструктуры никогда не зависит просто от технических императивов, но зависит от поддержания хрупкого равновесия между обслуживанием общих интересов и поддержанием экономической конкурентоспособности среди заинтересованных сторон в рамках этой инфраструктуры.

Несмотря на бурный темп развития ИТ, вычислительная инфраструктура развивается *очень медленно*.

Балансирование между противоречивыми требованиями заключено в самой основе вычислительной инфраструктуры. На инфраструктуру возлагается выполнение двух различных целей. Это, с одной стороны, посредничество в доступе приложений к физическим ресурсам вычислений -- обработке, хранению, а также подключение, -- через многоуровневые абстракции, а с другой стороны -- мультиплексирование этих ограниченных ресурсов таким образом, чтобы эффективно реагировать на конкурирующие требования от приложений. Неизбежные трудности при одновременном удовлетворении этих двух критериев приводят в движение ряд инфраструктурных динамик -- устойчивость, эффективность компромиссов, дефицит, сдвиги, -- что придает вычислительной инфраструктуре ее отличительные эволюционные особенности.

Инфраструктурная устойчивость. Несмотря на быстрый темп развития ИТ, вычислительная инфраструктура развивается очень медленно. Действительно, системные абстракции, которые доминируют в обработке информации, потоки информации и стеки ресурсов имеют повсеместное распространение, насчитывая более, чем 65 лет для модели фон Неймана и 45 лет для файлов и пакетов. Абстракции сохраняются по мере их воплощение в оборудовании (например, маршрутизаторы), программном обеспечении (например, протоколы) и письменных текстах (например, учебники по программированию). Кроме того, они сохраняются и в силу принимаемого требования жесткой модульной декомпозируемости, так как изменения в интерфейсе модуля потребовали бы пересмотра организации взаимодействия со всеми модулями связи. Долгосрочная устой-

чивость обеспечивает стабильность, необходимую для экономии на том масштабе, который требуется достигнуть. Эта экономия продолжительное время позволяет машинам с архитектурой фон Неймана иметь лучшее соотношение цена/вычислительная мощность, чем у многочисленных альтернативных архитектур, которые на протяжении десятилетий не смогли ее вытеснить. В то же время, высокая стоимость инвестиций в инфраструктуру обеспечения вычислительных ресурсов приводит к их перепрофилированию, а не просто к замене. Таким образом, стеки должны комплектоваться различными типами материальных ресурсов -- последовательной и параллельной архитектурой, витой парой и оптоволокном, магнитными лентами и флэш-памятью, -- для которых обеспечивается *совместимость*. Другими словами, инфраструктурные изменения происходит консервативно, через мутации и гибридизации, вместо того, чтобы непосредственно идти на разрыв с прошлым.

Все компромиссы имеют направленность сверху вниз. Общепринятая модульность на сегодня является мощной стратегией проектирования: от обеспечения независимости абстракции от реализации, она позволяет выполнять разбиение сложных систем на независимые, но еще скоординированные организационные единицы. Модульность также обеспечивает гибкость в принятии решений на осуществление тех или иных технических изменений. Вместе с тем, модульность рассматривается как первичный принцип для социальной и технической составляющей инфраструктуры. Примечательно, что в литературе расходы на модульность крайне редко отмечаются и/или рассматриваются. Как известно, гибкая модульность приводит к выработке и принятию абстракции, а реализация всегда осуществляется за счет эффективных компромиссов, например, известен компромисс принятия архитектуры фон Неймана и последовательных моделей программирования, который просто из раза в раз наследуется. Из-за отмеченных инфраструктурных предубеждений, существование и осуществление модульности всегда находятся под угрозой, под давлением. За счет модульности пытаются извлечь из текущей организации стеков как можно больше вычислительных заданий. Это противоречие становится особенно очевидным по мере того, как появляются новые типы вычислительных ресурсов. В рамках инфраструктуры они требуют интеграции разного вида. Примерами могут служить переходы от одноядерных к многоядерным процессорам, от магнитных накопителей к флэш-носителям, от проводных к беспроводным связям. Эти переходы влияют на все стеки ресурсов. Происходит это по мере того, как вновь и вновь достигаются эффективные компромиссы [1-2].

Таким образом, абстракция и реализации являются двумя сторонами одной медали, они всегда в напряжении и противоречии, которые обеспечивает ключевые точки входа для анализа инфраструктурных изменений [3].

Учебники по информатике (компьютерным наукам) в первую очередь сосредоточиваются на *абстракции* как фундаментальной практике в этой области.

Инфраструктура управляет ограничениями на ресурсы. Центры обработки данных уже потребляют до 3% от общей мировой выработанной электроэнергии, а это такой показатель, который может привязать число вычислительных циклов к добыче угля [5]. Однако, вычислительная инфраструктура не столько связана с управлением дефицитом электроэнергии, как с обработкой,

хранением данных и подключением ресурсов. Как известно, абстракции не только освобождают программистов от бремени отслеживания исчерпываемости ресурсов, но и от того, как ресурсы физически распределяются между конкурирующими приложениями и их обслуживают. В свою очередь, разделение ресурсов также неизбежно влечет за собой различные компромиссы эффективности, когда некоторые типы приложений получают преимущества над другими. Современная тенденция применения сетевых вычислений там, где это возможно и для все более широкого круга оказания основных услуг -- от телехирургии до интеллектуальных перевозок и национальной безопасности, -- потребует появления специальных поставщиков. Эти поставщики все более становятся занятыми разработкой политики приоритетов конкурирующих требований, предъявляемых к общим вычислительным ресурсам. Нарушения этих требований могут иметь гораздо более серьезные последствия, чем это происходит в случае музыкальных клипов.

Медленные сдвиги. Вычислительная инфраструктура является постоянно развивающейся системой, отвечая за интеграцию и рост в размере и трафике, техническое развитие и снятие с эксплуатации устаревших приложений, своевременный ввод в эксплуатацию новых приложений, сервисы и реализации, возникающие поведения и так далее. Это развитие сдерживается описанной ранее вариативной динамикой -- живучестью, эффективными компромиссами и необходимостью разделения ограниченных ресурсов. Такие ограничения приводят к сокращению набора возможных путей развития, в числе которых совместное проектирование слоев, инкапсуляция, введение новых слоев (так называемых промежуточных слоев). Следовательно, на практике развитие инфраструктуры никогда не начинается с некоторого чистого листа, а, скорее, начинается с учета столкновений интересов конкурирующих сторон, пытающихся развиваться в наиболее выгодных для себя направлениях. Даже модели OSI, дающие лучший пример проектирования сверху вниз, предполагающие модульное разбиение стека ресурсов, были немедленно оспорены на сложившемся рынке. Конкурентный путь заключался в более широком применения стека TCP/IP. Как оказалось, всегда лишь частично реагируя на рациональное управление, инфраструктурные эволюции характеризуются сдвигом, возможностями и импровизацией [4].

Заключение

В настоящее время инфраструктурно-центрированный подход выдвигает новые требования к студентам. От них все больше требуется использовать компьютинг в специальной манере, которая полностью интегрирует в сеть экономику, стандартизацию, взаимодействие человека с компьютером, модульность, материальные ресурсы вычислений, регулирование и проектирование систем. В такой перспективе, фундаментальные принципы, лежащие в основе вычислительного мышления, оказываются неотделимыми от большой черновой работы, необходимой для своей физической/материальной реализации в виде инфраструктуры. Известно несколько педагогических ресурсов, способствующих

продвижению такой точки зрения (см. [7]). По большому счету, в учебниках по компьютерным наукам основной упор делается на абстракции как основную практику в данной области. Однако, все нарастающая доля общественных отношений становится опосредованной информационными технологиями. Это приводит к необходимости понимать вычисления/компьютинг в равной степени связанными как с построением инфраструктуры, так и со всеми реальными сложностями и взаимодействием с социальным миром, которые это за собой влечет.

Благодарности

Работа является обобщением ряда результатов проектов 11-07-00305-а, 12-07-00554-а, 12-07-00646-а, частично поддержанных грантами РФФИ.

Литература

1. Blanchette, Jean-François. Computing as if Infrastructure Mattered. / Communications of the ACM, Vol. 55 No. 10, 2012, 32-34
2. Asanovic, K. et al. A view of the parallel computing landscape. Commun. ACM 52, 10 (Oct. 2009), 56–67.
3. Blanchette, J-F. A material history of bits. Journal of the American Society for Information Science and Technology 62, 6 (June 2011), 1042–1057.
4. Ciborra, C. From Control to Drift: The Dynamics of Corporate Information Infrastructures. Oxford University Press, 2000.
5. Cook, G. and Van Horn, J. How dirty is your data? Technical report, Greenpeace International. Amsterdam, 2011.
6. Lin, H.S., Editor. Report of a Workshop on the Scope and Nature of Computational Thinking. National Academies Press, Washington, D.C., 2010.
7. Messerschmitt, D.G. Networked Applications: A Guide to the New Computing Infrastructure. Morgan Kaufmann Publishers, 1999.
8. Pappano, L. The master's as the new bachelor's. The New York Times (July 22, 2011), page ED16.
9. Van Schewick, B. Internet Architecture and Innovation. The MIT Press, 2010.
10. Wing, J.M. Computational thinking. Commun. ACM 49, 3 (Mar. 2006), 33–35.

Pattern-Based Enterprise Systems: Models, Tools and Practices

Zykov S.V.

Dept. Of Business Informatics, NRU HSE,
Moscow, 119435 RF, Kirpichnaya ul., 33/5, office 526
szykov@hse.ru,
WWW: http://www.hse.ru/org/persons/3468544

Abstract. Building enterprise software is a dramatic challenge due to data size, complexity and rapid growth of the both in time. The issue becomes even more dramatic when it gets to integrating heterogeneous applications. Therewith, a uniform approach is required, which combines formal models and CASE tools. The suggested methodology is based on extracting common ERP module level patterns and applying them to series of heterogeneous implementations. The approach includes an innovative lifecycle model, which extends conventional models by: formal data representation/management models and DSL-based CASE tools supporting the formalisms. The approach has been implemented as a series of portal-based ERP systems in ITERA International Oil and Gas Corporation, and in a number of trading/banking enterprise applications elsewhere. The works in progress include semantic network-based airline dispatch system, and a 6D-model-driven nuclear power plant construction methodology.

Introduction

The paper outlines the new technology for large-scale integrated heterogeneous applications. Currently, multinational enterprises possess large, geographically distributed infrastructures. Each of the enterprises accumulates a huge and rapidly increasing data burden. In certain cases, the data bulk exceeds petabyte size; it tends to double every five years. Managing such data is an issue. The problem is even more complex due to heterogeneous data, which varies from well-structured relational databases to non-normalized trees and lists, and weak-structured multimedia data. The technology presented is focused at more efficient heterogeneous enterprise and uniform data management procedures. It involves a set of novel mathematical models, methods, and the supporting CASE tools for object-based representation and manipulation of heterogeneous enterprise systems data. The architecture is portal-based.

Managing the enterprise systems

Unfortunately, a brute force application of the so-called "industrial" enterprise software development methodologies (such as IBM RUP, Microsoft MSF, Oracle CDM etc.) to heterogeneous enterprise data management, without an object-based model-level theoretical basis, results either in unreasonably narrow "single-vendor" solutions, or in inadequate time-and-cost expenses. On the other hand, the existing generalized approaches to information systems modeling and integration (such as category and ontology-based approaches, Cyc and SYNTHESIS projects [2,7,8,10,12,13]) do not result in practically applicable (scalable, robust, ergonomic) implementations since they are separated from state-of-the-art industrial technologies. A number of international and federal research programs proves that the technological problems of heterogeneous enterprise data management are critical [11].

Thus, the suggested technology of integrated development and maintenance of heterogeneous internet-based enterprise software systems has been created. The approach is based on rigorous mathematical models and it is supported by software engineering tools, which provide integration to standard enterprise-scale CASE tools, commonly used with software development methodologies. The approach eliminates data duplication and contradiction within the integrated modules, thus increasing the robustness of the enterprise software systems (ESS). The technology takes into consideration a set of interrelated ESS development levels, such as data models, software applications, "industrial" methodologies, CASE, architecture, and database management.

The technology elements include: conceptual framework of ESS development; a set of object models for ESS data representation and management; engineering tools, which support semantic-oriented ESS development and intelligent content management, i.e., the ConceptModeller tool and the intelligent content management system (ICMS) [18,19]; portal architecture, ESS prototypes and full-scale implementations [16,19].

Modeling the enterprise lifecycle

For adequate modeling of heterogeneous ESS, a systematic approach has been developed, which includes object models for both data representation and data management [18-20]. The general technological framework of ESS development provides closed-loop, two-way construction with re-engineering. The latter feature is really critical for ESS verification, which critically increases system robustness and reliability.

The general technological framework of ESS development contains stages, which correspond to data representation forms for heterogeneous software system components, communicating in the global environment. Such data representation forms include natural language, mathematical models, engineering tools integration, and content management. The data representation forms are further detailed by the representation levels.

Content-oriented approach to ESS data management allows data/metadata generalization on the common model basis, uniform managing heterogeneous objects, and adequate modeling of the internet environment, which is critical for ESS robustness and reliability.

The object nature of the "class-object-value" model framework provides compatibility with traditional OOAD, as well as with other certain promising approaches ([15], [17]) and helps to extend the mentioned approaches to model the ESS internet-based environments. The following technological transformation sequence, according to the models developed, is suggested: (i) a finite sequence object, such as a λ-calculus term [1]; (ii) a logical predicate - higher order logics is used; (iii) a frame as a graphical representation [14]; (iv) an XML object, with the class declaration generated by the ConceptModeller engineering tool [18]; (v) a UML diagram, where the data scheme is as a part of the ESS (meta)data warehouse.

Therewith, the warehouse content representation is based on semantic network situation model, which provides intuitive transparency for problem domain analysts when they construct the problem domain description. The model can be ergonomically visualized through a frame-based notation. Warehouse content management is modeled as a state-based abstract machine and role assignments, which naturally generalizes the processes of similar engineering tools, such as (portal page template generation, portal page publication cycle, role/access management etc. Therewith, the major content management operations are modeled by the abstract machine language. The language has a formal syntax and denotation semantics in terms of variable domains.

Managing SSDL: sequential elaboration

The ConceptModeller engineering tool [18] assists in semantically-oriented visualized development of heterogeneous ESS data warehouse scheme. Therewith, a semantic network-based model is suggested, which works in nearly natural-language terms, intuitively transparent to problem domain analysts. Model visualization is based on frame representation of the warehouse data scheme.

Thus, due to deep integration with mathematical models and state-of-the-art CASE toolkits, the ConceptModeller tool provides a closed-loop, continuous ESS development cycle (from formal model to data warehouse scheme) with a re-engineering feature. Therewith, frames are mapped into specific ordered lists.

The ICMS tool is based on an abstract machine model, and it is used for problem-oriented visualized heterogeneous ESS content management and portal publication cycle. The ICMS tool features a flexible content management cycle and role-based mechanisms, which allow personalized content management based on dynamically adaptive access profiles and portal page templates. Due to scenario-oriented content management, the ICMS provides a unified portal representation of heterogeneous data and metadata objects, flexible content processing by various user groups, high data security, a higher ergonomics level and intuitively transparent complex data object management. Therewith, the data object classes of the ESS warehouse are

represented by order lists of <*attribute, type*> format, and templates – by ordered lists of <*attribute, type, value*> format.

Pattern-based development for enterprise systems

The general ESS development framework [19,20] potentially allows application of a "spiral-like" lifecycle to the general ESS development framework, which includes sequential elaboration of ESS warehouse scheme after each iteration of the development cycle. Another benefit is ESS "tuning", specifically, ESS software and data warehouse component-wise improvement, by applying a "spiral-like" lifecycle and subsequent verification. Also, requirement "tracing" implemented is possible through reverse engineering and/or verification, and followed by correction and/or optimization. As for building a repository of ESS "meta-snapshots", the system could be "reincarnated" to virtually any previous state using component-wise strategy. Also, building a "pattern catalogue" [6] for heterogeneous ESS, based on the integrated repository of various ESS state "meta-snapshots". Further, developing a repository of "branches" makes possible "cloning" slight ESS variations for the "basis". As for the DSLs, it is possible to develop a formal language specification [3] for ESS requirement specification; let us call it Requirement Specification Language or RSL. Finally, the existing ESS "meta-snapshot" repository components can be adjusted to match the new requirements, and the desired components can be reused.

Thus, the ESS development framework implies software lifecycle variations according to waterfall, spiral, evolution, and incremental approach. Though ESS development framework tends to be iterative, in certain cases, the waterfall model is possible and reasonable.

An essential feature of the general ESS development framework is its two-way organization. The approach provides reverse engineering possibility both for ESS in general, and their components in particular. The practical value of the approach is provided by the verifiability of heterogeneous ESS components at the uniform level of the problem domain model, which is practically independent upon the hardware and software environment of the particular component. Therewith, a major theoretical generalization is a possibility of mathematically rigorous verification of the heterogeneous ESS components by a function-based model. A critical issue for engineering practice of huge and complex ESS, is that the models suggested are oriented at a very promising "pure" objects approach, which is a strategy of the state-of-the-art enterprise-level component technologies of Microsoft .NET and Oracle Java, where any program entity is an object.

An essential benefit of the approach suggested is a possibility of adaptive, sequential "fine tuning" of ESS heterogeneous component management schemes in order to match the rapidly changing business requirements. Such benefit is possible due to the reverse engineering feature of the integrated general iterative framework of ESS development. The reverse engineering is possible down to model level, which allows rigorous component-wise ESS verification. Thus, conventional reengineering and verification can be enhanced by flexible correction and "optimization" of the target ESS in strict accordance with the specified business requirements. This is possible

due to the suggested model-level generalization of the iterative, evolutionary ESS development framework. Another benefit of the suggested ESS development framework is a possibility of building a "catalogue of templates for heterogeneous ESS", which is based on an integrated metadata warehouse, i.e., a "meta-snapshot" repository. Thus, the software development companies get a solution for storing relatively stable or frequently used configurations of heterogeneous ESS. The solution allows avoiding the integration problems of "standard" ESS components and/or combinations. The approach allows serious project savings for clients due to ESS developer's "meta-snapshot" repository with a number of similar integrated solutions to the system required.

The above consideration gives way for "meta-snapshot" repository development, which stores the chronological sequence of ESS solutions as a tree with the "baseline" version and slightly different "branches" for ESS variations. This is analogous to software engineering tools for version control. The approach allows a reasonable selection of most valuable deliverables of the ESS lifecycle phases, and organization of similar solution "cloning". Therewith, the "clones" may be created both for different client enterprises, and for different companies of a single enterprise.

Further discussion could cover the prospective areas of "meta-snapshot" repository development. First of all, to describe the metadata warehouses and the related enterprise-level business requirements it seems reasonable to develop new DSL-type problem-oriented meta-languages. Let us call them the MetaWarehouse Description Language (MWDL) and the Requirement Specification Language (RSL) respectively. Further, the formal models, outlined in the paper and given a more detailed coverage [19,20], allow interrelating the RSL and MWDL entities. Semantic-oriented search mechanisms based on semantic networks with frame visualization will assist in revealing the components of ESS "meta-snapshot" repository, which provide the closest matching to the new requirements. The approach potentially allows terms-and-cost-effective and adequate transforming of the existing ESS components in order to match the new requirements with minimum corrections effort and, consequently, with minimum labor expenses. Therewith, the global perspective it becomes possible to reuse certain ESS components for current or new clients. Selection criteria for such "basic" components may be percentage of reuse, ease of maintenance, client satisfaction, degree of matching business requirements etc.

Portal-Based Implementation: ITERA Oil-And-Gas Group

The methodology has been approved by internet and intranet portals in ITERA International Group of Companies. In terms of system architecture, the portals provide assignments with certain content management rights, e.g. view, modify, analyze, and generate reports. Problem-oriented form designer, report writer, online documentation and administration tools make an interactive interface toolkit. The enterprise warehouse supports integrated storage of data and metadata.

During the design stage, problem domain model specifications are transformed by the ConceptModeller SDK to UML diagrams, then by Oracle Developer/2000 integrated CASE tool – to ER diagrams and, finally, into target IS and enterprise content

warehouse storage schemes. Portal implementation process included fast prototyping and full-scale integrated Oracle-based implementation. The fast portal prototype has been designed to prove adequacy of the content-based data models, methods and algorithms. Upon prototype testing, a full-scale ESS portal-based toolkit has been implemented. Web pages automatically generated by the enterprise content management system are published at ITERA Group intranet portal and official internet site.

Portal architecture has been designed, implemented and customized according to technical specifications outlined by the author and tested for several years in a heterogeneous enterprise environment. Implementation terms and costs have been reduced about 40% compared to commercial software available, while features range has been essentially improved. Advanced personalization and content access level differentiation substantially reduces risks of the enterprise data damage or loss.

Upon customizing theoretical methods of finite sequences, categories, semantic networks, computations and abstract machines, a set of models have been constructed including problem domain conceptual model for enterprise content dynamics and statics as well as a model for development tools and computational environment in terms of state-based abstract machines, which provide integrated object-based content management in heterogeneous enterprise portals. For the model collection, a generalized development toolkit choice criteria set has been suggested for information system prototyping, design and implementation.

A set of SDKs has been implemented including ConceptModeller visual problem oriented CASE-tool and the content management system. According to the conceptual approach, a generalized interface solution has been designed for Internet-portal, which is based on content-oriented architecture with explicit division into front-end and back-end sides. To solve the task of building the architecture for enterprise content management, a fast event-driven prototype has been developed using ConceptModeller toolkit and PowerScript and Perl languages. After prototype testing, a full-scale object-oriented enterprise content management portal-based architecture has been implemented. The full-scale enterprise portal has been customized for content management and implemented in a 10,000 staff enterprise. The obtained results proved reducing implementation terms and costs as compared to commercial software available, and demonstrated high scalability, mobility, expandability and ergonomics. Portal design scheme is based on a set of data models integrating object-oriented management of (meta)data.

The data models used integrate methods of finite sequences, category theory, computation theory and semantic networks and they provide enterprise content management in heterogeneous interoperable globally distributed environments. Due to the approach, costs of enterprise content management, maintenance and integrity support have been essentially reduced, while portal modernization, customization and performance optimization procedures have been simplified. The results obtained have been used for development of a number of portals in ITERA Group: CMS, intranet/internet portals.

The models, methods and SDKs make a foundation for portal-based enterprise content management in ITERA International Group of Companies. According to ITERA experts, the portal implementation has resulted in a substantial annual cost reduction, while content management efficiency has increased essentially.

Domain-Driven Messaging System for a Distributed Trading Company

A trading corporation used to commercially operate a proprietary Microsoft .NET-based message delivery system for information exchange between the headquarters and the local shops. The system was client-server based. The client included a local database and a Windows-based messaging service, while the server side consisted of a Web service and central database. The operation/maintenance challenges were: complicated client-side code refactoring; difficult error localization/reduction; inadequate documentation; and decentralized configuration monitoring/management for remote shops. To solve the problems mentioned, an approach based on domain-driven development [5] and Domain Specific Languages (DSL) has been suggested. The approach included problem domain modeling and DSL development for managing problem domain objects.

The DSL-based model helped to conquer problem domain complexity, to filter and to structure the problem-specific information. It also provided a uniform approach to data representation and manipulation. We used an external XML-based DSL, which extended the scope of the enterprise application programming language [9]. The methodology instance included the following steps: DSL scope detection, problem domain modeling, DSL notation development, DSL restrictions development, and DSL testing. The approach was client-side focused, since this is the most changeable and challenging task. Lifecycle model is iterative, the solution is based on a redesigned architecture pattern. The Windows service is a constant part of the application, which contains a DSL parser. The DSL parser input is a current message transfer map.

The DSL scope included rules/parameters of message transfer, and new types of messages. Different shops may have different configuration instances, which made the client-side message processing/transfer structure.

The next methodology stage was building DSL-based semantic object model [9]. We got three object types: messages, message transfer channels and message transfer templates. DSL describes object metadata, i.e., configurations and manipulation rules. Templates were core model elements, and channels were links between template instances. Templates and channels together made message maps. DSL described the maps, i.e. the static part of the model, while messages referred to system dynamics and states.

Templates define actions with messages, i.e. transform or route them. Templates were grouped into the *IMessageProcessingPattern* interface. Standard routing templates were: content-based router, filter, receiver list, aggregator, splitter, and sorter. We also produced a number of domain-specific templates for system reconfiguration, server interaction, etc. Channels were used for message management. In the graph of map messaging, templates are represented as nodes, while channels are arcs between certain templates. In our case, two types of channels were implemented: “peer-to-peer” channel and error messages channel. Based on DSL classes, messaging maps were built, which were later used by parser to generate system configuration. At this stage, DSL syntax and semantics were built. Each messaging map, generally, a script, was instantiated by a file. Messaging map was built as an XML document, which

defined system configuration and contained templates for routing, message processing, transfer channels and their relationships.

While parsing messaging map, the parser creates channel objects based on DSL channel descriptions. Then it configures the messaging system by creating message processing objects in a similar way. Finally, the parser instantiates the I/O channels, and creates the required relationships between channels and message processor. The resulting DSL-based system configuration was functionally identical to the initial, C#-based one.

DSL-based refactoring resulted in an enterprise trade management system with transparent configuration and a standard object-based model. The DSL developed solved the problem of messaging management. Since changes are chiefly localized within the transfer configuration, the change management has been dramatically simplified. The DSL-based methodology conquered complexity, made the proprietary system an open, scalable, and maintainable solution. The approach is easily customized to fit a broad class of similar proprietary systems.

The Air Transportation Planning System for Russian Central Transportation Agency

Air traffic planning system is an area of work-in-progress. The problem is to develop remote access to the planning data. An operating solution currently exists. However, it is based on an outdated TAXXI-Baikonur technology, which is no longer evolving after early 2000s. The technology involves component-based visualized assembling of the server application. The ready-made VCL library components from Borland had been integrated with proprietary TAXXI components. The client side is an XML browser, i.e. a "thin" client.

The TAXXI technology is limited Microsoft Windows framework, which is the only possible basis for both client and server-side applications. According to the State Program of Planning System Updates, the Main Air Traffic Management Centre is going to create the new remote access solution. The internet-based architecture is to be implemented in Java technology and to operate on the Apache web server platform. The solution is to query Oracle-based data centre, process the query output and retrieve the results of the air traffic planned capacities to an intuitive and user-friendly GUI.

The practical application of the solution is the global enterprise-scale integrated system, which is providing a uniform and equal information access to all of the international air traffic participants. The similar globalization processes are underway in Europe and the U.S.A. The suggested pattern-based and component-wise approach is going to unify the issues of the architecture-level update and application migration in Russia. The methodology will also simplify the integration challenges of the global air traffic management software solution.

6D-modeling for nuclear power plants

Another challenging aspect of the methodology implementation is related to high-level template-based software re-engineering for nuclear power plants (NPP). To provide worldwide competitive level on the nuclear power plant production, it is necessary to meet the quality standards throughout the lifecycle, high security under long-term operation, terms-and-costs reduction for new generation facilities development. The above conditions could be satisfied only under a systematic approach, which combines state-of-the-art production potential, advanced control methods, and software engineering tools. Each stage of the NPP lifecycle is mapped into a set of business processes, where people and ESSs interact.

Identifying operation sequences, the systems form business process automation standards. For example, workflow mechanisms can assist in building enterprise standards on electronic documents validation and approval. During a certain NPP lifecycle, the enterprise systems acquire information on it. Finally, each of the enterprise systems reveals certain NPP aspects: design, technology, economics etc. Thus, various objects, the systems together describe NPP as a huge object. Heterogeneous nature of the data objects, and millions of units, make NPP a high complexity information object.

A major competitiveness criterion in nuclear power industry is a set of electronic manuals, which helps to assemble, troubleshoot, repair NPP etc. Such manual set provides transparent information models of NPP (units), which allow getting information on the object without directly contacting it. Such a versatile description, combined in a single data model is often referred to as a 6D model, which includes 3D-geometry, time and resources for operating the plant. Since mechanisms for information searching, scaling, filtering and linking, should provide complete and non-contradictory results, the information models should have well-defined semantics. The uniqueness of data entry assumes information model data acquisition by the enterprise systems throughout the lifecycle. While a single information model can be derived out of a single system, the 6D model should combine information models of a number of systems. The methodology for 6D model suggests portal-based system integration, which can be based on a “platform” capable of entire lifecycle support.

The further information model development assumes monitoring system state changes and their influence to the other parts of the system. This helps to immediately react on critical issues in NPP construction, which can be used for decision making. A wrong decision would be made otherwise under incomplete or incorrect information.

Among major nuclear industry issues is a concept of typical optimized nuclear reactor. The idea is to select typical invariant units for rapid “template-based” development of a slightly varying versions set. Applying the suggested methodology to the 6D information model of the reactor, is a promising approach to pattern-based component-wise NPP series development.

Conclusion

Implementation of the suggested approach allowed to developing a unified ESS, which integrates a number of heterogeneous components: state-of-the-art Oracle-based ERP modules for financial planning and management, a legacy HR management system and a weak-structured multimedia archive. The implementation of internet and intranet portals, which manage the heterogeneous ESS warehouse content, provided a number of successful implementations in diversified ITERA International Group of companies, which has around 10,000 employees in nearly 150 companies of over 20 countries. The systematic approach to ESS framework development provides integration with a wide range of state-of-the-art CASE tools and ESS development standards.

Other implementations and work-in-progress areas include: air transportation planning system, messaging system for a trading enterprise, a nuclear power plant and banking solutions. Each of the implementations is a domain-specific one, so the system cloning process is not straightforward, and it requires certain analytical and CASE re-engineering efforts. However, in most cases the approach reveals patterns for building similar implementation in series, which results in substantial term-and-cost reduction of 30% and over. The series can be applied both to subsidiaries, as it has been done in ITERA and is being done in Renaissance, and to different enterprises, as in case of the other clients.

References

1. Barendregt H.: *The lambda calculus (rev. ed.)*, Studies in Logic, 103, North Holland, Amsterdam, 1984
2. Birnbaum L., Forbus K., Wagner E. et al.: Combining analogy, intelligent information retrieval, and knowledge integration for analysis: A preliminary report. *In: ICIA 2005*, McLean, Virginia, USA, 2005
3. Cook S., Jones G., Kent S., Wills A.C.: *Domain-Specific Development with Visual Studio DSL Tools*, Pearson Education, Inc, 2008, 524 pp.
4. Curry H., Feys R.: *Combinatory logic*, Vol.1, North Holland, Amsterdam, 1958
5. Evans E.: *Domain-Driven Design: Tackling Complexity in the Heart of Software*. Addison Wesley, 2003, 560 pp.
6. Fowler M.: *Analysis Patterns: Reusable Object Models*, Addison Wesley, 1997, 223 pp.
7. Guha R., Lenat D.: *Building Large Knowledge-Based Systems: Representation and Inference in the Cyc Project*. Addison-Wesley, 1990
8. Güngördü Z., Masters J.: Structured Knowledge Source Integration: A Progress Report. In: *IKIMS 2003*, Cambridge, MA, USA, 2003
9. Hohpe G., Woolf B.: *Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions*. Addison Wesley, 2003, 736 pp.
10. Kalinichenko L., Stupnikov S.: Heterogeneous information model unification as a prerequisite to resource schema mapping. In: *ITAIS 2009*, Springer, 2009, pp.373-380
11. Kanazawa S., Fujiwara M. et al.: *R&D Trends for Future Networks in the USA, the EU, and Japan. NTT Technical Review*, Vol. 7, No.5, May 2009, p.p.1-6

12. Lenat D., Reed S. Mapping Ontologies into Cyc. In: *AAAI CWOSW 2002*, Edmonton, Canada, 2002
13. Panton K., Reed S., et al.: Automated OWL Annotation Assisted by a Large Knowledge Base. In: *ISWC 2004*, Hiroshima, Japan, 2004, pp. 71-80
14. Roussopulos N.: *A semantic network model of databases*. Toronto Univ., 1976
15. Scott D.: *Lectures on a mathematical theory of computations*. Oxford Computing Laboratory Technical Monograph. PRG-19, 1981, 148 pp.
16. Sushkov, N.; Zykov, S.: Message system refactoring using DSL. *In: CEE-SECR'09*, Moscow, Russia, 2009, pp.153-158
17. Wolfengagen V.: Event Driven Objects. In: *CSIT'99*, Moscow, Russia, 1999, pp.88-96
18. Zykov S.: Concept Modeller: A Frame-Based Toolkit for Modeling Complex Software Applications. *In: IMCIC 2010*, Orlando, FL, USA, pp. 468-473
19. Zykov S.: Pattern-Based Development of Enterprise Systems - from Conceptual Framework to Series of Implementations. *In: ICEIS 2011*, Beijing, China, 2011, SciTePress, Vol.4, pp. 475-478
20. Zykov S.: Pattern Development Technology for Heterogeneous Enterprise Software Systems. *Journal of Communication and Computer*, Vol.7, No4, David Publishing Co., 2010, pp.56-61

Some Approaches to Formalization of Subject Domains of Organizational Systems on the Basis of Applied Semiotics

Kulyanitsa A.L., Fomitcheva O.E.

Moscow State Mining University,
Moscow, RF
kulyanitsa@gmail.com

Abstract. Organizational systems are as complicated as a control process. These systems are characterized by almost absolute absence of the reliable mathematical models. The most important are requirements of ergonomics, naturalness of models from the point of view of human perception, simplicity of their support and updating. Actually before us there are problems of control in the conditions of a noosphere.

Introduction

Using of the applied semiotics methods, allowing to pull together representations of the person with computer realizations, is most expediently. On the other hand, it is necessary to define statuses of objects in organizational systems more strictly. There is a basic mismatch of formal language for the description and real knowledge which experts of this subject domain possess. Therefore there is a limitation of strict models possibilities for soft subject domains using. This knowledge is characterised of high anthropomorphic: uncertainty, ambiguity, discrepancy, a lot of qualitative and linguistic characteristics and estimations ("big", "considerable", "dangerous", "perspective", "steady", "proved", «negative impact makes on...», «undesirable consequences», etc.).There is and, so-called, «a human factor» (subjectivity, an emotionality, weariness, a carelessness, an illogicality, laziness, difficult and dynamical structure of interpersonal relations – liking, antipathies, trust, prejudices, insult, gratitude, revenge, family relations, «feeling of justice», "call of duty", «feeling of deep internal satisfaction», ability to work at a command, presence of own mechanism of purposes establishment and realisations). Existing methods of modeling practically completely ignore the human factor.

Scientifically proved approach to management problems in similar areas demands, first of all, development of a paradigm of representation of knowledge of the real world, and also methods of work with this knowledge: accumulation, integration, verification, telecommunication.

The world has entered into an epoch of an information society. However practically there are no results on working out of language means adequate to a task in view.

Serious steps to this direction have been made by V.I.Vernadsky who has suggested to use the term "noosphere" for the full, correct and effective account of all volume of the knowledge accumulated by a society, but during its time the applied semiotics, approaches to representation of knowledge was not work out.

Semiotics approach to a modeling problem

The semiotics approach to a modeling problem allows to keep high degree of correlation between the real world and model that helps to build model and to accompany it in the course of all life cycle of working out. Semiotics is a science, investigating structure, property and dynamics of formal symbolical systems, their relations with physical and culture the world from positions of the system world learning, construction and using of symbolical models of the validity, a rule of interpretation, manipulation and behaviour of such models.

The applied semiotics differs that in its frameworks is a question not about purely formal, but about quite real symbolical models realised by computer aids for modeling and management by real applied objects and systems.

Computer semiotics models possess semantic depth.

It allows to carry out on their basis procedures of assimilation of new knowledge, check on consistency and completeness, to plan purposeful behaviour, to carry out the substantial control of behaviour of systems, diagnosing, decision-making on management, to co-operate with modelling system in terms of behavioural concepts: actions, conditions, the purposes, scenarios, procedures, functions.These properties are not present at the compiled program decisions in languages of low level.

Semantics at this level is already lost, there are only commands of performance of those or other actions – without explanations of motivations.

Let's stop on the basic definitions. Universum – set of all entities (objects) of Universe (noosphere). An entity, or an object – something allocated with the observer from a universum from positions of certain pragmatical reasons, on what our attention is directed, what we speak or we think about, on what we refer in one way or another.

Any research, the analysis, modelling assumes exarticulation from all universum of its some subset which we will name subject domain (SD). Usually SD includes as directly object of research, modelling and-or management, and its surrounding named the environment.Their interaction and interference should be considered in a complex.

Subject domain (SD) – set of all entities (objects) having the essential relation to the solved problem of the analysis, modelling or management. First, acceptance noosphere the approach to control of organizational systems and to existence in the modern world is represented expedient. As in the world actually on the equal rights operate both material, and "ideal" objects, our picture of the world should consider it as much as possible adequately.

Secondly, that was impossible with application of classical philosophy, becomes possible now, on the basis of semiotics and researches in the field of representation of knowledge. Paradigm change in modern conditions is required, as soon as necessary for this means are exist. Thus, control intellectualization in organizational systems

demands revision and specification of problems of semiotics, approaches to construction of subject domain models in frameworks noosphere representations.

The purposes and research problems in this direction are specification of entities statuses, working out of methods of the description of purposeful activity on management of organizational systems taking into account concrete noosphere status of the considered entities.

Division of a subject domain into three components – physical, mental and culture are offered. Then there will be an explicit registration of these statuses at construction of semiotics models.

The important part of intellectual system is the module of "acquisition" or extraction of knowledge. Any system of knowledge acquisition from data is system of collective work, in which people of absolutely different trades and qualifications participate.

Such a complicated system should integrate in itself difficult processes of three kinds:

1. Directly subject activity of people proceeding in real or a material world.
2. Thought processes in consciousness of the people using this complicated system.
3. Symbol structures and their transformations occurring in the world of symbols, which occur in the functioning process of such system.

In the course of the activity (business activity) experts develop certain repeating samples or templates. As it is logical people often replace with their symbols structures. These symbol structures can be built on a uniform system basis on purpose not only the uniform description of corresponding activity, but also by analogy to computing process probably to construct semiotics process. Such process will allow carrying out symbol (semiotics) transformations with these symbol structures. Elementary units of such activity we will name by business processes. In other words, this system of paradigms which is called cognitive semiotics, should realize three functions:

The cognitive description of business processes (BP) in the form of symbol structures. Semiotics processing of such descriptions for the purpose of reception of new knowledge of the activity presented by these BP. Decision-making (including automatic) on the basis of new knowledge about BP.

Thus, the cognitive semiotics of business processes should carry out communication between symbol and cognitive representation of various structures - including BP. It would allow to build and maintain more effectively various systems managing BP.

There are many ways to raise efficiency BP, to make their more intellectual. Actually it is one of directions of modern computer science. In this process the big role is played by symbol (semiotics) procedures. So, BP are described by complex symbol systems – such as ARIS, IDEF, UML, etc. Routine parts BP can be carried out without intervention of the person.

Creative, non-standard parts BP to automate or computerise more difficult as they comprise the operations based on knowledge – in other words, are cognitive business processes. Many sciences – for example researches on an artificial intellect, management of knowledge, etc. are engaged in the description of such processes, attempts of their formalisation and a computerization. But, nevertheless, – many business

processes control systems still contain such components when the cognitive processes, leading to business decisions, to management BP proceed in a head of the person. Last years there is a rapid development of the mathematical apparatus underlying intellectual systems. One of perspective directions of researches in this area in our opinion is the further working out of the theory of the semiotics systems representing mathematical basis for construction of intellectual systems of qualitatively new level. So instead of formal system and its partial updatings it is offered to use the semiotics system allowing adequately to describe modern problem area which basic characteristics are an openness, dynamism, complex structure.

Successful projects and systems, which integrate BP and the cognitive processes which are passing in consciousness of the person and force them to be carried out on the computer, exist. In this case knowledge about BP is stored in so-called ontology and these ontologies are processed by cognitive processors which receive as a result new knowledge for BP management without participation of the person. But modern databases which support BP in various systems are inefficient for maintenance cognitive processes. Many BP control systems have much in common. Projects which try not to process the information, but to receive new knowledge, are known. These are systems of support of social networks, such as Semantic Web, Semiotic Web. They unite symbol representation of knowledge (special bases or structures of knowledge, ontologies) and systems of processing of this knowledge (cognitive processors) realised, for example, so-called agents or others cognitive "cursors". Thus, it is obviously possible:

- To connect symbol structures which describe real BP with knowledge, as result of some cognitive process.
- To develop symbol (semiotics) procedures for processing of these symbol structures for reception of new knowledge (including about BP).

It is possible to illustrate it on an example of the systems making frequently unpredictable knowledge or conclusions, based on social networks on the Internet. Sometimes it looks in the symbol form as sequence of formulas at the decision of a problem or the theorem proof. The knowledge received thus can be used as support of the decision by manager BP. The higher stage symbol and cognitive supports is independence decision making. It can be the program agent, which itself has decided to carry out business action (for example, purchase on this site), having only the most general instructions received from the person right at the beginning as first components of the knowledge, using the knowledge as the second to a component and having received new knowledge at information processing on a site to which it is connected as third base to a component cognitive process. There are already decisions of this triple problem which work already in some projects. . Systems of knowledge extraction from texts are a part of such systems. Working out of such systems is hardworking process. In quality really decision problems it is possible to lay down the aim – working out and the description symbol (semiotics) formalism which could form a basis for realization put above problems. Then we can see in these symbol structures reflexion of our knowledge about BP (objects making them and their transformations) which will serve, for the subsequent cognitive processings for the purpose of reception of new knowledge. Such the cognitive semiotics methodology can serve as the support tool by working out and perfection of various BP control systems. So, knowledge in con-

trol systems business by processes can be described as sign structures (semiotics) by means of the methodology developed in frameworks cognitive of semiotics which will allow to develop more effectively various IT systems, to do their more intellectual and independent, faster reacting to changes and, hence, by more effective. In particular, the offered approach, is represented expedient at the description of business processes in the field of organizational systems which HR control systems or personnel management systems concern. The given systems are characterized by stochastic hard formalized processes, with subjectivity of the person, making the decisions. Attempts of intellectualization of such systems are connected with difficulty of formalisation of the given subject domain, a part of problems probably to solve, applying the cognitive-semiotics approach to construction of knowledge bases, namely at the first stages of formalization of a subject domain.

Conclusion

Processes of automatization in organization systems induced revision and specification of problems of semiotics. It is reasonable to create models of a subject domain within noosferny representations. The goal of this article was to specify the statuses of objects of the world subject domain. We think that it is good idea to divide a subject domain into three components – the physical, mental and cultural world. Therefore it will be used in creation of semiotics models.

References

1. Vernadsky V. (1944) *Some words about a noosphere, from V.I.Vernakdsky's archive*: http://vernadsky.lib.ru/.
2. Pospelov D. (1975) *Big systems. Situational management*, Moscow, «Znanie», 64 p.
3. Pospelov D. (1975) *Situational management, Theory and practice*, Moscow, «Nauka», 284 p.
4. Fomitcheva O., Fedounets N. (2004) The object-production knowledge model for the construction of real time expert systems for active-dispatch mining control. The Paper was presented at Proc. of SSAKI, *International Control Conference*, Sept 23-26 2004, Glasgo, Great Britain.
5. Fomitcheva O., Fedounets N. (2004) The situation model of knowledge base for the construction of parallel expert systems. The Paper was presented at Proc. of SSAKI, *International Control Conference*, Sept 23-26 2004, Glasgo, Great Britain.
6. Kulyanitsa A., Bondarenko V. (2004) *Situation Assessment in Geographic Information Systems of Control Providing of multi component structures in problematic situations*. Workshop on Computer Science and Information Technologies (CSIT'2004), Budapest, Hungary.

Информационная поддержка принятия решений при управлении академической мобильностью с использованием инструментария инженерии знаний и интеграции информации

Юсупова Н. И., Сметанина О. Н., Ясинецкий С. П.

Кафедра ВМ и К, УГАТУ,
Уфа, 450000 РФ, К. Маркса ул., д. 12
yussupova@ugatu.ac.ru

Аннотация. Значительные объемы распределенной слабоструктурированной информации, предназначенной для обеспечения поддержки управленческих решений, требуют использования интеллектуальных технологий. Проведенный анализ работ по поддержке принятия решений при управлении академической мобильностью позволил выявить ряд нерешенных вопросов, в частности разработка моделей представления знаний в данной области и интеграция информации. Развитие процессов студенческой международной академической мобильности в России влечет за собой необходимость их решения для повышения эффективности управленческих решений.

Введение

Уровень развития академической мобильности (АМ) в вузе выступает одним из компонентов в официальной системе оценки вузов в России. Всестороннее изучение процессов развития студенческой АМ отражено в работах зарубежных и отечественных ученых П. Блюменталь, Н. Варгхис, В. Найду, Г. Нив, У. Тайхлер, Ф. Альтбах, М. Вейра, М. Келоу, Дж. Найт, Е. И. Артамонова, Н. И. Башмакова, В. А. Галичин, Л. А. Герасимова, П. С. Красовский, Т. В. Ляпина, С. С. Мкртчян, В. А. Садовничий, Р. В. Светлов и др.

Анализ информационных потоков при развитии международной студенческой академической мобильности позволил выявить основные источники информации, их общие свойства, такие как неоднородность и распределенность, слабоструктурированность информации. Принятие решений на основе такой информации должно базироваться на использовании интеллектуальных технологий. Современные методы поддержки принятия решений с использованием интеллектуальных технологий берут начало в работах таких российских и зарубежных ученых, как Н. А. Амосов, Т. А. Гаврилова, Л. Заде, Ю. И. Нечаев, Г. С. Осипов, Д. А. Поспелов, Э. В. Попов, В. Л. Стефанюк, Э. А. Трахтенгерц и др.

В данной статье рассматриваются вопросы интеллектуальной поддержки принятия решений при управлении АМ, требующие дальнейших исследований в связи с их недостаточной проработанностью.

Модели представления знаний для обеспечения интеллектуальной поддержки

На основе проведенного анализа процессов реализации АМ и данных об участии вузов России в программах АМ, выявлены: основные участники программ АМ, их цели и возможные для них типы программ АМ; масштаб развития АМ в российских вузах. Анализ задач и процессов управления АМ, организационных структур управления АМ, используемых в российских и зарубежных вузах показал многообразие и сложность решаемых задач.

По результатам проведенного контент-мониторинга информационных потоков при управлении АМ и на основе опыта разработки информационных систем предложено для обеспечения поддержки принятия решений (рис. 1) использовать технологии инженерии знаний и интеграции информации.

В данной статье не рассмотрены вопросы организации ситуационного управления АМ на базе иерархических ситуационных моделей (ИСМ).

Документальная база, используемая в системе поддержки принятия решений при управлении АМ: стандарты, образовательные программы, программы АМ и пр., выступает в роли информационного обеспечения. На основе анализа информации разработана комбинация сетевых и продукционных моделей представления знаний. В таких моделях декларативные знания описываются в сетевом компоненте знаний, а процедурные знания – в продукционном.

В общем виде сетевые модели имеют формальное представление: $H = <I, C_1, C_2, ..., C_n, \Gamma>$, где: I – множество информационных единиц; $C_1, C_2, ..., C_n$ – множество типов связей между информационными единицами. Отображение Γ задает между информационными единицами множества I, связи из заданного набора типов связей. В семантической сети допускаются связи различного типа, например, семантическая сеть программ АМ (рис. 2). Фрагмент семантической сети для представления знаний о программе АМ содержит следующие объекты: тип участников, тип мобильности, продолжительность, базовый и принимающий вузы, факультет, область знаний, сертификат, ссылка на официальную страницу, начало действия программы, срок подачи заявки. Отношения представлены таким образом: «имеет», «включает», «имеет частью» и др.

При планировании «включенной» академической мобильности формируется учебное соглашений путем сопоставления образовательных программ принимающего и базового вузов на основе семантических моделей (рис. 3). Сопоставление проводится с целью оценки дисциплин, как по количественным признакам (трудоемкости), так и по качественным (содержание дисциплин).

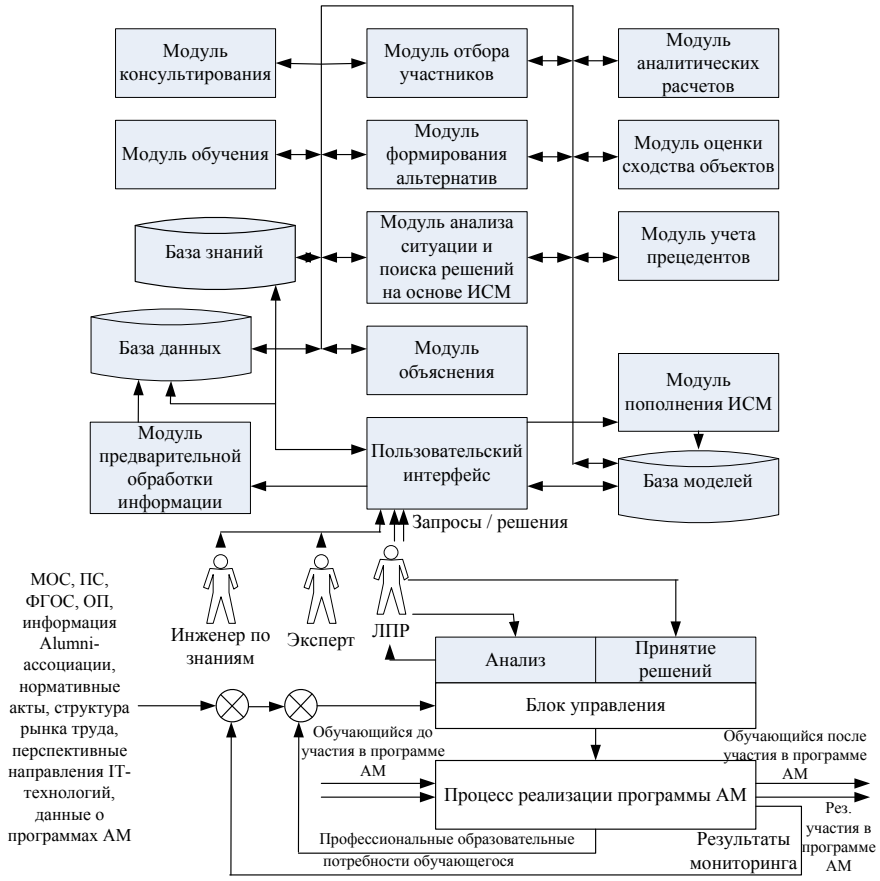


Рис. 1. Место СППР в системе управления академической мобильностью

В продукционных системах, основанных на знаниях, процесс обработки информации осуществляется двумя способами. Продукционные правила могут применяться к описанию состояния и описывать новые состояния (гипотезы) или же, напротив, использовать целевое состояние задачи как базу, когда система работает в обратном направлении. При этом продукционные правила применяются к целевому описанию для порождения подцелей.

На основе анализа заключений эксперта была определена совокупность правил логического вывода для определения семантически близких понятий между единицами сравнения (табл. 1).

При использовании продукционной модели база знаний состоит из набора правил. Программа, управляющая перебором правил, называется машиной вывода. Механизм выводов связывает знания воедино, а затем выводит из последовательности знаний заключение.

Рис. 2. Фрагмент семантической сети программы АМ

Таблица 1

Фрагмент таблицы правил логического вывода для определения семантически близких понятий между единицами сравнения

№	Правила логического вывода
*Rule*1	**Если** % совпадений <=20%, **то** совпадение = недостаточное, единицы сравнения считаются не совпавшими; осуществляется переход к следующей единице сравнения;
*Rule*2	**Если** % совпадений >20% и <=60%, **то** совпадение = норма. Эксперт принимает решение о включении единицы документальной базы в учебное соглашение;
*Rule*3	**Если** % совпадений >60% и <80%, **то** совпадение = высокое. Эксперт принимает решение о включении единицы документальной базы в учебное соглашение;

Интеграция информации из распределенных баз данных подсистем СППР при управлении АМ

Предлагаемая СППР при управлении АМ является надстройкой информационных систем и предназначена для обеспечения информационной поддержки персонала кафедры, факультета, международного отдела или отдела по организации академической мобильности. Подсистемы рассматриваемой СППР имеют разную функциональность, используют различные типы данных.

Рис. 3. Фрагменты семантических сетей образовательных программ УГАТУ (верхний) и университета *Stanford* (нижний)

Система может использовать как имеющиеся базы данных (БД), так и вновь создаваемые. Для принятия решений при управлении АМ информации, хранящейся в одной из БД недостаточно (рис. 4). Это влечет за собой необходимость организации единого информационного пространства.

Принимая во внимание территориальную распределенность информационных баз при решении задач управления АМ, особенно важным представляется такой аспект информационной поддержки, как автоматизированный поиск. В статье представлена система информационной поддержки управления АМ, обеспечивающая совместное использование различных информационных баз, имеющих значительные объемы.

Авторами рассматривается семантический уровень интеграции данных. Объединение информации достигается путем построения онтологий по отдельным БД (рис. 5) и последующем создании общей для них онтологии. Создаваемое единое информационное пространство должно позволять объединять информацию из различных БД и проводить поиск информации.

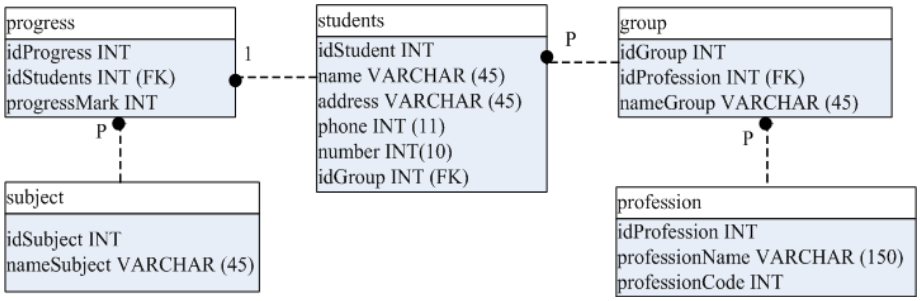


Рис. 4. Логическая модель базы данных

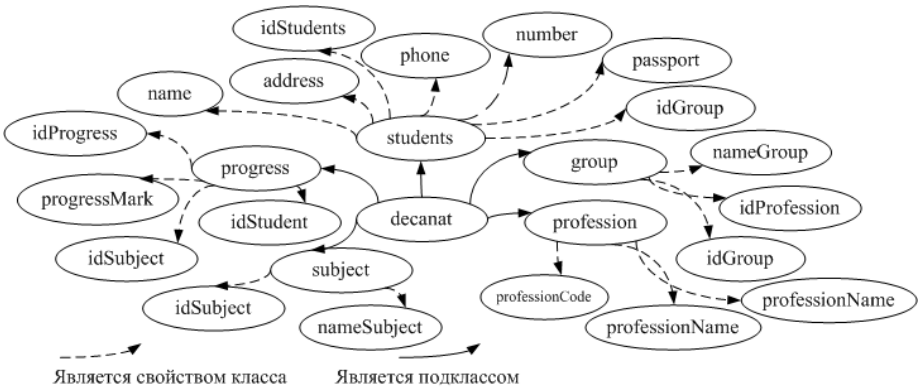


Рис. 5. Сетевая модель онтологии

Связывание данных из разных источников обеспечивается посредством способности языка описания онтологий *OWL* выражать онтологическую информацию об индивидах. Работа с информацией, содержащейся в общей онтологии (рис. 6), включает предоставление к ней доступа, поиска и вывода запрашиваемой информации пользователю. Объединение информации в онтологию из различных таблиц в БД достигается за счет использования *SWRL* правил. Фрагмент набора правил, используемых для этой цели, приведен в таблице 2.

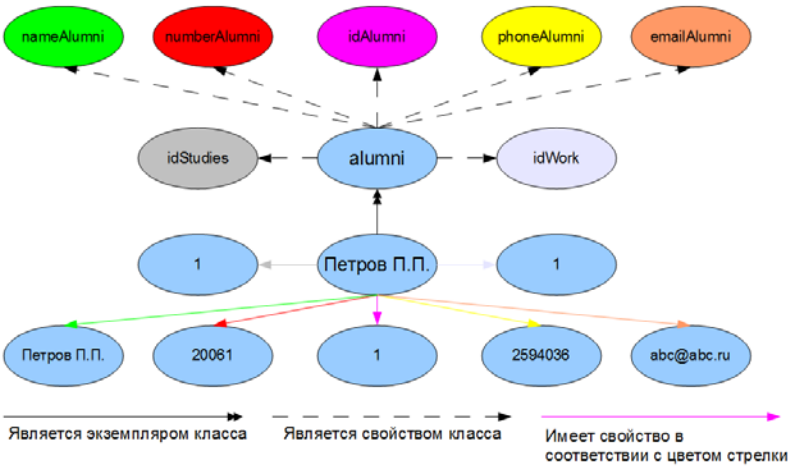


Рис. 6. Сетевая модель индивида из онтологии

Мультиагентная система, в основе которой лежит платформа *JADE* обеспечивает доступ к информации. Создаются два типа агентов. Агенты БД подключаются к базам данных, создают онтологии и предоставляют к ней доступ агентам пользователя. Агенты пользователей обеспечивают подключение пользова-

телей к единому информационному пространству и поиск необходимой информации.

Для каждого типа агентов прописываются свои сценарии (табл. 2, 3).

Поиск информации в онтологиях проводится с использованием специализированного языка запросов *SPARQL*, который используется для представления запросов к разнообразным источникам данных.

Таблица 2

Фрагмент таблицы правил объединения информации в онтологию

Правило	Описание
(?B "+SecondKey+" ?X) (?A "+PrimaryKey+" ?X) → (?B "+strURI+"/connected"+" ?A)	Если внешний ключ индивида B совпадает со значением первичного ключа индивида A, то добавляем индивиду B свойство http:// … /connected.
(?individB http:// … /connected ?individA) (?individA ?propertyA ?valueA) (?propertyA rdfs:label 'YES') → (?individB ?propertyA ?valueA)	Присоединяет индивиду B свойства индивида A. Если они связаны свойством http:// … /connected и свойства индивида A помечены, как разрешенные для присоединения.
(?individA http:// … /connected2 ?valueA) → (?individA http:// … /connected2 ?individA)	Добавляет дополнительное свойство к ответам. Необходимо для углубления поиска у данного агента.

Таблица 3

Фрагмент таблицы сценария работы пользовательского агента

Условие	Действие
Запуск	Запускается окно пользователя; запрашивается список доступных агентов БД; выбирается основной агент БД, отмечается основная БД; заполняет список дополнительных БД, заносятся прочие найденные агенты БД.
Отправка запроса на поиск	Создается и отправляется сообщение на поиск основному агенту БД.
Отправка запроса на углубление поиска	Создается и отправляется сообщение на углубленный поиск информации по индивиду. Получатель - агент от которого была получена информация с возможностью углубления поиска.
Отправка запроса на расширение поиска	Создается и отправляется сообщение на расширение поиска по индивиду. В получатели выбираются агенты выбранные пользователем в списке дополнительных БД.

Таблица 4

Фрагмент таблицы сценария работы агента БД

Условие	Действие
Первый запуск	Подключается к БД; создается файл настройки агента с расширением *.conf; завершение работы агента.
Повторный запуск	Подключается к базе данных; создается онтология согласно настройкам произведенным в файле *.conf; создаются правила работы агента; регистрируется в «Желтых страницах»;
Получение сообщения с запросом на поиск	Создание и выполнение запроса на *SPARQL* к онтологии; создание ответного сообщения агенту; преобразование результатов запроса в строку и добавление к телу ответного сообщения. Передача сообщения приславшему агенту.
Завершение работы	Сохранение онтологии БД на жесткий диск в файле с расширением *.xml; сохранение правил работы агента на жесткий диск в файлах с расширением *.rules; отмена регистрации агента в «Желтых страницах»; завершение работы агента.

Заключение

Формальные модели представления знаний в виде семантических сетей, представляющих основные понятия, определяющие классы объектов в процессе управления АМ и отношения между ними и математические модели поиска решений на основе продукционных правил позволяют получать обоснованные управленческие решения. Использование семантического уровня интеграции данных, при котором объединение информации достигается путем построения онтологий по отдельным базам данных с последующим созданием общей онтологии, позволяет объединять информацию из различных баз данных и проводить поиск информации, обеспечивая лицо, принимающее решение более полной информацией.

Литература

1. Сметанина О.Н., Маркелова А.В., Козырева В.А. Модели управления процессом реализации академической мобильности в вузе // Вестник НГУ. 2011. Т. 2, № 2. С. 55–66.
2. Гузаиров М.Б., Юсупова Н.И., Сметанина О.Н. Информационное и математическое обеспечение в системе поддержки принятия решений при управлении процессом разработки образовательной программы – М.: Машиностроение, 2011. 247 с.
3. Козырева В. А., Сметанина О. Н. Информационная поддержка при управлении образовательным маршрутом в вузе // Современные проблемы науки и образования. – 2012. – №1. URL: http://www.science-education.ru/101-5400

Использование кванторов в задаче оптимизации поиска в информационной системе по рукопашному бою.

Степанов А.Л., Балакирев А.А., Климаков А.В., Мамедов Н.В.

«Национальный исследовательский ядерный университет «МИФИ»
Москва, Каширское ш. д.31 каф. №15

Аннотация. Рассмотрены возможности применения кванторов для оптимизации поиска в информационной системе, представляющей собой базу данных приемов и перемещений рукопашного боя. Приведены примеры графического и фреймового представления ситуаций с учетом использования кванторов. Проанализированы типичные построения поисковых запросов ситуаций.

Введение

Основной задачей организации поисковой системы в базе данных приемов и перемещений по рукопашному бою является достижение высокой точности поиска и информативности получаемых результатов. Данная задача решается посредством развитой классификации технических элементов, выделения нескольких уровней поиска технических элементов, одновременного поиска технических элементов и их классов, поиска связанных с техническими элементами ситуаций по названиям технических элементов. Ранее для разработки БД приемов и перемещений рукопашного боя была использована РРМ [1], усиленная элементами комбинаторной логики. РРМ рассматривает предметы и ситуации, определяемые, как совокупность предметов и их свойств или предметов и отношений между ними. Предложенный подход [2] позволяет построить информационную систему для анализа технических действий участников спарринга. Названия свойств и отношений между участниками ситуации называются предикатами, а участники ситуации – аргументами предикатов. Кванторы ограничивают область значений аргументов предикатов. Для организации поисковой системы используются следующие виды кванторов:

$\forall$ – квантор всеобщности, $\exists$ – квантор существования, [n] – численный квантор существования n экземпляров аргументов предиката, [n} – численный квантор существования не меньше n экземпляров аргументов предиката, {n] – численный квантор существования не больше n экземпляров аргументов предиката.

Теоретический анализ

Среди различных уровней поиска в информационной системе можно перечислить следующие: 1)Поиск технического элемента по названию. 2) Поиск технических элементов по названию соответствующего класса. 3) Поиск технических элементов по названию класса и атрибута класса. 4) Поиск заданных ситуаций. 5) Поиск связанных ситуаций по названиям технических элементов. Применение кванторов важно для всех уровней поиска в информационной системе.

Для различных вариантов поиска необходимо рассмотреть возможные варианты формализации записи кванторов. Один из вариантов – фреймовое представление ситуации. Под фреймом понимается упорядоченное представление ситуации. Ситуация в данном представлении задана её участниками и отношениями между ними определяемыми падежами аблатив, пролатив, локатив, объектив. Падежи задают роли участников ситуации, что формируют отношения. T^1, T^2, T^3, T^4 – соответствующие участникам ситуации именные фразы, при этом в данном примере T^1 – связанная переменная, T^2, T^3, T^4 – свободные переменные. Свободная переменная – переменная без квантора, связанная переменная – переменная, находящаяся в диапазоне квантора. Фрейм, содержащий только свободные переменные, называется открытым. Закрытый фрейм – фрейм, содержащий хотя бы одну связанную переменную. Связывая переменные, кванторы ограничивают означивания фрейма, выражая действительное знание. Квантор всеобщности служит для указания в качестве области поиска всех возможных элементов. Квантор существования применяется для указания связанности соответствующей переменной. Так как роли участников ситуации определяются расположением термов в реляторе [3], то применение кванторов не изменяет ролей участников ситуации и сохраняет структуру предложения.

Ситуация «Прямой удар из произвольного ближнего дома в конечный дом через промежуточный дом» может быть записана фреймом (1).

$$F \cong [\forall T^1.T^4(\mathbf{a}:T^1, \mathbf{p}:T^2, \mathbf{l}:T^3)] \qquad (1)$$

Где $\mathbf{a}$, $\mathbf{p}$, $\mathbf{l}$ – аблатив, пролатив, локатив, T^1 – ближний дом, T^2 – промежуточный дом, T^3 – конечный дом, T^4 – прямой удар.

Используя РРМ данную ситуацию можно записать в виде (2).

$$S^1 = \forall T^1.\mathbf{R_{aplo}} T^1 T^2 T^3 T^4 \qquad (2)$$

Соответствие между РРМ и фреймовым представлением осуществляется по следующим правилам:

Термы константы или переменные РРМ соответствуют переменным или константам во фреймовом представлении квантификации. Предложения соответствуют простым функциональным фреймам во фреймовом представлении квантификации. Данный подход может быть использован при расширении области применения реляторно-ролевой модели в задаче моделирования динамики ролевого поведения участников ситуаций [4, 5].

Если известно число означиваний связанной переменной, то необходимо применить численный квантор. Рассмотрим соответствующий пример.

Ситуация «Перемещение выход из исходной позиции через две промежуточных позиции в позицию «Угол» может быть записана фреймом (3).

$$F \cong [[2]T^2, \exists T^3 . T^4 (\mathbf{a} : T^1, \mathbf{p} : T^2, \mathbf{l} : T^3)] \qquad (3)$$

Где T^1 – исходная позиция, T^2 – промежуточная позиция, T^3 – позиция «Род», T^4 – перемещение «выход». Используя РРМ данную ситуацию можно записать в виде (4).

$$S^2 = [2]T^2, \exists T^3 . \mathbf{R_{aplo}} T^1 T^2 T^3 T^4 \qquad (4)$$

Значение численного квантора имеет название кардинальности. В примере (3,4) кардинальность численного квантора равна 2.

Графическое представление ситуации достигается посредством использования семантической сети [6,7]. Связь между семантической сетью и фреймовым представлением достигается посредством указания в вершинах семантической сети аргументов с учетом соответствующих падежей, а также посредством отображения направленных связей. Семантическая сеть данной ситуации показана на рис.1.

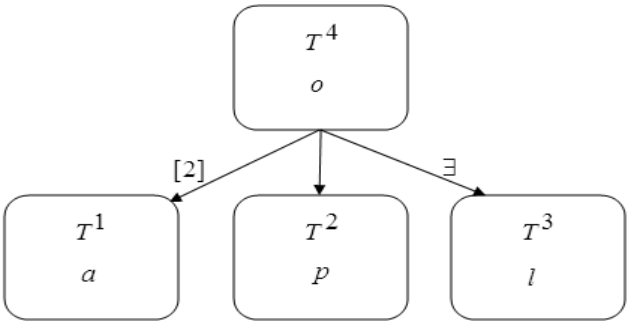


Рис 1. Семантическая сеть ситуации (3) . Падежи **a** , **p** , **l** , **o** – аблатив, пролатив, локатив, объектив.

Методика

В реляционной базе данных ситуацию (3) можно представить в виде четырех-местного отношения, что отображено в таблице №1:

Таблица №1

аблатив	пролатив	локатив	объектив
Исходная позиция	Промежуточная позиция 1	Позиция «Угол»	Перемещение «Выход»
Исходная позиция	Промежуточная позиция 2	Позиция «Угол»	Перемещение «Выход»

При этом кортеж в аблативе имеет всего два возможных значения, что задано численным квантором, кортеж локатив имеет одно возможное значение, заданное квантором существования. Значения кортежа пролатив – не определено, поскольку задано как свободная переменная. В случае если неизвестно число означиваний достаточно использовать численные кванторы, в которых количественные параметры не полностью определены. Такие кванторы обозначаются фигурными скобками. Тогда ситуация (3) преобразуется в ситуацию: «Перемещение выход из исходной позиции через не более чем две промежуточных позиции в позицию «Угол». Эта ситуация может быть записана в виде фрейма:

$$F \cong [\{2\}T^2, \exists T^3 T^4 (\mathbf{a}: T^1, \mathbf{p}: T^2, \mathbf{l}: T^3)] \qquad (5)$$

Где T^1 – исходная позиция, T^2 – промежуточная позиция, T^3 – позиция «Род», T^4 – перемещение «выход». Подобным образом, с помощью фреймового представления могут быть записаны любые ситуации, в дальнейшем для организации поиска выбирается фреймовое представление ситуации, область поиск

Экспериментальная часть

1) Рассмотрим пример организации поиска по названию технического элемента. Пусть необходимо найти «Зарочь» (обозначим данный технический элемент как T^1).

Фреймовое представление:

$$F \cong [T^1] \tag{6}$$

Выберем в качестве области поиска – таблицу «Защиты» соответствующую классу Защит. Соответствующий Sql запрос: «Select Защиты.* from Защиты where Защиты.Название = Зарочь». В результате отобразится строка о защите «Зарочь» из таблицы Защиты.

2)Поиск технических элементов определенного класса. Пусть необходимо найти все защиты (обозначим в виде S^1).

Фреймовое представление:

$$F \cong [\forall S^1] \tag{7}$$

Область поиска – таблица «Защиты». Соответствующий Sql запрос: «Select Защиты.* from Защиты». В результате отобразятся все строки о защитах из таблицы Защиты.

3) Поиск технических элементов по названию класса и атрибута класса. Пусть необходимо найти все позиции с передним расположением веса.

Фреймовое представление:

$$F \cong [\forall T^1 \exists T^2 (\mathbf{a} : T^2, \mathbf{o} : T^1)] \tag{8}$$

Где T^1 – позиция, T^2 – расположение веса. Область поиска – позиции, атрибут – расположение веса. Соответствующий Sql запрос: «Select Позиции.* from Позиции where Позиции.Расположение веса = переднее». В результате будут найдены все позиции с передним расположением веса.

Атрибутом может быть и параметр ограничение, например в запросе – «найти все комбинации, где количество действий не более 3». Фреймовое представление:

$$F \cong [\forall T^1 \{3\} T^2 (\mathbf{a} : T^2, \mathbf{o} : T^1)] \tag{9}$$

Где T^1 – комбинация, T^2 – действия. Область поиска – комбинации, атрибут – действия. Соответствующий SQL запрос фреймового представления будет выглядеть так: «Select кобинации.* from комбинации where number (комбина-

ции.действия)<=3». В результате поиска отобразятся все комбинации, где количество действий не превышает трех.

4)Поиск заданных ситуаций. Пусть необходимо найти ситуацию «Перемещение выход из исходной позиции через не более чем 2 промежуточных позиций в позицию «Угол». Фреймовое представление (5) уже было ранее рассмотрено. Область поиска – «Перемещения». Соответствующий Sql запрос: «Select Перемещение.* from Перемещение where Перемещения.конечная позиция = «Угол» and Перемещение.Название = «Выход» and number (Перемещение.промежуточная позиция<=2). В результате поиска отобразится искомая ситуация.

5)Кванторы решают задачи поиска связанных ситуаций по названию технических элементов, например: «Найти все защиты (S^1), из верхнего ближнего дома (T^2)». Такой запрос приведет к результату в виде выборки по связанной квантором переменной. Фреймовое представление данной ситуации:

$$F \cong [\exists T^1 \forall T^4 (\mathbf{a} : T^1, \mathbf{o} : T^4)] \qquad (10)$$

Где T^1 – верхний ближний дом, T^4 – защита. Область поиска защиты. Соответствующий SQL запрос фреймового представления будет выглядеть так: «Select Защиты.* from Защиты where Защиты.Исходный дом= верхний ближний дом».

Таким образом, было показано формирование основных типов поисковых запросов и возможности использования кванторов для оптимизации поиска.

Заключение

В данной работе показаны возможности развития поисковой системы на примере базы технических элементов рукопашного боя.

Представлены графическое представление ситуаций, что может быть в дальнейшем использовано в качестве средства навигации по ИС. Выполнена связь РРМ и фреймового представления ситуаций.

Проиллюстрированы на примерах применения различных кванторов для формирования поисковых запросов. Проиллюстрировано использование кванторов в логических операциях над фреймами для формирования поисковых запросов по связям технических элементов.

Литература

1. Вольфенгаген В.Э., Степанов А.Л., *Использование реляторов для описания движений в рукопашном бое*, Научная сессия МИФИ-2007, Сборник научных трудов. В 17 томах., М.: МИФИ, 2007. Т.10 «Телекоммуникации и новые информационные технологии в образовании», с.110-111
2. Вольфенгаген В.Э., Степанов А.Л., Климаков А.В., *Разработка модели спарринга в рукопашном бое*, Научная сессия МИФИ-2008, Сборник научных трудов. XII Выставка-конференция «Телекоммуникации и новые информационные технологии в образовании», М.: МИФИ, 2008. с.107-108
3. Stepanov A.L., Wolfengagen V.E., Klimakov A.V.Balakirev A.A., *Using of combinatory logic in the relator-role model*, CSIT'2009, Proceedings of the 11th International Workshop on Computer Science and Information Technologies, Crete, Greece, October 5-9, USATU, 2009, Vol. 2, pp. 11-13
4. Степанов А.Л., Климаков А.В., *Система моделирования динамики ролевого поведения в ключевых комбинациях действий для единоборств*, Научная сессия МИФИ-2010, Сборник научных трудов. XIV Выставка-конференция «Телекоммуникации и новые информационные технологии в образовании», М.: МИФИ, 2010. с.168-169
5. Степанов А.Л., *Система моделирования динамики ролевого поведения в ключевых комбинациях действий для единоборств*, Труды 2-й международной конференции по аппликативным вычислительным системам (ABC 2010), Москва, 29-31 октября 2010 г. / Под ред. Л.Ю.Исмаиловой. - М.: НОУ Институт Актуального образования ЮрИнфоР-МГУ, 2010, с.37
6. Stepanov A.L., Wolfengagen V.E., Klimakov A.V.Balakirev A.A., *Graphical representations of formal relator notations*, CSIT'2010, Proceedings of the 12th International Workshop on Computer Science and Information Technologies, Russia, Moscow-St. Petersburg, September 13-19, USATU, 2010, Vol. 2, pp. 145-148
7. Степанов А.Л., Балакирев А.А., Климаков А.В., *Использование семантических сетей в моделировании динамики ролевого поведения для единоборств*. Научная сессия МИФИ-2011, Сборник научных трудов. XV Выставка-конференция «Телекоммуникации и новые информационные технологии в образовании», М.: МИФИ, 2011. с.156-158

Проблемы повышения понятности знаний на семиотическом уровне в интеллектуальных системах

Куляница А.Л., Фомичева О.Е.

Кафедра Автоматизированных систем управления,
Московский государственный горный университет,
119991, Ленинский проспект, 6, Москва, Россия,
kulyanitsa@gmail.com,olga_fom@mail.ru

Аннотация. Последние годы характеризуются бурным развитием процессов обмена формализованными знаниями в интеллектуальных системах. Предполагается, что использование знаний позволяет повышать эффективность систем, обеспечивая лучшее понимание передаваемой и получаемой информации. Однако, понятность знаний не обеспечивается лишь использованием семантически определенного языка, для решения проблемы повышения «разумности», «понятности» и эффективности баз знаний предлагается использовать семиотический подход.

Введение

С развитием компьютерных сетей увеличивается обмен сложными знаниями, информацией и данными. Для достижения более точного и ясного описания знания должны быть представлены на формально определенных языках (от таких как XML –структурированных, обобщенных, но лишенных семантического смысла, до языков представления знаний – структурированных, определенных семантически, но с явно недостаточной степенью выразительности). Между источником части знаний и его приемником компьютер может реорганизовывать, манипулировать, фильтровать, комбинировать с другими фрагментами знаний и, наконец, представлять обработанные знания их получателю.

Трансформация знаний – это компьютерный способ последовательной генерации представлений, причем не всегда на том же самом языке. Трансформации знаний для типовых случаев могут храниться в базе знаний для их дальнейшего использования, или генерироваться специально для особенных случаев. Например, было бы полезно определить трансформацию, которая доставляет документацию заказчику, но в которой скрыта некоторая информация из первоисточника (т.е. если проектная документация включает стоимость исследований, то не благоразумно сообщать эту информацию субподрядчикам).

И наоборот, когда выработка представлений - это непрерывный процесс сотрудничества (в инженерной разработке с параллельно протекающими событиями), необходимо применять инструменты, которые не скрывают текущее положение дел. Это относится к сохранению сути, семантики процесса.

Однако интеллектуальность результирующего вывода (будучи свойством воспринимаемости знаний человеком) не обеспечивается в процессе логического вывода (трансформаций). Для того, чтобы сохранить понятность знаний для конечных пользователей, требуется совершенствовать абстрактное понимание знания от трансформации к трансформации.

Существует несколько уровней, требуемых для понимания сообщения на формальном языке (здесь описываются обобщенные значения смыслов, которые могут происходить из явной интерпретации терминов):

- **лексический** (или терминологический), на котором пользователи должны быть уверены, что они используют слова или термины с непротиворечивым смыслом. Этот уровень явно необходимо принять во внимание, что подтверждается работами по общей онтологии [3].

- **семантический**, на котором пользователи должны быть уверены, что они приписывают одинаковый смысл конструкциям используемого языка. Данная тема хорошо проработана в компьютерной науке;

- **семиотический** (или прагматический), на котором пользователи должны быть уверены, что они используют одинаковые правила интерпретации утверждений.

Последний уровень не является необходимым для использования компьютером, так как может быть указано, что нужно использовать определенные правила с соответствующей семантикой. Но когда компьютер должен взаимодействовать с человеком, то, что от него потребуется для взаимодействия, не может быть описано лишь на уровне семантики.

Приходится признавать, что компьютерная наука для обеспечения понятности на семантическом уровне предоставляет недостаточно средств, что человек при этом, интерпретируя смыслы, пользуется символами и знаками. Проанализированные примеры поясняют, что компьютерная наука должна рассматриваться на фундаменте компьютерной семиотики.

Компьютерная наука и семантика

Некоторым кажется, что если обозначить идентификаторы в программе более осмысленно (с их точки зрения) символьными именами вместо, скажем, id#356, id#873,то они «добавят больше семантики в систему». Однако, для системы, фактически, результат будет таким же. Более того, ничего не происходит с семантикой, все это относится к синтаксису! Фактически, семантика языка должна быть отражена посредством отношения к смыслу, значению, и это никак не может быть достигнуто посредством синтаксиса.

Формальная семантика была разработана для логических языков [7], и адаптирована к компьютерным языкам программирования [5] и системам представления знаний [6]. Теоретики разработали мощные средства обеспечения языков программирования семантикой, на основе теории моделирования. Грубо говоря, они обеспечили возможность интерпретировать предметную область путем структурирования. Функция интерпретации I определяется на языке предметной области D. В конечном счете, предложения могут быть определены на бу-

левских переменных (принимающих значения истина или ложь), но это не обязательно. Вместо назначения конкретной интерпретации того или иного термина, схема показывает *способ* описания значения. Таким образом, конкретный язык и конкретная предметная область могут иметь много интерпретаций.

Модель как множество выражений на языке, является некоторой интерпретацией, согласующейся со всеми выражениями (иногда большая точность достигается, когда каждое выражение принимает значение *истина*). Значение термина является, таким образом, пересечением всех интерпретаций во всех моделях, и можно сказать, выводится из набора выражений, если они истинны в каждой модели.

Интересной особенностью является то, что если пользователем рассматривается преимущественная интерпретация набора выражений, то это должно быть отражено в модели. Затем проводится обработка на языке моделирования всех моделей, с учетом интерпретации пользователя.

Данная схема подходит для применения в однопользовательской системе, так как ожидается, что пользователь интерпретирует модель в соответствии с терминами (идентификаторами), которые определил он. Мы исследовали возможности использования языка XML с точки зрения семантики. Для этой цели, язык был определен для связи, основанной на множестве модели с теоретической семантикой элементов, определенных в документе - описании. Так, форма может быть сохранена как можно больше раз (форма относится к человеческому пониманию), работоспособность обеспечивается через доступ к семантике (корректная компьютерная обработка в основном зависит от семантики). Описание семантик позволяет разработчикам инструментальных средств проверять работоспособность программного обеспечения и устранять логические ошибки.

Однако при работе нескольких пользователей, понимание не может быть обеспечено только лишь использованием семантики. Работа в области интерпретации терминов может быть успешно реализована с использованием понятия семантики, однако проблемы с реализацией интерпретации (логического вывода) остаются.

Данная проблема касается также чисто искусственных систем. Агенты, являющиеся автономными программами, способны взаимодействовать вместе. Конечно, агентный подход должен применяться к агентам, которые не были запрограммированы работать вместе. На основе прагматической теории речевых актов были построены языки типа ACL (Agent Communication Languages), практика применения которого показала, что конструкция языка была недостаточно ясна для компьютерной обработки. Следовательно стали определять семантику для таких речевых актов. Тогда и родилась идея построить грамматику речевых актов в виде формального исчисления. Но чем яснее становилась семантика, тем компактнее и лаконичнее общий язык. ACL не хватает универсальности естественных языков, которая позволяет мгновенно адаптироваться к новым ситуациям. Современной тенденцией развития ACL является разработка языков с широким значением и использованием протоколов (т.е. правил интерпретации) для корректного понимания значения предложений в контексте разговора.

Компьютерная семиотика

Семиотическое несоответствие может быть выявлено несколькими способами. Во-первых, это может быть несоответствие в «понимании» компьютера, когда он получает информацию от человека, назначившего (возможно неправильно) правила интерпретации для этого компьютера. Люди обычно составляют свои собственные правила для компьютера. Также возникает несоответствие, когда человек неверно интерпретирует получаемую от компьютера информацию. И в том и в другом случае, применение компьютерной семиотики помогло бы использовать знания о возможной интерпретации правил для понижения (если не устранения) возможности недопонимания.

Несоответствия могут быть в дальнейшем проанализированы с точки зрения ослабления или усиления того, что выражает компьютер. Обычно компьютеры запрограммированы на уменьшение погрешности и невозможно ухудшить выходные данные во время их интерпретации. Однако, человек ожидает, что компьютеры всегда показывают все, что они «знают». Это другое правило интерпретации, которое должно приниматься во внимание при конструировании систем старающихся адаптировать свои выходные данные под те, которые хочет видеть пользователь.

Термин «компьютерная семиотика» используется многими. Основная идея состоит в том, что компьютер (фактически любое компьютерное устройство) может быть использован для манипулирования значением выражений. Для некоторых авторов, это ограничивается манипулированием данными и отображением их понятным образом. Для других, это может отождествляться с искусственным интеллектом в целом. Для остальных, это имеет целью воспроизводить на компьютерах порождение новых значений символов, выявление новых признаков (семиозис).

Подход, рассматриваемый в этой статье, основывается на том, что компьютерная семиотика определяется «существованием алгоритма, данной формальной семиотической структурой и функцией отображения между ними» [1]. Исходя из этого, представляется возможным уменьшать семиотическое несоответствие между компьютером и человеком. Имея правила интерпретации, связанные с языками, подобно тому, как с языками связана семантика, можно обеспечить сохранение смысла, семантических значений в процессе логических выводов.

Идея алгебраической семиотики [2] была выдвинута Джозефом Гогеном в целях рассмотрения представлений (или спецификаций) в отношении с их использованием. В этом смысле представления называются знаковыми системами. Они включают синтаксис и семантику рассматриваемого языка и, насколько это возможно, информацию для интерпретации представления.

Семиотический морфизм трансформирует такую знаковую систему (источник) в другую (приемник)- предположительно более подходящую для конкретного использования. Алгебраическая семиотика изучает знаковые системы и морфизмы в рамках теории категорий.

Однако, существует нехватка общих способов формулирования правил интерпретации и объединения таких правил.

Заключение

Очевидно, что при взаимодействии человека с компьютером взаимопонимание не может быть обеспечено только лишь семантикой. Пользователи будут применять прагматические правила интерпретации, которые должны приниматься во внимание в процессе обработки знаков компьютером.

Литература

1. Gerd H. (1998) *A possible scientific framework for computational semiotic systems.* http://www.imm.de/kip/semiotics/cs-framework-19.11.98.html
2. Joseph Goguen. (1999) *An Introduction to Algebraic Semiotics, With Applications to User Interfase Design*, Lecture notes in computer science 1562:242-291.
3. Nicola Guarino. (1998) *Formal ontologies in information systems*, Amsterdam.
4. Yannis Labrou, others. (1999) *The current landscape of agent communications languages*. IEEE:Intelligent systems 14(2);45-52.
5. Peter Mosses. (1990) *Denotational semantics*. In Handbook of theoretical computer science, 575-631.
6. Bernhard Nebel. (1990) *Reasoning and revision in hybrid representation systems.* Springer Verlag, Heidelberg.
7. Alfred Tarski. (1956) *Logic, semantics, metamathematics*.Clarendon press, Oxford.

Клиентоориентированные информационные технологии в налоговом администрировании

Исмаилова Л.Ю., Журавлева О.О.

Кафедра ПКИ и ИТ, Институт ``ЮрИнфоР-МГУ",
Москва, 119435 РФ, М. Пироговская ул., д. 5, офис 37
ozhura@gmail.com, lyu.ismailova@gmail.com

Аннотация. Использование современных массмедийных технологий способно существенно повысить качество налогового администрирования и сократить издержки налогоплательщика на реализацию своих законных прав и исполнение обязанностей в сфере налогообложения. Однако широкое внедрение новых сервисов должно сопровождаться надлежащей правовой регламентацией, позволяющей гарантировать основные конституционные права и свободы граждан. При этом должен учитываться передовой опыт стран-членов ОЭСР.

Введение

Концепция электронного правительства в первую очередь связана с с новыми способами применения информационных технологий в сфере государственного управления, в том числе в налоговом администрировании[1]. Федеральная налоговая служба (ФНС) РФ последовательно осуществляет шаги по расширению возможностей электронного взаимодействия с налогоплательщиками и государственными и муниципальными органами власти и иными организациями за счет внедрения электронных услуг, в частности в форме клиентоориентированных сервисов.

В этой связи следует отметить: 1) развитие и улучшение электронных сервисов, связанных с регистрацией налогоплательщиков[2] и представлением налого-

[1] В настоящее время только 3 государства мира не имеют своих официальных сайтов в сети Интернет. Как отмечается в недавно опубликованном ООН обзоре «Электронное правительство Обзор 2012», в настоящее время 179 стран предоставляют возможность заполнения специальных формам различных документов на соответствующих порталах в сети интернет, связанных с финансами, большую часть из которых составляют налоговые документы. (P. 40).См.: United Nations E-Government Survey 2012. UN 2012.unpan1.un.org/intradoc/groups/public/.../un/unpan048065.pdf

[2] В Основных направлениях налоговой политики РФ на 2013 год и на плановый период 2014 и 2015 годов отмечается, что «трансформируются подходы к работе с налогоплательщиками - это клиентоориентированность, упрощение способов взаимодействия с налоговыми органами, создание более комфортных условий для исполнения налогопла-

вой отчетности в электронном виде; 2) создание специализированных информационных ресурсов для налогоплательщиков; 3) попытки развития программных средств мониторинга правоприменения, а 4) также развитие средств электронного консультирования. Использование новых информационных массмедийных технологий для российской налоговой правоприменительной практики не характерно. Это в первую очередь объясняется отсутствием регламентов их использования. В то время каканализ международной практики применения информационных технологий в сфере налогового администрирования свидетельствует об эффективности применения массмедийных информационных технологий в сфере налогового администрирования.

Подходы к информационному обеспечению

В целом при выработке эффективных подходов к информационному обеспечению налогового администрирования следует учитывать растущую сложность и мобильность бизнеса (организационно-правовых форм, схем ведения совместной деятельности, выполнения работ и оказания услуг, получения предоставлений и их оформления), физических лиц и их союзов (в том числе семейных) и вытекающие отсюда проблемы моделирования бизнес-процессов и их трансформации.

Факторы, обусловливающие сложный характер современного бизнеса подробно рассмотрены ранее[3]. Значительно усложняют процедуры обеспечения информационного сопровождения деятельности налоговых органов, налогоплательщиков и налоговых агентов, такие факторы как: (1) глобализация деятельности субъектов; (2) появление сложных товаров; (3) ведение деятельности посредством телекоммуникационных каналов связи (в том числе заключение сделок посредством сети Интернет); (4) развитые финансовые механизмы/инструменты, в том числе использующие телекоммуникационные сети; (5) возможности рассредоточения центров прибыли с потенциалом динамической активизации при смене требований налогового администрирования (6) развитие дистанционных форм выполнения и предоставления работ и услуг (в том числе по налоговому консультированию); и другие. Как следствие возникает требование большой адаптивности программ поддержки налогового администрирова-

---

тельщиками своих обязанностей». В рамках этой деятельности предусматривается «дальнейшее развитие доступной информационной поддержки, системы электронного документооборота и построение интерактивных сервисов, включая новый Интернет-сервис "Создай свой бизнес", являющийся пошаговой инструкцией по созданию и ведению бизнеса с точки зрения налогообложения, а также сервис "Личный кабинет налогоплательщика для юридического лица".». См.: Основные направления налоговой политики Российской Федерации на 2013 год и на плановый период 2014 и 2015 годов, утв. Правительством Российской Федерации// http://www.minfin.ru/ru/

[3] Журавлева О.О., Исмаилова Л. Ю. Новые информационные массмедийные технологии в налоговом администрировании. Мировые тренды и отечественная практика. // Финансовое право №__, 2012. С. __-__.

ния к изменяющимся условиям и развития стандартов обмена налоговой информации между налоговыми органами различных публичных субъектов.

Аналогичные факторы обусловливают и сложности, возникающие в связи с информационным обеспечением администрирования налогов, уплачиваемых физическими лицами, поскольку факты, лежащие в основании возникновения, изменения, прекращения обязанности по уплате налога, различным образом могут квалифицироваться и устанавливаться в зависимости от юрисдикционного[4], экономического и правового контекста. Кроме того, в разных правовых системах наблюдаются процессы трансформации даже базовых институтов семьи и иных союзов/объединений физических лиц с вытекающими из этого экономико-правовыми последствиями, что может кардинально менять саму концепцию клиентоориентированного информационного обеспечения.

Особые требования к разрабатываемым информационным продуктам (ИП), регламентам их использования и технологии сопровождения связаны и с тем, что большие изменения имеют место в определении налоговозначимых признаков и обстоятельств. Так, например, большие изменения претерпевают квалифицирующие признаки при определении статуса налогового резидента по отдельным видам налогов, приобретающие особое значение при перемещении специалистов или иных работников и членов их семей.

В законодательстве различных публичных образований по-своему определяются налоговые периоды, статусы лиц, в том числе статусы[5]специалиста, режим работы (стационарный, дистанционный, разъездной, надомный и др.), что

---

[4]Исторически первоначально налоговые системы развивались изолированно друг от друга, заключавшиеся впоследствии межгосударственные соглашения в налоговой сфере имели целью исключить двойное налогообложение, возникающее в силу различий национальных налоговых систем. Однако положения соглашений могут использоваться для избежания налогообложения в обоих договаривающихся государствах, более того может возникать право на получение налоговых кредитов. В силу того, что подобные ситуации перестали быть экзотикой налоговые администрации различных государств ищут общие подходы к решению данной проблемы. Так, 8-10 мая 2012 года в Монреале состоялась встреча глав налоговых администраций стран участников ОЭСР темой которой была борьба с непреднамеренным двойным избежанием налогообложения. См.: Fighting unintended double non-taxation. 11/05/2012. http://www.oecd.org/document/54/0,3746,en_2649_37427_50276086_1_1_1_37427,00.html

[5] Например, в РФ статус высококвалифицированного специалиста, определенный в соответствии с Федеральным законом от 25 июля 2002 года № 115-ФЗ «О правовом положении иностранных граждан в Российской Федерации» порождает особый режим налогообложения налогом на доходы физических лиц доходов, полученных от осуществления трудовой деятельности в таком качестве. Так названные доходы подлежат налогообложению по ставке 13% независимо от факта наличия/отсутствия налогового резидентства физического лица.(п.3 статьи 224 НК РФ). См.: Налоговый кодекс Российской Федерации (часть первая) от 31.07.1998, ред. от ред. от 30.03.2012, Налоговый кодекс Российской Федерации (часть вторая) от 05.08.2000, в ред. от 03.05.2012// СЗ РФ. 1998. N 31. Ст. 3824, СЗ РФ. 2002. N 32. Ст. 3340 (далее по тексту - НК РФ). Аналогичного рода режимы существуют и в других государствах.

также усложняет разработку и сопровождение ИП, ориентированных на поддержку электронного взаимодействия органов, на которые возложено администрирование налогов. Различные критерии могут использоваться и при квалификации доходов как: (1) регулярных, (2) получаемых на постоянной основе и (3) разовых.

Различные налоговые последствия могут устанавливаться при квалификации (1) расходов, возникающих в силу выполнения работником различных видов работ (принимаемых для целей налогообложения в качестве уменьшающих налогооблагаемую базу по налогу на прибыль/ доход или нет), а также (2) расходов, возникающих в связи с определенным режимом выполнения работ (сезонные работы, командировка, вахтовый метод и т.д.). Например, возможна не только дифференциация видов расходов, которые относятся к уменьшающим налогооблагаемую базу по конкретному налогу, но и применение различных подходов к определению норм их учета.

Однако, при наличие фактов лавинообразного изменения экономических условий и моделей налогообложения и администрирования, влекущих усложнение взаимного влияния налоговых систем и механизмов взаимодействия налоговых администраций различных стран, требуется конструктивная основа для внедрения современных информационных подходов, ориентированных на выравнивании возможностей налогоплательщиков. Так для выработки возможных реакций в сфере налогового администрирования следует шире использовать современные информационные подходы и средства электронного взаимодействия, как с налогоплательщиками, так и государственными и муниципальными органами власти и иными организациями. Последнее представляется наиболее эффективным в сферах мультиналогового и кросс-территориального налогового менеджмента. Не последнюю роль играет возможность реализации сервисов для информационного обеспечения функциональной экспертизы.

Исходя их того факта, что при современной мобильности субъектов хозяйственной деятельности налоги и способы их администрирования зачастую играют роль решающих факторов при перемещении центров прибыли, только «глобальная» прозрачность отчетов и совершенствование форм взаимодействия налоговых администраций могут способствовать обеспечению баланса интересов. Последние, как отмечалось ранее, могут быть достигнуты только за счет применения новых информационных технологий.

В числе общих тенденций в изменении налогового администрирования следует назвать: (1) внедрение электронных форм регистрации: субъектов экономической деятельности и их органов управления; юридических лиц, лиц, занимающихся определенными видами деятельности (электронный уведомительный порядок); налогоплательщиков по определенным видам налогов; (2) внедрение электронных форм налогового декларирования; (3) внедрение процедур электронного взаимодействия между различными государственными органами и реестрами (обеспечение перекрестного контроля); (4) внедрение электронных форм обжалования действия/ бездействия налоговых органов и должностных лиц; (5) расширение взаимодействия с налогоплательщиками, в том числе по-

средством электронного оповещения налогоплательщиков[6] и проведения опросов относительно работы налоговых органов (в том числе с использованием электронных массмедийных форм); (6) внедрение электронных форм консультирования налогоплательщиков.

В таблицах 1 и 2 приведены на основе данных ОЭСР[7] сведения, которые иллюстрируют степень использования социальных массмедийных информационных технологий для внешних коммуникаций в налоговой сфере зарубежными государствами. Анализ этих данных позволяет условно разделить государства на три группы: 1) страны, налоговые администрации которых активно используют весь инструментарий современных массмедийных средств (Австралия, Дания, Португалия, Сингапур, США); 2) страны, налоговые администрации которых только начинают их осваивать (Австрия, Германия, Испания, Ирландия, Канада, Мексика, Новая Зеландия, Норвегия, Соединенное Королевство, Финляндия, Эстония, Япония);3) страны, в которых массмедийные технологии налоговыми органами пока не используются (Бельгия, Италия, Китай, Чили, Франция, Швеция, Швейцария, ЮАР).

Таблица 1. Использование социальных массмедийных технологий для внешних связей (коммуникаций) по состоянию на январь 2011 г.[8]

Страна	Использование социальных массмедийных технологий		
	Существование формальной стратегии	Использование для внешнего взаимодействия с налогоплательщиками	Использование для внутренних коммуникаций (связи)
Австралия	V	V	V
Австрия	X	V	V
Бельгия	X	X	X
Канада	X	V	V
Чили	X	X	X
Китай	X	X	X
Дания	V	V	V
Эстония	X	V	V
Финляндия	X	V	X
Франция	X	X	X
Германия	X	V	X
Ирландия	Разрабатывается	V	V
Италия	X	X	X
Япония	X	V	X

[6]Например, электронная почта, SMS-информирование могут требовать изменения форматов государственных реестров или ведения отдельных баз данных.

[7]ОЭСР (OECD) –- организация экономического сотрудничества и развития (Organisation for Economic Co-operation and Development)

[8]Forum on Tax Administration: Taxpayer Services Sub-Group Information note Social Media Technologies and Tax Administration. October.2011. P.15. // http://www.oecd.org/dataoecd/50/5/48870427.pdf.

Мексика	X	V	V
Новая Зеландия	Разрабатывается	V	V
Норвегия	X	V	X
Португалия	V	V	V
Сингапур	V	V	V
ЮАР	X	X	X
Испания	X	V	V
Швеция	X	X	X
Швейцария	X	X	X
Соединенное Королевство Великобритании	Разрабатывается	V	V
США	V	V	V

Условные обозначения:

«X» – соответствующий тип социальной массмедийной технологии не применяется.

«V» - соответствующий тип социальной массмедийной технологии применяется.

Таблица 2. Использование социальных массмедийных технологий для внешних связей (коммуникаций) по состоянию на январь 2011 года [продолжение][9]

Страна	Типы медийных средств/ технологий					
	Твиттер	Фейсбук	ЮТюб	РСС	Вебинары (Веб-семинары)	Иное
Австралия	V	V	V(эксп-т)	V	V	V/1
Австрия	X	V	V			
Бельгия	-	-	-	-	-	-
Канада	V	X	V	V		V/1
Чили	X	X	X	V		
Китай[10]	-	-	-	-	-	-
Дания	V	V	X	X	X	X
Эстония	V	X	V	X	X	V/1
Финляндия	X	X	X			V/1
Франция	X	X	X	X	X	X

[9]Forum on Tax Administration: Taxpayer Services Sub-Group Information note Social Media Technologies and Tax Administration. October.2011// http://www.oecd.org/dataoecd/50/5/48870427.pdf. P.15.

[10]В настоящее время в провинции Нанкин, КНР, налоговые службы внедряют различные сервисы для налогоплательщиков, в том числе, специально разработанные для мобильных телефонов. См.: 南京地税十大服务举措助推科技创新2012-04-12 // http://life.jschina.com.cn/system/2012/04/12/013124579.shtml

Германия	X	X	V	X	X	X
Ирландия	X	X	V	X	X	V/1
Италия	-	-	-	-	-	-
Япония	V	X	V	X	V	X/1
Мексика	V	X	V	V	X	V/1
Новая Зеландия	V	V	X	X	X	X
Норвегия	V	X	V	X	X	X
Португалия	V	X	V	X	X	X
Сингапур	V	X	X	X	X	V/1
Южная Африка	X	X	X	X	X	X
Испания	V	X	V	V		X
Швеция	X	X	X	X	X	X
Швейцария	X	X	X	X	X	X
Соединенное Королевство Великобритании	V	V	V	X	X	V/1
США	V	V	V	X	X	V/1

Условные обозначения:

«X» – соответствующий тип медийной технологии, средства не применяется.

«V» - соответствующий тип медийной технологии, средства применяется.

«/1» по странам соответственно:

Австралия – включая подкасты, GovDex (инициатива Правительства Австралии) и форум среднего, малого бизнеса и самозанятых налогоплательщиков; *Ирландия* – в отчете упоминаются и форумы для ограниченного внешнего пользования; *Канада* – аудио- и видеокасты; *Мексика* применяет Flickr и блоги с контролируемым путем доступа (для специфических целей). Использование РСС не рассматривалось при составлении национального отчета как работа в Интернет; *Сингапур* запустил специальное приложение для айфоновIRAS_SG, которое за первые два месяца было загружено более чем 25000 пользователями. Приложение позволяет пользователям контролировать статус исполнения налоговой декларации по налогу на доходы физических лиц, рассчитать указанный налог, а также налог на имущество. Приложение имеет ссылку на раздел налоговой службы Сингапура в Твиттере; *Соединенное Королевство Великобритании*– применяет LinkedIn (используется 500 пользователями и дополнительно персоналом), форумы внешних сообществ (сетевых) и интранет гражданских служб для отправки сообщений и wikis; *США*использует следующие социальные сети: (DIGG (HR), SecondLife (HR), Quora); *Финляндия*– в отчете упоминается использование LinkedIn и экспериментальных проектов (Suomi - большое сетевое сообщество) и Aamulehti (газетный блог); *Эстония* – в отчете данного государства описано использование форумов; *Япония* – Национальная налоговая администрация Японии создает много видео-программ для своего сайта[11],

[11]Опыт использования видеороликов в сети Интернет имеет и ФНС России См.: Налоговики Архангельской области используют новые способы информирования налогоплательщиков. Дата публикации: 21.05.2012. http://www.nalog.ru/rub_mns_news/3929789/. (дата доступа 23.05.2012).

которые разъясняют налоговые процедуры при помощи видео и диаграмм. Видео также загружается с Youtube (http://www.nta.go.jp/webtaxtv/all_list.html).

Как отмечается ОЭСР, использование современных массмедийных технологий способно существенно повысить качество налогового администрирования и сократить издержки налогоплательщика на реализацию своих законных прав и исполнение обязанностей в сфере налогообложения.

Действующий административный регламент ФНС по исполнению государственной функции по бесплатному информированию (в том числе в письменной форме) налогоплательщиков, плательщиков сборов и налоговых агентов о действующих налогах и сборах, о законодательстве о налогах и сборах и принятых в соответствии с ним нормативных правовых актах, порядке исчисления и уплаты налогов и сборов, правах и обязанностях налогоплательщиков, плательщиков сборов и налоговых агентов, полномочиях налоговых органов и их должностных лиц, а также представлению форм налоговых деклараций (расчетов) и разъяснению порядка их заполнения, утвержденный приказом Министерства финансов РФ от 18.01.2008 № 9н[12] (далее по тексту Административный регламент), не предусматривает возможностей осуществления функции информирования налогоплательщиков посредством средств мобильной связи (SMS- и MMS-сообщений), размещения информации в специальных социальных сетях и иных возможностей информирования налогоплательщиков, налоговых агентов и их представителей посредством использования современных информационно-телекоммуникационных и массмедийных технологий. Однако, в некоторых инспекциях в субъектах РФ реализуются экспериментальные проекты.

Так, по сообщению ФНС, кировские налоговики активно используют SMS-информирование налогоплательщиков[13]. Планируется использовать оставленные налогоплательщиками телефонные номера после предоставления информации о задолженности налогоплательщиков для рассылки информационных SMS-сообщений. Кроме того, собственники движимого и недвижимого имущества, налогообложение которого осуществляется в данном регионе, получат SMS-напоминания о сроках уплаты земельного и транспортного налога[14]. Однако территориальное подразделение ФНС не учитывает проблем идентификации отправителя и удостоверения его полномочий на отправку соответствующего сообщения. В настоящее время, невозможно доказать факт доставки со-

---

[12]Приказ Минфина РФ от 18.01.2008 № 9н «Об утверждении Административного регламента Федеральной налоговой службы по исполнению государственной функции по бесплатному информированию (в том числе в письменной форме) налогоплательщиков, плательщиков сборов и налоговых агентов о действующих налогах и сборах, законодательстве о налогах и сборах и принятых в соответствии с ним нормативных правовых актах, порядке исчисления и уплаты налогов и сборов, правах иобязанностях налогоплательщиков, плательщиков сборов и налоговых агентов, полномочиях налоговых органов и их должностных лиц, а также представлению форм налоговых деклараций (расчетов) и разъяснению порядка их заполнения», с изм. от 14.07.2011 / /РГ. 13.05.2008. № 100.

[13] См.: http//www.nalog.ru.

[14] См.: http//www.nalog.ru.

общения надлежащему в юридическом смысле адресату, то есть данный сервис является факультативным и не может рассматриваться в качестве средства доставки уведомления и требования об уплате соответствующих налогов.

Кроме того, если указанная информация без получения предварительного согласия налогоплательщика будет направляться на такие номера, то могут быть нарушены положения ст. 103 НК РФ о налоговой тайне, а также положения статьи 23 Конституции РФ[15]. Такие сведения могут стать не только достоянием членов семьи налогоплательщика, в том числе несовершеннолетних, но и третьих лиц. Как следствие раскрытие информации ненадлежащему субъекту может создавать предпосылки для злоупотреблений и преступлений.

Опыт другого региона также свидетельствует о том, что налоговые органы пытаются создать специализированные массмедийные механизмы. Так, управление ФНС России по Забайкальскому краю[16] предлагает налогоплательщикам в числе прочих такие виды предоставляемых услуг: SMS-информирование о задолженности и виртуальный налоговый помощник.В рамках первой услуги предоставляется бесплатноеSMS-информирование о задолженности по имущественным налогам (транспортный налог, налог на имущество физических лиц и земельный налог), а также осуществляется отправка SMS-сообщений по иным вопросам. Круг их в настоящее время исчерпывающим образом не определен. Существенным является тот факт, что подключение сервиса осуществляется только на основании лично представленной налогоплательщиком в налоговую инспекцию заявки. В предлагаемом на сайте образце документа не определены сроки подключения сервиса, в силу чего подобная заявка носит бессрочный характер. Также не определены порядок и основания отказа от данной услуги, не определены последствия для случаев смены пользователя телефонного номера. Таким образом, отсутствует надлежащая нормативно-правовой базы предоставления данной услуги, что не позволяет гарантировать соблюдение прав и законных интересов налогоплательщиков и сокращает эффект от внедрения ИП.

Рассматриваемый виртуальный помощник (ВП)[17] еще не очень развит[18]. Например, выдав по соответствующему запросу определение налога на землю, на вопрос: «сроки уплаты» ВП уже не дает ответа и предлагает обратиться к сервису позже (обращение спустя две недели также не дало результата).Правовое регулирование предоставления названных услуг в форме специализированного административного регламента не осуществляется. Необходимо

---

[15]Данная статья предусматривает право на неприкосновенность частной жизни, личную и семейную тайну. См.: Конституция Российской Федерации принята всенародным голосованием 12.12.1993, с учетом поправок, внесенных Законами РФ о поправках к Конституции РФ от 30.12.2008 № 6-ФКЗ, от 30.12.2008 № 7-ФКЗ// СЗ РФ. 2009. № 4. Ст. 445. Далее по тексту Конституция РФ.

[16] См.: http://www.zabnalog.ru/

[17] См.: http://www.zabnalog.ru/vnp

[18]При этом по данным статистики, предоставляемой самим сервисом, с начала функционирования по май 2012 года зафиксировано более 450 тысяч обращений к данному ресурсу.

отметить, что даже, если ИП внедряется в порядке правового эксперимента, то его надлежащее правовое обеспечение должно опираться на положения Конституции РФ и гарантировать соблюдение прав и налогоплательщика (по крайней мере предупреждать о возможных неблагоприятных последствиях). Ни использование электронных сервисов, ни использование информационных массмедийных технологий не должно лишать налогоплательщика права на правовую определенность и осведомленность о правовых последствиях совершаемых им юридически-значимых действий или их отсутствии. Отсутствие надлежащей правовой регламентации способно дискредитировать любую информационную новацию.

В целом рассмотренные сервисы и практика их внедрения показывают, что, не смотря на имеющийся значительный интерес налогоплательщиков, сфера их налогового информирования весьма ограничена и нормативно не обеспечена. Кроме того, такого рода сервисы пока не ориентированы на разрешение большинства из перечисленных в настоящей статье проблем и требуют дальнейшего развития, в том числе с учетом положительных практик стран-членов ОЭСР. Имеющие место: 1) ориентированность сервисов данного класса (а) на ограниченный круг налогоплательщиков и (б) ограниченный перечень налогов, а также 2) их региональная обособленность предопределяют ограничения сферы их применения.

К числу основных проблем применения сервисов данного класса для решения задачи повышения собираемости налогов следует отнести отсутствие комплексного подхода к их развитию и сопровождению и отсутствие ориентированности на определенный возможный уровень мобильности налогоплательщиков. Появление даже «пилотных», «экспериментальных» налоговых сервисов должно опираться на надлежащую нормативно-правовую основу и стандарты представления налоговозначимой информации.

Литература

1. Forum on Tax Administration: Taxpayer Services Sub-Group Information note Social Media Technologies and Tax Administration. October. 2011. 51P. http://www.oecd.org/dataoecd/50/5/48870427.pdf
2. United Nations E-Government Survey 2012. UN 2012. unpan1.un.org/intradoc/groups/public/.../un/unpan048065.pdf
3. Журавлева О.О., Исмаилова Л. Ю. *Новые информационные массмедийные технологии в налоговом администрировании. Мировые тренды и отечественная практика.* // Финансовое право № 9, 2012. С. 21-27
4. 南京地税十大服务举措助推科技创新2012-04-12 // http://life.jschina.com.cn /system/ 2012/04/12/013124579.shtml

Правовое регулирование налогообложения инновационного сектора КНР: сравнительно-правовой анализ

Журавлева О.О., Севальнев В.В.

Кафедра ПКИ и ИТ, Институт “ЮрИнфоР-МГУ”,
Москва, 119435 РФ, М. Пироговская ул., д. 5, офис 37
ozhura@gmail.com

Аннотация. Исследование правового регулирования налогообложения инновационного сектора КНР представляет интерес для большинства современных государств, поскольку позволяет выработать национальную стратегию поддержки инновационного развития. При осуществлении исследований важно не только описать применяемые льготы, но и учесть практику их применения субъектами инновационной деятельности с учетом особенностей правовой и налоговой систем, а также правовой культуры. Это позволит не просто копировать существующие образцы, но создавать реальные конкурентные преимущества в национальной правовой налоговой системе.

В кризисные периоды развития мировой экономики наряду с предложениями осуществлять финансирование стратегических направлений будущего роста звучат предложения по сокращению государственной поддержки, в том числе и в данной сфере, основанные на вере во всесильность рынка, его способность самостоятельно придать импульс развитию инновационного бизнеса без участия государства. Исходя из этого, на данном этапе важно выбрать правильный баланс между публичной и частной поддержкой и финансированием инновационных проектов с учетом зарубежного опыта.

Построение современного, гармоничного, креативного и высокодоходного общества заявлено в качестве основного приоритета развития КНР до 2030 года[1]. В связи исследованием, проведенным Всемирным банком совместно с КНР, сделан вывод о необходимости ускорения инновационного пути развития и создании открытой инновационной системы, в которой бы честная конкуренция побуждала китайские фирмы осуществлять продуктовые инновации и инновации процессов не только в результате проведения ими самостоятельных научных исследований и осуществления разработок, но и в результате участия в мировом научном сообществе, осуществляющем такие проекты.

[1] Исследование проводилось совместно Всемирным банком с Министерством финансов КНР, Центром исследования развития Госсовета КНР. См.: China 2030. Building a Modern, Harmonious and Creative High-Income Society. World Bank. Development Research Center of State Council, the People’s Republic of China. Conference edition. Word Bank. 2012. P. XV.

Исследование правового регулирования налогообложения инновационного сектора КНР представляет интерес для большинства современных государств, в том числе и таких достаточно успешных в сфере инновационного развития как Великобритания и США, поскольку позволяет выработать национальную стратегию поддержки инновационного развития. В процессе его проведения были выявлены источники, субъекты и уровни правового регулирования налогообложения инновационного сектора КНР и Российской Федерации, определены: роль государства в осуществлении поддержки не только государственных, но и частных субъектов, а также применяемые налогово-правовые формы. При этом были учтены границы распределения полномочий между субъектами (органами центральной и местной государственной власти КНР), осуществляющими правовое регулирование налогообложения указанной сферы. Несмотря на определенные различия в ориентации и уровне развития экономики, в политике большинства современных государств выделяются общие приоритетные направления, без которых будущее экономическое развитие не возможно. Знание правовых принципов, которые лежат в основе налогово-правовой поддержки инновационного сектора КНР, сопоставление его с применяемыми и предлагаемыми подходами в России, позволило оценить возможности имплементации зарубежных правовых институтов, к каковым можно отнести и налоговые льготы. Была исследована правовая природа налоговых льгот, направленных на поддержку инновационного сектора КНР, выявлены этапы в правовом регулировании налогообложения инновационного сектора КНР и России. Сформулированы общие доктринальные принципы и некоторые специальные начала правового регулирования налогообложения инновационного сектора КНР, которые являются своего рода импульсом, позволяющим экономической модели принести реальные плоды, и получить преимущества в конкуренции на международных рынках научно-технической продукции, в том числе в информационной сфере. Кроме того, обоснована классификация налоговых льгот, направленных на развитие инновационного сектора.

Из анализа формы предоставления налоговые льготы инновационному сектору КНР можно выделить такие формы как: пониженная ставка налогообложения; налоговые освобождения – налоговые каникулы; ускоренная амортизация основных фондов и специального оборудования для проведения научно-исследовательских, опытно-конструкторских и технологических работ[2]. Также вызывает интерес опыт КНР в части определения адресатов льгот и направлений поддержки, в том числе, в части поддержки инновационной деятельности, направленной на преобразование традиционных отраслей экономики[3].

Было выявлено, что местные правительства в КНР обладают достаточно широкими полномочиями, которые позволяют регионам не только конкурировать

---

[2]Севальнев В.В. Правовое регулирование налогообложения инновационного сектора КНР: дисс. на соиск. уч. степени канд. юрид. наук. М., 2012. С.14.

[3] Следует отметить, что для части российских регионов – это достаточно серьезная проблема, не нашедшая пока адекватного решения (традиционно решается путем предоставления государственных субсидий).

между собой за научные умы, ресурсы и технологии, но и успешно развиваться. В то же время, в ряде случаев имеет место фактически «ручное регулирование».

Поскольку в различных странах в настоящее время широко применяются в налоговой сфере так называемые «пилотные проекты», «эксперименты», то был проанализирован, в том числе с точки зрения возможного будущего применения опыт КНР (экспериментальное правовое регулирование).

Оценка информационной надежности адаптивного аппаратного обеспечения, использующего RAM на основе элементов памяти с фазовыми переходами

Егармин К.Н.[1], Воронков Э.Н.[2]

[1]Кафедра полупроводниковой электроники, НИУ МЭИ,
Москва, 111250 РФ, Красноказарменная ул., д. 14
egarmin@bk.ru,
WWW: http://www.mpei.ru

[2]Институт информационных и инновационных технологий, МАБиУ,
Москва, 129594 РФ, 5-й проезд Марьиной Рощи, д. 15-а
voronkoven@mail.ru,
WWW: http://mabiu.ru

Аннотация. Одной из основных задач, стоящих перед разработчиками программного обеспечения является использование адаптивного оборудования, которое конфигурировалось бы под решаемую задачу. Действительно, любой алгоритм, насколько бы эффективным он ни был, потеряет свои достоинства, если будет выполняться в последовательной системе обработки данных. Поэтому повышение эффективности решения задач, при сохранении традиционных программных подходов, переходит в плоскость поиска аппаратного обеспечения (АО) с динамическим конфигурированием аппаратуры. В настоящее время выделяется две концепции подобного АО: программируемые логические интегральные схемы (ПЛИС) на основе технологии Flash и на основе технологии PcRAM (англ., Phase change Random Access Memory). Каждый из этих подходов имеет свои достоинства и недостатки. В данной работе рассматривается информационная надежность PcRAM на основе моделей, впервые предложенной А.Н. Колмогоровым.

Введение

В особый класс адаптивных компьютерных систем попадает адаптивное аппаратное обеспечение. К такому классу систем относятся ПЛИС, которые позволяют производить конфигурирование «на лету», что необходимо для построения обучаемой нейронной сети и для быстрого решения ряда узкоспециализированных задач. Организация ПЛИС на основе Flash обладает слишком малым быстродействием, для того чтобы хотя бы в отдаленном будущем можно было говорить о скоростях, соизмеримых со скоростью конфигурирования нейронной сети живого мозга. Однако в настоящее время на смену технологии Flash, приходит технология PcRAM, значительно превосходящая Flash по многим параметрам, в частности по быстродействию. Однако кроме экономических

причин, сдерживающих появление на рынке этой технологии, существует и проблема оценки информационной надежности, которая для PcRAM еще не окончательно разрешена.

Ключевая проблематика

К оценке информационной надежности PcRAM применимы разные подходы. К ним относятся численные и аналитические методы. Применение численных методов всегда несет в себе трудности выделения ключевых параметров влияющих на конечный результат. Этого нельзя сказать об аналитических методах, однако последние – сложнее в построении. В этой связи построение адекватных аналитических моделей становится ключевой задачей.

Развитие технологии

PcRAM – принадлежат к новому поколению микросхем энергонезависимой памяти, запись информации в ячейках которой происходит за счет изменения фазового состояния активной области, сопровождающегося изменением ее проводимости от близкой к металлической (состояние "set" - ключ замкнут), до близкой к проводимости диэлектрика (состояние "reset" - ключ разомкнут). В отличие от микросхем "Flash" памяти, PcRAM обладают возможностью побитовой выборки, более высоким быстродействием и ресурсом, что позволяет их успешно использовать в схемах динамического конфигурирования компьютера, адаптируя его под конкретные, решаемые в данный момент задачи. Однако так же как и для Flash памяти для PcRAM одним из основных моментов становится информационная надежность, обеспечивающая сохранность записанных данных.

В настоящей работе основываясь на экспериментах, выполненных на тестовых образцах, литературных данных, а так же расчетах, даны оценки информационной надежности ячеек памяти PcRAM.

Из анализа физических процессов, приводящих к потере информации следует, что наиболее вероятной является потеря информации состоянием, обладающим большей энтропией и большей энергией ("reset").

Показано, что основным параметром, влияющим на надежность хранения информации, при заданных электрофизических параметрах активной области, , является температура в процессе эксплуатации микросхемы. На основе подхода А. Н. Колмогорова [1] была разработана модель информационной надежности и получены соответствующие аналитические формулы для статистического распределения отказов. Приведены зависимости вероятности информационно отказа при различных температурах окружающей среды. Было показано, что вероятность отказа повышается по мере увеличения суммарного времени прибивания микросхемы при повышенной температуры. На основе, рассмотренной в работе модели, выполнен расчет распределения Вейбулла. Установлена связь характеристических параметров статистического распределения с термодина-

мическими параметрами активной области ячейки памяти PcRAM. Результаты компьютерного статистического расчета удовлетворяют экспериментальным данным [2-4]. Показано, что вероятность отказа изменяется с размером ячеек PcRAM. Это означает, что при оценке перспектив этих микросхем, следует учитывать не только ограничения вызванные снижением технологического размера (закон Мура), но так же и изменение надежности микросхем этого типа.

Заключение

В работе разработана модель информационной надежности микросхем PcRAM, предназначенных для применения в устройствах динамического конфигурирования компьютера. На основе разработанной модели отказа при хранении информации, сделан вывод о том, что информационная надежность PcRAM и сильно зависит от времени пребывания микросхемы при повышенной температуре, технологической нормы и свойств материала, использованного в качестве активной среды запоминающих ячеек.

Литература

1. Колмогоров А. Н. *Теория вероятностей и математическая статистика*: [Сб. статей].– М.: Наука, 1986.– с. 178 – 182
2. U. Russo, D. Ielmini, A. L. Lacaita. *A physics-based crystallization model for retention in phase-change memories*. IEEE 07CH37867 45th Annual International Reliability Physics Symposium, Phoenix, 2007, P. 547 – 553
3. A. Redaelli, A. Pirovano, I. Tortorelli, D. Ielmini, A. L. Lacaita. *A Reliable Technique for Experimental Evaluation of Crystallization Activation Energy in PCMs*. IEEE Electron Device Letters, Vol. 29, № 1, January 2008, P. 41 – 43
4. Ugo Russo, Daniele Ielmini, Andrea L. Lacaita. *Analytical Modeling of Chalcogenide Crystallization for PCM Data-Retention Extrapolation*. IEEE transactions on electron devices, vol. 54, No. 10, October 2007, p. 2769 - 2777

Гибридные информационные технологии для анализа и классификации сложноструктурируемых биомедицинских изображений

Томакова Р.А.

канд. техн. наук, доцент Юго-Западный государственный университет, г. Курск
Tomakova@rambler.ru

Брежнева А.Н.

Международная Академия Бизнеса и Управления,
институт информационных и инновационных технологий, г. Москва
anbr1@rambler.ru

Аннотация. Проблема, связанная с автоматической классификацией объектов по их изображениям, весьма характерна для анализа биомедицинских изображений. Эту задачу решают посредством обучаемых интеллектуальных систем, для построения которых наиболее часто используют нейросетевые технологии. Многие изображения не имеют четкого морфологического описания сегментов и относятся к классу сложноструктурируемых изображений. При этом сложности структурирования таких изображений обусловлены как природой самого изображения, так и зашумленностью хорошо структурированного изображения. Поэтому для выбора эффективных методов сегментации и классификации требуется использование гибридных информационных технологий.

Введение

Проблема, связанная с автоматической классификацией объектов по их изображениям, весьма характерна для анализа биомедицинских изображений. Эту задачу решают посредством обучаемых интеллектуальных систем, для построения которых наиболее часто используют нейросетевые технологии.

Многие изображения не имеют четкого морфологического описания сегментов и относятся к классу сложноструктурируемых изображений. При этом сложности структурирования таких изображений обусловлены как природой самого изображения, так и зашумленностью хорошо структурированного изображения. Поэтому для выбора эффективных методов сегментации и классификации требуется использование гибридных информационных технологий.

Для реализации гибридных технологий сегментации и классификации сложноструктурируемых объектов предложено использовать нейросетевые и нечеткие технологии [2]. Наиболее типичным примером сегментации сложностру-

турируемых изображений является задача определения межклеточных соотношений в мазках периферической крови.

Ключевая проблематика

Большинство способов сегментации включает последовательную обработку изображения градиентным и пороговым операторами [1]. В качестве градиентного оператора используют два анизотропных градиентных оператора, в результате чего получают векторное изображение, каждый пиксель которого характеризуется двумя параметрами, определяемыми как

$$G1_x = \partial G / \partial x \quad , \tag{1}$$

$$G1_y = \partial G / \partial y \quad , \tag{2}$$

где G- исходное изображение, G1-изображение, полученное в результате обработки изображения G градиентным оператором, x и y - направления, характеризующие анизотропию градиентного оператора.

Для сегментации сложноструктурируемых изображений разработан метод, реализующий три последовательных этапа: на первом этапе получают векторизованное изображение согласно операторам (1) и (2); на втором этапе – препарированное векторизованное изображение G2; на третьем этапе выполняют переход к бинарному изображению, отражающему контуры границ сегментов исходного изображения, осуществляемый посредством интеллектуального морфологического оператора, построенного на основе нечеткой нейросетевой модели.

Для преобразования изображения G1 в изображение G3 предложено использовать пятислойную модель нечеткой сети, которая реализует интеллектуальный оператор выделением границ сегмента [3].

За основу формирования слоя взята девятиэлементная маска с восьмью направлениями θ $(\theta = \overline{0,7})$. Для выбора пикселя границы сегмента альтернативных решений сформированы три решающих правила. Эти правила реализуются тремя слоями сети, четвертый слой агрегирует решающие правила для каждой из восьми альтернатив, а пятый слой, включающий всего один узел, активирует только один пиксель из восьми, для которого функция активации Z, полученная в результате агрегации трех решающих правил, имеет наибольшее значение.

Первое решающее правило основано на том, что движение границы сегмента должно осуществляться в направлении, перпендикулярном направлению градиента яркости.

Второе решающее правило основано на том, что наибольшую уверенность активации имеет тот пиксель, градиент которого Ψ_i наиболее близок к градиенту текущего активизированного пикселя Ψ_9, то есть необходимо проверить условие:

$$\left|\psi_9 - \psi_i\right| = \min, \quad i = \overline{1,8}\,. \tag{3}$$

Третье решающее правило основано на том положении, что яркость I_i активируемого пикселя должна быть близка к яркости предшествующего пикселя границы сегмента

$$\left|I_9 - I_i\right| = \min, \quad i = \overline{1,8}. \tag{4}$$

В третьем слое осуществляется формирование нечетких решающих правил. Каждое из трех решающих правил выполняет нелинейное преобразование величин, вычисленных в узлах второго слоя, в соответствующее число, лежащее в диапазоне 0 … v. Величина v определяет вес решающего правила, задаваемый эмпирически, с последующим его уточнением и принимает значения $\nu \in [0, 1]$.

В качестве нелинейных преобразований используем нелинейные преобразования кусочно-линейного типа или гауссианы.

Таким образом, имеем три решающих правила:

$$S1 = f_1(u1_i) \ , \tag{5}$$

$$S2 = f_2(u2_i) \ , \tag{6}$$

$$S3 = f_3(u3_i) \ , \tag{7}$$

где $u1_i, u2_i, u3_i$ - выходы узлов второго слоя, $i = \overline{1,8}$.

Нелинейное преобразование $S1$ описывается следующим уравнением

$$S1 = \begin{cases} 0 \ , & при \quad u1 > -\pi/4; \\ (4 \cdot v1/\pi) \cdot u1 + v1 \ , & при \quad -\pi/4 \le u1 < 0; \\ -(4 \cdot v1/\pi) \cdot u1 + v1 \ , & при \quad 0 \le u1 \le \pi/4; \\ 0 \ , & при \quad u1 > \pi/4. \end{cases} \tag{8}$$

Нелинейное преобразование $S2$ позволяет оценить уверенность в выборе пикселя границы сегмента на основе сравнения градиента в текущем и выбираемом пикселе. Аппроксимация функции $S2$ аналогична (8).

Нелинейное преобразование $S3$ позволяет оценить уверенность в выборе границы пикселя на основе того, насколько его яркость близка к яркости текущего пикселя. Нелинейная зависимость $S3$ имеет кусочно-линейную аппроксимацию, аналогичную $S1$ и $S2$.

В четвертом слое сети осуществляется агрегация решающих правил $S1, S2, S3$ для каждого пикселя 1…8 маски. В пятом слое выбирается один из восьми пикселей, агрегирующая функция Z для которого имеет максимальное значение. Агрегатор определяется как

$$Z = S1 + S2 + S3 \ . \tag{9}$$

В качестве детектора конца границы сегмента используем следующее решающее правило:

Если { (последующая координата пикселя границы вышла за кадр изображения) или (последующая координата пикселя границы сегмента уже выбрана) }, то «конец границы сегмента», иначе « текущий пиксель принадлежит границе сегмента».

Разработанное специализированное программное обеспечение предназначено для автоматизированного анализа микроскопических фотографий сложноструктурируемых изображений и реализовано в среде Matlab 7.10. Оно отличается функциональной последовательностью включения каждого модуля в технологический процесс гематологического анализа и гибким управлением процессом интерактивного взаимодействия программного обеспечения, предназначенного для обучения решающих модулей, с лицом, принимающим решение.

Работа выполнена в рамках реализации Федеральной целевой программы «Научные и научно-педагогические кадры инновационной России» на 2009 – 2013 годы (контракт № 424).

Литература

1. Гонсалес Р., Вудс Р., Эддинс С. Цифровая обработка изображений в среде MATLAB. М.: Техносфера, 2006. 616 с.
2. Филист С.А. Анализ гистологических изображений посредством морфологических операторов, синтезированных на основе преобразования Фурье и нейросетевого моделирования/С.А. Филист, Томакова Р.А., Горбатенко С.А., Швецова Н.А.// Биотехносфера, №3,(10), 2010. С. 54-60.
3. Томакова Р.А., Филист С.А. Метод обработки и анализа сложноструктурируемых изображений на основе встроенных функций среды MATLAB // Вестник Читинского государственного университета. 2012. № 1 (80). С.3-9.

Совершенствование технической подготовки студентов, обучающихся по направлению «Информационная безопасность» в Центре подготовки Институт Безопасности бизнеса МЭИ (ТУ)

Невский А.Ю.

кафедра «Комплексная безопасность бизнеса», Институт безопасности бизнеса Национального исследовательского университета «МЭИ»
111250, г. Москва, ул. Красноказарменная д. 14
nevskyay@mpei.ru,
WWW: http://www.mpei.ru

Аннотация. Анализ текущего уровня подготовленности студентов из результатов итоговой государственной аттестации выпускников, а также ознакомления с отзывами реальных и потенциальных работодателей свидетельствует об их недостаточной технической подготовке. В данной статье рассматриваются пути устранения дисбаланса между теоретической и технической подготовкой студентов в пользу последней, с конкретными предложениями по проведению учебных занятия по дисциплине «Технические средства охраны».

Введение

Работу по повышению уровня технической подготовки студентов 4 курса покажем на примере преподавания учебной дисциплины «Технические средства охраны», где можно отметить следующие особенности:

- данная учебная дисциплина является одной из наименее нагруженных в теоретическом плане дисциплин;
- ее освоение базируется на ранее изученных материалах, а, следовательно, носит интегральный характер;
- в процессе изучения данной дисциплины существуют широкие возможности использования технических средств и программных комплексов, реализующих основные функции систем безопасности.

Если говорить о компетентностном подходе в образовательной деятельности, то освоение данной дисциплины, в соответствии с федеральным государственным образовательным стандартом высшего профессионального образования по направлению подготовки «Информационная безопасность», ведет к овладению таких компетенций эксплуатационной и проектно-технологической деятельности, как ПК-9, 11, 12-15, 18, а также других

Как же осуществляется реализация компетентностного подхода при изучении одной из тем данной учебной дисциплины, а именно, темы «Системы охранного телевидения (CCTV)».

Ключевая проблематика

Проведем анализ содержания занятий, обеспечивающих техническую подготовку студентов.

Основным, на наш взгляд, здесь является освоение студентами компетенций, связанных с моделированием системы охранного телевидения для некоторого хозяйствующего субъекта, оценка ее эффективности по выбранным критериям и оценка экономических затрат на ее создание. Работы, связанные с моделированием, в настоящее время целесообразно проводить с использованием специального программного обеспечения.

В плане совершенствования материально-технической базы преподавания данной дисциплины в целом и упомянутой темы, в частности, нами приобретены следующие программные и технические средства:

- графический редактор VideoCAD-6.1-Лайт, предназначенный для проектирования систем охранного телевидения;
- программное обеспечение, представляющее собой надстройку над офисным приложением Microsoft Office Visio, для быстрого графического проектирования систем безопасности;
- программное обеспечение XLink, предназначенное для автоматизации расчетов сметной стоимости монтажа, наладки и технического обслуживания систем безопасности ресурсным методом;
- программное обеспечение EasySpec, предназначенное для автоматизации процесса разработки спецификаций на оборудование и материалы для систем безопасности;
- видеорегистратор «Vigilant M-JPEG»;
- DVD-рекордер «Samsung DVD Recorder & VCR DVD-VR330»;
- видеокамеры различных типов в количестве 4 шт.

Кроме этого нами приобретена достаточно обширная информационная база по всем вопросам безопасности.

Теоретические материалы, необходимые для освоения учебной дисциплины и конкретно данной темы размещаются на образовательном информационном ресурсе института по адресу http://www.i-b-b.su. Доступ к образовательным ресурсам требует аутентификации на сайте, что соответствует политике безопасности, разработанной в институте.

Рассмотрим, как организуется практикум для студентов по теме «Системы охранного телевидения (CCTV)». Общая характеристика практикума показана в табл.2.

Таблица 2

Практикум студентов по изучению темы «Системы охранного телевидения (CCTV)»

№ п/п	Тема занятия	Часов на занятие	Вид занятия	Программное и техническое обеспечение занятия
1	2	3	4	5
1	Моделирование системы охранного телевидения для типового объекта и оценка ее эффективности	4	Практич.	- лаборатория «Технические средства безопасности»; - графический редактор VideoCAD-6.1-Лайт; - офисное приложение Microsoft Office Visio, с надстройкой для проектирования систем безопасности.
2	Выбор технических средств для системы видеонаблюдения и оценка затрат на ее создание	2	Практич.	- лаборатория «Технические средства безопасности»; - программное обеспечение XLink; - программное обеспечение EasySpec; - поисковые сервисы Интернет.
3	Развертывание системы охранного телевидения и управление режимами ее функционирования с использованием стендового оборудования	2	Лабор. работа	- лаборатория «Технические средства безопасности»; - видеорегистратор «Vigilant M-JPEG»; - DVD-рекордер «Samsung DVD Recorder & VCR DVD-VR330»; - комплект видеокамер.
4	Разработка системы охранного телевидения для типового объекта	4	Контр. работа	- программное обеспечение, освоенное на занятиях.

Развитие технологии

На первом практическом занятии каждый студент получает задание на разработку системы охранного телевидения для одного из типовых объектов из следующего перечня: отдельная 3-4 комнатная квартира, коттедж, офисное помещение, торговое предприятие, склад.

Кроме типа объекта в задании студентам даются исходные данные для моделирования.

Задание выполняется в следующем порядке.

Студент создает графический план типового объекта с заданными размерами и в заданном масштабе. При этом используется офисное приложение Microsoft Office Visio, основы практического использования которого у наших студентов уже приобретены в процессе прохождения информационно-технологической практики в 6 семестре обучения.

Несмотря на то, что создание графического плана объекта может показаться избыточным, мы считаем, что это полезно в интересах повышения общей инженерной подготовки студентов.

В дальнейшем студенты, используя надстройку над приложением Microsoft Office Visio для графического проектирования систем безопасности, «накладывают» на план объекта технические средства видеонаблюдения.

В результате этого этапа создается графический проект моделируемой системы. Данный проект студент представляет преподавателю или ассистенту (лаборанту) и, после обсуждения и утверждения продолжает работу. При этом преподаватель (ассистент) обязательно обращает внимание на правильное, в соответствии с требованиями ГОСТа оформление чертежа, что также способствует повышению инженерной грамотности и культуры студента.

Следующим этапом является непосредственно моделирование системы охранного телевидения и оценка ее эффективности

Для этой работы студенты используют ранее упомянутый графический редактор VideoCAD-6.1-Лайт, предназначенный для проектирования систем охранного телевидения. Необходимо отметить, что это достаточно мощное программное средство, которое студент видит впервые, поэтому необходимым этапом освоения является знакомство с интерфейсом редактора и основными приемами работы. Для этого каждый студент получает практическое руководство по использованию VideoCAD, разработанное на кафедре КББ.

На втором практическом занятии студент продолжает работу над созданной моделью. При этом в соответствии с результатами моделирования технических средств видеонаблюдения осуществляется их выбор и оценка общих затрат на создание системы.

Здесь технология выполнения задания предполагает работу в несколько этапов.

1. Подбор видеокамер и других приборов для разработанной модели системы охранного телевидения с использованием поисковых сервисов Интернет и возможных их технических характеристик и создание собственной базы данных в приложении EasySpec;

2. Разработка спецификации на оборудование для системы охранного телевидения с использованием программного обеспечения EasySpec;

3. Расчет сметной стоимости оборудования для системы охранного телевидения, его монтажа, наладки и технического обслуживания ресурсным методом с использованием программного обеспечения XLink.

Третье занятие представляет собой *лабораторную работу*, на которой студенты завершают формирование практических навыков работы с образцами технических средств видеонаблюдения.

Лабораторная работа заключается в выполнении практических действий по развертыванию реальной системы охранного телевидения и управлению режимами ее функционирования. Для этого используется лабораторный стенд.

В завершение изучения данной темы (табл.2) предусматривается контрольная работа, которая выполняется в часы самостоятельной подготовки студентов. При этом каждым студентом выбирается типовой объект при условии, что на предыдущих занятиях им таковой не рассматривался, и выполняет весь комплекс работ, начиная от моделирования системы охранного телевидения и, заканчивая созданием электронной ее сметы на развертывание.

Так как технология работы студентами изучена, эти работы, по опыту выполняются студентом среднего уровня подготовки за 2-2,5 часа. Остальное время требуется на защиту своего решения у преподавателя.

Заключение

В заключении можно сказать, что в случае реализации описанной технологии проведения занятий по данной теме мы охватываем полный перечень общепрофессиональных компетенций обучаемых, заключающихся в их общей способности и готовности выполнять практические работы в интересах систем безопасности по:

- проектированию:
  - проведение предпроектного обследования;
  - экономическая оценка проекта;
  - разработка технического задания;
  - проектирование системы;
  - приемка системы;
  - организация внедрения;
- эксплуатации и аудиту:
  - моделирование угроз и уязвимостей;
  - разработка предложений по совершенствованию системы и их экономическая оценка;
- исследованию уязвимостей:
  - выбор программных средств;
  - проведение исследований;
- и других.

Можно с уверенностью сказать, что проведение учебных занятий с использованием технологий и технических средств, которые были описаны в настоящей статье, приведут к прорыву в области повышения инженерно-технической подготовки наших студентов.

Литература

1. Невский А.Ю. *Технические средства охраны. Учебно-методический комплекс.* М.: Издательский дом «МЭИ», 2010.
2. ГОСТ Р 51558-2008 Средства и системы охранные телевизионные.
3. Графический редактор VideoCAD-6.1-Лайт Руководство пользователя. Компания «Орбита-Союз», http://orbita.perm.ru.

Защита конфиденциальной информации при использовании «облачных» вычислений

Невский А.Ю.

кафедра «Комплексная безопасность бизнеса», Институт безопасности бизнеса Национального исследовательского университета «МЭИ»
111250, г. Москва, ул. Красноказарменная д. 14
nevskyay@mpei.ru,
WWW: http://www.mpei.ru

Аннотация. В век информационных технологий вопрос обработки больших объемов информации является одним из наиболее важных, в том числе и в нашей стране. Начиная от предприятий малого бизнеса, заканчивая международными корпорациями – всем приходится иметь дело с конфиденциальной информацией. Постоянный рост объёмов информации во времени приводит к необходимости увеличения вычислительных мощностей информационных систем, что заставляет бизнес постоянно вкладывать финансовые ресурсы в модернизацию информационных систем. Однако не все хозяйствующие субъекты могут позволить себе такие вложения капитала, чтобы постоянно поддерживать свои технические средства обработки на уровне современных требований. Не так давно появилась реальная альтернатива в виде возможности обращения к удаленным ресурсам и сервисам, предоставляемым специализированными компаниями, с целью обработки своих информационных активов.

Введение

Наиболее перспективным направлением являются системы так называемых «облачных» вычислений, под которыми понимается технология распределённой обработки данных, в которой компьютерные ресурсы и мощности предоставляются пользователям в виде некоторого интернет-сервиса.

В настоящее время компании уже могут перенести если не все, то большую часть своих вычислительных задач в «облако», экономя при этом на модернизации оборудования, покупке дорогостоящего программного обеспечения и на поисках квалифицированного персонала для внедрения этого оборудования и программного обеспечения.

В опубликованном в декабре 2010 года отчете «Облачные дивиденды - 2011» специалисты Центра экономических и бизнес-исследований (CEBR) утверждают, что к 2015 году благодаря «облачным» вычислениям экономика развитых европейских стран будет получать дополнительно по 177,3 млрд. евро в год [1].

Специалисты CEBR отмечают, что основная доля этих средств будет обеспечена за счет освоения частной и гибридной моделей «облачных» вычислений.

CEBR подсчитал, что годовой экономический эффект от «облачных» вычислений для каждого европейского государства к 2015 году составит:

- ФРГ - 49,6 млрд. евро;
- Франция - 37,4 млрд. евро;
- Италия - 35,1 млрд. евро;
- Великобритания - 30,0 млрд. евро;
- Испания - 25,2 млрд. евро.

Данный отчет является практически первой официальной оценкой значения освоения облачных вычислений на макроэкономическом уровне для пяти крупнейших экономик Европы.

Однако, несмотря на очевидные преимущества использования таких технологий, она не получает широкого распространения ни в ведущих мировых экономиках, ни в России. Главной причиной является достаточно низкая степень доверия к ним, то есть, у пользователя нет уверенности в том, что при использовании «облачных» сервисов не произойдет утечки или утраты ценной информации, а также он не в состоянии контролировать все действия, происходящие с ней.

Ключевая проблематика

Облачные вычисления – особый вид распределенных вычислений, которые появились еще в 1973 году. Данный вид вычислений представляет собой клиент-серверную технологию, которая базируется на работе группы серверов, которая представляется клиенту в виде единого виртуального сервера. Совокупность серверов позволяет:

- повысить доступность такой системы за счет масштабируемости: возможно увеличение количества используемых источников данного ресурса пропорционально увеличению потребности в нем, а также перенос работающего виртуального сервера на более мощный источник (так называемая «живая миграция»);
- понизить риск отказа в обслуживании в случае выхода из строя одного из серверов, обслуживающих систему, так как он автоматически заменится другим, резервным сервером.

В соответствии с взглядами специалистов Национального института стандартизации и технологий США NIST (the National Institute of Standards and Technology) системы облачных вычислений представляют собой совокупность пяти важных характеристик, трех моделей предоставления услуг и четырех моделей предоставления, как показано на рис. 1 [2].

Характеристики систем «облачных» вычислений:

а) широкий доступ. Вычислительные мощности предоставляются по сети, доступ к ним осуществляется с помощью стандартных механизмов, доступных клиенту с любых платформ (ПК, ноутбуки, смартфоны и т.д.);

б) масштабируемость. Дополнительная вычислительная мощность предоставляется легко и быстро, сразу же после оплаты ее клиентом;

в) распределение услуг. Системы облачных вычислений автоматически распределяют ресурсы между клиентами по мере их потребности;

г) самообслуживание. Клиент может получить дополнительные вычислительные мощности в любой момент, когда ему это необходимо, не связываясь с представителем услуг;

д) пул ресурсов. Вычислительные мощности компании-провайдера представляют собой совокупность ресурсов, которые своевременно предоставляются клиентам по их требованию [2].

Развитие технологии

Системы облачных вычислений обладают рядом достоинств и недостатков. Среди достоинств необходимо отметить:

1. Экономическая выгода. За дополнительное хранилище данных компания при использовании «облака» платит только за использование «облачного» сервера. Альтернатива – расходы на покупку оборудования, выделение места, оплату труда персонала и др.

2. Гибкость. Возможность предоставления клиентам дополнительных вычислительные мощностей в реальном времени.

3. Надежность. «Облачная» компания организует распределенное хранение данных и принимает ряд мер по повышению надежности своих сервисов.

Однако, несмотря на вышеперечисленные достоинства, существуют определенные опасения, связанные с надежностью сервисов и возможностью обрабатывать конфиденциальную информацию в таких системах.

Одним из ярких инцидентов информационной безопасности является проблема, возникшая у провайдера Dropbox летом 2012 года. Злоумышленники получили доступ к учетным записям, пароли от которых они узнали благодаря массовой утечке паролей с других сайтов. Среди прочих, злоумышленники получили доступ к учетной записи одного из разработчиков Dropbox, у которого в папке лежал документ с почтовыми адресами пользователей - именно эта база позволила злоумышленникам осуществить рассылку спама [3].

В мировой практике «облачных» вычислений известны случаи, когда потребитель в течение длительного времени не мог получить доступ к приложениям. Кроме этого, простое «отключение Интернета» по вине провайдера (и не только) непосредственно обслуживающего заказчика, может сделать работу с «облачными» ресурсами невозможной в принципе.

К потенциальным угрозам использования «облачных» услуг относят:

1. Угроза конфиденциальности информации, обусловленная необходимостью передавать данные, условно говоря, в «никуда».

2. Сложности, связанные с самим провайдером. Возможная сложность миграции от одного провайдера к другому или в собственное облако. Банкротство или поглощение провайдера.

3. Угроза гарантированному удалению информации в случае прекращения ее обработки или отношений с Заказчиком услуг.

4. Неполучение или несвоевременное получение услуг вследствие отказа или ошибок оборудования.

5. Уязвимость «облачных» систем всем традиционным типам компьютерных атак в сочетании с использованием обычных средств защиты.

6. Угрозы атак типа «человек посередине» при транспортировании информации к «облаку».

7. Несоответствие законодательным требованиям при обработке, например, персональных данных.

8. Угроза виртуализации, заключающейся в появлении в системе дополнительных целей атак, таких как:

- гипервизор, т.е. на ключевой элемент, который обеспечивает разделение ресурсов физического компьютера между виртуальными машинами;
- виртуальная машина, перенос которой с одного узла на другой предполагается в системах управления виртуальными машинами. При этом виртуальной машиной, представляющей собой специальный файл, можно манипулировать в широких пределах;
- системы управления виртуальными машинами, атаки на которые могут привести к возможности получения контроля над отдельными виртуальными машинами, частью «облака», или «облаком» в целом [4].

Нами был проведен анализ политик конфиденциальности наиболее крупных и популярных «облачных» сервисов, в результате которого можно сделать вывод о том, что в них речь идет в основном об обеспечении целостности и доступности самих сервисов посредством, например резервирования серверов.

Например, политика конфиденциальности сервиса iCloud от Apple описывает:

- какой тип личной информации компания будет собирать;
- как они ее будут использовать;
- как они ее будут защищать.

Защита будет заключаться в принятии ряда мер, в том числе организационных, технических и физических. Эти меры направлены на противодействие утрате, краже, утечке, несанкционированному доступу, разглашению, изменению и уничтожению. Также в политике упоминается награда TRUSTe за соблюдение всех требований этой программы [10]. Единственное названное в данной политике средство защиты, которым предполагается обеспечить конфиденциальность информации, является использование протокола SSL, о котором было сказано выше.

Для ответа на вопрос: «Что может предпринять Заказчик услуг?», необходимо иметь в виду информационные задачи, решаемые в «облаке».

1. Хранение. При решении этой задачи, вопроса доверия к системам облачных вычислений не возникает. Компания обрабатывает информацию у себя, а в «облаке» она ее только хранит, следовательно, можно воспользоваться криптографическими средствами защиты.

2. Обработка и хранение. Здесь уже начинают возникать некоторые затруднения, ведь для обработки информацию нужно посылать в открытом виде, что порождает уязвимость.

3. Обработка. Здесь также есть затруднения, аналогичные п. 2.

Таким образом, проблемы безопасности связаны с обработкой информации в «облаке». При этом шифрование данных на стороне Заказчика не представляется возможным, так как оно делает невозможным саму обработку информации.

Исходя из того, что наиболее часто встречающимся видом данных, пригодных для обработки, являются табличные данные, нами предлагаются следующие, наиболее просто реализуемые решения:

- обезличивание информации;
- кодирование (легендирование) информации.

Заключение

Необходимость обработки значительных объемов информации – это реалии сегодняшнего бизнеса. Следствием этого является постоянный рост требований к вычислительным мощностям информационных систем организаций.

Далеко не все предприятия могут себе позволить постоянную модернизацию оборудования.

В этом случае использование систем «облачных» вычислений представляется обоснованным вследствие значительного сокращения расходов на оборудование, программное обеспечение, обучение персонала и других.

Однако использование «облачных» сервисов и технологий представляется не таким «безоблачным», что связано с невозможностью контроля информации со стороны владельца после ее передачи в «облако». В случае необходимости обработки конфиденциальной информации перед владельцем встает дилемма «экономия, удобство – утечка, ущерб». Таким образом, встает вопрос доверия, точнее говоря – отсутствие такового, к таким системам и вопрос этот очень важен.

В настоящей работе предпринята попытка обоснования некоторых подходов к обеспечению конфиденциальности информации, обрабатываемой в «облачных» сервисах, таких как обезличивание и кодирование, которые за счет использования достаточно простых приемов позволят поднять уровень доверия Клиентов к таким системам, особенно в начальный период их широкого использования.

Литература

1. Облачные вычисления: [Электронный документ] (http://www.tadviser.ru/).
2. Cloud Security Alliance. Security Guidance v2: [Электронный документ] (https://cloudsecurityalliance.org).
3. Dropbox вводит двухфакторную аутентификацию. [Электронный документ] (http://www.xakep.ru/post/59084/).
4. E-executive – сообщество менеджеров. Владислав Шелобаев: Детство «облачной» безопасности, или Встреча с пугалом: [Электронный документ] (http://www.e-xecutive.ru/).

5. Cloud computing is a trap, warns GNU founder Richard Stallman. Bobbie Johnson: Guardian, 29 сентября 2008 г.
6. Biting the hand, than feeds. Hackers break SSL encryption used by millions of sites. Dan Goodin: [Электронный документ] (http://www.theregister.co.uk/).
7. Safe Harbor. Frequently asked questions: [Электронный документ] (http://export.gov/safeharbor/).
8. Symantec. Virtualization and Evolution to the Cloud Survey [Электронный документ] (http://symantec.com)
9. РИА-Новости. Требования к облачным платформам могут быть разработаны до конца года. Иван Шадрин: [Электронный документ] (http://www.ria.ru/).
10. Apple Inc. Apple Costumer Privacy Policy: [Электронный документ] (http://www.apple.com).
11. Amazon Help. Amazon Privacy Notice: [Электронный документ] (http://www.amazon.com).

Моделирование объекта автоматизации в процессе проектирования информационных систем

Ковтун И.И.

Кафедра ИиИТ, Институт «МАБиУ»,
Москва, 129594, г. Москва, 5-й проезд Марьиной Рощи, д. 15-а
Kovtoon@mail.ru,
WWW: http://www.mabiu.ru

Аннотация. Порядок проектирования и разработки информационных систем, а также их жизненный цикл предопределяют многократное описание, как всей системы, так и ее различных компонент, как в проектной, так и в эксплуатационной документации. Работа посвящена поиску оптимального сочетания методологий функционального моделирования, реляционного подхода и гипертекстовых описаний.

Введение

В качестве универсальной модели описания информационной системы можно выбрать текст на естественном языке, т.е. на русском, английском и т.п., или дальнейшее развитие данной модели – гипертекст, т.е. текст с выделенными словами, где инициация выделенного слова ведет к переходу на другой, семантически связанный фрагмент текста. Как модель для описания текст на естественном языке обладает одним неоспоримым преимуществом – на нем можно описать практически любой нюанс человеческой практики. Недостатком текста является сложность компьютерного анализа, а именно, сложность выбора из него статистической информации для принятия некоторого управленческого решения. Поэтому наряду с использованием текста постоянно встает необходимость применения других моделей и средств их реализации для получения более подробной и качественной информации об объекте исследования. Так, для представления информации, наряду с текстом и гипертекстом широко используется реляционная модель данных, а в процессе проектирования, разработки и создания информационных систем все больший оборот набирают семейство методологий IDEF и методология DFD. Работа посвящена поиску оптимального сочетания в процессе построения информационных систем методологий функционального моделирования, реляционного подхода и гипертекстовых описаний.

Ключевая проблематика

Анализ существующих подходов к построению сложных информационных систем показал, что они не в должной мере обеспечивают одновременное выполнение двух важных требований:
- необходимости представления данных и знаний в форме, обеспечивающей компактное хранение и оперативную обработку информации;
- необходимости наглядного отображения проектных решений в текстовой и графической форме, в ряде случаев наиболее предпочтительной, как для заказчика, так и для разработчика.

Развитие технологии

Для создания системы описания информации, удовлетворяющей указанным требованиям, предложена модель,в основу которой положены следующие решения:

1. Данные предложено представлять семантической сетью, вершинами и ребрами которой являются слова – конечные последовательности символов или числа, и их совокупности – объекты. Обоснована возможность применения к такой семантической сети известных средств обработки семантических сетей.
2. В состав объектов включены страницы. Информация на страницах представлена в виде гипертекста. Гипертекст использован как средство управления взаимодействием с базой данных, так как переход по выделенному слову определяется не только семантикой слова, но и некоторым правилом, установленным экспертом базы данных.
3. Объекты семантической сети разбиты на множество двумерных таблиц. Показана возможность применения к таким таблицам операций реляционной алгебры.
4. Страницы переменной длины предложено хранить в нескольких файлах, в каждом из которых записываются куски текста строго определенной длины. Поставлена и формализована задача оптимизации количества файлов и размеров кусков непрерывной записи в них. Решение сведено к стандартным методам исследования операций.

Заключение

Основными научными и практическими результатами работы являются:
- постановка и исследование задачи интеграции методов и моделей организации данных в единую модель, совмещающую наглядность отображения информации с простотой ее хранения и обработки;

- разработка подхода к решению поставленной задачи, основанного на моделировании данных семантической сетью объектов, организованных в двумерные таблицы и допускающих в своей структуре гипертекст;
- разработка машинного представления предложенных логических структур данных;
- создание информационно-справочной системы, основанной на предложенных средствах хранения и обработки информации.

Литература

1. Мейер Д. Теория реляционных баз данных: Пер. с англ. – М.: Мир, 1987. – 608 с., ил.
2. Мартин Дж. Организация баз данных в вычислительных системах. Издательство «Мир» 1980.
3. C. Gane, T. Sarson. «Structured Systems Analysis: Tools and Techniques». New York: IST, Inc., 1977.
4. Carlo Batini, StefanoCeri, Shamkant B. Navathe. «Conceptual Database Design». Redwood City, Ca.: The Benjamin/Cummings Publishing Company, Inc., 1992.

Проблемы обеспечения защиты информации при использовании информационных технологий

Баронов О.Р.

Кафедра КББ ИББ, ``ФГБОУ НИУ «МЭИ»",
Москва, 111250 РФ, Красноказарменная ул., д. 14
BaronovOR@mail.ru,
WWW: http://www.i-b-b.su

Аннотация. Внедрение информационных технологий во все сферы нашей жизни создало новые проблемы в обеспечении информационной безопасности. Одной из актуальных проблем является похищение денежных средств со счетов отдельных граждан, которые для оплаты товаров и услуг, получения наличных денег в банкоматах, используют пластиковые кредитные и дебетовые карты. Существующее положение дел с защитой персональных данных и реквизитов банковской карты, относительно существующих угроз не отвечает требованиям сегодняшнего дня. Для их защиты предлагается к внедрению система комплексного противодействия кардингу, которая содержит в обеспечивающих подсистемах приоритетные меры, способы и средства защиты.

Введение

В апреле 2010 г. руководство Сбербанка России предупредило своих клиентов владеющих банковскими картами, о всплеске мошенничества в основе которого лежат действия злоумышленников по сбору конфиденциальной информации с использованием SMS-сообщений, ложных телефонных звонков из «банка» с мобильного телефона, посредством использования клиентами интернет-банкинга и др. Несмотря на то, что самым лучшим методом защиты является внимательность владельца кредитной карты и его осведомлённость, в сети злоумышленников попадают далеко не самые легковерные люди.

Ключевая проблематика

Проведенный анализ показывает, что на сегодняшний день возможны мошенничества с использованием любых типов банковских карт. Чтобы узнать номера счетов банковских карт и персональные данные их держателей для проведения несанкционированных операций по счету карты, мошенники изобретают различные способы съема информации. Способы могут быть самыми разно-

образными и изощренными. При этом сбор конфиденциальной информации осуществляется как на основе психологического воздействия, так и за счет использования технических средств. Поэтому решать проблемы защиты необходимо с использованием комплексного подхода, сочетающего в себе такие аспекты, как совершенствование законодательной базы банковской системы, аппаратных средств и строгого соблюдения определенных рекомендаций и инструкций.

Развитие технологии

Надежная защита пластиковой карты, а, следовательно, и ее владельца от любых видов кардинга возможна только при эффективном функционировании системы комплексного противодействия кардингу, функционирующей в составе подсистем психологической и воспитательной работы, административного и правового обеспечения, организационного обеспечения и современного технического обеспечения.

Целью системы комплексного противодействия кардингу и борьбы с ним является предотвращение кардинга или снижение вероятности его успеха. Достижение цели возможно при обеспечении эффективного функционирования всех, перечисленных ранее подсистем. Развитие подсистемы психологической и воспитательной работы должно основываться на совершенствовании подготовки кадров в банковской системе, информировании населения об угрозе кардинга с использованием средств массовой информации, а также выпуска специальной литературы и познавательных телевизионных программ. Подсистема административного и правового обеспечения должна развиваться по пути совершенствования законодательной базы, страхования пластиковых карт, разработки актуальных рекомендаций и инструкций пользователям пластиковых карт по их использованию. Подсистема технического обеспечения должна развиваться за счет совершенствования аппаратных и программных средств защиты на основе модернизации каналов связи, реализации технологии обязательного использования PIN-кодов, внедрение систем своевременного мониторинга транзакций карты, отказа от применения магнитных карт и использовании технологий чипованных пластиковых карт и бесконтактных смарт-карт.

Заключение

Оценивая возможность эффективно решить проблемы обеспечения информационной безопасности при использовании информационных технологий можно прийти к выводу, что необходим комплексный подход - перспективы и возможности которого связаны с противодействием картингу по следующим основным направлениям: психологическое и воспитательное, правовое и административное, организационное и техническое. При этом дальнейшее совершенствование подсистем системы комплексного противодействия кардингу

должно осуществляться в следующих направлениях: всестороннее просвещение населения в этой области; совершенствование законодательной базы РФ; совершенствование мероприятий, направленных на использование новых информационных технологий по реализации транзакций по кредитным картам; совершенствование технических устройств (банкоматов) и программного обеспечения.

Литература

1. Савин Г.Е. Экономические преступления, совершаемые с использованием информационных технологий, и способы их предовращения. // Вестник Академии экономической безопасности МВД России. – 2010. – № 10. – С. 70–77.

Информационно-аналитическая платформа «Дедуктор» для решения задач бизнес аналитики

Нестерович С.А.

Кафедра ИТ института ИиИТ МАБиУ
Москва, РФ, 5-й проезд Мариной рощи д. 15.

Аннотация. Любая организация в процессе своей деятельности стремится максимизировать прибыль. Для этого ей необходимо правильно организовать работу, ориентируясь на те вещи, которые приносят наибольшую прибыль с наименьшими затратами. Некоторое время назад считалось, что товар или предоставляемые организацией услуги приносят ей прибыль. Сейчас же все большее распространение приобретает клиентно-ориентированная стратегия, где во главу угла ставится клиент и механизмы взаимодействия с ним. Данная стратегия получила название CRM - Customer Relationship Management. На рынке существует большое разнообразие CRM систем [1], однако почти все они предназначены для автоматизации работ по сбору и систематизации данных о клиентах и практически не обладают развитыми средствами анализа. Только в наиболее дорогих имеются средства OLAP. Из-за игнорирования вопросов анализа CRM данных, часто организации даже не подозревают о каких-то закономерностях в поведении клиентов, в то время как знание подобных закономерностей и их учет в своих действиях может принести значительную практическую пользу.

Введение

Информационно-аналитическая платформа Deductor предлагает готовое аналитическое решение для анализа информации о клиентах на базе платформы Deductor 5.2. Решение включает в себя хранилище данных о клиентах, готовые механизмы аналитической отчетности, а также сценарии решения более сложных задач: анализ воздействия рекламы, сегментация клиентов, установление признаков клиентов, приносящих наибольшую прибыль, определение предпочтения товаров, прогнозирования успеха сделки и объемов продаж.

Хранилище данных по клиентам

В хранилище данных информация хранится в измерениях и процессах. Измерение - это объект анализа, который может характеризоваться свойствами, присущими только ему и имеет уникальный идентификатор. Процесс представляет собой звезду, в центре которой хранятся факты, а лучи являются измерениями.

В хранилище данных будет содержать следующую информацию о клиентах: название, вид (юридическое или физическое лицо), сфера деятельности, географический регион расположения клиента, тип клиента по классификации ABC, тип клиента по классификации XYZ, потенциальный клиент или клиент.

Процесс отображает сделку с клиентом, то есть продажи клиенту некоторого товара. Фактом процесса будет сумма сделки и количество закупаемого товара. Измерения – клиент, менеджер организации, курирующий сделку, дата совершения сделки, состояние сделки (открыта, отказ, успех), причина отказа в случае неуспешной сделки, источник информации о приобретаемом товаре, товар.

Например, процесс «Сделка» содержит следующее **измерения**:

- номер клиента;
- состояние сделки;
- причина отказа, в случае неуспешной сделки;
- менеджер, курирующий сделку;
- дата сделки;
- источник информации, способствующий сделке;
- номер покупаемого товара;
- сфера деятельности клиента.

и **факты**:

- количество купленного по сделке товара;
- сумма сделки.

Данные в ХД могут собираться из любых источников, например из учетной CRM системы, бухгалтерской или складской программы.

Когда хранилище данных подготовлено, можно приступать к анализу данных.

Создание аналитической отчетности

Благодаря аналитической отчетности, данные из хранилища представляются в виде, удобном для дальнейшего анализа. Наиболее удобным инструментом для получения аналитической отчетности являются OLAP-кубы (On Line Analytical Processing - оперативная аналитическая обработка данных). OLAP дает возможность в реальном времени генерировать описательные и сравнительные сводки данных и получать ответы на различные другие аналитические запросы. OLAP-кубы представляют собой проекцию исходного куба данных на куб данных меньшей размерности. При этом значения ячеек объединяются. Такие проекции или срезы исходного куба представляются на плоскости в виде кросс-таблицы.

На рисунке 1 представлена кросс таблица с двумя измерениями: менеджер и статус сделки. По такой таблице легко определить наиболее успешных менеджеров.

	DEAL_STATE			
MANAGER	Отказ	Открыта	Успех	Итого
Баранов	8 105 280,00	7 724 140,00	46 459 960,00	62 289 380,00
Дмитриева	22 065 220,00	19 748 960,00	15 393 080,00	57 207 260,00
Иванова	20 820 540,00	23 102 400,00	14 714 560,00	58 637 500,00
Костина	20 866 180,00	23 381 780,00	14 251 360,00	58 499 320,00
Медведев	7 356 480,00	8 457 580,00	45 713 460,00	61 527 520,00
Чернова	7 990 900,00	8 014 340,00	46 028 440,00	62 033 680,00
Итого	87 204 600,00	90 429 200,00	182 560 860,00	360 194 660,00

Рисунок 1.

Deductor в OLAP-кубе с помощью инструмента «селектор» можно агрегировать факты по какому-либо измерению, оставляя только те объекты, которые соответствуют указанному условию. Например, можно объединить сумму сделок по клиентам, оставив только тех, которые в сумме приносят 50% прибыли. На рисунках 2 и 3 представлены селектор и клиенты, приносящие 50% прибыли.

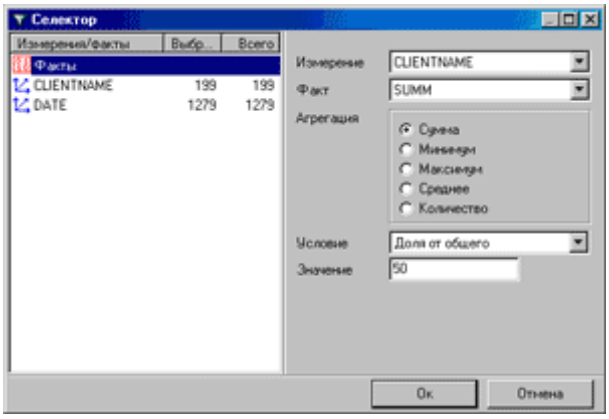


Рисунок 2. Клиенты, приносящие 50% прибыли

CLIENTNAME	Σ SUMM
Покупатель 1	3 044 620,00
Покупатель 10	2 735 880,00
Покупатель 12	3 144 920,00
Покупатель 2	15 923 320,00
Покупатель 3	5 767 360,00
Покупатель 35	2 811 700,00
Покупатель 4	10 285 840,00
Покупатель 5	5 326 060,00
Покупатель 6	14 545 360,00
Покупатель 7	8 896 760,00
Покупатель 8	19 803 680,00
Итого	92 285 500,00

Рисунок 3.

Сегментация клиентов

Наиболее активные фирмы, занимающиеся торговлей, используют как пассивную рекламу для привлечения новых клиентов, то есть, например, рекламу на телевидении, на радио, в прессе, так и рассылку с прямыми коммерческими предложениями. Для повышения эффективности подобных мероприятий необ-

ходимо учитывать интересы клиентов, объектов воздействия. То есть предлагать клиентам именно тот товар, который они предпочитают. Но нельзя учесть предпочтения каждого клиента. Необходимо выделять некоторые группы - сегменты - клиентов и уже этим группам предлагать конкретную категорию товаров.

Выделять сегменты клиентов можно по нескольким группам признаков. Это могут быть сегменты по сфере деятельности, по географическому расположению. После сегментирования можно узнать, какие именно сегменты являются наиболее активными, какие приносят наибольшую прибыль и где находятся наиболее лояльные клиенты, выделить характерные для них признаки. Для решения этой задачи, мощным механизмом кластуризации – воспользуемся самоорганизующимися картами Кохонена.

На рисунке 4 представлены карты Кохонена, полученные после кластеризации клиентов по характеристикам: вид (KIND), удаленность региона (DISTANCE), сфера деятельности (BRANCH), среднегодовые суммы сделок (SUMM) и частота сделок – количество сделок в неделю (DEAL_STATE). Поле BRANCH является строковым и использовалось как выходное, поэтому в процессе обучения нейросети оно не участвовало. Но после группировки по остальным полям, значения поля BRANCH также могли быть сгруппированы на карте.

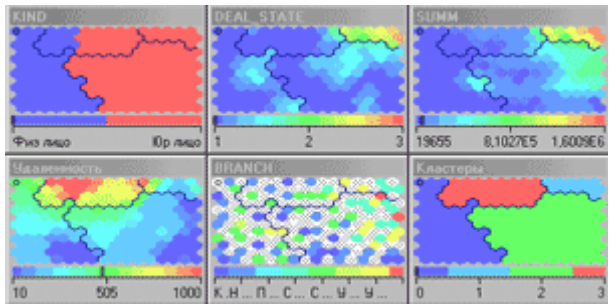


Рисунок 4. Сегментация клиентов.

Полученные сегменты можно интерпретировать следующим образом. Сегмент 0 – физические лица, расположенные ближе 600 км – клиенты, совершающие сделки на небольшую сумму и не очень часто. Сегмент 1 – клиенты, приносящие большую прибыль и часто совершающие сделки – это юридические лица в сфере связи, расположенные от 175 до 600 км. Клиенты сегмента 3 редко совершают сделки и приносят маленькие прибыли. Эти клиенты расположены дальше 600 км. Сегмент 2 обладает средними показателями.

Прогнозирование объемов продаж

Торговые организации стремятся свести к минимуму время, которое товар лежит на складе, а также место, которое он там занимает. С другой стороны, необходимо чтобы на складе всегда лежал требуемый в настоящее время товар.

Прогнозирование объемов продаж является важным шагом на пути принятия решения по оптимизации работы предприятия.

Важными данными для построения прогноза объемов продаж - это статистика продаж за предыдущие периоды. Такая информация есть в нашем хранилище. Не всегда имеет смысл делать общий прогноз продаж. Дело в том, что существуют случайные клиенты, которые нарушают общую тенденцию, покупая товар в «неподходящий» сезон или случайный товар. Поэтому перед прогнозированием нужно выделить сделки по соответствующему сегменту клиентов и определенной группе товара, например, предпочитаемого этим сегментом.

Для примера сделаем фильтрацию всех сделок по третьему сегменту клиентов, группе товара номер 1 (предпочитаемый третьим сегментом), оставив только успешные сделки. Прогнозировать будем количество продаваемого товара. Рассматривать сделки по дням не имеет смысла, так как каждый день может очень сильно отличаться от другого по объему продаж. Но продажи в среднем по неделям или по месяцам отличаются не сильно. Сгруппируем все сделки по неделям. После такой группировки тенденция наблюдается лучше, но все еще существуют различные выбросы и шумы на кривой продаж. Для их устранения в Deductor применяется инструмент «парциальная обработка».

Прогноз строится на предположении сезонности продаж и общего развития рынка. Для оценки сезонности используют автокорреляцию. Она показывает степень зависимости некоторого значения от предыдущих значений этой величины.

После оценки сезонности строится модель. В нашем примере, когда объемы продаж зависят от предыдущих продаж, можно использовать линейную регрессию. Однако, в более сложных ситуациях лучшие результаты даст использование нейронных сетей для построения более адекватных моделей. При обучении линейной регрессии будем подавать на ее вход объем продаж за две предыдущие недели текущего года, чтобы учесть общее развитие рынка, и две недели, соответствующие им год назад, чтобы учесть сезонность.

После того, как построена модель можно построить прогноз, прогнав через нее историю продаж. Сделаем прогноз на 4 недели вперед. Прогноз на более длительный период будет в гораздо меньшей степени соответствовать действительности. Результат можно посмотреть на диаграмме прогноза (рисунок 5).

Рисунок 5 Прогноз объемов продаж на 4 недели (диаграмма прогноза).

Сделав таким способом прогноз по всем сегментам клиентов и всем группам товаров, можно получить общий прогноз объемов продаж. Такой прогноз учи-

тывает сезонность продаж каждой группы товаров и активность каждого сегмента клиентов в разные периоды времени.

Заключение

Заменив в рассмотренных выше примерах товар на услуги можно распространить проведенный анализ на организации в сфере услуг. Но нельзя построить универсальную аналитическую систему, которая бы подходила любой клиентно-ориентированной организации. Для успешной деятельности одних организации необходима, например, информация о личных характеристиках человека, таких как пол, вес, рост, возраст. Для других необходима информация о характеристиках организаций, таких как удаленность, число сотрудников, сфера деятельности, занимаемая площадь. Поэтому создание аналитической системы начинается уже при создании хранилища данных, специализированного для аккумулирования специфических данных о клиентах. Отличие может быть и в зависимостях между данными. Например, сезонность продаж зависит от продаваемого товара. Такая сезонность может наблюдаться при торговле одеждой, но ее может не быть при торговле продуктами питания. Для каждого конкретного случая необходимо строить свою модель. Даже, если у двух различных организаций абсолютно одинаковая деятельность, модели, построенные для одной, могут не работать для другой. Необходимо их переобучать на данных именно этой организации. Именно поэтому данное решение реализовано на платформе Deductor, обладающего достаточной гибкостью, удобными механизмами интеграции со сторонними системами и большим набором инструментов анализа. Это позволяет быстро адаптировать решение в соответствии с изменяющимися требованиями.

Литература

1. Дюк В., Самойленко А. Data mining: учебный курс. – СПб: Питер, 2011. – 368 с.
2. Круглов В.В., Борисов В.В. Искусственные нейронные сети. –М.: Горячая линия – Телеком, 2002. – 382 с.
3. Neural Networks for Instrumentation, Measurement and Related ndustrial Applications. Proceedings of the NATO Advanced Study Institute on Neural Networks for Instrumentation, Measurement, and Related Industrial Applications (9-20 October 2001, Crema, Italy) / ed. by Sergey Ablameyko, Liviu Goras, Marco Gori and Vincenzo Piuri, IOS Press, Series 3: Computer and Systems Sciences, 2003,Vol. 185, Amsterdam.
4. Аналитическая платформа Deductor: www.basegroup.ru

Системный анализ проблемы управления качеством обучения студентов за рубежом и известных подходов к ее решению

Ташани Н.А.

Кафедра АСУ, Московский государственный горный университет,
Москва, 119991 РФ, М. Октябрьская, Ленинский проспект д. 6,
n_tashani@hotmail.com,
WWW: http://www.msmu.ru

Аннотация. В данной статье характеризуется процесс управления обучением за рубежом. Был выбран подход к решению проблемы управления качеством обучения студентов за рубежом с учетом их психологических способностей на основе современных информационных технологий. Были сформулированы цели и задачи исследований.

Введение

Многие развивающиеся государства официально учредили образовательную стипендию для подготовки кадров за рубежом. Не все тогда понимали, зачем таким государствам делать такие большие инвестиции в образование на фоне низкой экономики, безработицы, сокращения социальных расходов. Это понимание пришло со временем, когда стало ясно, что для дальнейшего качественного роста не только рыночной экономики, но и государства в целом, нужны высококлассные специалисты, обладающие не только знанием языков, основ современного менеджмента, но и другой ментальностью, видением проблем. Сейчас перед этими государствами стоят другие масштабные задачи. Для того, чтобы делать дальнейшие шаги вперед, нужен большой качественный рост. В связи с этим большую важность имеет разработка общего подхода к управлению качеством обучения студентов за рубежом, поскольку многие, кто учился за рубежом, не смогли адаптироваться к новым условиям жизни или к учебной деятельности [6].

Обучение за рубежом - сложный и многоплановый процесс. Обучаемый в процессе образования сталкивается с проблемами, которые могут носить не только учебный, но и психологический характер. Под психологическими проблемами следует понимать психологическую адаптацию обучаемых за рубежом при смене социокультурной ситуации [1]. Межкультурная адаптация понимается как сложный процесс, благодаря которому молодой человек достигает соответствия («совместимости») с новой культурной средой [2]. Актуальными для исследования ставятся вопросы о том, как происходит включение личности в

разнообразные ситуации нового социокультурного окружения, как человек преобразует, переосмысливает для себя действительность, как преодолевает преграды, стоящие на пути реализации его привычного поведения, как переживаются личностью конфликты, возникающие в процессе адаптации [3].

Однако и существуют и другие психологические проблемы связаны с вопросами усвоения знаний, снижение его самооценки, интереса к работе, мотивации и эмоционального настроя.

Ключевая проблематика

В результате исследования специфики рассматриваемой предметной области необходимо применение методов искусственного интеллекта для решения поставленной задачи из-за отсутствия прямого математического аппарата для решения данной проблемы и наличия «творческой» составляющей. То есть факты и знания данной предметной области неструктурированны и неполны, а задачи решаются методом неформальных рассуждений. Но огромную важность имеет разработка общего подхода к контролю качества обучения студентов за рубежом и поддержке обучаемых с учетом психологических особенностей, поскольку многие проблемы связаны с вопросом адаптации в новой социальной среде.

Принципиально важную роль в процессе управления качеством обучения студентов за рубежом играет полная и достоверная информация о состоянии процесса обучения. При этом получение подобной информации должно постоянно возобновляться в течение достаточно короткого промежутка времени (например: семестр). В противном случае управляющие воздействия могут оказаться неэффективными или даже бесполезными. Поэтому такая задача должна быть возложена на Бюро культуры в посольстве данного государства за рубежом.

Необходимым условием обеспечения эффективности процесса управления качеством обучения является использование современных информационных технологий, а существующие компьютерные системы в данной области (например, для оценивания понятийного состава знаний обучаемого, для оценки качества знаний в системе открытого образования, для обучения, для профориентации и др.) строятся, в основном, в классе информационно-справочных систем. Современные уровни развития информационных технологий позволяют использовать вычислительную технику для повышения эффективности данного процесса. Поэтому создание интеллектуальных информационных обучающих и советующих систем, предназначенных для управления, контроля и самоорганизации процесса обучения, является актуальной задачей. Управление качеством обучения за рубежом с использованием психологического аппарата не решено и построение необходимой системы, помогающей обучаемому решать психологические проблемы, возникающие в процессе обучения.

Система должна содержать модели различных ситуаций и модели принятия решений в этих ситуациях. В системе должны храниться результаты успеваемости и большого количества тестов по каждому обучаемому - тесты на опреде-

ление психологического состояния обучаемого за рубежом, уровня адаптации и др. [5].

Развитие технологии

Следует признать, что управление качеством обучения студентов - это давно устоявшийся вопрос в учебных заведениях, где общие подходы в оценке качества обучения построены на отдельных качественных показателях при оценке образовательной деятельности, а также проблемой психологической поддержке обучаемых в высших школах и университетах начали заниматься еще в прошлом веке. Однако управление качеством обучения студентов за рубежом и психологическая поддержка обучаемых является достаточно узкой проблемой и не существует формальных методов для её решения.

Государства, которые отправляют своих студентов за рубежом, обычно принимают следующие меры для управления качеством обучения студентов за рубежом:

- Выбирают лучших из лучших по уровню знаний;
- Требуют получения высшего образования по специальностям, востребованным на Родине студентов;
- Требуют от них ежегодных отчетов для контроля их успеваемости;
- Предоставляют студентам государственное финансирование.

Государство имеет право потребовать возвращения всех средств, затраченных им на обучение студента за рубежом в одном из следующих случаев:

- При несоблюдении правил и сроков оформления регистрации;
- Если результаты исследования показали неспособность студента обучаться за рубежом или при отчислении его из института, в который он был послан своим государством;
- Если студент прервал своё обучение, не имея на то веских причин;
- Если студент поменял специальность без предварительного согласования с Министерством образования или иным учреждениями-инвесторами.

В случае болезни или смерти студента соответствующие органы принимают решения в каждом конкретном случае индивидуально.

24 июля 2011 года Правительство РФ сообщило о своих планах выделить финансовые средства студентам, самостоятельно поступившим в ведущие зарубежные университеты, но не имеющим возможность самим оплатить стоимость обучения. Согласно данной инициативе в 2012 году за рубежом смогут обучаться уже 2000 студентов и их число, увеличиваясь с каждым годом, предположительно достигнет в 2015 году 10 000 человек.

Планируется, что программа финансирования будет охватывать два направления – обучение по магистерской программе и получение докторской степени (PhD). В общей сложности в 2012 году для этой цели будет выделено около 60 млн. долларов. В первые годы действия программы студенты смогут сами выбирать специальности, по которым они хотели бы обучаться, однако, в даль-

нейшем существует вероятность того, что правительство будет само определять приоритетные направления.

Студенты, получившие государственное финансирование, должны будут обязательно вернуться на родину после окончания учебы. В случае если студент не вернется в положенный срок, он обязан будет вернуть потраченные на него финансовые средства и выплатить неустойку[7].

В России и в зарубежных государствах вопрос управления качеством обучения студентов за рубежом раскрыт не в том аспекте и поэтому нуждается в дальнейшей разработке [5]. На мировом рынке имеется большой выбор программных продуктов, позволяющих обеспечивать контроль качества обучения студентов в учебных заведениях или по дистанционному обучению. Эти программные продукты позволяют обеспечивать весь цикл разработки учебных материалов и управления процессом дистанционного обучения, а также разработано достаточно много систем в области образования и психодиагностики.

В связи с этим возникает актуальная задача разработки моделей и алгоритмов для интеллектуальной системы управления качеством обучения студентов за рубежом (ИС УКО) с учетом психологических аспектов личности студентов. Под психологическими аспектами личности понимается его психотип, уровни мотивации, перспективности, знания иностранного языка, организации учебного процесса и организации свободного времени. А также исследование личности на эмоциональное состояние в процессе всего обучения.

Основными задачами компьютерной системы управления качеством обучения студентов за рубежом являются следующие:

- Психологическая поддержка: оценка внутреннего состояния обучаемого, сопереживание ему, вселение в него уверенности, стимулирование к самосовершенствованию, разрешение возникающих противоречий и проблем в процессе обучения за рубежом;
- Повышение качества обучения путем регулярного запроса об успеваемости студентов у иностранных деканатов и предоставления Бюро культуры разнообразных отчетов, связанных с эмоционально-психическим состоянием студента и его успеваемостью. Осуществление постоянного контроля над процессом обучения позволит прогнозировать критические ситуации, а также адаптировать студента к новым условиям за рубежом, что увеличит продуктивность работы студента и в итоге повысит качества обучения студентов;
- Быстрая социализация обучаемого за рубежом;
- Выдача студенту рекомендаций по результатам и проблемам обучения.
- Формирование индивидуального плана управленческих действий над студентом с учетом психотипа личности и его эмоционального состояния;

Предлагаемая система должна выполнять роль советчика, и студент будет обращаться к ней по мере возникновения вопросов, как учебного, так и социально-психологического характера.

Целью работы является разработка обоснованного подхода к построению интеллектуальной системы задачи «Управление качеством обучения студентов за рубежом» [5].

Для достижения цели в работе необходимо решить следующие задачи:

1. Разработать подход к созданию интеллектуальной системы управления качеством обучения студентов за рубежом ИС УКО на основе методов инженерных знаний с учетом психологических особенностей.

2. Разработать формальные модели ИС УКО для решения поставленной задачи на основе объектно-ориентированного анализа с использованием UML-моделирования и ORM- диаграмм для построения БД интеллектуальной системы.

3. Разработать модели представления знаний для базы знаний ИС УКО, а также алгоритмы и методику наполнения БЗ.

4. Разработать информационное, алгоритмическое, программное обеспечение для реализации предложенных алгоритмов и моделей в компьютерной среде, исследовать эффективность предложенного подхода.

Заключение

В заключении настоящей работы можно резюмировать, что анализ известных подходов к решению проблемы управления качеством обучения студентов за рубежом, как в России, так и в зарубежных государствах показал, что эти подходы не рассматривают вопрос использования эффективного аппарата учета психологических и психофизиологических особенностей студентов при создании систем поддержки обучаемых.

Проведенный анализ проблемы управления качеством обучения студентов за рубежом показал, что существующие центры психологической поддержки обучаемых в Бюро культуры являются важными факторами в процессе управления качеством обучения студентов за рубежом.

Выбранный подход к решению проблемы управления качеством обучения студентов за рубежом заключается в контроле качества обучения студентов за рубежом с учетом их психологических особенностей на основе современных информационных технологий.

Предлагаемая система должна выполнять роль советчика, и студент будет обращаться к ней по мере возникновения вопросов, как учебного, так и социально-психологического характера. Внедрение данной системы позволит реализовать принцип индивидуального подхода к каждому обучаемому, повысит качество обучения, создаст комфортные условия для достижения поставленной цели с учетом своих психологических особенностей. Анализ психологического климата в среде обучаемых позволит Бюро культуры своевременно принимать необходимые меры по предотвращению проблемных ситуаций в процессе обучения за рубежом, а также выявлять слабо подготовленных студентов. Система позволит выявлять социально-психологические проблемы, возникающие на отдельных этапах процесса обучения за рубежом, и находить способы их разрешения, что обеспечит повышение качества и эффективности обучения.

Разработанные рекомендации дадут возможность профессионалам, работающим в Бюро культуры, грамотно информировать молодых людей о типичных проблемах адаптации к иностранной культуре и учебной деятельности за рубежом и использовать эти знания при психологическом сопровождении уча-

щейся молодежи в процессе адаптации к иностранной культуре и учебной деятельности [5].

В работе сформулированы задачи и цели исследований.

Литература

1. Шолохов И. А. Научная статья «Психологические проблемы адаптации учащейся молодежи за рубежом». // Развитие личности. - М., 1999, № 2. - С. 72-85. - 0,9 п. л.
2. Шолохов И. А. *Проблемы адаптации учащейся молодежи за рубежом. // Личность на пороге XXI века*. Тезисы международной конференции психологов, педагогов, историков, философов, филологов, культурологов. - М.: Прометей, 2001. - С. 307-311. - 0,3 п. л.
3. Шолохов И. А. автореферат «Психологические особенности адаптации учащейся молодежи за рубежом» Год: 2002 к.п.н. - М.: 2002. - 235 с.
4. Сунанова, М.В. *Интеллектуальная система поддержки обучаемых с учетом психологических особенностей личности*, diss.rsl.ru 2005.
5. Ташани Н. А.- Магистерская диссертация «Интеллектуальные модели и методы управления качеством обучения студентов за рубежом». //М.: 2011.
6. Фомичева О.Е., Ташани Н. А. Научная статья «Разработка модели для решения задачи управления качеством обучения студентов за рубежом». // МГАУ. - М, 2012. - 14 с.
7. ссылка на: http://www.eurogates.ru/news/a/7010/.

UML- модели акторов и их взаимодействие с интеллектуальной системой управления качеством обучения студентов за рубежом (ИС УКО)

Ташани Н.А.

Кафедра АСУ, Московский государственный горный университет,
Москва, 119991 РФ, М. Октябрьская, Ленинский проспект д. 6,
n_tashani@hotmail.com,
WWW: http://www.msmu.ru

Аннотация. В данной статье определены требования интеллектуальной системы (ИС УКО) для решения проблемы «Управление качеством обучения студентов за рубежом», сделано обоснование выбора методологии объектно-ориентированного анализа для разработки системных моделей ИС УКО. Выделены основные акторы системы, показано их взаимодействие с ИС УКО.

Введение

Интеллектуальная система «Управление качеством обучения студентов за рубежом» (УКО) позволит реализовать принцип индивидуального взаимодействия с каждым обучаемым, что повысит качество обучения. Предлагаемая система должна содержать модели различных ситуаций и модели принятия решений в этих ситуациях. Система должна содержать результаты успеваемости по каждому обучаемому студенту, а также результаты большого количества тестов на определение психотипа личности обучаемого, психофизического и эмоционального состояния и др. (за каждый семестр). Если учесть, что количество обучаемых исчисляется сотнями и тысячами, а связь осуществляется через телекоммуникации, то очевидно, что без применения новейших компьютерных технологий, реализовать указанную систему невозможно.

Принципиально важную роль в процессе управления качеством обучения студентов за рубежом играет полная и достоверная информация о состоянии процесса обучения. Поэтому такая задача должна быть возложена на Бюро культуры в посольстве данного государства за рубежом [1].

Ключевая проблематика

Необходимость создания интеллектуальной системы управления качеством обучения студентов за рубежом (ИС УКО) обуславливается:

- Возможностью существенной экономии времени (средств) Бюро культуры, чтобы контролировать процесс обучения студентов за рубежом;
- Повышением качества процесса обучения.

Внедрение данной системы позволит вырабатывать рациональный режим управление качеством обучения за рубежом, который создаст комфортные условия для контроля процесса обучения студентов. Данная система поможет Бюро культуры выбрать наилучшее управленческое решение для повышения качества обучения студентов с учетом их психологических особенностей. Система позволит реализовать принцип индивидуального подхода к каждому обучаемому.

Эффективность данной системы зависит от того, насколько эффективен механизм работы данных компонентов. Система должна определять условия, в которых та или иная ситуация или событие могут быть предсказаны. Это позволяет управлять процессом обучения. В процессе обучения возникает целый ряд разноплановых проблем: социально-психологических, учебно-профессиональных, организационных и т.д. Знания о характеристиках каждого вида проблем обучаемого помогают в эффективной организации процесса контроля качества обучения.

ИС УКО должна, содержать модели различных ситуаций и модели принятия решений в этих ситуациях. Система должна работать с большими объемами информации (данные о студентах со всеми результатами успеваемости и тестирования психолого-физиологического состояния, рекомендации, выданные системой, база знаний, необходимая для формирования рекомендаций, разнообразные виды тестов и пр.). В связи с этим правильная форма организации данных является определяющей для производительности ИС УКО. Предлагается реализовать модели работы системы с использованием объектно-ориентированного подхода.

Для построения модели ИС УКО был выбран объектный подход. Объектная модель принципиально отличается от моделей, которые связаны с более традиционными методами структурного анализа, проектирования и программирования. Объектный подход обеспечивает ряд существенных удобств, которые другими моделями не предусматривались. Наиболее важно, что объектный подход позволяет создавать системы, которые удовлетворяют пяти признакам хорошо структурированных сложных систем.

Во-первых, объектная модель позволяет в полной мере использовать выразительные возможности объектных и объектно-ориентированных языков программирования.

Во-вторых, использование объектного подхода существенно повышает уровень унификации разработки и пригодность для повторного использования не только программ, но и проектов. Объектно-ориентированные системы часто получаются более компактными, чем их не объектно-ориентированные эквива-

ленты. А это означает не только уменьшение объема кода программ, но и удешевление проекта за счет использования предыдущих разработок, что дает выигрыш в стоимости и времени.

В-третьих, использование объектной модели приводит к построению систем на основе стабильных промежуточных описаний, что упрощает процесс внесения изменений. Это дает системе возможность развиваться постепенно и не приводит к полной ее переработке даже в случае существенных изменений исходных требований.

В-четвертых, объектная модель уменьшает риск при разработке сложных систем, так как процесс интеграции растягивается на все время разработки, а не превращается в единовременное событие. Объектный подход состоит из ряда хорошо продуманных этапов проектирования, что также уменьшает степень риска и повышает уверенность в правильности принимаемых решений.

Наконец, объектная модель ориентирована на человеческое восприятие мира.

Преимущества применения объектного подхода доказаны для задач самого разного характера. В настоящее время объектно-ориентированное проектирование - единственная методология, позволяющая справиться со сложностью, присущей очень большим системам.

В качестве "языка для описания объектной модели в данной работе был использован унифицированный язык моделирования (Unified Modeling Language - UML).

Язык UML был создан для того, чтобы участники процесса создания программного обеспечения могли строить модели визуализации системы, определения ее структуры и поведения, сборки системы и документирования решений, пронимаемых в процессе разработки.

Главной целью любого проекта по разработке программного обеспечения является работающий код. Большое количество конструкций UML имеют отношение к конструкциям популярных языков программирования, например, C++ или Java. Другие элементы UML подходят для задач моделирования БД [3].

Развитие технологии

В ходе анализа требований к интеллектуальной системе «Управление качеством обучения студентов за рубежом» были выявлены 4 актора, взаимодействующих с ИС УКО.

Акторами могут быть

- люди, непосредственно использующие систему,
- люди, ответственные за поддержание системы,
- внешнее оборудование, используемое системой,
- другие системы, взаимодействующие с данной системы.

В нашем случае акторами являются обучаемый (студент), психолог Бюро культуры, администратор Бюро культуры и служащий деканата. Все эти акторы имеют некие общие черты (наличие у каждого из них имени и пароля для входа

в систему) и взаимодействуют с системой одинаковым образом (перед началом работы они должны пройти процедуру авторизации на сервере). Это выражается отношением обобщения - существует абстрактный актор (пользователь системы), имеющий имя и пароль для входа в систему, а вышеперечисленные акторы являются его потомками [2].

На рисунке 1 показано взаимодействие акторов ИС УКО.

Рисунок 1 UML-модель «Взаимодействие акторов интеллектуальной системы управления качеством обучения студентов за рубежом»

При определении функциональных требований к системе были выделены прецеденты использования системы управление качеством обучения студентов за рубежом. **Прецедент** (use case, вариант использования) - это функциональная часть системы, необходимая для получения актором значимого для него, ощутимого и измеримого результата. Прецеденты обеспечивают функциональные требования к системе и описывают последовательности действий, которые может осуществлять система, взаимодействуя с внешними акторами. Прецеденты отображаются на диаграмме прецедентов (use case diagram, диаграмма использования) представляющей собой граф акторов, на котором отражено множество вариантов использования, ассоциации между актантами и вариантами использования.

В связи с большим количеством прецедентов разделим их по принципу ассоциации с акторами. Рассмотрим основных акторов подробнее.

Актор «Студент»:

Студент является наиболее массовой группой в системе. Студент должен периодический получать тесты с сервера для проверки его эмоционального состояния. Также он может выбрать тот тест, который он готовь отвечать. По окончании тестирования студенту будет выдано заключение. На рисунке 2 представлена UML- модель актора «Студент», как одного из главных элементов системы.

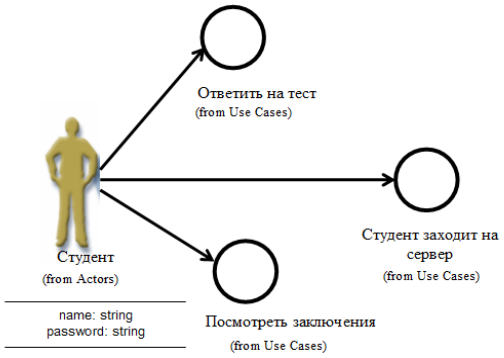


Рисунок 2 UML-модель актора «Студент»

Студент запускает приложение, после чего начинается процедура авторизации на сервере. Если имя и пароль введены, верно, то сервер выдает студенту разрешение на работу с системой (имя и пароль вводятся только при первой авторизации на сервере, затем они запоминаются приложением студента и при последующих сеансах авторизация происходит автоматически). Если приложение обнаруживает на сервере новые тесты для данного студента, оно скачивает их и сохраняет для последующего использования студентом.

Актор «Психолог Бюро культуры»:

На рисунке 3 представлена UML-модель актора «Психолог Бюро культуры». Психолог Бюро культуры следит за ходом процесса тестирования студентов и корректирует его, изменяя тесты и планы тестов.

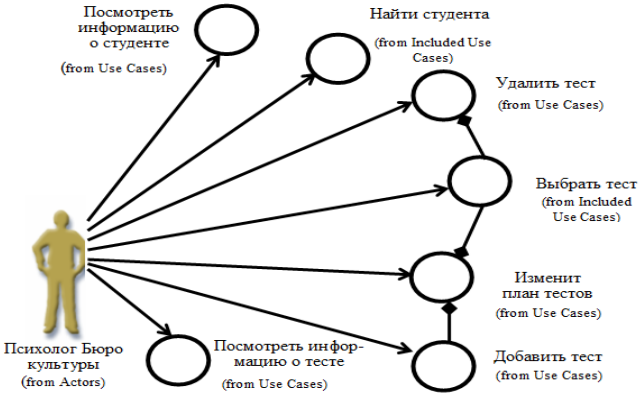


Рисунок 3 UML-модель актора «Психолог Бюро культуры »

Психолог Бюро культуры после процедуры авторизации на сервере, аналогичной процедуре авторизации студента, может выполнять следующие действия: добавлять и удалять тесты, просматривать результаты тестирования; производить поиск студентов, просматривать информацию о них и изменять план тестов студентов.

Актор «Служащий деканата»:

На рисунке 4 представлена UML-модель актора «Служащий деканата». Служащий деканата после процедуры авторизации на сервере, может выполнять следующие действия: занести информацию об успеваемости студента и изменить ее, занести информацию о состоянии студента в Вузе и его посещаемость, производить поиск студентов, а также просматривать определенную информацию о них.

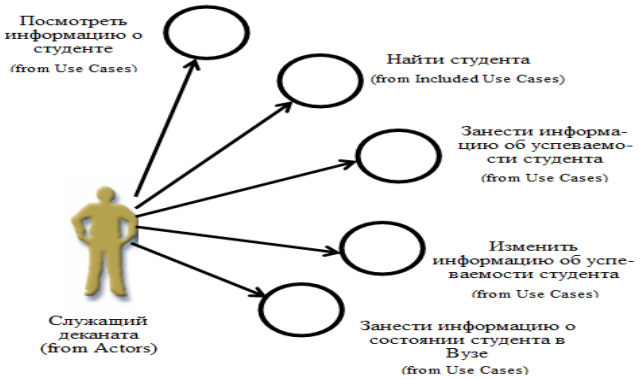


Рисунок 4 UML-модель актора «Служащий деканата»

Актор «Администратор Бюро культуры»:

Основная обязанность администратора (Сотрудник Бюро культуры по работе со студентами)- следить за работоспособностью системы и ее целостностью, а также добавлять и удалять пользователей. Для удаления администратор находит по критериям поиска (имя, пароль, права) пользователей и производит удаление. На рисунке 4 представлена UML-модель актора «Администратор Бюро культуры» (В наше системе администратор- Сотрудник Бюро культуры по работе со студен

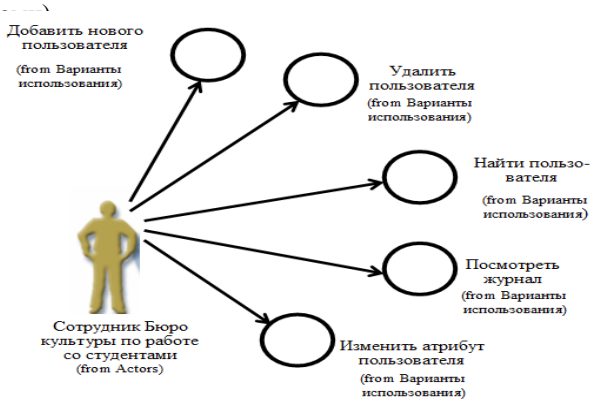


Рисунок 5 UML-модель актора
«Сотрудник Бюро культуры по работе со студентами»

Также он может изменять атрибуты пользователя, внося необходимые изменения в атрибуты пользователя, присланные сервером. Для обеспечения цело-

стности и работоспособности системы администратор может просматривать журнал событий, ведущийся на сервере.

Заключение

Для разработки системных моделей системы управления качеством обучения студентов за рубежом (ИС УКО) выбран метод объектно-ориентированного анализа. Даны основные характеристики объектного подхода, обосновано применение данной методологии к построению и показаны преимущества метода объектно-ориентированного анализа для разработки моделей ИС УКО. В соответствии методологией объектно-ориентированного анализа в качестве акторов выделены 4: студент, психолог Бюро культуры, служащий деканата и администратор Бюро культуры. На стадии моделирования построены UML-модели всех акторов системы.

Литература

1. Ташани Н. А.- Магистерская диссертация «Интеллектуальные модели и методы управления качеством обучения студентов за рубежом». // М. 2011.
2. Фомичева О.Е., Ташани Н. А., Научная статья «Разработка алгоритма и модели интеллектуальной системы управления качеством обучения студентов за рубежом». // МГАУ. - М, 2011. - 8 с.
3. Кендал Скотт. UML. Основные концепции.: Пер. С англ.- М.: Издательский дом «Вильяме», 2002. - 144с.

Разработка электронного курса «Языки и методы программирования»

Шабунин Л.В., Сидорова Е.Б.

Кафедра прикладной математики,
Чувашский государственный университет,
Чебоксары, 428015 РФ, Московский пр-т, д. 15
lvsh@mail.ru, sidorova.math@gmail.com

Аннотация. Электронная форма обучения становится всё более популярной и зрелой. В развитых странах электронное образование широко используется не только в университетах, но также и в средней школе, и в организации послевузовского образования. В работе предлагается подход к построению среды для изучения дисциплины «Языки и методы программирования». В качестве платформы выбрана система Moodle, представляющая собой свободное веб-приложение, обеспечивающее возможность создания сайтов для онлайн-обучения.

В связи со стремительным развитием компьютерных технологий обучение с помощью компьютеров все больше используется в образовании.

Одной из новых образовательных технологий является электронное образование (E-learning) — процесс обучения в электронной форме через сеть Интернет.

Электронная форма обучения становится всё более популярной и зрелой. В развитых странах электронное образование широко используется не только в университетах, но также и в средней школе, и в организации послевузовского образования.

В настоящее время электронное образование (ЭО) находит все большее применение в России. Появляются качественные российские разработки (Lektorium.tv, TeachVideo.ru, UniverTV.tv, Intuit.ru), в ВУЗах и школах разрабатывается большое количество курсов.

Учебный контент для ЭО может быть представлен как простыми статическими страницами (HTML страницы, тексты), так и сложными системами управления обучением (Learning Management Systems — LMS) и системами управления учебным контентом (Learning Content Management Systems — LCMS).

Успешное внедрение ЭО основывается не только на правильном выборе программного обеспечения, но также и на разработке методик создания учебных материалов, отвечающих российским и международным стандартам.

По нашему мнению, наилучшие результаты обучения обеспечиваются при смешанном обучении, когда традиционная подача материала дополняется технологиями электронного обучения. ЭО позволяет переложить функцию

передачи обучающимся новой информации на компьютер и основные усилия преподавателя направить на обсуждении более трудных вопросов.

Использование электронных информационных ресурсов способствуют дополнительной мотивации обучающихся, увеличивает процент направленной интенсивной самостоятельной работы студента, позволяет использовать дифференцированный подход на практических и лабораторных занятиях.

В работе предлагается подход к построению среды для изучения дисциплины «Языки и методы программирования» (в качестве изучаемого рассматривается язык Pascal). Предполагается создание следующих материалов: программа обучения (программа дисциплины, академический календарь и т.д.), информационные материалы (основная информация, учебное пособие), дополнительные информационные материалы (статьи, справки, комментарии преподавателя, ответы на часто задаваемые вопросы и т.д.), библиотека ресурсов (рекомендованная литература, списки веб-ресурсов и т.д.), диагностические материалы (тестовые и практические задания, критерии выполнения и эталонные ответы), рефлексивные материалы (анкеты), коммуникативные материалы (форумы, чаты, e-mail переписка и т.д.).

Одним из наиболее технологичных и функциональных способов организации масштабного дистанционного учебного процесса на сегодняшний день является использование Learning Management System (LMS). В качестве платформы выбрана система Moodle, представляющая собой свободное веб-приложение, обеспечивающее возможность создания сайтов для онлайн-обучения. LMS Moodle ориентирована на организацию взаимодействия между преподавателем и учениками, совместима со сборником спецификаций и стандартов SCORM.

Литература

1. *Дистанционные образовательные технологии: проектирование и реализация учебных курсов* / Лебедева М.Б., Агапонов С.В., Горюнова М.А.и др. / Под общ. ред. М.Б. Лебедевой. – СПб.: БХВ-Петербург, 2010. – 336 с.
2. Гильмутдинов А.Х., Ибрагимов Р.А., Цивильский И.В. *Электронное образование на платформе Moodle.* — Казань: Изд-во КГУ, 2008. – 169 с.
3. Андреев А.В., Андреева С.В., Доценко И.Б. *Практика электронного обучения с использованием Moodle.* – Таганрог: Изд-во ТТИ ЮФУ, 2008. – 146 с.

Абстрактные алгебры различных классов программ

Непейвода Н. Н.

Институт программных систем им. А. К. Айламазяна РАН,
152021 РФ, Ярославская обл. Переславский р-н, с. Веськово, ул. Петра I, 4а
nepejvodann@gmail.com

Аннотация Рассматриваются алгебраические структуры различных классов программ как бигруппоиды над полугруппами. Эти классы характеризуются не способами конструирования. а свойствами пространства преобразований и действий. Показано, что действия над программами и внутри программ не всегда выражаются как функции, и что свойства действий тесно связаны со свойствами вычисляемых функций (обратимые функции требуют обратимых действий, как требовал Беннетт, и так далее)

Ключевые слова: алгебраические структуры, классы программ.

Содержательная постановка задачи

Данная работа инициирована исследованиями в области перспективных направлений в компьютерной технике: обратимые вычисления. Принципиально обратимые вычисления дают возможность преодолеть предел Ландауэра тепловыделения и повысить быстродействие вычислительных устройств. Но при исследовании возможности аппаратной реализации обратимых вычислений выяснилось, что обратимость может дать выигрыш лишь в том случае, если элементная база не двоична: обратимые элементы естественно вычисляют некоторую группу преобразований.

Далее, первые опыты по построению алгоритмического языка для обратимых вычислений выявили, что он сразу же становится полностью нетрадиционным. В частности, и на аппаратном, и на программном уровне возникает операция обращения программ (пропустить сигнал в обратном направлении). Поскольку естественной математической формализацией обратимых преобразований являются группы, первые опыты в данном направлении базировались на группах.

Но еще Беннетт заметил, что для преодоления предела Ландауэра не только операции, но и система управления должна базироваться на обратимых преобразованиях. А для совместного описания команд и данных групп не хватает, поскольку уже операция обращения не является групповой. Поиск исследованных математических структур, описывающих такие преобразования, привел к отрицательным результатам. Поэтому возникает задача абстрактного алгебраического описания таких программ и она требует новых математических структур.

Далее, выяснилось, что аналогичные же проблемы возникают для частично обратимых программ и даже в обычном программировании, если мы не зашориваемся стандартными операторами языков, а рассматриваем реальные дей-

ствия, например, UNDO. Описание и исследование математической алгебраической структуры программ и действий, подходящей для данных случаев — основная цель данной работы.

Программные и алгоритмические алгебры практически одновременно начали рассматриваться в СССР и США. Глушков [1] определил алгоритмические алгебры. Маурер [2] — абстрактные программные алгебры вычислений PGA. Позже появились разные виды алгоритмических и динамических алгебр, но все они следуют двум основным направлениям, заложенным Глушковым и Маурером. Для знакомства с направлением Глушкова рекомендуем [3,4]. Для последних результатов в PGA — [5,6]. Алгебрам программ посвящены сотни работ (см., например, [8], где даны ссылки на большинство важнейших работ до середины 90-х). Практически все эти работы посвящены полным по Тьюрингу системам. Это находится в полном соответствии с доказавшим на многих примерах свою порочность, а порою уже и гибельность, духом современной науки и информатики (чем больше возможностей, тем лучше! Не дай Бог что-то пропустить, сразу некомпетентные «peer reviewers» и просто коллеги, не желающие разбираться в сути работы, закричат: “У Вас нет того-то! Он даже факториал не может запрограммировать! У него нет функции Аккермана!”).

В связи с этим невозможно отделить принципиально важные и общие аспекты от частностей тьюринговской модели вычислений. Более того, как показано, например, в [9,10], во многих случаях практически важные классы вычислений приводят к заведомо неполным по Тьюрингу системам. Поэтому в данной работе строятся алгебраические модели программ таким образом, чтобы они могли единообразно охватить все многообразие возможных программных систем: от автоматных до полных по Тьюрингу, и далее, до использующих физические устройства и физические датчики и тем самым заведомо выходящих за пределы тьюринговской вычислимости.

Далее, существующие работы по алгебрам программ сразу же привязываются к конкретным выразительным средствам существующих языков программирования (например, включая в себя объединения как образ условных операторов и замыкания как образ циклов). Это известная принципиальная ошибка: поставить средства впереди целей — диагностируемая как таковая в теории творческого мышления и в системном анализе. Мы попытаемся полностью абстрагироваться от конкретных способов программирования, тем самым опять-таки ограничивая себя и, как это неразрывно связано друг с другом в алгебре, заодно обобщая.

Но ограничивая себя, мы часто многое приобретаем. Чтобы выразить глубокое чувство, мы ограничиваем свой язык и выражаем это чувство в стихах. Отказавшись от иллюзии всезнания, интуиционисты получили возможность выражать и явно использовать незнание. Здесь мы сразу добавляем к программам операции метапрограммирования, преобразующие программы. Это не соответствует функциональному программированию, где программа используется как целостный модуль для ее применения либо построения на базе ее новой программы, не изменяющей исходную, а использующей ее как параметр.

Поскольку прослеживается, что в ближайшее время будут играть важнейшую роль инъективные и обратимые программы, эти классы программ анализируются

отдельно. Кроме того, наш подход позволяет бросить новый взгляд на нильпотентные (доказуемо завершающиеся) программы и на автоматное программирование.

Вместо подозрительного термина «множество» мы часто будем использовать понятие «тип элементов». Слово «множество» резервируется для тех случаев, когда построенная совокупность заведомо не может иметь вычислительной интерпретации, либо когда на самом деле она не построена, а доказана грязная теорема существования, использующая, например, аксиому выбора.

Внимание! Для большей естественности соотношения с алгебраическими понятиями и для того, чтобы сразу выбить читателя из привычной численной и бинарной парадигмы, мы пишем функцию после ее аргумента и используем принятую в λ-исчислении и функциональном программировании запись применений функций как списков. $(x\ f)$ — применение функции f к аргументу x. При описании конкретных отображений мы пользуемся общепринятыми в современной теоретической информатике λ-обозначениями. Это не означает, что рассматриваемые нами программные и прочие системы обязаны базироваться на λ-исчислении.

1 Общие алгебраические структуры с точки зрения программиста

Каждой программе сопоставим (частичное) отображение данных, производимое ею. В самом общем случае алгебраическая структура вычисляемых программами отображений, образует полугруппу, операцией в которой является композиция программ (или их последовательное исполнение, что в данном случае то же самое). Напомним элементарные алгебраические определения.

Определение 1. Сигнатура σ — список символов: функций (бинарные функции могут записываться между их аргументами, как операции), констант и предикатов. Каждой функции и предикату сопоставлена их арность — количество аргументов. Если в сигнатуре присутствует предикат $=$, он обязательно интерпретируется как настоящее равенство, то есть как совпадение элементов в интерпретации. Для обозначения принадлежности символа сигнатуре будем использовать обычное обозначение $\in$.

Алгебраическая система $\mathbf{S}$ сигнатуры σ — интерпретация этой сигнатуры, то есть тип данных либо множество (носитель системы) S, и функция второго типа, сопоставляющая каждому символу функции функцию $S^{\mathrm{arity}(f)} \to S$, символу константы — элемент носителя, символу предиката — булевозначную функцию $S^{\mathrm{arity}(P)} \to \{\mathrm{false}, \mathrm{true}\}$.

Определение 2. *Группоид* — тип элементов (множество) с заданной на нем бинарной операцией $\circledast$. Таким образом, сигнатура группоида — $\langle \circledast, = \rangle$ Элемент называется нейтральным и обозначается e, если $x \circledast e = e \circledast x = x$. Элемент называется нулевым и обозначается 0, если $x \circledast 0 = 0 \circledast x = 0$. Группоид называется *полугруппой*, если операция ассоциативна. Группоид коммутативен, если операция коммутативна. Группоид инъективен, или обладает левыми сокращениями,

если

$$x \circledast f = y \circledast f \Rightarrow x = y.$$

Соответственно, он обладает правыми сокращениями, если

$$z \circledast x = z \circledast y \Rightarrow x = y.$$

Группоид обладает правыми обратными, если для любого f найдется такое g, что $f \circledast g = e$. Группоид обладает левыми обратными, если для любого f найдется такое g, что $g \circledast f = e$. *Моноид* — полугруппа с единицей. *Группа* — моноид с левыми и правыми обратными. *Бигруппоид* — алгебраическая структура с двумя бинарными операциями, определенными на одном и том же множестве. *Мультигруппоид* — структура с несколькими бинарными операциями.

Напомним, что гомоморфизм алгебраических систем — отображение, сохраняющее все операции и константы данной сигнатуры и истинность всех ее предикатов. Изоморфизм — взаимно-однозначный гомоморфизм, или, что эквивалентно, такой гомоморфизм, что обратное отображение также гомоморфизм. Говорить о гомоморфизмах или изоморфизмах можно лишь для алгебраических систем с одним и тем же набором операций и констант (одной и той же сигнатурой).

Прокомментируем смысл этих понятий с точки зрения информатики. Произвольная неассоциативная операция задает на выражениях структуру упорядоченного бинарного дерева или же списков в смысле Lisp'а. В самом деле, выражение $((a \circledast (b \circledast c)) \circledast (a \circledast d))$ задает список Lisp ((a b c) a d), если интерпретировать операцию $\circledast$ как CONS, и бинарное дерево на рисунке 1. В случае коммута-

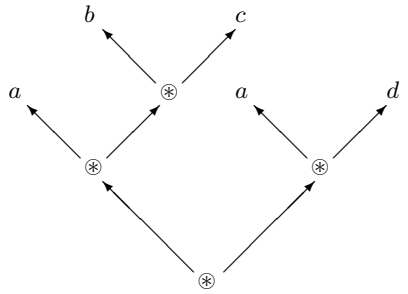


Рис. 1. $((a \circledast (b \circledast c)) \circledast (a \circledast d))$ как дерево

тивности дерево становится неупорядоченным, а прямые аналогии со списками пропадают.

Если же операция ассоциативна, все скобки можно убрать, список становится линейным. Более того, любая ассоциативная операция допускает интерпретацию как композиция функций. Это следует из классической теоремы.

Теорема 1. (Теорема о представлении для полугрупп) Каждая полугруппа изоморфна полугруппе функций с операцией композиции. [7, стр. 90]

Доказательство. Возьмем произвольную полугруппу G с операцией $*$. Рассмотрим тип всех отображений $G \to G$ вида $f_a = \lambda x.\, x \star a$, обозначим его F_G. Тогда имеем

$$((a\ f_b)\ f_c) = ((a\ \lambda x.\,(x * b))\ \lambda x.\,(x * c)) =$$
$$((a * b) * c) = (a * (b * c)) = (a\ \lambda x.\,(x * (b * c)) = (a\ f_{b*c})$$

Чтобы различить отображения, которые могут совпасть, добавим к области определения каждой функции еще одно значение name и определим $(\mathrm{name}\ f_a) = a$.

$$((\mathrm{name}\ f_a)\ f_b) = (a\ f_b) = a * b = (\mathrm{name}\ f_{a*b}).$$

□

Таким образом, говорить о том, что у нас имеется совокупность отображений, которые мы можем применять последовательно, то же самое, что говорить: у нас есть полугруппа. Тем самым мы получаем заодно и критерий, чтобы проверить, можно ли данную совокупность программных действий рассматривать как функции: ассоциативность композиции.

Пример 1. Рассмотрим совокупность действий программной системы, *пока не формализуя понятие действия*. Пусть для некоторых действий a определено правое обратное a^{-1}: отмена только что произведенного действия a. Если же предшествующее действие было не a, то a^{-1} ничего не делает. Тогда $a * a^{-1} = e$, но $a * (e * a^{-1}) = a * e = a$, $(a * e) * a^{-1} = a * a^{-1} = e$.

Аналогичными же рассуждениями можно показать, что общая операция undo (отменить предыдущее действие) также не может рассматриваться как функция.

Тем самым мы видим, что действия — понятия другой природы, чем преобразования, и нуждаются в других формализмах.

Таким образом, *абстрагирование программ до отображений, до вычисляемых ими функций, не всегда является корректным действием*. Отображение, определяемое программой, дает порою искаженное представление о программе.

Сами понятия гомоморфизма и изоморфизма систем также являются важнейшими для информатики. Изоморфизм — совершенно корректный переход от одного представления данных к другому (например, от алгебраической формулы к дереву), не теряющий и не добавляющий никакой информации, не теряющий и не добавляющий никаких свойств. формулируемых на чисто математическом языке. Поэтому математики обычно высокомерно говорят в случае изоморфных структур: «Выбор структуры не имеет никакого значения». Но, как известно, в практических задачах выбор адекватного представления данных — три четверти успеха. Это еще одно следствие того, что алгоритм далеко не полностью описывается функцией, вычисляемой программой. Важны также его сложностные характеристики и так далее.

Гомоморфизм — аналогия, переход к новому представлению, упрощающему данные за счет потери части информации, возможно, разрушающему некоторые

свойства, но сохраняющему основные преобразования данных. Это более рискованный, но часто исключительно эффективный метод действий.

Теперь вернемся к полугруппам, являющимся важным классом алгебраических систем в наших рассмотрениях. Поскольку в полугруппах операция всегда может быть интерпретирована как композиция функций, она обычно обозначается $\circ$. Насколько широка область применения полугрупп, можно видеть из следующих примеров.

Пример 2. Любая совокупность функций, замкнутая относительно композиции, является полугруппой, поскольку операция композиции функций ассоциативна. Любая совокупность отношений (то есть частичных недетерминированных функций), замкнутая по композиции, также является полугруппой. Это позволяет рассматривать как единую полугруппу отображения данных в результаты, даже если программы работают в типизированном языке. Введем, как сделано, скажем, у Митчелла [8], единый всеобъемлющий тип $\mathbb{U}$. Программы рассматриваются как функции, не определенные на исходных данных, не принадлежащих типу их параметров.

Пример 3. Строки образуют полугруппу. Программы с операцией последовательной композиции (обычно обозначаемой точкой с запятой) образуют полугруппу как синтаксические структуры, даже если их действия не являются функциональными.

Каждый конечный автомат может рассматриваться как конечная полугруппа, порожденная действиями входных символов на состояниях автомата. И каждая конечная полугруппа может рассматриваться как конечный автомат.

Полугруппа заодно порождает любопытный вариант функционального программирования. Каждый элемент a полугруппы может рассматриваться и как данное, и как функция $\lambda x.\, x \circ a$. Соответственно, полугруппа изоморфна полугруппе своих естественных гомоморфизмов, и теперь мы можем рассматривать функции как данные. Так что операция умножения может быть интерпретирована и как применение функции, и как применение функционала к функции, и так далее.

Теперь рассмотрим коммутативные неассоциативные действия (пока что лишь для полноты перебора вариантов; проблема формализации понятия действия трудна, и ниже делаются первые шаги к ее постановке и решению). Область их применения намного уже, чем неассоциативных и некоммутативных либо ассоциативных. Структура данных, порождаемая ими — неупорядоченное двоичное дерево.

Ассоциативная коммутативная операция порождает коммутативную полугруппу. С точки зрения функциональной интерпретации, коммутативность полугруппы означает, что наши действия независимы. С точки зрения вычислительной она означает, что они могут исполняться параллельно. С точки зрения логики — что их можно рассматривать как траты некоторого ресурса, автоматически «списываемого» с центрального счета, а еще лучше — с нескольких независимых счетов. Каждая операция не зависит по потраченным ресурсам от других, исполнявшихся ранее или исполняющихся параллельно.

Из-за столь важной роли полугрупп, частные понятия из теории полугрупп имеют фундаментальный смысл в информатике. Единица e означает действие «ничего не делать». Поскольку в реальности такое действие не всегда возможно (например, мы не можем остановить время), не всегда нужно предполагать, что полугруппа является моноидом. В синтаксических полугруппах e означает пустой элемент, добавление которого ничего не меняет. В автомате — это входной символ, который не меняет состояний автомата. 0 в функциональных полугруппах естественно интерпретируется как ошибка. В полугруппах отношений и частичных функций это — нигде не определенная функция и, соответственно, пустое отношение. Если в полугруппе более одного элемента, то 0 и e различны.

Если функции взаимно-однозначны, то появляются обратные элементы и полугруппа превращается в группу. Более того, любая полугруппа взаимно-однозначных функций легко пополняется до группы (добавим обратные элементы для всех и замкнем получившийся тип функций относительно операций композиции и обращения). Группа означает, что для любого действия есть полное обратное, которое может и предотвратить его, и аннулировать его результаты после выполнения

$$a \circ a^{-1} = e; \qquad a^{-1} \circ a = e. \tag{1}$$

Область применения групп также широка. Изоморфизмы алгебраической структуры на себя образуют группу. Симметричность объекта означает, что есть группа преобразований, при которой он сохраняется (например, сдвиги и повороты для мозаики, перестановки переменных для фрагмента программы или формулы и так далее...). Кристаллы описываются группами симметрий. Динамические системы — группами инвариантов и сдвигов. Примеры неисчислимы...

Не удивляйтесь, что мы не переопределили понятия гомоморфизма и изоморфизма для полугрупп и групп. В алгебраических системах гомоморфизмы сохраняют все тождества [7]. Но, естественно, в группе появляется новая операция взятия обратного, которая тоже должна сохраняться при гомоморфизмах (даже это выполнится автоматически, поскольку обратный определяется двумя тождествами (1)).

Пример 4. Рассмотрим гомоморфизм кольца целых чисел в поле остатков по большому простому модулю p: $\lambda x : \mathrm{int}.\, x \ \mathbf{mod}\ p$. По определению гомоморфизма, он сохраняет свойства операций сложения, умножения, вычитания и деления (когда оно определено) для всех натуральных чисел, меньших p (или целых, меньших по модулю $p/2$, если решим использовать большие числа как коды отрицательных, как и сделано в стандартной машинной арифметике). Да и значения этих операций тоже сохраняет, если не происходит «переполнения». А вот если воспользоваться гомоморфизмом $\lambda x : \mathrm{int}.\, x \ \mathbf{mod}\ 2^n$, то появляются делители нуля и другие гадости. Тем не менее бездумно и автоматически этот порочный гомоморфизм используется в «железе» всех современных компьютеров. Таким образом, назрела постановка вопроса: а не лучше ли система остаточных классов, уже применявшаяся ранее?

Общая алгебра программ

Из рассмотрений в предыдущем параграфе видно, что не любая совокупность программ и действий может быть описана их функциями. Таким образом, пространство программ в общем случае не является полугруппой. Но синтаксическая структура программ, которая важна при их преобразованиях, полугруппой является.

Заметим, что любой группоид порождает полугруппу действий (которая также не полностью описывает действия программ).

Определение 3. Действие последовательности $f_1, \ldots, f_n$ элементов группоида — функция

$$\varphi_{f_1;\ldots;f_n} = \lambda x.\,(\ldots((x \circledast f_1) \circledast f_2)\ldots) \circledast f_n).$$

Предложение 1. Действия любого группоида являются полугруппой относительно операции композиции функций.

Доказательство. Композицией $\varphi_{f_1;\ldots;f_n}$ и $\varphi_{g_1;\ldots;g_k}$ является $\varphi_{f_1;\ldots;f_n;g_1;\ldots;g_k}$. Композиция функций всегда ассоциативна.
□

На данном примере видно, почему мы предпочли несколько необычный порядок записи функции и ее аргументов: в сложных случаях он намного естественнее.

Рассмотрим теперь с содержательной точки зрения якобы совершенно ясное понятие: применение программы к аргументу $x \star f$. Эта операция для большинства программных систем неассоциативна, как было показано выше, она согласована с композицией, но не сводится к ней. Поскольку любой элемент у нас может играть роль и функции, и данного ($f \star x$ столь же законная операция), алгебра применений заодно дает возможность описать преобразования программ.

Мы приходим к алгебре, являющейся бигруппоидом с двумя операциями. Одна из них полугрупповая операция композиции $\circ$, вторая — неассоциативная операция применения программы $\star$. Получающуюся полугруппу по композиции будем обозначать большой жирной буквой, а всю алгебру — прозрачной. Они связаны следующими тождествами.

$$((x \star f) \star g) = (x \star (f \circ g)) \tag{2}$$

$$(0 \star x) = (x \star 0) = 0 \tag{3}$$

$$(x \star e) = x \tag{4}$$

Как бы мы ни определили понятия и композиции и применения, они должны быть связаны вышеприведенными тождествами. В остальном мы, как и всегда при алгебраическом подходе, свободны в интерпретации. Во всех известных автору программных, человеко-машинных и аппаратных системах эти свойства выполнены.

Такие алгебры назовем GAPS (Generic algebraic program structure).

Первое тождество выполняется всегда, оно является определением последовательного исполнения. Два последних выполнены лишь в том случае, если синтаксическая полугруппа содержит 0 или e. Если хоть одна из этих констант присутствует, она сохраняет свой смысл и для операции $\star$.

Напомним понятие обогащения алгебраической системы. Пусть дана сигнатура σ_1, расширяющая сигнатуру σ. Алгебра $\mathbb{A}_1$ сигнатуры σ_1 является обогащением алгебры $\mathbb{A}$ сигнатуры σ, если носители, константы и операции из σ остаются без изменения.

Пусть имеется некоторый гомоморфизм $\alpha : \mathbb{G} \to (G \to G)$ нашей полугруппы $\mathbb{G}$ в полугруппу отображений ее носителя, такой, что $(e\ \alpha) = \lambda x.\, x$, $(0\ \alpha) = \lambda x.\, 0$. С точки зрения программиста, этот гомоморфизм задает интерпретатор или транслятор наших программ, перерабатывающий запись программы в исполняемый модуль. Синтаксически различные программы могут порождать один и тот же исполняемый модуль, так что считать это отображение α изоморфизмом нельзя.

Предложение 2. Каждая полугруппа $\mathbb{G}$ может быть обогащена до GAPS таким образом, что $x \star f = (x\ (f\ \alpha))$.

Доказательство. Проверим тождества для GAPS. Второе и третье выполнены по построению. Убедимся в том, что и первое выполнено.

$$((x \star f) \star g) = ((x\ (f\ \alpha))\ (g\ \alpha)) = (x\ ((f\ \alpha) \circ (g\ \alpha))) =$$
$$(x\ (f \circ g\ \alpha)) = (x \star (f \circ g))$$

Имеется еще одна маленькая тонкость: всегда ли такой гомоморфизм существует? Достаточно рассмотреть тривиальный пример: определим $(x\ \alpha) = \lambda x.\, 0$ для всех делителей нуля, то есть таких x, для которых имеется $y \neq 0$, такое, что $x \circ y = 0$, а для всех остальных: $(x\ \alpha) = \lambda x.\, x$. Второй тривиальный вариант — просто определить $x \star y = x \circ y$.
□

Предложение 3. В любой GAPS для любого действия группоида φ имеется элемент α, такой, что $(x \star \alpha) = (x\ \varphi)$.

Доказательство. Рассмотрим произвольное действие $f_1, \ \ldots, \ f_n$. Применяя (2) и ассоциативность умножения в полугруппе, получаем:

$$(\ldots((x \circledast f_1) \circledast f_2) \ldots) \circledast f_n) = (x \circledast f_1 \circ f_2 \cdots \circ f_n).$$

Таким образом, в GAPS любая последовательность действий программ выразима действием одной программы.
□

Пример 5. Заметим, что некоторые «естественные» предположения о GAPS разрушают систему. Например, почему нет тождества $(e \star f) = f$? Потому что тогда обе операции совпадают:

$$(x \star y) = ((e \star x) \star y) = (e \star (x \circ y)) = x \circ y.$$

Рассмотрим возникшую ситуацию содержательно: почему это 0 может оставаться нулем, а единица при действиях оставаться сама собой не всегда может? 0 с точки зрения интерпретации — ошибка. Дальше либо с нею внутри алгебры делать уже нечего: система вылетела. e может быть интерпретирована как программа, ничего не делающая и не тратящая ресурсов (например, пустая). Но преобразователь программ вполне может выдать какой-то сложный код, исходя из пустых данных. Смотри дальше пример 10.

Теперь рассмотрим важный вопрос: когда можно некоторую систему преобразований программ соединить с данной синтаксической полугруппой программ, тем самым образовав единый язык метапрограммирования? Эта задача естественно разбивается на три подпроблемы: соединение языка с системой преобразований без изменения языка (то есть полугруппы); сохранение некоторого подъязыка (подполугруппы) и превращение ранее неинтерпретированных или не нужных нам более синтаксических конструкций в действия преобразователей; пополнение существующего языка до метаязыка путем расширения полугруппы без изменения исходной.

Здесь начинает действовать еще одно важное алгебраическое понятие: полугруппа, порожденная совокупностью функций (или элементов полугруппы, или отношений, что с точки зрения математика в данном случае одно и то же). Неявно оно уже было использовано в определении действий группоида.

Определение 4. Полугруппа $\mathbb{F}$, порожденная совокупностью функций $\mathbf{F}$ — минимальный тип функций, включающий $\mathbf{F}$ и замкнутый относительно композиций, то есть

$$\forall f, g\,(f \in \mathbb{F} \& g \in \mathbb{F} \supset f \circ g \in \mathbb{F})$$

Содержательно это означает, что $\mathbb{F}$ состоит из всех функций, представимых как $f_1 \circ f_2 \cdots \circ f_n$, где $f_1, f_2, \ldots, f_n$ — элементы (не обязательно различные) $\mathbf{F}$.

Определение 5. Пусть задана некоторая совокупность действий $\mathbb{A}$ как функции на носителе полугруппы $\mathbb{G}$. Она *консервативна*, если имеется GAPS $\mathbb{AG}$, которая является обогащением $\mathbb{G}$, такая, что каждое $\varphi \in \mathbb{A}$ представляется в ней как действие группоида: $(x \star \alpha) = (x\ \varphi)$ для некоторого α. Она *консервативна над подполугруппой* $\mathbb{G}_0$, если она консервативна и в $\mathbb{AG}$ $a \star b = a \circ b$ для всех элементов $\mathbb{G}_0$. Она *слабо допустима*, если имеется полугруппа $\mathbb{G}_1$, подполугруппой которой является $\mathbb{G}$, и функции из $\mathbb{A}$ могут быть продолжены на $\mathbb{G}_1$ таким образом, что полученная $\mathbb{A}_1$ консервативна над $\mathbb{G}$. Она *допустима*, если она слабо допустима и в $\mathbb{A}_1$ остаются истинными все факты вида $f_1 \circ \cdots \circ f_n = g_1 \circ \cdots \circ g_k$, истинные в $\mathbb{A}$.

Очевидно, что самое сильное из этих понятий — консервативность над полугруппой. Далее идут консервативность, допустимость, слабая допустимость. Рассмотрим простейшие условия возможности всех четырех обогащений и расширений.

Предложение 4. Совокупность действий $\mathbb{A}$ консервативна тогда и только тогда, когда ее замыкание изоморфно подполугруппе $\mathbb{G}$.

Доказательство. По предложению 3, если обогащение удалось осуществить, то любой элемент полугруппы, порожденной $\mathbb{A}$, представляется как действие некоторого элемента $\mathbb{G}$. Значит, мы изоморфно вложили замыкание $\mathbb{A}$ в $\mathbb{G}$.

Обратно, если замыкание $\mathbb{A}$ вкладывается в $\mathbb{G}$, то просто сопоставим каждому действию из $\mathbb{A}$ соответствующий ему при изоморфизме элемент $\mathbb{G}$.
□

Заметим, что важнейшим фактором является вложение структуры полугруппы действий, а не самих действий.

Пример 6. Если у нас имеется всего одно действие сильной инверсии программ $\mathbf{M}$, обладающее свойством $((a\ \mathbf{M})\ \mathbf{M}) = a$, то обогатить полугруппу программ этим действием удается тогда и только тогда, когда у нас имеется элемент порядка 2, то есть такая программа, что $f \neq e \& f \circ f = e$. Как эта f действует на данные, нам все равно. Важно, что мы можем доопределить ее как операцию метапрограммирования без разрушения общей структуры вычислений.

Предложение 5. Совокупность действий $\mathbb{A}$ консервативна над $\mathbb{G}_0$ тогда и только тогда, когда имеется такое изоморфное вложение ψ ее замыкания в $\mathbb{G}$, что для каждого действия f, такого, что $(f\ \psi) \in \mathbb{G}_0$, выполнено $(a\ f) = (a \circ (f\ \psi))$.

Доказательство. Если такое обогащение удалось осуществить, то действие, попавшее в $\mathbb{G}_0$, обязано по условию сохранения подполугруппы, удовлетворять $(a\ f) = (a \circ (f\ \psi))$.

Обратно, если такой изоморфизм имеется, то просто сопоставим каждому действию из $\mathbb{A}$ соответствующий ему при изоморфизме элемент $\mathbb{G}$.
□

Пример 7. Например, пользуясь этим критерием, можно проверить возможность обогащения языка до языка метавычислений, рассматривая строки как программы, а подъязык, работающий с числами, оставляя неизменным.

Теорема 2. Любая совокупность действий $\mathbb{A}$ допустима над любой $\mathbb{G}$.

Для построений необходимы понятия свободной алгебраической системы, фактов, матрицы системы, тождеств и факторизации (приведенные, в частности, в книге Мальцева [7]).

Определение 6. *Свободная алгебраическая система* сигнатуры σ над совокупностью констант $\mathbb{C}$ — система всех выражений, которые могут быть построены из констант σ и $\mathbb{C}$ при помощи операций σ. Выражения в свободной системе считаются равными, если они совпали текстуально.

Конгруэнтность — отношение эквивалентности, такое, что функции и все предикаты, кроме равенства, сохраняют его, то есть

$$\forall x_1, \ldots, x_n, y_1, \ldots, y_n\, (x_1 \sim y_1 \& \cdots \& x_n \sim y_n \supset (x_1\ \ldots\ x_n\ f) \sim (y_1\ \ldots\ y_n\ f))$$
$$\forall x_1, \ldots, x_n, y_1, \ldots, y_n\, (x_1 \sim y_1 \& \cdots \& x_n \sim y_n \supset (P(x_1, \ldots, x_n) \equiv P(y_1, \ldots, y_n)))$$

Фактор-система $\Sigma/\!\sim$ системы Σ по конгруэнтности $\sim$ — система, носителем которой является множество классов эквивалентности, то есть множеств $\hat{a} =$

$\{x \mid a \sim x\}$. Функции и предикаты переносятся на фактор-систему следующим образом:

$$(\hat{x_1}\ \ldots\ \hat{x_n}\ f) = \{z \mid \exists y_1, \ldots, y_n\, (x_1 \sim y_1 \& \cdots \& x_n \sim y_n \& z = (y_1\ y_n\ f))\};$$
$$P(\hat{x_1}, \ldots, \hat{x_n}) \equiv \exists y_1, \ldots, y_n\, (x_1 \sim y_1 \& \cdots \& x_n \sim y_n \& P(y_1, \ldots, y_n))\}$$

Определение 7. [7, стр. 172] *Факт* — выражение вида $P(t_1, \ldots, t_n)$ либо $\neg P(t_1, \ldots, t_n)$, не содержащее переменных, в частности, $t = u$ или $t \neq u$. Факт *позитивный*, если он не начинается с отрицания. *Диаграмма системы* — совокупность всех истинных в ней фактов, при условии, что сигнатура пополнена константами для всех элементов носителя системы.

Тождество — формула вида $\forall x_k, \ldots, x_k\, P(t_1, \ldots, t_n)$, где x_i — все свободные переменные элементарной формулы.

Понятие свободной системы близко каждому программисту. Фактор-система образуется, когда у одного и того же объекта много представлений. Один из важнейших для информатики видов фактор-систем — системы, образующиеся в результате доказательства тождеств и фактов.

Предложение 6. Если Th — некоторая теория, то все выводимые в ней факты типа $t = u$ задают отношение конгруэнтности на свободной алгебре, построенной на константах произвольной модели данной теории.

Очевидно. $\square$

Систему, образующуюся в результате факторизации по такому отношению конгруэнтности, назовем Σ/Th.

Доказательство теоремы 2. Построим соответствующее расширение и обогащение. Возьмем свободную алгебру $\mathbf{GA}$ на базе наших двух операций, всех элементов полугруппы $\mathbb{G}$ и действий $\mathbb{A}$. Зададим теорию $\mathrm{Th}\{\mathbb{GA}\}$ со следующими аксиомами:

1. Диаграмма полугруппы $\mathbb{G}$.
2. Все заданные факты про $\mathbb{A}$, переведенные в форму тождеств
$$\forall x\, (x \star t) = (x \star u).$$
3. $\forall x, y, z \quad x \circ (y \circ z) = (x \circ y) \circ z$.
4. $\forall x, f, g \quad ((x \star f) \star g) = (x \star (f \circ g))$.
5. $\forall x\, x \star 0 = 0$, $\forall x\, 0 \star x = 0$, $\forall x\, x \circ 0 = 0$, $\forall x\, 0 \circ x = 0$, если 0 присутствует в нашей полугруппе.
6. $\forall x\, x \star e = x$, $\forall x\, e \star x = x$, $\forall x\, x \circ e = x$, если e присутствует в нашей полугруппе.

Если эта теория непротиворечива, то положим $\mathbb{GA} \triangleq \mathbf{GA}/\mathrm{Th}\{\mathbb{GA}\}$. Эта алгебра будет искомой.

Теперь покажем, что $\mathrm{Th}\{\mathbb{GA}\}$ всегда непротиворечива. Поскольку заданные нами факты для $\mathbb{A}$ являются переводом на язык нашей алгебры свойств полугруппы функций, они совместимы с аксиомой ассоциативности операции $\star$. Тем

самым они совместимы с аксиомой (2) и с исходной полугруппой и не задают никаких новых равенств на ней.
□

Следовательно, понятие слабой допустимости является излишним. Оно введено, чтобы лучше структурировать определения. Но наше доказательство на самом деле позволяет дать полный общий теоретический критерий пополнимости некоторой системы операций над программами с заданными свойствами операций до GAPS над уже имеющейся у нас программной системой.

Заменим в построении теории $\mathrm{Th}\{\mathbb{GA}\}$ из доказательства 2 пункт 2 на произвольную совокупность аксиом Th_0 в сигнатуре GAPS$\cup\{\mathbb{G}, \mathbb{A}\}$, пополненной константами для всех элементов полугруппы и всех заданных действий.

Теорема 3. Совокупность действий $\mathbb{A}$, описываемая теорией Th_0, допустима над $\mathbb{G}$ в том и только в том случае, когда теория $\mathrm{Th}_0\{\mathbb{GA}\}$ непротиворечива.

Доказательство Согласно [7, стр. 172], если теория содержит матрицу алгебраической системы Σ, эта система изоморфно вложима в любую ее модель.
□

Принципиально данное решение полное. Но на практике оно, конечно же, редко применимо. Однако есть один практический критерий, который следует из данной теоремы.

Практическое следствие. *Если система преобразований программ разрушает свойства каких-то уже имеющихся программ или заставляет отождествить разные программы, она принципиально некорректна*.

Но модель конечной теории не всегда вычислима, хотя процесс ее построения для разрешимых теорий не столь уж сложен: трансфинитная индукция до ω^2 и получающаяся модель определима формулой класса $\forall\exists$, то есть арифметической формулой вида $\forall x \exists y\, R(x, y)$, где $R(x, y)$ — разрешимый предикат. Соответственно, алгебра $\mathbf{GA}/\mathrm{Th}\{\mathbb{GA}\}$ теперь уже не всегда является искомой GAPS. Максимальное обобщение метода, данного в теореме 2, когда носитель разрешим, а отношение тождества максимум перечислимо, возможно для хорновских теорий (или теорий из квазитождеств, как их еще раньше открыл и назвал Мальцев).

Определение 8. Квазитождество — формула вида

$$\forall x_1, \ldots, x_k\, (Q_1 \& \cdots \& Q_n \supset P),$$

где x_i — все входящие в нее переменные, и все Q_i, P — предикаты.

Квазитождества сохраняются при прямых произведениях и даже при фильтрованных произведениях (что такое последнее — пока что можно для программистов пояснить чисто содержательно: на множестве индексов перемножаемых систем вводится понятие большинства[1], формулы, истинные на большинстве членов произведения, считаются истинными).

[1] Скажем: «Один человек вместе с Богом составляют большинство».

Пусть класс алгебраических систем Σ описан квазитождествами QTh. Пусть задана некоторая совокупность исходных элементов V. Среди алгебр $\Sigma\{V\}$, которые получаются факторизацией свободной алгебры над V, есть хоть одна модель QTh (а именно, единичная, получающаяся отождествлением всех выражений). Возникает вопрос, можно ли среди моделей выделить какую-то «стандартную»? Таковой считается инициальная модель QTh, из которой любая другая модель из $\Sigma\{V\}$, получается факторизацией, то есть отождествлением по некоторой конгруэнции. Она строится следующим образом: отождествляются те и только те термы, равенство которых доказуемо в QTh. Следовательно, недоказуемые равенства считаются ложными. В нашем случае инициальная модель — это $\mathbf{GA}/\mathrm{Th}\{\mathbb{GA}\}$.

Выход за пределы квазитождеств моментально разрушает алгоритмичность либо сохраняемость. Но и квазитождества уже не обязательно сохраняются при гомоморфизмах. Рассмотрим простейшие примеры.

Пример 8. Рассмотрим мультипликативную полугруппу положительных целых чисел $\mathbb{Z}^+$. В ней имеет место свойство инъективности (разнозначности, сокращения) для умножения:

$$\forall x, y, z\,(x \cdot z = y \cdot z \supset x = y).$$

Если отобразить его в полугруппу остатков по модулю n, то в ней появится 0 (числа, делящиеся на n), и свойство инъективности уже не будет выполнено.

Теперь рассмотрим теорию из единственной аксиомы $a = b \vee a = c$. В ней не выводимы ни $a = b$, ни $a = c$, и поэтому построенная инициальная модель не будет моделью данной теории.

На самом деле, поскольку высокоуровневые понятия и сложные логические формулы в программировании практически всегда играют важную и полезную роль призраков, облегчающих понимание и преобразование программы, но как минимум бесполезных в исполняемом тексте, можно предложить следующую методику «приближенного моделирования».

Поскольку реальными понятиями в программе могут быть факты или в крайнем случае квазитождества, будем рассматривать теорию как способ порождения реальных формул. Th_Q — теория, состоящая из всех квазитождеств, доказуемых в Th. Как правило, сама Th просто и красиво формулируется, доказательства в ней кратки и естественны, а ее «ползучий» фрагмент сложен, и доказывать в нем отвратительно, если вообще возможно. В таком случае Th естественно играет роль идеального понятия для Th_Q. Д. Гильберт в своей программе подчеркнул важность идеальных понятий для практических задач и невозможность решить сложные задачи без использования подходящих сильных идеальных понятий. Этот вывод Гильберта был подтвержден содержательным анализом доказательств Д. Пойа (парадокс изобретателя) и формальными результатами Хао Вана, Оревкова и всего направления reverse mathematics, вскрывающего идеальные понятия, стоящие за «реальными» теоремами.

Поэтому будем во многих случаях считать, что нам на самом деле нужна Th_Q и строить алгебру именно для нее.

Пример 9. (Постановка задачи принадлежит С. Мешвелиани) Поскольку λ-исчисление лежит в основе современной математический теории полных по Тьюрингу программных систем, построим модель стратифицированного λ-исчисления как GAPS. Напомним, что λ-выражение называется стратифицированным. если в нем можно корректно расставить типы всех переменных и констант.

Две естественных точки зрения: рассматривать полугруппу как множество λ-термов в нормальной форме или как множество отображений нормальных форм t в нормальные формы $(f\ t)$, происходящих при конверсии выражений. Первый подход не дает возможности естественно определить композицию, поскольку нет операции соединения λ-выражений. При втором композиция естественно определяется, но теряется суть программной системы: разные выражения могут производить одну и ту же функцию преобразования.

Чтобы приблизиться к GAPS, перейдем к эквивалентному представлению λ-термов как комбинаторной логики. Комбинаторная логика естественно и взаимно-однозначно формулируется как группоид действий: применение терма к терму $(t\ r)$ записывается в наших обозначениях $(r \star t)$. Определяющие равенства для констант $\mathbf{K}$ и $\mathbf{S}$ переписываются в форме

$$(x \star (y \star \mathbf{K})) = y; \tag{5}$$

$$(x \star (y \star (z \star \mathbf{S})) = ((x \star y) \star (x \star z)). \tag{6}$$

В комбинаторной алгебре определяется комбинатор композиции $\mathbf{B}$:

$$\mathbf{B} \triangleq (\mathbf{K} \star ((\mathbf{S} \star \mathbf{K}) \star \mathbf{S})). \tag{7}$$

Преобразуя согласно определяющим равенствам для $\mathbf{K}$ и $\mathbf{S}$, получаем:

$$\begin{gathered}(x \star (f \star (g \star (\mathbf{K} \star ((\mathbf{S} \star \mathbf{K}) \star \mathbf{S})))) = \\ (x \star (f \star ((g \star \mathbf{K}) \star (f \star (\mathbf{S} \star \mathbf{K}))))) = \\ (x \star (f \star ((g \star \mathbf{K}) \star \mathbf{S}))) = \\ ((x \star f) \star (x \star (g \star \mathbf{K}))) = \\ ((x \star f) \star g).\end{gathered}$$

Таким образом, $(f \star (g \star \mathbf{B}))$ определяет последовательное применение (композицию) f и g.

Теперь можно было перейти к другому известному базису комбинаторной логики, включающему $\mathbf{B}$, но мы проиллюстрируем другой путь, показывающий способ перехода от теории или системы, в которой композиции как корректно определенной с нашей точки зрения операции не было, к теории или системе, с практической и теоретической точки зрения эквивалентной, но представимой как алгебра GAPS.

Прежде всего, добавим $\mathbf{B}$ как новую константу с определяющим равенством (7). Такое расширение теории с привычной точки зрения эквивалентно, поскольку добавили определимое понятие, но оно избавляет от нежелательного эффекта: как бы мы ни определили $\mathbf{B}$, $(f \star (g \star \mathbf{B}))$ не будет нормальной формой, поскольку последняя константа в определении заведомо применится (даже если она $\mathbf{S}$,

требующая три аргумента, то, поскольку определение не сводится к самой $\mathbf{S}$, третий аргумент будет взят из текста определения).

Далее, два выражения

$$(f \star ((g \star (h \star \mathbf{B})) \star \mathbf{B})) \qquad ((f \star (g \star \mathbf{B})) \star (h \star \mathbf{B}))$$

функционально и вычислительно эквивалентны, но находятся в нормальной форме и поэтому не могут быть преобразованы друг в друга. Равенство между ними также постулируем как тождество.

$$(f \star ((g \star (h \star \mathbf{B})) \star \mathbf{B})) = ((f \star (g \star \mathbf{B})) \star (h \star \mathbf{B})) \tag{8}$$

Это тождество может быть эффективно реализовано как одностороннее и заведомо фундированное правило переписывания левой части в правую. Таким образом, носитель алгебры у нас множество нормальных форм выражений в комбинаторной логике, пополненной комбинатором $\mathbf{B}$, нормализованных применением правила, соответствующего равенству (8). Композицией двух выражений считается $(f \star (g \star \mathbf{B}))$.

Это определение корректно, поскольку стратифицированные λ-выражения всегда имеют нормальную форму, и композиция стратифицированных выражений стратифицирована.

Построение завершено.

Таким образом, мы показали, что GAPS позволяет адекватно описать одну из самых общих теоретических концепций современной информатики.

Если перейти к бестиповому λ-исчислению, то предложенная конструкция так же хорошо работает, если значением выражения, не имеющего нормальной формы, считать 0.

Заметим, что в стратифицированной GAPS нуля нет.

В построенных нами GAPS нет единицы. Она может быть добавлена как новая константа с правилами конверсии

$$(t \star e) = t, \quad (f \star (e \star \mathbf{B})) = f, \quad (e \star (f \star \mathbf{B})) = f.$$

Тем самым на этапе принципиальной проработки теоретических концепций представление «конкретных» и «практических» алгоритмических языков становится излишним: они теоретически сводимы к λ-исчислению.

Пример 10. Рассмотрим еще один пример превращения программной системы в абстрактную алгебру. Пусть у нас есть полный по Тьюрингу алгоритмический язык, в котором программы и данные являются строками символов в одном и том же алфавите. Пусть соединение строк дает композицию функций программ. Пусть пустая строка — ничего не делающая программа. Примером такого языка может служить Brainfuck. Добавим константы 0 и e. Первая из них является значением вылетающей или зацикливающейся программы, вторая — пустая строка.

Композиция программ определяется как соединение строк. А вот действие строки $a \star f$ определим следующим образом. Вычисляется строка, являющаяся значением f на a. Если f выдала ошибку или получившаяся строка синтаксически некорректна как программа, значением является 0. В противном случае получившаяся программа.

В этом примере видно, что результатом действия над пустой программой может быть любая другая программа. А вот результат действия пустой программы — та же программа, что была.

Теперь пара замечаний, показывающих некоторое ехидство нашего аппарата.

Если в полугруппе есть единица, то универсальная функция, описываемая в нашей алгебре тождеством

$$(x \star (f \star \mathbf{U})) = (x \star f),$$

тривиальна: $\mathbf{U} = e$. Зато специализатор программ, описываемый как

$$(f \star (x \star \mathbf{PE})) = (x \star f),$$

уже нетривиален почти всегда. И на этом тождестве видно, что специализатор является одним из видов универсальной функции.

Неподвижная точка программ, определяемая как

$$((f \star \mathbf{Y}) \star f) = (f \star \mathbf{Y}),$$

тривиальна, если в полугруппе есть 0: она просто тождественно равна 0.

Еще об одном. В принципе GAPS дают общее понятие автомата высших типов, значительно более абстрактное и более общее, чем то, которое дано, например, в [11]. Но непосредственно реализуемы как автоматы здесь могут быть лишь конечные GAPS.

Алгебра полностью обратимых программ (AFIP)

Как уже говорилось, обратимые действия составляют группу. Группа сохраняет основное свойство полугруппы: возможность рассматривать любое данное как функцию. В ней имеется дополнительная операция взятия обратного действия (которая может применяться как до, так и после исходного действия): a^{-1}. Таким образом, в данном случае GAPS основан на группе.

Преобразования программ также должны быть в этой обстановке обратимы, и мы получаем характеристику алгебр полностью обратимых программ как GAPS с дополнительной константой M (обращение программ) и дополнительными аксиомами:

$$x \circ (x \star M) = e \qquad (x \star M) \circ x = e \tag{9}$$

Из этих аксиом автоматически следует биективность операции применения программ $\star$:

Предложение 7. Отображение $\lambda x.\,(x \star f)$ биективно для любого f.

Доказательство. Инъективность. Пусть $(x \star f) = (y \star f)$. Тогда

$$((x \star f) \star (f \star M)) = ((y \star f) \star (f \star M))$$
$$(x \star (f \circ (f \star M))) = ((y \star (f \circ (f \star M)))$$
$$(x \star e) = (y \star e) = x = y$$

Сюръективность. Найдем для каждого x и f такое y, что $(y \star f) = y$.

$$((x \star f) \star (f \star M)) = ((y \star f) \star (f \star M))$$
$$((x \star (f \star M)) \star f) = (x \star ((f \star M) \circ f)) = (x \star e) = x.$$

□

Наличие этих двух аксиом автоматически превращает полугруппу в группу: $(x \star M)$ оказывается обратным элементом для x. Оно же окончательно различает операции и функции: в большинстве групп нет такого элемента, который при умножении на все остальные давал бы обратные им.

Предложение 8. Каждая группа $\mathbb{G}$, в которой имеется элемент порядка 2 (то есть такой x_0, что $x_0 \circ x_0 = e$) может быть обогащена до AFIP.

Доказательство. Пусть G — носитель группы. Зададим для M в качестве значения какой-нибудь элемент порядка 2. Пусть GG — группа всех взаимно однозначных отображений G. Пусть φ — гомоморфизм $\mathbf{G}$ в GG, такой, что $(M\ \varphi) = \lambda x.\, x^{-1}$. Определив $(x \star f) = (x(f\varphi))$, получаем искомую алгебру.

Возникает второй вопрос: а обязательно ли существует такой гомоморфизм? Простейшим из них является следующий: $(M\ \varphi) = \lambda x.\, x^{-1}$, а все остальные элементы отображаются в $\lambda x.\, x$, то есть в единицу.

□

На самом деле этот тривиальный гомоморфизм задает практически важное наблюдение: оператор обращения программ M можно добавить в значительной степени независимо от других преобразований программ, и во многих случаях многие из преобразователей можно пока что оставлять тождественными для дальнейших расширений системы.

Предложение 9. Группа $\mathbb{G}$ обогащается до AFIP с сохранением групповой операции $\circ = \star$ тогда и только тогда, когда она коммутативна и каждый элемент в ней имеет порядок 2.

Доказательство. Прежде всего, покажем, что в этом случае $a \circ a = e$ для любого элемента. Для этого возьмем его обратный.

$$(a \circ a)^{-1} = ((a \circ a) \star M) = ((a \circ a) \circ M) = (a \circ (a \circ M)) = a \circ a^{-1} = e.$$

Таким образом, в группе, являющейся также AFIP, $\forall x\, x = x^{-1}$ и $M = e$. Далее, известно из теории групп (см. например [13,14]), что любая группа порядка 2 коммутативна. Покажем это с помощью M.

$$a \circ b = ((a \circ b) \star M) = (a \circ b)^{-1} = b^{-1} \circ a^{-1} = b \circ a.$$

□

Из известных результатов теории групп следует утверждение:

Предложение 10. Группа $\mathbb{G}$ обогащается до AFIP с сохранением групповой операции $\circ = \star$ тогда и только тогда, когда она изоморфна полному прямому произведению некоторого числа групп вычетов по модулю 2 $\mathbb{Z}_2$ либо, в бесконечном случае, его подгруппе, в которой множества индексов, где стоят нули перемножаемых групп, принадлежит некоторому фильтру.

К несчастью, в данном случае вычисления в бинарной памяти оказываются слишком ограниченными: мы не имеем права организовывать взаимодействие разных битов памяти между собой.

Здесь мы столкнулись с еще одной важной конструкцией алгебры, полезной программисту. Если алгебраическая структура представлена как прямое произведение, то действия в разных членах этого произведения могут быть как угодно распараллелены или переставлены между собой, поскольку работа внутри каждого из элементов прямого произведения локальна.

Заметим также, что предложение 8 является следствием и конкретизацией для нашего случая общей теоремы 3. Можно было бы дать его абстрактное и косвенное доказательство через эту теорему, но мы предпочли прямое построение, показывающее индуцируемые конструкции.

Критерий обогащаемости группы до программной алгебры может быть расширен до критерия обогащаемости с сохранением выделенной подгруппы.

Предложение 11. Пусть группа $\mathbb{G}$ не является группой порядка 2. Пусть G_0 — ее подгруппа, не совпадающая с $\{e\}$ и не являющаяся группой порядка 2 (хотя она может содержать элементы порядка 2). Существует алгебра AFIP $\mathbb{G}_0$ такая, что для всех $f \in G_0$ $(x \star f) = (x \circ f)$ тогда и только тогда, когда есть элемент порядка 2, не входящий в $\mathbb{G}_0$.

Доказательство. Возьмем некоторое $m \in G \setminus G_0$, имеющее порядок 2. Пусть $(g\ \varphi) = \lambda x.\,(x \circ g)$ для $g \in G_0$, Расширим его, положив $m\ \varphi) = \lambda x.\,x^{-1}$. Если x принадлежит замыканию $G_0 \cup \{m\}$ относительно групповых операций $\circ$ и $^{-1}$, то $x\ \varphi$ вычисляется как композиция определенных ранее функций и их обращений.

Если остался x_1, не принадлежащий замыканию $G_0 \cup \{m\}$, то выберем $x_1\ \varphi$ произвольным взаимно-однозначным отображением, сохраняющим e, и продолжим этот процесс по ординалам, пока не исчерпается G.

Необходимость условий на группу следует из предложений 9 и 8.

□

Заметим еще одну коварную особенность алгебраических определений. Маленькая «эквивалентная» переформулировка может полностью изменить свойства алгебраической системы. Например, если заменим равенства композиций программы и ее обращения единице на равенства их вычислительного эффекта тождественному преобразованию

$$((x \star f) \star (f \star M)) = x \qquad ((x \star (f \star M)) \star f) = x, \tag{10}$$

то доказательство биективности всех действий сохранится, а вот условия расширяемости (предложения 8 и 11) станут более «комфортными»: необходимо и достаточно наличие хотя бы одного элемента, не имеющего нечетный порядок.

Алгебра инъективных частично обратимых программ (APIP)

Эта алгебра соответствует классу инъективных программ, описанному в работе Глюка и Аксельсена [10].

Поскольку программы не предполагаются всюду определенными (иначе для инъективных программ невозможно было бы определить обращение), в полугруппе обязательно имеется константа 0 (**ошибка**). Поскольку есть ничего не делающие программы, полугруппа является моноидом и имеется единица e. Полугруппы частично определенных инъективных отображений в свое время интенсивно исследовались (энциклопедический сборник классических результатов смотри [16,17]; в последние годы эта тема практически выпала из рассмотрения алгебраистов как «исчерпанная и неактуальная»).

Дадим краткую сводку необходимых нам результатов. Обратные друг другу элементы можно определить даже в полугруппе без единицы или в таковой для делителей нуля, которые ни в каком произведении с другими элементами не могут дать e.

Определение 9. Элементы a и b *взаимно обратны*, если $a \circ b \circ a = a$, $b \circ a \circ b = b$. Полугруппа *инверсная*, если у каждого ненулевого элемента есть единственный обратный. Тогда для обратного используется традиционное обозначение a^{-1}. Элемент a *полный*, если $a \circ a^{-1} = e$. Если $a \neq 0$, $b \neq 0$, $a \circ b = 0$, то a называется *левым делителем нуля*, b — *правым*. a — *делитель нуля*, если есть такие b и c, что $a \circ b \circ c = 0$. *Частичная функция* — такая функция, что для некоторого ненулевого a $(a \star f) = 0$. В инверсной полугруппе a является ограничением b $(a \preccurlyeq b)$, если $a \circ b^{-1} = a \circ a^{-1}$. a — частичная единица, если $a = b \circ b^{-1}$ для некоторого $b \neq 0$.

Построим теперь абстрактную алгебру частичных инъективных отображений с операцией обращения программ (APIP).

$$(0 \star x) = (x \star 0) = 0 \tag{11}$$

$$f \neq 0 \supset (f \star M) = f^{-1} \tag{12}$$

$$\forall x, f((x \star f) \neq 0 \supset (x \star (f \circ f^{-1})) = x \circ f \circ f^{-1}) \tag{13}$$

$$\tag{14}$$

Свойства (11) как раз достаточны для описания взаимодействия программной алгебры с нулем. А вот (13) устанавливает взаимоотношения действий с единицей. Это гарантирует инъективность действий. Установим некоторые свойства частичных единиц и отношения порядка.

Предложение 12. Свойства частичных единиц и отношения порядка.

1. $a \neq 0 \& a \preccurlyeq e \equiv a = a \circ a^{-1}$.
2. $a \neq 0 \& a \preccurlyeq e \equiv a = a^{-1} \circ a$.
3. $a \neq 0 \& a \preccurlyeq e \equiv \exists b\, a = b \circ b^{-1}$.
4. $a \neq 0 \& a \preccurlyeq e \supset a = a^{-1}$.
5. $a \neq 0 \& a \preccurlyeq e \supset a^2 = a$.
6. $a \preccurlyeq b \supset a \circ c \preccurlyeq b \circ c$.
7. $a \preccurlyeq b \supset c \circ a \preccurlyeq c \circ b$.
8. $a \preccurlyeq b \supset a^{-1} \preccurlyeq b^{-1}$.
9. $a \preccurlyeq b \equiv a \circ b^{-1} = a \circ a^{-1}$.
10. $a \preccurlyeq b \equiv a \circ b^{-1} \circ a = a$.
11. $a \preccurlyeq b \equiv a^{-1} \circ b \circ a^{-1} = a^{-1}$.
12. $a \preccurlyeq b$ — отношение частичного порядка на полугруппе без 0.
13. Максимальные элементы по отношению $a \preccurlyeq b$ — полные элементы.

Доказательство. Большинство из этих результатов можно найти в книге [17, §7.1]. Поэтому мы докажем лишь те из них, которые касаются отсутствующего в книге сравнения с e. В самом деле, цепочка равенств доказывает первое утверждение.

$$a \circ a^{-1} = a \circ e^{-1} = a \circ e = a.$$

Второе устанавливается следующим образом:

$$a^{-1} \circ a = a^{-1} \circ a \circ a^{-1} = a.$$

Третье утверждение доказывается следующими преобразованиями. Пусть для некоторого b выполнено $a = b \circ b^{-1}$. Тогда

$$a \circ a^{-1} = b \circ b^{-1} \circ \left(b \circ b^{-1}\right)^{-1} = b \circ b^{-1} \circ b \circ b^{-1} = b \circ b^{-1}.$$

Теперь установим четвертое равенство.

$$a^{-1} = a \circ a^{-1} = a \circ a^{-1} \circ a^{-1} = a \circ a^{-1} \circ a = a.$$

Пятое очевидно следует из первого и четвертого.
□

Заметим, что для обычного обращения следовало бы ожидать обратной перестановки членов в отношении порядка, а здесь порядок сохраняется.

Из доказанного следует монотонность композиции по обоим аргументам, а действия — по второму. Монотонности действий по первому аргументу ожидать не приходится, поскольку, скажем, метапреобразования могут расширять область определения преобразуемой программы, причем существенно зависимо от ее конкретной записи.

Теперь установим аналог инъективности для действий.

Предложение 13. Выполнено следующее свойство относительной инъективности:

$$\forall a, b, f\, ((a \star f) = (b \star f) \equiv\ (a \star (f \circ f^{-1})) = (b \star (f \circ f^{-1}))).$$

Очевидно.
□

Пример 11. Полной инъективности ожидать нельзя. Рассмотрим программы, частично определенные на типах данных. Пусть f как функция определена лишь на натуральных числах, y совпадает с x на натуральных числах, но различается на отрицательных, хотя все функции инъективны. Тогда

$$(x \circ f) = (y \circ f) \& x \neq y.$$

Предложение 14. Левые и правые делители нуля являются делителями нуля.

Очевидно.
□

Предложение 15.

$$\forall x, f\,((x \star f) = 0 \equiv x \circ f = 0).$$

Доказательство. Пусть $(x \star f) = 0$, $f \neq 0$. Тогда

$$0 = (0 \star f^{-1}) = ((x \star f) \star f^{-1}) = ((x \star (f \circ f^{-1})) = x \circ f \circ f^{-1} = x \circ f \circ f^{-1} \circ f = x \circ f.$$

Повторяя эту последовательность преобразований в обратную сторону, получаем вторую часть доказываемой эквивалентности.
□

Мы видим, что в отношении равенства нулю результата две наших операции совпадают.

Как отметил Е. Кочуров в личной переписке: “Раньше длинно говорил «источники наведенных ошибок программы при сборке». Теперь буду говорить коротко и ясно: «делители нуля».”

Теперь методами аналогичными предыдущим параграфам, можно доказать два утверждения

Предложение 16. Каждая инверсная полугруппа с нулем, единицей и хотя бы одним элементом второго порядка, может быть продолжена до APIP.

Предложение 17. Пусть G_0 — подмоноид инверсной полугруппы с нулем $\mathbb{G}$, не совпадающий с нею самой и с $\{0, e\}$ и в $G \setminus G_0$ есть хотя бы один элемент второго порядка. Тогда его можно продолжить до APIP $\mathbb{G}_0$, такой, что для всех $f \in G_0$ $(x \star f) = (x \circ f)$.

Нильпотентная логика как алгебраическая структура (ANP)

Нильпотентная логика была введена как логика существенно необратимых действий. Таковы, например, любые действия (даже бездействие), если принимать в расчет текущее время. Рассмотрим понятие строго нильпотентной полугруппы, подобное тому, которое исследовалось в работе А. И. Мальцева [18].

Определение 10. $\mathbb{G}$ — строго нильпотентная полугруппа, если для каждой бесконечной последовательности элементов a_i существует такое n, что

$$\prod_{i=1}^{n} a_i = 0.$$

Нильпотентная программная алгебра (ANP) над $\mathbb{G}$ — GAPS над $\mathbb{G}$, сохраняющая 0.

В строго нильпотентной полугруппе не может быть единицы и идемпотентов, кроме 0 (то есть таких x, что $x \circ x = x$). Все элементы необратимы, даже в полугрупповом смысле. Из определения GAPS следует, что каждая бесконечная цепочка действий также в некоторый момент дает ошибку.

Методами предыдущих параграфов доказываются два утверждения:

Предложение 18. Каждая строго нильпотентная полугруппа с нулем $\mathbb{G}$ может быть обогащена до APIP.

Предложение 19. Пусть G_0 — подполугруппа $\mathbb{G}$. Тогда существут APIP $\mathbb{G}_0$, такая, что для всех $f \in G_0$ $(x \star f) = (x \circ f)$.

Поскольку здесь нет выделенных действий, леммы не имеют ограничений.

Ценность таких алгебр в том, что необратимые действия порождают автоматное программирование, а программные алгебры, соответственно — автоматы высших типов.

Некоторые замечания о связях алгебр с программированием

Прежде всего, об истоках неассоциативности в программировании. В функциональном программировании все очевидно: применить оператор к программе, а затем результирующую программу к данным, не то же самое, что применить обе эти программы друг за другом. Если у нас есть преобразования программ, тоже более или менее ясно: преобразовав программу, мы получаем новое действие. В якобы линейных последовательностях действий с частичной обратимостью (undo) неассоциативность замаскирована под «медвежьи услуги». Система может неявно объединить несколько действий в блок и отменить блок целиком (так, например, делает Word при отмене операций посимвольного редактирования). Явно блочность и неассоциативность возникает, когда имеются действия типа отката к предыдущей контрольной точке.

Даже простая групповая реализация полностью обратимых действий выходит за пределы группы, поскольку с необходимостью возникает операция обращения сегмента программы, не являющаяся внутренней групповой [12].

Теперь о необходимости сохранять ассоциативную структуру. Во-первых, она гарантирует возможность объединения блока действий в одно действие. Во-вторых, она лучше соответствует структуре текста программы или записи последовательности действий.

В последнее время намечается необходимость перехода от примата численных данных к данным нечисленным. Это означает переход от численных функций, математического анализа и дифференциальных уравнений к алгебраическим и логическим структурам. Даже в современной качественной теории дифференциальных уравнений основные преобразования алгебраические, но по традиции исходные данные и конечные результаты «для лучшего понимания» представляются в численной форме. Тем самым теоретическая математика в принципе подготовлена к переходу от аппарата численных моделей к аппарату моделей алгебро-логических. Самое главное, что уже накоплен багаж методов перевода задач из одной формы в другую.

Алгебраические вычисления представляются гораздо более радикальным шагом в супервычислениях, чем «параллелизм». Даже в идеальном случае распараллеливание дает линейное ускорение программы, а переход к более адекватным высокоуровневым структурам данных может давать выигрыш в башню экспонент раз и по длине программы, и по скорости ее исполнения. Алгебраические вычисления «распараллелены» с самого начала, что делает излишней задачу преобразования к параллельной форме программы, изначально написанной в терминах последовательных действий. С точки зрения системного анализа последняя задача была просто неправильно поставлена, что диагностирует неизлечимую генетическую болезнь вечно молодого и перспективного направления параллельных вычислений.

Теперь рассмотрим более высокоуровневые аспекты взаимодействия программирования и алгебры. А именно, одной из важнейших групп понятий в алгебре являются преобразования алгебраических структур. Посмотрим, как они отражаются в программах.

О простейшем из методов соединения алгебраических структур — прямом произведении — уже говорилось. Оно делит программу и данные на две невзаимодействующие части, что может быть выигрышно для эффективных вычислений, но бедновато в реальных случаях. Рассмотрим некоторую модификацию произведения.

В программах обычно целесообразно выделять команды и данные. В традиционных формализациях команды являются константами и программа совершает пробег по этим константам. В обратимом и полуобратимом программировании, а также в программировании, где действия могут влиять на команды, такой подход недостаточен. Команды в обратимом программировании должны также работать полностью обратимо. В других случаях команды должны обрабатываться более аккуратно, чем данные. Этим условиям удовлетворяет конструкция полупрямого произведения, известная для групп (см., напр., [15]) и обобщаемая здесь на полугруппы и программные алгебры.

Определение 11. Полупрямое произведение полугруппы команд $\mathbf{C}$ и полугруппы данных $\mathbf{D}$ есть полугруппа $\mathbf{C} \rtimes \mathbf{D}$, носитель которой $C \times D$, а операция умножения определяется следующим образом. Задан гомоморфизм $\varphi : \mathbf{C} \to \mathrm{Hom}(\mathbf{D}, \mathbf{D})$ полугруппы команд в полугруппу гомоморфизмов полугруппы данных.

$$\langle c_1, d_1 \rangle \circ \langle c_2, d_2 \rangle = \langle c_1 \circ c_2, d_1 \circ (d_2\ (c_2\ \varphi)) \rangle\,.$$

Полупрямое произведение программных алгебр $\mathbb{C}$ и $\mathbb{D}$ определяется аналогично с дополнительным условием на $\star$:

$$\langle c_1, d_1\rangle \star \langle c_2, d_2\rangle = \langle c_1 \star c_2, d_1 \star (d_2\ (c_2\ \varphi))\rangle\,.$$

Полупрямое произведение является естественным методом структурирования алгебры на программу и данные. Оно, в частности, возникает при моделировании условных операторов и циклов в обратимых вычислениях, базирующихся на группах (смотри, например, [12]). Оно часто позволяет совместное исполнение в широких пределах (его критерием является, в частности, равенство единичному отображению $(c\ \varphi)$ для некоторого подкласса команд).

Еще одним важным понятием является перемежающееся произведение: полугруппа (группа) выражений вида

$$[a_1\circ]b_1 \circ a_2 \circ b_2 \cdots \circ a_n[\circ b_n].$$

В полугруппах без единицы первый и последний члены могут опускаться. Такая конструкция соответствует прямой интерпретации программ с неограниченными условными операторами. С ограниченными бесконечное разрастание выражений удается преодолеть, используя полупрямое произведение. Рассмотрим интерпретацию условного предложения, вырабатывающего новое действие над группой **if** P **then** t **else** r **fi** Пусть G — исходная группа команд, H — группа значений альтернатив. Чтобы обратимо его вычислить, нужна группа с носителем $\mathbb{Z}_2 \times G \times G \times H$ и с операцией

$$\begin{gathered}\langle z, a_1, b_1, c_1\rangle \circ \langle 0, a_2, b_2, c_2\rangle = \langle z, a_1 \circ a_2, b_1 \circ b_2, c_1 \circ c_2\rangle \\ \langle z, a_1, b_1, c_1\rangle \circ \langle 1, a_2, b_2, c_2\rangle = \\ \langle z \oplus 1, a_1 \circ b_2, b_1 \circ a_2, c_1 \circ c_2\rangle\end{gathered} \tag{15}$$

представляемая как $(G \times G) \rtimes (\mathbb{Z}_2 \times H)$. Требуемый объем памяти и программ при этом увеличивается в случае обратимости или полуобратимости в два раза для каждого исполнения условного оператора.

В полугруппах имеется понятие подпрямого произведения: подгруппа прямого произведения, содержащая для каждого элемента i-того сомножителя a_i по крайней мере один элемент, i-тая проекция которого есть a_i. Оно соответствует сборке внешне независимых модулей, исключающей их нежелательные взаимодействия.

Как ни странно, казалось бы, самая «безобидная» из алгебраических конструкций — факторизация — может полностью разрушить структуру вычислений, например, их обратимость или полуобратимость. Если нужна конструкция, ее нужно собрать из допустимых «кирпичиков» или получить отрезанием лишнего, как в подпрямом произведении.

И, наконец, исключительная гибкость алгебраических понятий (они могут интерпретироваться самыми разными способами) открывает перед алгебраическим программированием невиданные для обычного операторного перспективы.

Заключение

Предельно общий а абстрактный алгебраический подход позволил выделить несколько важнейших структур, привязанных не к конкретным методам программирования, а к классам получающихся вычислений и преобразований. В частности, AFIP показывают класс вычислений более широких, чем групповые, но тем не менее обладающих свойством полной обратимости, которые могут быть использованы для преодоления предела Ландауэра.

Автор благодарен за неоднократные стимулирующие обсуждения и нахождение неточностей Р. Глюку, С. Мешвелиани, А. Непейвода.

Список литературы

1. Глушков В. М. Синтез цифровых автоматов. — М.: Физматгиз, 1962. — 476 с.
2. Maurer W. D. A theory of computer instructions. Journal of the ACM, 13(2): 226–235, 1966.
3. Глушков В. М., Цейтлин Г. Е., Ющенко Е. Л. Алгебра, языки. программирование. — Киев, Наукова думка, 1978. — 320 с.
4. Иванов П. М. Алгебраическое моделирование сложных систем. — М.: Наука. Физматлит, 1996. — 274 с.
5. Alban Ponse and Mark B. van der Zwaag. An Introduction to Program and Thread Algebra. LNCS 3988, 2006, pp 445–488.
6. Bergstra J. A., Bethke I, Ponse A. Program Algebra and Thread Algebra. Amsterdam, 2006, 114p.
7. Мальцев А. И. Алгебраические системы. М.: Наука, 1970. — 392с.
8. Митчелл Дж. Основания языков программирования. М.— Ижевск, РХД, 2009 (Mitchell J. C. Foundationfor programming languages, MIT, 1996.)
9. Непейвода Н. Н. Реверсивные конструктивные логики. Логические исследования, **15**, 150–169 (2008).
10. Axelsen H. G., Glück R. What do reversible programs compute? FOCSSACS 2011, LNCS 6604, pp. 42–56, 2011
11. Непейвода Н.Н.: Уроки конструктивизма. Geidelberg: Lambert Academic Publishing, 98 pp. (2011)
12. Nepejvoda N. N. Reversivity, reversibility and retractability. Third international Valentin Turchin workshop on metacomputation. Pereslavl: УГП, 2012. pp 203–215.
13. Курош А. Г. Теория групп. СПб.: Лань, 2005, 648с.
14. Каргаполов М. И., Мерздяков Ю. И. Основы теории групп. М.: Наука, 1977, 240 с.
15. А.В. Михалев, А.А. Михалев Введение в алгебру. 2007. М: ИНТУИТ, ISBN: 978-5-9556-0038-3, www.intuit.ru/department/mathematics/intalgebra/
16. A. H. Clifford, G. B. Preston. The algebraic theory of semigroups. v. I. Providence RI: AMS, 1964. Русский перевод А. Клиффорд, Г. Престон. Алгебраическая теория полугрупп. т. 1., М: Мир, 1972
17. A. H. Clifford, G. B. Preston. The algebraic theory of semigroups. v. II. Providence RI: AMS, 1967. Русский перевод А. Клиффорд, Г. Престон. Алгебраическая теория полугрупп. т. 2., М: Мир, 1972
18. Мальцев А. И., ‘Уч. зап. Ивановского гос. пед. ин-та’, 1953, т. 4, с. 107 – 118;

Применение двойственных сетей для построения вычислений в сетевых моделях сложных систем

Петров А.Е.

Московский государственный горный университет,
Москва, 119991 РФ, М. Октябрьская, Ленинский проспект д. 6,,
helen_pet@mail.ru

Аннотация. В докладе рассматривается применение тензорного метода двойственных сетей для создания сетевых моделей предметных областей и алгоритмов параллельных вычислений. Двойственные сети обладают инвариантом преобразования их структуры, который является математическим выражением закона сохранения потока энергии. Найденный автором инвариант решает проблему инварианта мощности Г. Крона. Инвариант выражает фундаментальное свойство, которым обладает структура сетей, независимо от их физического или логического содержания.

Введение

В докладе рассматривается применение тензорного метода двойственных сетей для создания сетевых моделей предметных областей и алгоритмов параллельных вычислений. Двойственные сети обладают инвариантом преобразования их структуры, который является математическим выражением закона сохранения потока энергии. Найденный автором инвариант решает проблему инварианта мощности Г. Крона. Инвариант выражает фундаментальное свойство, которым обладает структура сетей, независимо от их физического или логического содержания.

Ключевая проблематика

Сети обобщают метод графов, характеризационного анализа, и позволяют представлять одновременно процессы и структуру сложных систем. Представлены сетевые модели межотраслевого баланса и социально-экономической системы, с двойственными сетями потоков продуктов и денежных средств. При моделировании использованы свойства ортогональности внешних и внутренних воздействий, базисом для которых в пространстве сети являются, соответственно, открытые и замкнутые пути. Это является аналогом моделирования поведения живого организма с помощью неживых систем, и может применяться для построения нейросетевых моделей и нейрокомпьютеров.

Замкнутые и разомкнутые пути имеют фундаментальное свойство асимметрии – замкнутые пути можно выразить через разомкнутые пути и (или) замкнутые пути, а разомкнутые пути нельзя выразить через замкнутые пути. Свойства путей аналогичны поведению Х-У хромосом при размножении. Матрицы преобразования путей в сетях имеют нулевой квадрант, соответствующий выражению разомкнутых путей через замкнутые пути. Инвариант двойственности преобразования структуры лежит в основе разработанных автором алгоритмов декомпозиции и расчета сетевых моделей сложных систем по частям без итераций, за счет использования сети соединений. Это позволяет проводить одним методом вычисления в сетях с внешними и (или) внутренними воздействиями, в которых распространяются потоки энергии или другие измеримые сущности, при любых изменениях связи элементов.

Заключение

Для реализации данных алгоритмов к построению вычислений в сетевых моделях сложных физических, технических, биологических системах могут применяться ЭВМ с параллельной архитектурой. Полученная система алгоритмов является обобщением диакоптики Г.Крона. Обсуждаются результаты попытки построения двойственных сетей для непланарных графов. На эту проблему автору указал В.А. Горбатов в 1984 году на семинаре кафедры «Кибернетика» в МИФИ, который проводился под руководством Л.Т. Кузина.

Литература

1. Петров А.Е. *Тензорная методология в теории систем*. -- М.: Радио и связь. Редакция литературы по кибернетике и вычислительной технике, 1985
2. Карминский А. М., Пересецкий А. А., Петров А. Е. *Рейтинги в экономике: методология и практика*. -- М.: Финансы и статистика, 2005. — 240 с.
3. Арменский А. Е., Гусев В. С., Петров А. Е. *Информационная и экономическая безопасность государства*. -- М.: Мобиле, 2003. — 228 с.

Асимптотика вероятности связности графа с низко надежными ребрами

Цициашвили Г.Ш.

Лаборатория ВМ и СА, Институт ``ИПМ ДВО РАН",
Владивосток, 690041 РФ, Радио ул., д. 7, офис 316
guram@iam.dvo.ru

Аннотация. В настоящей работе решается задача асимптотической оценки вероятности связности неориентированного графа общего вида с кратными и низко надежными ребрами. Строится алгоритм вычисления асимптотических констант, имеющий кубическую по числу вершин графа сложность.

Введение

Задача вычисления вероятности связности графа с ненадежными ребрами рассматривалась во многих работах. В [1], [2] построены верхние и нижние оценки вероятности связности (полинома надежности) для сетей общего вида на основе использования максимальных систем попарно непересекающихся разрезов или остовов. Указаны сети, для которых эти оценки становятся равенствами. В работе [3] вероятность связности исследуется для мультиграфов, т.е. графов с кратными ребрами. При относительно небольшом числе ребер в графе (с независимыми ребрами) в [4] приведены ускоренные алгоритмы вычисления коэффициентов полинома надежности, показавшие хорошие результаты в сравнении с Maple 11. В [5] к решению данной задачи был применен метод Монте-Карло в сочетании со специфическими комбинаторными формулами.

Однако при увеличении числа ребер в графе задача существенно усложняется. Поэтому появляется необходимость в разработке удобных в вычислительном плане асимптотических формул для вероятности связности графа с высоконадежными или с низконадежными ребрами. Первая задача была решена в работе [6] для плоских графов с помощью асимптотической формулы Буртина-Питтеля [7], теоремы Уитни о соответствии разрезов плоского графа циклам в двойственном графе [8] и формул Харари [9] для числа простых (без самопересечения) циклов длины не более 5 в неориентированном связном графе.

В настоящей работе решается задача асимптотической оценки вероятности связности неориентированного графа общего вида с кратными и низко надежными ребрами. Строится алгоритм вычисления асимптотических констант, имеющий кубическую по числу вершин графа сложность. Предлагаемое решение основывается на понятии остовного дерева и на известной теореме Кирхго-

фа-Трента для вычисления количества остовных деревьев [10] и связанных с ними характеристик. Следует отметить, что в последние годы данная теорема и ее многочисленные обобщения используются в различных дисциплинах: физике элементарных частиц, электромагнетизме, акустике, теории упругости, социологии, биологии и т.д. [10], [11]. В настоящей работе эта теорема нашла применение в задачах теории надежности.

Основные результаты

Предположим, что неориентированный связный простой (без петель и кратных ребер) граф G состоит из множества узлов $U = \{1,...,n\}$ и множества ребер W. Каждое ребро $w = (i, j),\ 1 \le i \ne j \le n,$ графа G с вероятностью $p(w)$ работоспособно и с вероятностью $\overline{p} = 1 - p(w)$ отказывает, причем все ребра функционируют независимо.

Предположим, что G - связный неориентированный граф (без петель и кратных ребер). Нас будет интересовать вероятность связности P графа G со случайно функционирующими и низко надежными ребрами. Определим матрицу Кирхгофа $K = \|k(i, j)\|_{i,j=1}^{m}$ графа G:

$$k(i,j) = \begin{cases} \textit{степень узла } i,\ i = j, \\ -1,\ (i,j) \in W, \\ 0,\ \textit{иначе.} \end{cases}$$

Здесь степень узла - это число инцидентных этому узлу ребер. Под остовным деревом графа G понимается дерево с множеством узлов U и с множеством ребер, содержащемся в W. Обозначим m число остовных деревьев в графе G.

Теорема 1. *Если* $p(w) = h,\ w \in W,$ *то при* $h \to 0$ *справедливо равенство* $P = mh^{n-1}(1 + O(h)),$ *причем вычислительная сложность определения коэффициента m имеет вид* $O(n^3)$.

Следствие. *Если при некоторых натуральных l(w) вероятности* $p(w) = h^{l(w)},\ w \in W,$ *то каждое из ребер этого графа можно заменить на l(w) последовательно соединенных ребер и к построенному таким образом графу вновь применить теорему 1.*

Это следствие можно использовать при исследовании распределения случайного времени потери связности сети, полагая $p(w) = P(\tau(w) > T),\ T \to \infty.$ Здесь $\tau(w)$ случайное время до отказа ребра w.

Теорема 2. *Если при некоторых положительных c(w) вероятности* $p(w) \sim c(w)h,\quad h \to 0,\ w \in W,$ *то* $P = m_1 h^{n-1}(1 + O(h)).$ *Здесь* m_1 *– алгебраическое дополнение любого элемента (они совпадают) матрицы* $K_1 = \|k_1(i, j)\|_{i,j=1}^{m}$ *вида*

$$k_1(i,j)=\begin{cases}\sum\limits_{t\in U,(i,t)\in W} c(i,t),\ i=j,\\ -c(i,j),\ (i,j)\in W,\\ 0,\ \textit{иначе},\end{cases}$$

причем вычислительная сложность определения коэффициента m_1 *имеет вид* $O(n^3)$.

Замечание. *Утверждение теоремы 2 автоматически переносится на мультиграф, полученный из графа G заменой каждого ребра* $w\in W$ *на c(w) параллельно соединенных и независимо функционирующих ребер с вероятностью работоспособности h. Такое параллельное соединение имеет вероятность работоспособности* $\sim c(w)h,\ h\to 0$.

Литература

1. Ломоносов М.В., Полесский В.П. Верхняя граница надежности информационных сетей // Проблемы передачи информации. – 1971. – Т. 7. – № 4. – С. 78–81.
2. Ломоносов М.В., Полесский В.П. Нижняя оценка надежности сетей // Проблемы передачи информации. – 1972. – Т. 8. – № 2. – С. 47–53.
3. Попков В.К. Математические модели связности. – Новосибирск: ИВМ и МГ СО РАН, 2006.
4. Rodionova O.K., Rodionov A.S., Choo H. Network Probabilistic ConnectivExact Calculation with Uase of Chains // ICCSA-2004. Springer LNCS. – Vol. 3046. – P. 315–324.
5. Gertsbakh I., Shpungin Y. Models of Network Reliability. Analysis, Combinatorics and Monte-Carlo. CRC Press. Taylor and Francis Group, 2010.
6. Tsitsiashvili G.Sh. Complete calculation of disconnection probability in planar graphs // Reliability: Theory and Applications. – 2012. – Vol. 1. – No 1. – P. 154–159.
7. Буртин Ю., Питтель Б. Асимптотические оценки надежности сложных систем // Техническая кибернетика. – 1972. – Т. 10. – № 3. – С. 90–96.
8. Whithney H. Nonseparable and planar graphs // Transactions of American Mathematical Society. – 1972. – Vol. 34. – P. 339–362.
9. Харари Ф. Теория графов. – М.: Мир, 1973.
10. Чеботарев П.Ю., Шамис Е.В. Матричная теорема о лесах и измерение связей в малых социальных группах // Автоматика и телемеханика. – 1997. – Вып. 9. – С. 125-137.
11. Kozma R., Puljic M., Balister P., Bollobas B., Frreeman W. Phase transition in the neuropercolation model of neural populations with mixed local and non-local interactions // Biological Cybernetics. – 2005. – vol. 92. – P. 367–379.

Аппликативная среда с ограничением для функций трехзначной логики

Шабунин Л.В., Шабунин А.Л.

Кафедра прикладной математики,
Чувашский государственный университет,
Чебоксары, 428015 РФ, Московский пр-т, д. 15
lvsh@mail.ru, a.shabunin@gmail.com

Аннотация. Функции k-значной логики широко используются при изучении дискретных преобразователей информации. В настоящей работе рассматривается вычисление функций трехзначной логики в аппликативной среде с ограничением.

Пусть E={0,1,2}, F={f_1, f_2} — бинарные операции, заданные таблицей Кэли:

f_1	0	1	2
0	1	1	1
1	2	2	0
2	0	0	2

f_2	0	1	2
0	1	1	1
1	0	2	2
2	2	0	0

Например, $f_1(0,1)=1$, $f_1(1,0)=2$. Обозначим через P_3 множество всех функций трехзначной логики.

Пусть X — множество переменных. Определим по индукции понятие α-терма над X:

1) переменная x из X есть α-терм;
2) элемент c из E есть α-терм;
3) если символ f обозначает бинарную операцию из F, Φ есть α-терм и x — переменная из X, то слово ((fΦ)x) есть α-терм.

При построении α-термов используется ограниченная суперпозиция: α-терм Φ может подставляться только на место первого аргумента функционального символа. Например,

$$f_2(f_1(f_2xy)y)x,$$

где x, y — переменные, есть α-терм, а выражение

$$f_2(f_1y(f_2xy))x$$

не является α-термом, так как есть подстановка на место второго аргумента символа f_1. Для α-терма Φ запись $\Phi(x_1, \ldots, x_n)$ означает, что все переменные, входящие в Φ, принадлежат множеству $\{x_1, \ldots, x_n\}$.

Аппликативная F-среда содержит объекты 0, 1, 2, представляющие элементы множества E, объекты f_1 и f_2, представляющие одноименные бинарные операции и характеризующиеся правилами редукции

$$(f_1a)b \to f_1(a,b), \qquad (f_2a)b \to f_2(a,b)$$

для любых a, b из E, а также набор объектов-атомов, которые будут играть роль параметров. Указанные объекты будем называть исходными. Объекты-атомы будут соответствовать переменным в определении α-термов.

Из исходных объектов с помощью ограниченной аппликации строятся другие объекты. Ограниченность аппликации состоит в том, чтобы получались только такие составные объекты, которые естественным образом соответствуют α-термам.

Объект, построенный из параметров $a_1, \ldots, a_n$ и соответствующий α-терму $\Phi(x_1, \ldots, x_n)$, обозначим через $\Phi(a_1, \ldots, a_n)$. Говорим, что α-терм $\Phi(x_1, \ldots, x_n)$ определяет объект $\Phi(a_1, \ldots, a_n)$. Если в качестве параметров $a_1, \ldots, a_n$ берутся объекты из множества $\{0, 1, 2\}$, то объект $\Phi(a_1, \ldots, a_n)$ редуцируется снова в объект из этого множества. Тем самым объект $\Phi(a_1, \ldots, a_n)$ задает некоторую функцию трехзначной логики. Будем говорить, что объект $\Phi(a_1, \ldots, a_n)$ реализует эту функцию.

Теорема. *Для любой функции* $f(x_1, \ldots, x_n)$ *из* P_3 *могут быть построены* α-*терм* $\Phi(x_1, \ldots, x_n)$ *и определяемый им объект* $\Phi(a_1, \ldots, a_n)$, *который реализует функцию* $f(x_1, \ldots, x_n)$.

Литература

1. Вольфенгаген В.Э. *Комбинаторы: объекты, помогающие понять строение компьютинга* // Труды 1-й международной конференции по аппликативным вычислительным системам (АВС’2008), Москва, 29-30 мая 2008 г. / Под ред. В.Э. Вольфенгагена. – М.: НОУ Институт Актуального образования «ЮрИнфоР-МГУ», 2008. – С. 1–21.
2. Барендрегт Х. *Ламбда-исчисление. Его синтаксис и семантика.* – М.: Мир, 1985. – 606 с.
3. Яблонский С.В. *Введение в дискретную математику.* – М.: Наука, 1979. – 272 с.
4. Шабунин А.Л. *Об α-суперпозиции функций k-значной логики* // Сиб. матем. журнал. – 2007. – Т. 48. – № 2. – С. 441–457.

Средства специализации концептуальных моделей на основе определённых дескрипций*

Исмаилова Л. Ю., Косиков С. В.

НИЯУ МИФИ / Группа компаний “ЮрИнфоР”,
Москва, 119435 РФ, М. Пироговская ул., д. 5, офис 37
jurinfor.2012@gmail.com
WWW: http://www.jurinfor.ru

Аннотация Предлагаются формализованные средства специализации концептуальных моделей для предметных областей (ПО), предполагающих построение и исследование интенсиональных сущностей. В качестве основного средства специализации предлагаются пропозициональные концепты интенсионального вида. Выполняется исследование различных конструкций, используемых при моделировании, в том числе экстенсиональных и интенсиональных предикатов. Вводятся определённые дескрипции как синтаксические конструкции, задаётся их интерпретация и изучаются её свойства.

Ключевые слова: предметная область, концептуальная модель, дескрипция, экстенсионал, интенсионал.

Введение

Задачи описания предметных областей (ПО), для которых существенно выражение процесса именования, требуют развития средств концептуального описания. Традиционно в качестве основного средства описания используется понятие интенсионала [Карнап], однако общей теории концептов, охватывающей весь круг задач моделирования, до настоящего времени не предложено.

Одна из узловых проблем возникает при анализе процесса замены выражения на тождественное ему по предметному значению в системах, допускающих подстановку. Имеющиеся подходы либо формализуют частные случаи успешности подстановок [Крипке], либо недостаточно формализованы для того, чтобы давать адекватную основу для технологичного описания ПО, обладающих нетривиальным отношением именования, с точки зрения последующего построения информационных систем [4,5,6].

Перспективным представляется использование различных неклассических логических систем, в том числе интенсиональных логик [3,8]. В работе предлагается формализация базисной системы интенсиональной логики, содержащая операторы определённых дескрипций. Дескрипции используются для определения объектов ПО по заданным свойствам, что позволяет расширить возможности

* Посвящается 70-летию Национального исследовательского ядерного университета «МИФИ»

языков прикладных систем. Таким образом, специализация концептуальных моделей может быть обеспечена за счёт использования определённых дескрипций.

Ключевая проблематика

Язык и метаязык. При построении семантики будем использовать стандартный логический метод [3] — будем выделять логику, которая изучается (предмет исследования, предметная логика) и логику, посредством которой выполняется исследование.

Принимаем также стандартный тезис — субъект-исследователь имеет дело не с самой реальностью (с ПО), а с языковой действительностью — всё фиксируется в языке [3]. Изучаемая логика формулируется на некотором языке (предметном языке, языке-объекте), который далее будем называть просто *язык*. Предметный язык исследуется на языке исследователя, который далее будем называть *метаязык*. Относительно метаязыка будем принимать, что в нём справедливы законы классической логики.

Соглашение. Связка «тогда и только тогда, когда» в метаязыке будет обозначаться через $\overset{\circ}{\Longleftrightarrow}$.

В дальнейшем одиночные символы языка-объекта в тех случаях, когда типографские соглашения затрудняют их выделение из текста, будем заключать в одинарные кавычки ‘’.

Язык. Алфавит используемого языка включает следующие символы:

(1) предметные константы a_1, a_2, …, a_n, … Константы также будут обозначаться буквами b, c, …, возможно, с индексами;

(2) предметные переменные x_1, x_2, …, x_n, … Переменные также будут обозначаться буквами y, z, …, возможно, с индексами;

(3) предикатные константы P_1, P_2, …, P_n, … Предикатные константы также будут обозначаться буквами Q, R, …, возможно, с индексами;

(4) логические символы $\neg$, $\wedge$, $\vee$, $\forall$, $\exists$, $\mathbf{I}$, $=$;

(5) скобки ‘(’, ‘)’и запятая ‘,’;

Будем использовать также некоторые символы, значение которых в рамках рассматриваемой семантики может быть определено. Среди таких «производных» символов будем выделять «ограниченные» кванторы ‘$\forall$.’, ‘$\exists$.’и оператор дескрипции ‘$\mathbf{I}$.’. Характер этого ограничения будет подробнее описан ниже.

Возможно также проведение существенных расширений введённого языка за счёт введения дополнительных синтаксических конструкций, например, таких, как пропозициональные операторы [8]. Некоторые способы такого расширения составляют тему отдельной статьи.

Термы и формулы. Термы и формулы языка определяются в виде индуктивных классов. Поскольку возможны случаи как включения терма в формулу, так и обратного включения, классы термов и формул должны определяться одновременно. С учётом этого замечания классы термов и формул строятся следующим образом.

Определение. Термами называются члены индуктивного класса, определяемые следующим образом.

(i) (базис индукции)

1) если a — предметная константа, то a — терм;

2) если x — предметная переменная, то x — терм.

(ii) (шаг индукции)

1) если Φ — формула, то $\mathbf{I}x.\Phi(x)$ — терм.

(iii) (замкнутость)

других термов нет.

Для обозначения термов будем использовать буквы $r, s, t, \ldots$, возможно, с индексами.

Принятый способ построения алфавита ФПЯз предполагает задание его элементов при помощи индексации. Содержательно индексация имеет несколько целей. Нижние индексы служат для нумерации соответствующих элементов (констант, переменных и т. д.). Верхние индексы предикатных констант показывают число аргументных мест соответствующей константы. Формально эти содержательные соображения отражены в следующем определении.

Определение. Формулами называются члены индуктивного класса, определяемого следующим образом.

(i) (базис индукции)

1) если P_k^n — предикатный символ, и $t_1, \ldots, t_n$ — термы, $n \geqslant 0$, то

$$P_k^n(t_1, \ldots, t_n) -$$

формула;

(ii) (шаг индукции)

1) если Φ и Ψ — формулы, x — переменная, то

$$\neg\Phi, (\Phi \wedge \Psi), (\Phi \vee \Psi), \forall x\Phi, \exists x\Phi -$$

формулы;

(iii) (замкнутость)

других формул нет.

Заметим, что принятое определение обеспечивает однозначный разбор формулы в соответствии с её скобочной структурой. Далее примем обычное соглашение о скобках [Карри?], позволяющее опускать избыточные пары скобок.

Семантические домены. В работе будем использовать семантический метод[4]. Именно, будем строить интерпретацию конструкций языка в виде отображения в элементы некоторого набора сущностей, находящихся вне рассматриваемого языка. Такие сущности также должны быть определены в рамках выбранной заранее формальной системы. Будем далее говорить, что выбранная формальная система используется для построения семантики языка, а сущности определяемого набора соответственно называть семантическими.

Замечание. Было бы неверно говорить, что формальная система задаёт семантику языка, так как в рамках одной и той же формальной системы возможно построение существенно различных отображений, задающих интерпретацию.

В качестве базисной примем теоретико-множественную модель, согласно которой семантические сущности рассматриваются как множества. При этом дополнительная структура, которой наделяются рассматриваемые множества, каждый раз задаётся явным образом. Интерпретация, рассматриваемая как семантическое отображение, может отображать различных классы языковых выражений в различные выделенные множества. Возможно также рассмотрение «дополнительных» множеств, которые не служат сами в качестве областей интерпретации, но используются для построения таких областей. В дальнейшем множества, служащие для построения интерпретации, будем называть *семантическими доменами*.

Замечание. Возможно рассмотрение альтернативных моделей, в которых семантические домены изначально наделяются определённой дополнительной структурой (например, структурой булевых алгебр) или, наоборот, рассматриваются как конструктивы формальной теории, независимой от теоретико-множественной парадигмы (такой, как теория категорий). В этом смысле термин «семантический домен» является нейтральным, так как не ведёт к принятию ограничительных теоретико-модельных предположений.

При построении предлагаемой интерпретации будем использовать следующие семантические домены:

D — множество возможных индивидов;

V — множество виртуальных индивидов;

Asg — множество точек соотнесения;

2 — множество значений истинности.

Использование доменов при построении интерпретации далее будут подробно обсуждаться при задании семантических отображений.

Индивиды. Предполагаем, что индивид — исходное, неопределяемое понятие. Множество индивидов будем обозначать D.

Принципиальной особенностью предлагаемой семантики считается то, что множество индивидов D дано заранее. Это требование накладывается, так как в случае, когда множество индивидов D расширяется в ходе исследования (т. е. производится переход от D к некоторому D' такому, что $D \subseteq D'$) возможно получение парадоксальных результатов. В частности, затрудняется применение классической логики в метаязыке. На самом деле принятие такого предположения не ограничивает общности, так как вводимая семантика будет содержать специальные (описанные ниже) средства описания изменения области интерпретации.

При рассмотрении индивидов будем различать действительные, возможные и виртуальные индивиды, причём область D будем считать областью возможных индивидов. Будем также рассматривать область виртуальных индивидов V, которые будем понимать как некоторые идеальные объекты, вводимые для того, чтобы увеличить закономерность в рассматриваемом языке. Примерами виртуальных объектов могут являться $\pm\infty$ в теории действительных чисел.

В некоторых случаях виртуальные объекты могут быть элиминированы (т. е. каждому выражению, содержащему виртуальный объект, может быть сопоставлено эквивалентное выражение, не содержащее такого объекта). Однако такая

элиминация в общей интенсиональной логике может порождать проблемы, так как при наличии нескольких вхождений такого объекта в выражение он может в различных контекстах вести себя по-разному. Элиминация в этом случае может искажать семантику, что неоднократно отмечалась, в частности, при попытках построения реляционных операторов над частично определёнными данными. Для обеспечения аккуратного исследования возникающих семантических возможностей не будем заранее предполагать возможность элиминации объектов; наоборот, будем считать выражения для виртуальных объектов действительно именующими абстрактные объекты.

Одним из выявленных узких мест построения моделей оказывается обеспечение кванторных операций. В предлагаемой модели квантификация допускается только по возможным индивидам. Квантификация по части виртуальных индивидов возможна, если такая часть может быть индексирована возможными индивидами с помощью некоторой определимой операции. Однако по этому методу квантификация никогда не будет производиться по всем виртуальным индивидам.

В итоге в модели различаются действительные, возможные и виртуальные индивиды. Обозначим область действительных индивидов через A. Разница между A и D становится решающей в модальной логике, в которой A должно быть заменено семейством областей A_i. Тогда:

$$A_i \subseteq D \subseteq V.$$

Соглашение. Множество действительных индивидов в точке соотнесения i_n будем обозначать A_n вместо A_{i_n}.

Пример.

$$V = \{a_1, a_2, a_3, a_\infty\}$$
$$D = \{a_1, a_2, a_3\}$$
$$A_1 = \{a_1, a_2\}$$
$$A_2 = \{a_1, a_3\}$$

Предложение. $\cup_{i \in Asg} A_i \neq D$ тогда и только тогда, когда существует возможный индивид, который ни в одной точке соотнесения не является действительным.

Доказательство. ($\Rightarrow$). Пусть

$$(*) \cup_{i \in Asg} A_i \neq D.$$

Поскольку $\forall i \in Asg\ A_i \subseteq D$, то $\cup_{i \in Asg} A_i \subseteq D$. В силу $(*)$ $D \setminus \cup_{i \in Asg} A_i$ непусто.

Рассмотрим $x \in (D \setminus \cup_{i \in Asg} A_i)$. Очевидно, $x \in D$, т. е. x — возможный индивид. Предположим, что $x \in A_i$ для некоторого $i \in Asg$. Тогда $x \in \cup_{i \in Asg} A_i$, что находится в противоречии с выбором x. Полученное противоречие доказывает, что $\forall i \in Asg\ x \notin A_i$, т. е. x не является действительным индивидом ни в одной точке соотнесения.

($\Leftarrow$). Пусть $\exists x \in D\ \forall i \in Asg\ x \notin A_i$. Тогда $x \notin \cup_{i \in Asg} A_i$. Поэтому множества D и $\cup_{i \in Asg} A_i$ различаются в элементе x (он входит в первое множество и не входит во второе), т. е. $\cup_{i \in Asg} A_i \neq D$, ч. т. д.

Точки соотнесения. Идея возможных миров является выражением принципа индексных выражений Карнапа-Бар-Хиллела. Возможные миры, понимаемые как различные собрания индивидов с дополнительной структурой или без неё, выражают только один аспект этой идеи. Любая система структур может быть индексирована элементами подходящего множества, причём различными способами. Возьмём фиксированное множество индексов, которые будем далее называть точками соотнесения [Исмаилова, Скотт], так как для определения значения выражения необходимо «соотнести» его с точкой. Множество точек соотнесения будем обозначать Asg, отдельные точки соотнесения будем обозначать $i \in Asg$.

Излагаемая семантика позволяет рассматривать различные аспекты динамичных предметных областей естественным и удобным путём: надо только выбрать подходящие координаты

$$i = (w, t, p, a, \dots),$$

где w есть *мир*, t — *время*, $p\{xyz\}$ — (трёхмерное) *положение* в мире, a — *субъект* и т. д. Все эти координаты могут изменяться, возможно, независимо и таким образом влиять на истинностное значение утверждений, которые косвенно относятся к данным координатам.

Прежде всего отметим индексами систему действительных индивидов, положив, возможно различные, $A_i \subseteq D$ для каждого $i \in Asg$. Важно, чтобы не предполагалось одно-однозначного соответствия между A_i и i, так как элементы множества Asg могут обладать структурой, не полностью отражающейся в отличиях одного A_i от другого.

Пример. $Asg = \{i_1, i_2\}$.

Посмотрим на эти индексы в действии. Суть состоит в том, что в самих выражениях языка делается неопределённая ссылка на индекс. В качестве примера рассмотрим выражение с квантификацией. Пусть имеется квантор $\forall x$, пробегающий по области D. Теперь введём квантор по «действительным» индивидам $\forall .x$, где точка обозначает указание на неопределённый индекс. Таким образом, даже если известно значение предиката $\mathcal{P}$, невозможно сказать, является ли значение

$$\forall .x\, \mathcal{P}(x)$$

истинным или ложным. Однако если выделить некоторое $i \in Asg$, то относительно этого индекса предложение имеет определённое истинностное значение; а именно, оно истинно, если $P(a)$ истинно для всех $a \in A_i$. Но пока не будет указано на некоторое $i \in Asg$, область значения квантифицированной переменной остаётся неизвестной.

Интерпретация. Определим интерпретацию термов τ и формул φ в точке соотнесения i. Интерпретация будет далее определяться для каждого конкретного вида термов и формул на протяжении настоящей работы. Интерпретацию формулы φ в точке соотнесения i будем обозначать

$$||\varphi||i = v.$$

Далее будем рассматривать интерпретации различных языковых конструкций и вводить семантические домены, необходимые для построения интерпретаций.

Утверждения. Примем обозначения истинностных значений: будем писать 1 для обозначения *истины* и 0 для обозначения *лжи*. Этот выбор основан на том, что $2 = \{0, 1\}$ есть простой и удобный символ для обозначения множества из двух истинностных значений. Далее, утверждению Φ сопоставим функцию $||\Phi||$, определённую на Asg, со значениями из множества 2. Иначе говоря, будем писать

$$||\Phi||i = 1,$$

когда Φ *истинно относительно* i.

Соглашение. Множество функций, определённых на A со значениями из B будет обозначаться $A \to B$.

Тогда

$$||\Phi|| \in Asg \to 2,$$

если

$$||\Phi||i \in 2,$$

где $i \in Asg$.

Замечание. Для приложений множество функций $A \to B$, возможно, потребуется ограничить. Укажем два специальных способа ограничения:

1) ограничение множество наложением на него условий алгебраического характера. Пример такого ограничения: построение семантики ламбда-исчисления при помощи рассмотрения полной решётки D и ограничения множество $D \to D$ до множества непрерывных функций $[D \to D]$, что дало возможность обеспечить изоморфизм $D \cong [D \to D]$;

2) ограничение областей, по которым пробегают переменные высших типов при интерпретации. Это даёт возможность построения семантики, обеспечивающую полноту системы.

Пропозициональные связки. Для объяснения значения пропозициональных связок дадим обычные семантические определения.

$$\begin{array}{lll}
(\neg)\ ||\neg\Phi||i = 1 & \overset{\circ}{\Longleftrightarrow} & ||\Phi||i = 0 \\
(\wedge)\ ||\Phi \wedge \Psi||i = 1 & \overset{\circ}{\Longleftrightarrow} & ||\Phi||i = 1 \text{ и } ||\Psi||i = 1 \\
(\vee)\ ||\Phi \vee \Psi||i = 1 & \overset{\circ}{\Longleftrightarrow} & ||\Phi||i = 1 \text{ или } ||\Psi||i = 1
\end{array}$$

Отметим, что принятая интерпретация пропозициональных связок в данном случае *экстенсиональна* — в каждой точке соотнесения значение формулы, главным знаком которой является пропозициональная связка, является функций значений подформул этой формулы в той же точке соотнесения.

Константы. Введём в язык (для интерпретации кванторов и дескрипций) константы

$$\bar{a} \text{ для всех } a \in V.$$

Обратим особое внимание на то, что константы вводятся для всех виртуальных индивидов (не только возможных).

Предполагаем далее, что соответствие между a и $\bar{a}$ является взаимно однозначным.

Замечание. Таким образом, имеем два предположения:

1) $\bar{a}$ определено для всех a, т. е. ни один виртуальный объект не пропущен — все виртуальные объекты можем поименовать;

2) $\bar{a}$ определено однозначно, т. е. каждый виртуальный объект получает уникальное имя — нет двух констант, которые соответствуют одному и тому же объекту.

Для задач проектирования информационных систем специального вида необходимо иметь в виду возможность принятия других предположений. Представляется, однако, что они могут быть формализованы на основе принятой выше простейшей модели.

Кванторы. При интерпретации кванторов вводим два квантора с разными свойствами.

$$\begin{array}{lll} (\forall) & ||\forall x \Phi(x)||i = 1 & \overset{\circ}{\Longleftrightarrow} \ ||\Phi(\bar{a})||i = 1 \text{ для всех } a \in D \\ (\forall.) & ||\forall . x \Phi(x)||i = 1 & \overset{\circ}{\Longleftrightarrow} \ ||\Phi(\bar{a})||i = 1 \text{ для всех } a \in A_i \end{array}$$

и аналогично для $(\exists)$ и $(\exists.)$.

Замечание. Отметим, что в этом месте происходит отступление от классического определения Тарского — не вводятся последовательности переменных и т. д., в результате чего семантика определяется только для замкнутых формул.

Определённые дескрипции. Определённая дескрипция представляет собой терм $(\mathbf{I}x\,\Phi(x))$.

$$(\mathbf{I}) \quad ||\mathbf{I}x\,\Phi(x)||i = a \quad \overset{\circ}{\Longleftrightarrow} \quad \{a\} = \{b \in D \ : \ ||\Phi(\bar{b})||i = 1\}$$

Проиллюстрируем некоторые свойства введённой интерпретации дескрипций на примерах.

Пример 1. Пусть $Asg = \{i_1, i_2\}$, $D = \{a_1, a_2\}$ и $A_1 = D, A_2 = D$ (т. е. в этом примере опущен эффект изменения области действительных индивидов в зависимости от соотнесений). Для определённых выше данных может быть сконструировано 9 индивидных концептов, заданных следующим образом (концепты будем обозначать через c_i).

Концепт	i_1	i_2
c_1	a_1	a_1
c_2	a_1	a_2
c_3	a_1	$\perp$
c_4	a_2	a_1
c_5	a_2	a_2
c_6	a_2	$\perp$
c_7	$\perp$	a_1
c_8	$\perp$	a_2
c_9	$\perp$	$\perp$

Здесь через $\perp$ обозначено, что концепт не определён в заданной точке соотнесения (напомним, что концепт представляет собой частичную функцию). Отметим, что

$$||\bar{a_1}|| = c_1$$
$$||\bar{a_2}|| = c_5$$

Зададим интерпретацию Φ, которая в соответствии с подходом к семантике одноместных предикатов должна иметь тип $C \to (Asg \to 2)$, где $C = (Asg) \to D$ — тип концепта. Для большего удобства задание $||\Phi||$ выполним в некаррированной форме, т. е. в виде функции типа $C \times Asg \to 2$.

$$||\Phi||(x, i_1) = [x = c_2]; \qquad ||\Phi||(x, i_2) = [x = c_4].$$

Напомним, что это значит, например, что $\Phi(x, i_1) = 1$, если $x = c_2$, и $\Phi(x, i_1) = 0$ в противном случае.

При таком задании $||\Phi||$ в каждой точке соотнесения i существует такой единственный x, что $(||\Phi||i)(x) = 1$, причём $x(i) = a_1$. Тем не менее дескрипция $\mathbf{I}x\Phi(x)$ не определена ни в одной точке соотнесения (т. е. $||\mathbf{I}x\Phi(x)|| = c_9$). Это происходит потому, что при подстановке в формулу любой константы (их две: $\bar{a}_1$ и $\bar{a}_2$) имеем $||\Phi(\bar{a})|| = 0$.

Этот пример иллюстрирует, что при вычислении дескрипции фактически учитывается только малая часть значений предиката, определяющего заданную дескрипцию. Именно, при вычислении на результат оказывают влияние только значения предиката на константах (из области D).

Пример 2. Теперь зададим дескрипцию, имеющую определённое значение в каждой точке соотнесения.

Задача. Пусть необходимо построить терм τ, значение которого в точках соотнесения принятого Asg заданы следующим образом:

Точка соотнесения	Значение
i_1	a_2
i_2	a_1

Решение. Построение терма будем выполнять в виде определённой дескрипции $\mathbf{I}x\Phi x$. Для построения интерпретации соответствующего интенсионального предиката Φ введём соответствующие пропозициональные концепты:

Концепт	i_1	i_2
f_1	1	1
f_2	1	0
f_3	0	1
f_4	0	0

Заметим, что в отличие от индивидных пропозициональные концепты определены в каждой точке соотнесения.

В соответствии с ($\mathbf{I}$) значения Φ зададим следующим образом:

$$||\Phi||(c_1) = f_3$$
$$||\Phi||(c_5) = f_2,$$

где индивидные концепты c_i определены выше. Значения Φ на остальных индивидных концептах могут быть заданы произвольно, так как они не влияют на значение дескрипции $\mathbf{I}x\Phi x$.

Проверим, что заданная интерпретация предиката действительно даёт решение задачи. Для этого вычислим $\{b \in D \ : \ ||\Phi(\bar{b})||i = 1\}$ в каждой точке соотнесения. Имеем:

$$||\Phi(\bar{a}_1)||i_1 = ||\Phi||(c_1)i_1 = f_3 i_1 = 0$$
$$||\Phi(\bar{a}_2)||i_1 = ||\Phi||(c_5)i_1 = f_2 i_1 = 1.$$

Поэтому $\{b \in D \ : \ ||\Phi(\bar{b})||i_1 = 1\} = \{a_2\}$. Следовательно, $||\mathbf{I}x\Phi x||i_1 = a_2$. Аналогично $||\mathbf{I}x\Phi x||i_2 = a_1$.

Свойства определённых дескрипций — полностью определённая дескрипция. Рассмотрим множество предикатов, задающих одну и ту же определённую дескрипцию, и изучим его структуру.

Предложение. Пусть Φ и Ψ таковы, что

$$||\mathbf{I}x\Phi(x)|| = ||\mathbf{I}x\Psi(x)|| = c,$$

где $t : Asg \to D$ — концепт, полностью определённый в каждой $i \in Asg$. Тогда

$$||\mathbf{I}x\Phi(x) \wedge \Psi(x)|| = c;$$
$$||\mathbf{I}x\Phi(x) \vee \Psi(x)|| = c.$$

Доказательство. Зафиксируем $i \in Asg$; пусть для этого i значение $t(i) = a$. По определению интерпретации дескрипции для этого i имеем:

$$\{a\} = \{b \in D \ : \ ||\Phi(\bar{b})||i = 1\};$$
$$\{a\} = \{b \in D \ : \ ||\Psi(\bar{b})||i = 1\}.$$

Поэтому

$$||\Phi(\bar{a})||i = 1; \quad ||\Phi(\bar{b})||i = 0 \text{ при } b \neq a;$$
$$||\Psi(\bar{a})||i = 1; \quad ||\Psi(\bar{b})||i = 0 \text{ при } b \neq a;.$$

Теперь имеем

$$||\Phi(\bar{a}) \wedge \Psi(\bar{a})||i = 1; \quad ||\Phi(\bar{b}) \wedge \Psi(\bar{b})||i = 0 \text{ при } b \neq a$$

и аналогично для $\vee$. Поэтому

$$\{a\} = \{b \in D \ : \ ||\Phi(\bar{b}) \wedge \Psi(\bar{b})||i = 1\}$$

и

$$||\mathbf{I}x\Phi(x) \wedge \Psi(x)||i = a = c(i),$$

ч. т. д.

Замечание. Предположение о полной определённости концепта t в условии предложения существенно.

Свойства определённых дескрипций — общий случай. Пусть теперь рассматриваемая дескрипция не определена в некоторых точках соотнесения. Рассмотрим подробнее одну из таких точек соотнесения (далее будем обозначать её i_0 и считать фиксированной) и изучим поведение дескрипции в этой точке.

Пусть $||\mathbf{I}x\Phi(x)||i_0 = \perp$. Это может произойти в двух случаях.

1. Для всех $a \in D$ имеем $||\Phi(\bar{a})||i_0 = 0$. Будем говорить в этом случае, что формула $\Phi(x)$ (с выделенной свободной переменной x) недоопределяет дескрипцию $\mathbf{I}x\Phi(x)$ в точке соотнесения i_0.

2. Существуют по крайней мере две различные константы $a_1, a_2 \in D$, для которых $||\Phi(\bar{a}_k)||i_0 = 1$, где $k = 1, 2$. Будем говорить в этом случае, что формула $\Phi(x)$ (с выделенной свободной переменной x) переопределяет дескрипцию $\mathbf{I}x\Phi(x)$ в точке соотнесения i_0.

В противном случае $||\mathbf{I}x\Phi(x)||i_0 \neq \perp$. Будем в этом случае говорить, что формула $\Phi(x)$ (с выделенной свободной переменной x) полностью определяет дескрипцию $\mathbf{I}x\Phi(x)$ в точке соотнесения i_0.

Соглашение. Будем опускать указание на дескрипцию, которую недоопределяет (переопределяет, полностью определяет) формула в тех случаях, когда она ясна из контекста.

Рассмотрим две формулы $\Phi(x)$ и $\Psi(x)$ (с выделенной свободной переменной x) такие, что $||\mathbf{I}x\Phi(x)||i_0 = ||\mathbf{I}x\Psi(x)||i_0$. Будем называть их эквивалентными относительно определения дескрипции в точке соотнесения i_0, если в этой точке соотнесения $\Phi(x)$ недоопределяет (переопределяет, полностью определяет) дескрипцию $\mathbf{I}x\Phi(x)$ тогда и только тогда, когда $\Psi(x)$ недоопределяет (переопределяет, полностью определяет) дескрипцию $\mathbf{I}x\Psi(x)$.

Предложение. Если для всех $b \in D$

$$||\Phi(\bar{b})||i_0 = ||\Phi(\bar{b})||i_0,$$

то формулы $\Phi(x)$ и $\Psi(x)$ эквивалентны относительно определения дескрипции в точке соотнесения i_0.

Доказательство. 1. Пусть дескрипция $\mathbf{I}x\Phi(x)$ полностью определена. Обозначим

$$||\mathbf{I}x\Phi(x)||i_0 = a.$$

Тогда по определению интерпретации дескрипции $\{a\} = \{b \in D : ||\Phi(\bar{b})||i = 1\}$, откуда следует:

$$\Phi(\bar{a}) = 1$$
$$\Phi(\bar{b}) = 0, b \neq a.$$

Из предположения имеем

$$\Psi(\bar{a}) = 1$$
$$\Psi(\bar{b}) = 0, b \neq a.$$

Поэтому дескрипция $\mathbf{I}x\Psi(x)$ также полностью определена, и $||\mathbf{I}x\Phi(x)||i_0 = a$.

2. Пусть дескрипция $\mathbf{I}x\Phi(x)$ недоопределена. Тогда по определению недоопределённой дескрипции для всех $a \in D$

$$\Phi(\bar{a}) = 0.$$

Из предположения имеем

$$\Psi(\bar{a}) = 0.$$

Поэтому дескрипция $\mathbf{I}x\Psi(x)$ также недоопределена.

3. Пусть дескрипция $\mathbf{I}x\Phi(x)$ переопределена. Тогда по определению переопределённой дескрипции существуют по крайней мере две различные константы $a_1, a_2 \in D$, для которых

$$||\Phi(\bar{a}_k)||i_0 = 1, \text{ где } k = 1, 2;$$
$$\Phi(\bar{b}) = 0, b \neq a_k.$$

Из предположения имеем

$$||\Psi(\bar{a}_k)||i_0 = 1, \text{ где } k = 1, 2; \Psi(\bar{b}) = 0, b \neq a_k.$$

Поэтому дескрипция $\mathbf{I}x\Psi(x)$ также переопределена, ч. т. д.

Для полностью определённых дескрипций указанное предложение допускает обращение.

Предложение. Если формулы $\Phi(x)$ и $\Psi(x)$ эквивалентны относительно определения дескрипции в точке соотнесения i_0, и соответствующие дескрипции полностью определены, то для всех $b \in D$ имеем

$$||\Phi(\bar{b})||i_0 = ||\Psi(\bar{b})||i_0.$$

Доказательство. Обозначим

$$||\mathbf{I}x\Phi(x)||i_0 = a.$$

Тогда по определению интерпретации дескрипции $\{a\} = \{b \in D : ||\Phi(\bar{b})||i = 1\}$, откуда следует:

$$\Phi(\bar{a}) = 1$$
$$\Phi(\bar{b}) = 0, b \neq a.$$

Поскольку $\Phi(x)$ и $\Psi(x)$ эквивалентны относительно определения дескрипции в точке соотнесения i_0, имеем

$$||\mathbf{I}x\Psi(x)||i_0 = ||\mathbf{I}x\Phi(x)||i_0 = a.$$

Следовательно,

$$\Psi(\bar{a}) = 1$$
$$\Psi(\bar{b}) = 0, b \neq a.$$

Сравнивая значения, получаем $||\Phi(\bar{b})||i_0 = ||\Psi(\bar{b})||i_0$, ч. т. д.

Для переопределённых дескрипций указанное обращение уже не имеет места. Таким образом, введённое отношение эквивалентности оказывается совместимым с отношением равенства формул, но слабее этого отношения.

Сформулируем теперь некоторые свойства введённых понятий.

Предложение 1. Пусть $\Phi(x)$ недоопределяет дескрипцию. Тогда $\Phi(x) \wedge \Psi(x)$ также недоопределяет дескрипцию.

Доказательство. По определению имеем $\forall a \in D\ ||\Phi(\bar{a})||i_0 = 0$. Тогда в соответствии с определением $||\wedge||$ имеем $\forall a \in D\ ||\Phi(\bar{a}) \wedge \Psi(\bar{a})||i_0 = 0$, ч. т. д.

Предложение 2. Пусть $\Phi(x)$ недоопределяет дескрипцию. Тогда формула $\Phi(x) \vee \Psi(x)$ эквивалентна относительно определения дескрипции формуле $\Psi(x)$.

Доказательство. По определению имеем $\forall a \in D\ ||\Phi(\bar{a})||i_0 = 0$. Тогда в соответствии с определением $||\vee||$ имеем $\forall a \in D\ ||\Phi(\bar{a}) \vee \Psi(\bar{a})||i_0 = ||\Psi(\bar{a})||i_0$, ч. т. д.

Предложение 3. Пусть $\Phi(x)$ переопределяет дескрипцию. Тогда формула $\Phi(x) \vee \Psi(x)$ также переопределяет дескрипцию.

Доказательство. По определению имеем $\exists a_1, a_2 \in D\ a_1 \neq a_2 \wedge ||\Phi(\bar{a}_k)||i_0 = 1$. Тогда в соответствии с определением $||\vee||$ имеем для этих $a_1, a_2\ ||\Phi(\bar{a}_k) \vee \Psi(\bar{a}_k)||i_0 = 1$, ч. т. д.

Отметим, что если формула $\Phi(x)$ переопределяет дескрипцию, то формула $\Phi(x) \wedge \Psi(x)$ может недоопределять, полностью определять либо переопределять дескрипцию. Это обосновывает обычный путь построения объекта с заданными свойствами при помощи дескрипций, когда объект строится последовательным указанием его свойств до тех пор, пока их не станет достаточно для определения объекта.

Дескрипции на действительных индивидах. Наряду с оператором '**I**'имеется также оператор '**I**.'(дескрипция с точкой), который определяется следующим образом:

$$(\mathbf{I}.) \quad ||\mathbf{I}.x\,\Phi(x)||i = a \quad \overset{\circ}{\Longleftrightarrow} \quad \{a\} = \{b \in A_i\ :\ ||\Phi(\bar{b})||i = 1\}$$

причём принимается, что значение этого терма неопределённо, если нет a, удовлетворяющего условию, заданному справа.

Оператор '**I**.'может быть выражен через оператор '**I**'следующим образом:

$$\mathbf{I}.x\,\Phi(x) = \mathbf{I}x\exists.y\,[x = y \wedge \Phi(y)]$$

Заключение

В работе предложены формализованные средства специализации концептуальных моделей, предполагающих построение и исследование интенсиональных сущностей. Акцент сделан на ПО, для которых существенно отношение именования. Средства специализации построены на основе использования пропозициональных концептов интенсионального вида. Выполнено исследование различных конструкций, используемых при концептуальном моделировании:

— выделение совокупности индивидов с разделением их на действительные, возможные и виртуальные обеспечивает средства описания частично определённых объектов и описания процессов динамического доопределения;

— использование точек соотнесения обеспечивает рассмотрение различных видов динамики ПО;

— использование определённых дескрипций, включающих полностью определённые, недоопределённые и переопределённые дескрипции, развивает традиционную технику определения объектов и обеспечивает задание объектов с контролируемой степенью определённости.

Исследование показывает возможность построения моделей с управляемой степенью интенсиональности за счёт согласованного использования экстенсиональных и интенсиональных конструкций. В частности, специализация концептуальных моделей может быть обеспечена за счёт использования определённых дескрипций.

Благодарности

Работа является обобщением ряда результатов проектов 10-07-00417-а, 11-07-00305-а, 12-07-00554-а, частично поддержанных грантами РФФИ.

Список литературы

1. Вольфенгаген В. Э. Комбинаторная логика в программировании. Вычисления с объектами в примерах и задачах. — М.: Институт ЮрИнфоР-МГУ, 2008.
2. Вольфенгаген В. Э. Логика. Конспект лекций: техника рассуждений. — М.: JurInfoR Ltd., АО "Центр ЮрИнфоР 2004.
3. Вольфенгаген В. Э. Методы и средства вычислений с объектами. Аппликативные вычислительные системы. — М.: JurInfoR Ltd., АО "Центр ЮрИнфоР 2004.
4. Исмаилова Л. Ю., Косиков С. В. Информационное моделирование динамичных рассуждений на примере юридической деятельности. — Технологии информационного общества – Интернет и современное общество: Материалы Всероссийской объединенной конференции. Санкт-Петербург, 20-23 ноября 2001 г. СПб.: Изд-во С.-Петерб. ун-та, 2001. с. 36-43.
5. Исмаилова Л. Ю., Косиков С. В. От поисковых систем к информационному моделированию рассуждений в Интернет-среде. — Технологии информационного общества – Интернет и современное общество: труды V Всероссийской объединенной конференции. Санкт-Петербург, 25-29 ноября 2002 г. СПб.: Изд-во С.-Петерб. ун-та, 2002. с. 97-100.
6. Исмаилова Л. Ю., Косиков С. В. Соотнесение семантик против семантики соотнесений. Технологии информационного общества – Интернет и современное общество: труды VI Всероссийской объединенной конференции. Санкт-Петербург, 3-6 ноября 2003 г. СПб.: Изд-во Филологического факультета СПбГУ, 2003. с. 17-22.
7. Монтегю Р. Прагматика и интенсиональная логика. — в кн. Семантика модальных и интенсиональных логик. Под ред. В. А. Смирнова. - М., Прогресс, 1981.
8. Скотт Д. Советы по модальной логике. — в кн. Семантика модальных и интенсиональных логик. Под ред. В. А. Смирнова. - М., Прогресс, 1981.

Специализация концептуальных моделей на основе интенсиональных операторов*

Исмаилова Л. Ю., Косиков С. В.

НИЯУ МИФИ / Группа компаний “ЮрИнфоР”,
Москва, 119435 РФ, М. Пироговская ул., д. 5, офис 37
jurinfor.2012@gmail.com
WWW: http://www.jurinfor.ru

Аннотация Предлагаются формализованные средства специализации концептуальных моделей для предметных областей (ПО), предполагающих построение и исследование интенсиональных сущностей. В качестве основного средства специализации предлагаются пропозициональные концепты интенсионального вида. Выполняется исследование различных конструкций, используемых при моделировании, в том числе экстенсиональных и интенсиональных предикатов. Вводятся и изучаются различные виды интенсиональных операторов.

Ключевые слова: предметная область, концептуальная модель, дескрипция, экстенсионал, интенсионал.

Введение

При разработке информационных систем (ИС), включающих развитые средства моделирования понятий, в большинстве случаев ограничиваются довольно простым набором операций над интенсиональными объектами. В частности, корректное выполнение кванторных операций, как правило, затруднительно, в том числе вследствие ограничений на использование более одного модального оператора и комбинаций операторов.

Зачастую отсутствие возможности корректного отграничения различных видов кванторных операций приводит к получению неадекватных информационных моделей. Так, при работе с интенсиональными операторами трудно интерпретируемой оказывается ситуация экстенсионала, пустого в заданной точке соотнесения, что порождает проблемы при вычислении значений кванторных выражений.

В работе вводятся и изучаются различные способы определения кванторных операций на основе использования экстенсиональных и интенсиональных предикатов. Исследуются операторы, в том числе позволяющие переносить результаты кванторных операций из одной точки соотнесения в другие. Формулируются ограничения, при которых некоторые виды кванторных операций становятся реализуемыми средствами поддержки разработки моделирующих систем.

---

* Посвящается 70-летию Национального исследовательского ядерного университета «МИФИ»

Ключевая проблематика

Язык и метаязык. При построении семантики будем использовать стандартный логический метод [3] — будем выделять логику, которая изучается (предмет исследования, предметная логика) и логику, посредством которой выполняется исследование.

Принимаем также стандартный тезис — субъект-исследователь имеет дело не с самой реальностью (с ПО), а с языковой действительностью — всё фиксируется в языке. Изучаемая логика формулируется на некотором языке (предметном языке, языке-объекте), который далее будем называть просто *язык*. Предметный язык исследуется на языке исследователя, который далее будем называть *метаязык*. Относительно метаязыка будем принимать, что в нём справедливы законы классической логики.

Соглашение. Связка «тогда и только тогда, когда» в метаязыке будет обозначаться через $\overset{\circ}{\Longleftrightarrow}$.

В дальнейшем одиночные символы языка-объекта в тех случаях, когда типографские соглашения затрудняют их выделение из текста, будем заключать в одинарные кавычки ‘’.

Язык. Алфавит используемого языка включает следующие символы:

(1) предметные константы a_1, a_2, …, a_n, … Константы также будут обозначаться буквами b, c, …, возможно, с индексами;

(2) предметные переменные x_1, x_2, …, x_n, … Переменные также будут обозначаться буквами y, z, …, возможно, с индексами;

(3) предикатные константы P_1, P_2, …, P_n, … Предикатные константы также будут обозначаться буквами Q, R, …, возможно, с индексами;

(4) логические символы $\neg$, $\wedge$, $\vee$, $\forall$, $\exists$, $=$;

(5) скобки ‘(’, ‘)’и запятая ‘,’;

(6) операторы:

- оператор необходимости $\Box$ и двойственный оператор возможности $\Diamond$;
- пропозициональные операторы N_1, N_2, …, N_k, … . Для некоторых пропозициональных операторов будут использоваться специальные обозначения вида $\Box_t$ и т. д.
- операторы, связывающие переменные S_1, S_2, …, S_n, … . Связывающие операторы будут также обозначаться символом \$, возможно, с индексами.

Термы и формулы языка определяются в виде индуктивных классов. Поскольку возможны случаи как включения терма в формулу, так и обратного включения, классы термов и формул должны определяться одновременно. С учётом этого замечания классы термов и формул строятся обычным образом.

Далее будем рассматривать интерпретации различных языковых конструкций и вводить семантические домены, необходимые для построения интерпретаций.

Использование комбинаций операторов. Существенной особенностью предлагаемых средств специализации является возможность одновременного использования нескольких модальных операторов. Как указано в [8], одна из самых больших ошибок всей модальной логики заключается в концентрации внимания на

системах лишь с одним модальным оператором. Для систем моделирования ПО, в частности, содержащих оценочные понятия, такое ограничение представляется особенно критическим.

Отметим, что комбинирование модальных операторов с временными операторами позволяет сформулировать принцип изменения; с логическими — позволяет сравнивать относительное и абсолютное; с операторами типа исторической или физической необходимости — обеспечивает связь субъекта и его окружения и т. д.

Излагаемая семантика позволяет рассматривать сразу несколько операторов естественным и удобным путём: надо только выбрать подходящие координаты

$$i = (w, t, p, a, \dots),$$

где w есть *мир*, t — *время*, $p\{xyz\}$ — (трёхмерное) *положение* в мире, a — *субъект* и т. д. Все эти координаты могут изменяться, возможно, независимо и таким образом влиять на истинностное значение утверждений, которые косвенно относятся к данным координатам.

Индивиды. Предполагаем, что индивид — исходное, неопределяемое понятие. Множество индивидов будем обозначать D.

Принципиальной особенностью предлагаемой семантики считается то, что множество индивидов D дано заранее. Это требование накладывается, так как в случае, когда множество индивидов D расширяется в ходе исследования (т. е. производится переход от D к некоторому D' такому, что $D \subseteq D'$) возможно получение парадоксальных результатов. В частности, затрудняется применение классической логики в метаязыке. На самом деле принятие такого предположения не ограничивает общности, так как вводимая семантика будет содержать специальные (описанные ниже) средства описания изменения области интерпретации.

При рассмотрении индивидов будем различать действительные, возможные и виртуальные индивиды, причём область D будем считать областью возможных индивидов. Будем также рассматривать область виртуальных индивидов V, которые будем понимать как некоторые идеальные объекты, вводимые для того, чтобы увеличить закономерность в рассматриваемом языке. Примерами виртуальных объектов могут являться $\pm\infty$ в теории действительных чисел.

В некоторых случаях виртуальные объекты могут быть элиминированы (т. е. каждому выражению, содержащему виртуальный объект, может быть сопоставлено эквивалентное выражение, не содержащее такого объекта). Однако такая элиминация в общей интенсиональной логике может порождать проблемы, так как при наличии нескольких вхождений такого объекта в выражение он может в различных контекстах вести себя по-разному. Элиминация в этом случае может искажать семантику, что неоднократно отмечалась, в частности, при попытках построения реляционных операторов над частично определёнными данными. Для обеспечения аккуратного исследования возникающих семантических возможностей не будем заранее предполагать возможность элиминации объектов; наоборот, будем считать выражения для виртуальных объектов действительно именующими абстрактные объекты.

Одним из выявленных узких мест построения моделей оказывается обеспечение кванторных операций. В предлагаемой модели квантификация допускается только по возможным индивидам. Квантификация по части виртуальных индивидов возможна, если такая часть может быть индексирована возможными индивидами с помощью некоторой определимой операции. Однако по этому методу квантификация никогда не будет производиться по всем виртуальным индивидам.

В итоге в модели различаются действительные, возможные и виртуальные индивиды. Обозначим область действительных индивидов через A. Разница между A и D становится решающей в модальной логике, в которой A должно быть заменено семейством областей A_i. Тогда:

$$A_i \subseteq D \subseteq V.$$

Соглашение. Множество действительных индивидов в точке соотнесения i_n будем обозначать A_n вместо A_{i_n}.

Пример.

$$V = \{a_1, a_2, a_3, a_\infty\}$$
$$D = \{a_1, a_2, a_3\}$$
$$A_1 = \{a_1, a_2\}$$
$$A_2 = \{a_1, a_3\}$$

Предложение. $\cup_{i \in Asg} A_i \neq D$ тогда и только тогда, когда существует возможный индивид, который ни в одной точке соотнесения не является действительным.

Доказательство. ($\Rightarrow$). Пусть

$$(*) \cup_{i \in Asg} A_i \neq D.$$

Поскольку $\forall i \in Asg\ A_i \subseteq D$, то $\cup_{i \in Asg} A_i \subseteq D$. В силу $(*)$ $D \setminus \cup_{i \in Asg} A_i$ непусто.

Рассмотрим $x \in (D \setminus \cup_{i \in Asg} A_i)$. Очевидно, $x \in D$, т. е. x — возможный индивид. Предположим, что $x \in A_i$ для некоторого $i \in Asg$. Тогда $x \in \cup_{i \in Asg} A_i$, что находится в противоречии с выбором x. Полученное противоречие доказывает, что $\forall i \in Asg\ x \notin A_i$, т. е. x не является действительным индивидом ни в одной точке соотнесения.

($\Leftarrow$). Пусть $\exists x \in D\ \forall i \in Asg\ x \notin A_i$. Тогда $x \notin \cup_{i \in Asg} A_i$. Поэтому множества D и $\cup_{i \in Asg} A_i$ различаются в элементе x (он входит в первое множество и не входит во второе), т. е. $\cup_{i \in Asg} A_i \neq D$, ч. т. д.

Точки соотнесения. Идея возможных миров является выражением принципа индексных выражений Карнапа-Бар-Хиллела. Возможные миры, понимаемые как различные собрания индивидов с дополнительной структурой или без неё, выражают только один аспект этой идеи. Любая система структур может быть индексирована элементами подходящего множества, причём различными способами. Возьмём фиксированное множество индексов, которые будем далее называть точками соотнесения [Исмаилова, Скотт], так как для определения значения выражения необходимо «соотнести» его с точкой. Множество точек соотнесения будем обозначать Asg, отдельные точки соотнесения будем обозначать $i \in Asg$.

Прежде всего отметим индексами систему действительных индивидов, положив, возможно различные, $A_i \subseteq D$ для каждого $i \in Asg$. Важно, чтобы не предполагалось одно-однозначного соответствия между A_i и i, так как элементы множества Asg могут обладать структурой, не полностью отражающейся в отличиях одного A_i от другого.

Пример. $Asg = \{i_1, i_2\}$.

Посмотрим на эти индексы в действии. Суть состоит в том, что в самих выражениях языка делается неопределённая ссылка на индекс. В качестве примера рассмотрим выражение с квантификацией. Пусть имеется квантор $\forall x$, пробегающий по области D. Теперь введём квантор по «действительным» индивидам $\forall .x$, где точка обозначает указание на неопределённый индекс. Таким образом, даже если известно значение предиката $\mathcal{P}$, невозможно сказать, является ли значение

$$\forall .x\, \mathcal{P}(x)$$

истинным или ложным. Однако если выделить некоторое $i \in Asg$, то относительно этого индекса предложение имеет определённое истинностное значение; а именно, оно истинно, если $P(a)$ истинно для всех $a \in A_i$. Но пока не будет указано на некоторое $i \in Asg$, область значения квантифицированной переменной остаётся неизвестной.

Интерпретация. Определим интерпретацию термов τ и формул φ в точке соотнесения i. Интерпретация будет далее определяться для каждого конкретного вида термов и формул на протяжении настоящей работы. Интерпретацию формулы φ в точке соотнесения i будем обозначать

$$||\varphi||i = v.$$

Утверждения. Примем обозначения истинностных значений: будем писать 1 для обозначения *истины* и 0 для обозначения *лжи*. Этот выбор основан на том, что $2 = \{0, 1\}$ есть простой и удобный символ для обозначения множества из двух истинностных значений. Далее, утверждению Φ сопоставим функцию $||\Phi||$, определённую на Asg, со значениями из множества 2. Иначе говоря, будем писать

$$||\Phi||i = 1,$$

когда Φ *истинно относительно* i.

Соглашение. Множество функций, определённых на A со значениями из B будет обозначаться $A \to B$.

Тогда

$$||\Phi|| \in Asg \to 2,$$

если

$$||\Phi||i \in 2,$$

где $i \in Asg$.

Замечание. Для приложений множество функций $A \to B$, возможно, потребуется ограничить. Укажем два специальных способа ограничения:

1) ограничение множество наложением на него условий алгебраического характера. Пример такого ограничения: построение семантики ламбда-исчисления при помощи рассмотрения полной решётки D и ограничения множество $D \to D$ до множества непрерывных функций $[D \to D]$, что дало возможность обеспечить изоморфизм $D \cong [D \to D]$;

2) ограничение областей, по которым пробегают переменные высших типов при интерпретации. Это даёт возможность построения семантики, обеспечивающую полноту системы.

Предикаты — бинарные отношения. Будем допускать в языке выражения вида $\tau R \sigma$, где τ, σ — термы. Это позволяет поставить вопрос, как интерпретировать R. Интерпретацию R по общим правилам обозначим через $||R||$. Принимаем, что она является отношением между индивидными концептами. Таким образом,

$$||R|| \in (((Asg) \to V) \times ((Asg) \to V)) \to (Asg \to 2),$$

так что

$$||\tau R \sigma|| = ||R||(||\tau||, ||\sigma||).$$

Аналогичным образом вводится интерпретация для предикатов с большим или меньшим числом аргументных мест.

Замечание. Исключительно существенный момент предлагаемого определения состоит в том, что $||\tau R \sigma||$ **не** означает, что $||\tau R \sigma||i = ||R||i(||\tau||i, ||\sigma||i)$. Последнее справедливо **только** для экстенсиональных предикатов (см. далее).

Принятый способ интерпретации R позволяет гарантировать, что если формула *логически общезначима* (то есть истинна во всех интерпретациях и относительно всех точек соотнесения в интерпретации), то она остаётся общезначимой и тогда, когда в неё вместо *предикатных букв* подставлены *формулы*.

Частные виды предикатов. Обыкновенные (постоянные) отношения. Рассмотрим некоторые частные виды предикатов. Одним из практически наиболее интересных видов являются *экстенсиональные* предикаты.

Лучшим примером экстенсиональных предикатов могут служить обыкновенные отношения $R \subseteq V \times V$. По аналогии с индивидами ту константу языка, которая соответствует отношению R, будем обозначать $\bar{R}$. Тогда

$$(\bar{R}) \quad ||\tau \bar{R} \sigma||i = 1 \quad \overset{\circ}{\Longleftrightarrow} \quad (||\tau||i, ||\sigma||i) \in R.$$

Для того, чтобы эта элементарная формула была истинной, должны быть определены как $||\tau||i$, так и $||\sigma||i$. Однако для равенства принимается другое соглашение, потому что особая роль равенства обуславливает различные соглашения, принимаемые для большего единообразия среди общезначимых формул.

Интересная деталь: если все атомарные формулы построены с помощью $=$ или различных $\bar{R}$ и если единственными термами являются переменные или дескрипции, то дескрипции могут быть полностью элиминированы. В общем случае это не так.

Принцип экстенсиональности. Принцип экстенсиональности для $\bar{R}$ формулируется следующим образом:

$$\tau = \tau' \wedge \sigma = \sigma' \wedge \tau \bar{R} \sigma \Rightarrow \tau' \bar{R} \sigma'.$$

Не каждый экстенсиональный в этом смысле предикат происходит от обычного R (см. далее).

Экстенсиональные предикаты — общий случай. Предположим, что каждому $i \in Asg$ сопоставлено отношение $R_i \subseteq V \times V$ с возможно различными R_i для различных i. Можно рассматривать R как функцию, значениями которой служат отношения. Затем можно ввести в язык константу $\tilde{R}$ (черточка $(-)$ обозначает *константное* отношения, а тильда $(\sim)$ — *переменное* отношение), значение которой задаётся так:

$$(\tilde{R}) \quad ||\tau \tilde{R} \sigma||i = 1 \quad \overset{\circ}{\Longleftrightarrow} \quad (||\tau||i, ||\sigma||i) \in R_i.$$

Замечание. Можно показать, что указанные выше переменные отношения действительно представляют наиболее общий вид экстенсионального предиката.

Структура экстенсионального предиката — пример. Рассмотрим приведённый выше пример построения дескрипций. Построим в условиях этого примера экстенсиональный предикат общего вида.

Пусть $\Phi(x)$ экстенсиональна. Это значит, что

$$||\tau||i = ||\sigma||i \quad \Rightarrow \quad ||\Phi\tau||i = ||\Phi\sigma||i.$$

Пусть $||\tau||i = ||\sigma||i = b \in V^+$. В соответствии с тем, что $||\Phi|| : C \to (Asg \to 2)$, где $C = Asg \to V^+$ — семантический домен концептов, будем строить $||\Phi||$ в виде двуместной функции $C \times Asg \to 2$. Обозначим эту функцию через φ.

Поскольку $c_1 i_1 = c_2 i_1 = c_3 i_1 = a_1$, должно быть $\varphi(c_1, i_1) = \varphi(c_2, i_1) = \varphi(c_3, i_1) = t_1$. Применяя аналогичные соображения к другим значениям функции, имеем следующую таблицу значений φ.

φ	i_1	i_2
c_1	t_1	t_4
c_2	t_1	t_5
c_3	t_1	t_6
c_4	t_2	t_4
c_5	t_2	t_5
c_6	t_2	t_6
c_7	t_3	t_4
c_8	t_3	t_5
c_9	t_3	t_6

Отметим, что принятое соглашение относительно равенства требует отождествления значений функции на элементе $\perp$ (т. е. невозможно принять $\varphi(c_7, i_1) \neq \varphi(c_8, i_1)$ и т. д.).

Замечание. Представленный вид экстенсионального предиката показывает, что в общем случае он возникает из переменного отношения, но не на множестве D (или V), а на множестве D^+ (или V^+).

Предикат существования. Предикат $A_i \subseteq D$ как раз является предикатом отмеченного выше вида. Поэтому можно ввести в язык константу $\tilde{A}$, описывающую существование в данной точке соотнесения:

$$(\tilde{A}) \quad ||\tilde{A}\tau||i = 1 \quad \overset{\circ}{\Longleftrightarrow} \quad ||\tau||i \in A_i.$$

Далее заметим, что

$$\forall.x\ \Phi(x) \Leftrightarrow \forall x\ [\tilde{A}(x) \Rightarrow \Phi(x)]$$

истинно, а это и означает, что $\forall.$ есть ограниченный квантор. Теперь легко заметить, что

$$[\tilde{A}(x) \Rightarrow \Box\Phi(x)]$$

и

$$\Box[\tilde{A}(x) \Rightarrow \Phi(x)]$$

являются независимыми формулами. Это проясняет явления, возникающие при интерпретации кванторов.

Пропозициональные связки. Для объяснения значения пропозициональных связок дадим обычные семантические определения.

$$\begin{array}{lll}
(\neg)\ ||\neg\Phi||i = 1 & \overset{\circ}{\Longleftrightarrow} & ||\Phi||i = 0 \\
(\wedge)\ ||\Phi \wedge \Psi||i = 1 & \overset{\circ}{\Longleftrightarrow} & ||\Phi||i = 1 \text{ и } ||\Psi||i = 1 \\
(\vee)\ ||\Phi \vee \Psi||i = 1 & \overset{\circ}{\Longleftrightarrow} & ||\Phi||i = 1 \text{ или } ||\Psi||i = 1
\end{array}$$

Отметим, что принятая интерпретация пропозициональных связок в данном случае *экстенсиональна* — в каждой точке соотнесения значение формулы, главным знаком которой является пропозициональная связка, является функций значений подформул этой формулы в той же точке соотнесения.

Константы. Введём в язык (для интерпретации кванторов и дескрипций) константы

$$\bar{a} \text{ для всех } a \in V.$$

Обратим особое внимание на то, что константы вводятся для всех виртуальных индивидов (не только возможных).

Предполагаем далее, что соответствие между a и $\bar{a}$ является взаимно однозначным.

Замечание. Таким образом, имеем два предположения:

1) $\bar{a}$ определено для всех a, т. е. ни один виртуальный объект не пропущен — все виртуальные объекты можем поименовать;

2) $\bar{a}$ определено однозначно, т. е. каждый виртуальный объект получает уникальное имя — нет двух констант, которые соответствуют одному и тому же объекту.

Для задач проектирования информационных систем специального вида необходимо иметь в виду возможность принятия других предположений. Представляется, однако, что они могут быть формализованы на основе принятой выше простейшей модели.

Кванторы. При интерпретации кванторов вводим два квантора с разными свойствами.

$$(\forall)\ \ ||\forall x\Phi(x)||i = 1 \quad \overset{\circ}{\Longleftrightarrow}\ ||\Phi(\bar{a})||i = 1 \text{ для всех } a \in D$$
$$(\forall.)\ ||\forall .x\Phi(x)||i = 1 \quad \overset{\circ}{\Longleftrightarrow}\ ||\Phi(\bar{a})||i = 1 \text{ для всех } a \in A_i$$

и аналогично для $(\exists)$ и $(\exists.)$.

Замечание. Отметим, что в этом месте происходит отступление от классического определения Тарского — не вводятся последовательности переменных и т. д., в результате чего семантика определяется только для замкнутых формул.

Значения термов. Пусть τ — терм. Его значением будем считать частично определённую функцию с областью определения Asg. Какова область значения этой функции? Считаем, что значения дескрипций должны принадлежать только D, а значения индивидных термов в общем случае должны лежать в V. Используем $(Asg) \to V$ для обозначения множества *частичных* функций, в то время как $Asg \to V$ есть множество *всюду определённых* функций. Тогда

$$||\tau|| \in (Asg) \to V.$$

Замечание. Есть три способа образования термов:

1) константы;

2) дескрипции;

3) интенсиональные связывающие операторы.

Только последний способ образования может приводить к появлению виртуальных объектов общего вида.

Замечание. Обозначим неопределённое значение терма через $\perp$. Тогда факт, что терм τ в точке соотнесения i имеет неопределённое значение, может быть выражен через $||\tau||i = \perp$.

Теперь частично определённые функции на V могут рассматриваться как полностью определённые функции на $V \cup \{\perp\}$. Обозначим $V^+ = V \cup \{\perp\}$, тогда, например, частично определённая функция из Asg в V может рассматриваться как полностью определённая функция из Asg в V^+. Примем также аналогичное соглашение для D (т. е. $D^+ = D \cup \{\perp\}$).

Установим далее на виртуальных индивидах отношение порядка $\sqsubseteq$, считая $\perp \sqsubseteq b$ для любого $b \in V$. Это отношение продолжается на концепты, т. е. функции $Asg \to V^+$, что позволяет сформулировать следующее предложение.

Предложение. Концепты образуют решётку, в которой отношение $\sqsubseteq$ задаёт частичный порядок.

Замечание. В случае определения топологии на Asg продолжение отношения частичного порядка на концепты может быть согласовано с этой топологией. Далее принимается простейшее определение

$$f \sqsubseteq g \quad \Leftrightarrow \quad \forall b \in V^+ \quad f(b) \sqsubseteq g(b),$$

согласованное с дискретной топологией на Asg.

Равенство. Дополним определение формулы с целью дальнейшего анализа индивидов. Будем считать, что из двух термов σ и τ может быть образовано равенство $\sigma = \tau$, представляющее собой формулу.

$$(=) \quad ||\sigma = \tau||i = 1 \quad \overset{\circ}{\Longleftrightarrow} \quad ||\sigma||i \text{ и } ||\tau||i \text{ оба либо определены и равны, либо неопределённы.}$$

На основе $(=)$ можно доказать, что следующие две формулы всегда истинны для всех $i \in Asg$:

$$\tau = \tau \text{ и } (\sigma = \tau \wedge \tau = \theta \Rightarrow \sigma = \theta).$$

Следующая формула

$$\sigma = \tau \wedge \Phi(\sigma) \Rightarrow \Phi(\tau)$$

в общем случае не проходит.

Замечание. Проходит только для экстенсиональных Φ: так как $||\Phi\tau|| = ||\Phi||(||\tau||)$, может быть $||\Phi\sigma||i = ||\Phi\tau||i$ но при этом $||\Phi\sigma|| \neq ||\Phi\tau||$, так как $||\sigma|| \neq ||\tau||$.

Замечание. Соглашение относительно равенства (термы, имеющие неопределённое значение в заданной точке соотнесения, считаем равными в этой точке) приводит к тому, что в семантике оказывается допустимым рассматривать *только одно* неопределённое значение (т. е. не вводятся $\perp_1$, $\perp_2$ и т. д.). Это согласуется с замечанием к предыдущему пункту.

Оператор необходимости. Простейшим пропозициональным интенсиональным оператором является оператор необходимости $\Box$.

$$(\Box) \quad ||\Box\Phi|| = 1 \quad \overset{\circ}{\Longleftrightarrow} \quad ||\Phi||j = 1 \text{ для всех } j \in Asg.$$

Возможность $\Diamond$ определяется двойственным образом.

При желании можно сказать, что $\Box$ имеет в качестве интерпретации $||\Box||$. В самом деле, $||\Box||$ имеет тот же логический тип, что и $||\neg||$:

$$||\Box||, ||\neg|| \in (Asg \to 2) \to (Asg \to 2).$$

Они являются отображениями из множества пропозициональных концептов на множество пропозициональных концептов. Разница в том, что один из них экстенсионален, а другой нет. Можно написать

$$||\Box\Phi|| = ||\Box||(||\Phi||),$$

что позволяет, в частности, представить интерпретацию операторов в аппликативной форме [1].

Тождество. Теперь, когда логический словарь стал более богатым, можно рассмотреть некоторые сложные фразы, например, $\Box(\tau = \sigma)$. По определению:

$$||\Box(\tau = \sigma)||i = 1 \quad \overset{\circ}{\Longleftrightarrow} \quad ||\tau||j = ||\sigma||j \text{ для всех } j \in Asg.$$

Причём правое равенство выполняется и в случае, когда оба его терма неопределённы. Это означает, что $||\Box(\tau = \sigma)||$ является концептом истинного выражения

тогда и только тогда, когда выполняется $||\tau|| = ||\sigma||$, т. е. если $||\tau||$ и $||\sigma||$ есть один и тот же индивидный концепт. Введём для этого понятия особый символ:

$$\tau \equiv \sigma \quad \overset{\circ}{\leftrightarrow} \quad \Box(\tau = \sigma).$$

Так различается экстенсиональное равенство $(\tau = \sigma)$ и интенсиональное $(\tau \equiv \sigma)$. Назовём это более строгое понятие *тождеством*.

Подстановочность. На основе конвенции относительно интенсионалов (которая может быть обоснована семантикой) легко может быть выведено, что если τ и σ не имеют свободных переменных, то

$$\tau \equiv \sigma \wedge \Phi(\tau) \to \Phi(\sigma)$$

верно для тождества, тогда как аналогичный принцип для равенства не выполняется.

Замечание. *Построить описание отражения этого свойства в программных системах, получаемых на его основе.*

Соответственно если Φ и Ψ не содержат свободных переменных и Θ' получается из Θ заменой вхождений Φ на Ψ, то

$$\Box(\Phi \leftrightarrow \Psi) \wedge \Theta \to \Theta'$$

также выполняется. Таким образом, предлагаемая логическая система имеет известную степень экстенсиональности: выполняется подстановочность для *логически эквивалентных* формул.

Необходимость и кванторы. Интересная комбинация операторов и кванторов содержит $\Box$ и $\forall$. Верна, например, такая формула:

$$\Box\forall x\Phi(x) \leftrightarrow \forall x\Box\Phi(x),$$

а такая

$$\Box\forall.x\Phi(x) \to \forall.x\Box\Phi(x),$$

не верна; и, таким образом, для '$\forall$.'не верна импликация в обе стороны.

Отношения альтернативности. Более общо, пусть $\rho \subseteq sg \times Asg$ будет бинарным отношением на Asg, тогда соответствующий ρ оператор $\boxed{\rho}$ определяется так:

$$(\boxed{\rho}) \quad ||\boxed{\rho}\Phi|| = 1 \quad \overset{\circ}{\Longleftrightarrow} \quad ||\Phi||j = 1 \text{ для всех } j \in Asg, \text{ где } i\ \rho\ j.$$

Другими словами, ρ есть отношение альтернативности, заданное на возможных мирах.

Получается чрезвычайно интересный оператор. В действительности это оператор довольно специального вида — он может стать предметом рассмотрения, например, во временной логике. И это, конечно, только один из множества различных типов операторов.

Общий монадический пропозициональный оператор. Теперь оказывается достаточно легко перейти к самому общему монадическому пропозициональному оператору. Сопоставим каждому $i \in Asg$ семейство G_i подмножеств Asg. Тогда можно определить следующий оператор:

$$(\boxed{\mathscr{G}}) \quad ||\boxed{\mathscr{G}}\Phi|| = 1 \quad \overset{\circ}{\Longleftrightarrow} \quad \{j \in Asg : ||\Phi||j = 1\} \in G_i,$$

что является просто тривиальной переформулировкой утверждения

$$||\boxed{\mathscr{G}}|| : (Asg \to 2) \to (Asg \to 2).$$

Операторы, связывающие переменные. До сих пор говорилось только об интенсиональных отношениях или интенсиональных операторах, действующих на предложения. Однако для описания практически интересных примеров требуется бóльшая степень общности, хотя достичь её сложно. В частности, требуются интенсиональные двойники экстенсиональных операторов типа кванторов и операторов дескрипции. Рассмотрим теперь прежде всего оператор $, который связывает одну переменную и навешивается на один терм и одну формулу; можно использовать его в языке в следующих выражениях:

$$\$x[\tau(x), \Phi(x)].$$

Теперь необходимо выбрать: считать ли это термом или формулой.

Примем первое, то есть будем считать $ термообразующим оператором. Значениями этого сложного терма могут быть и виртуальные объекты (ср. выражение

$$\bigcup_{x \in A} \tau(x)$$

в теории классов: оно существенно зависит от A и $\tau(x)$). Принимается, что его значениями будут (виртуальные) индивидные концепты. Так, принимается, что

$$||\$x[\tau(x), \Phi(x)]|| \in (Asg) \to V.$$

Каким же образом это выражение зависит от $\tau(x)$ и $\Phi(x)$? Эти выражения являются функциями от x, а все связанные переменные пробегают по D. Поэтому можно утверждать, что

$$||\$|| \in U \to ((Asg) \to V),$$

где

$$U = (D \to ((Asg) \to V)) \times (D \to (Asg \to 2)),$$

причём $D \to ((Asg) \to V)$ есть множество всех функций, значениями которых являются концепты, а областью определения — D; $D \to (Asg \to 2)$ есть множество заданных на D функций, значениями которых служат пропозициональные концепты. Таким образом,

$$(\$) \quad ||\$x[\tau(x), \Phi(x)]||i = ||\$||(f, F)i,$$

где $finD \to ((Asg) \to V)$ и $F \in D \to (Asg \to 2)$ определяются так:

$$f(a)i = ||\tau(\bar{a})||i,$$
$$F(a)i = ||\Phi(\bar{a})]||i \text{ для всех } a \in D.$$

Немного затруднительно привести действительно яркие и актуальные примеры таких операторов, которые существуют сами по себе: обычно они бывают составлены из других, более простых операторов. Например,

$$\mathbf{I}y\,\forall x\,\Box[\Phi(x) \to y = \tau(x)]$$

как раз и является сложным оператором.

Не исключено, что тот или иной оператор всегда можно свести к такому сложному оператору, в котором переменные связываются только квантором и операторами дескрипции. Эта проблема порождает дальнейшие размышления.

В случае, когда $ есть формулообразующий оператор, надо внести только одно изменение: сделать так, чтобы его значениями были пропозициональные концепты.

Чтобы ещё раз указать на то, какие принципы взаимозаменимости существуют в этой логике, укажем, что здесь имеют место следующие формулы:

$$\forall x\,\Box\tau(x) = \tau'(x) \wedge \forall x\,\Box[\Phi(x) \leftrightarrow \Phi'(x)] \to$$
$$\$x\,[\tau(x), \Phi(x)] = \$x\,[\tau'(x), \Phi'(x)].$$

Заметим, что $\Box$ есть единственный модальный оператор, который подходит для выражения условий взаимозаменимости — это одна из главных причин для включения его в число *логических* констант системы.

Заключение

В работе предложены формализованные средства специализации концептуальных моделей, в том числе используемых при разработке ИС. Исследуются возможности корректного отграничения различных видов кванторных операций для повышения адекватности получаемых информационных моделей.

Средства специализации построены на основе использования пропозициональных концептов различной степени интенсиональности. Выполнено исследование различных конструкций, используемых при концептуальном моделировании:

— введён интенсиональный язык описания ПО, включающий интенсиональные описания, как основа для последующей специализации, определены его синтаксис и семантика;

— определение экстенсиональных и интенсиональных предикатов обеспечивает возможность комбинирования различных способов оперирования объектами, включающих как чисто экстенсиональные способы, соответствующие пропозициональным связкам, так и специализированные кванторные операции, для которых формулируются условия применения;

— введены специализированные операторы, позволяющие связывать результаты вычисления операций в различных точках соотнесения;

— введены связывающие операторы, обеспечивающие возможность построения интенсиональных двойников рассматриваемых экстенсиональных понятий, в том числе кванторных операций. Операции включают возможности традиционных средств работы с данными (репликация и т. д.), но не ограничиваются ими.

Исследование показывает возможность построения моделей с управляемой степенью интенсиональности за счёт согласованного использования экстенсиональных и интенсиональных конструкций. В частности, специализация концептуальных моделей может быть обеспечена за счёт использования комбинаций интенсиональных операторов. Дальнейшее исследование таких операторов, как ожидается, позволит определить специализированные языковые конструкции и методы их интерпретации, обеспечивающие возможности интенсионального программирования.

Благодарности

Работа является обобщением ряда результатов проектов 10-07-00417-а, 11-07-00305-а, 12-07-00554-а, частично поддержанных грантами РФФИ.

Список литературы

1. Вольфенгаген В. Э. Комбинаторная логика в программировании. Вычисления с объектами в примерах и задачах. — М.: Институт ЮрИнфоР-МГУ, 2008.
2. Вольфенгаген В. Э. Логика. Конспект лекций: техника рассуждений. — М.: JurInfoR Ltd., АО "Центр ЮрИнфоР 2004.
3. Вольфенгаген В. Э. Методы и средства вычислений с объектами. Аппликативные вычислительные системы. — М.: JurInfoR Ltd., АО "Центр ЮрИнфоР 2004.
4. Исмаилова Л. Ю., Косиков С. В. Информационное моделирование динамичных рассуждений на примере юридической деятельности. — Технологии информационного общества – Интернет и современное общество: Материалы Всероссийской объединенной конференции. Санкт-Петербург, 20-23 ноября 2001 г. СПб.: Изд-во С.-Петерб. ун-та, 2001. с. 36-43.
5. Исмаилова Л. Ю., Косиков С. В. От поисковых систем к информационному моделированию рассуждений в Интернет-среде. — Технологии информационного общества – Интернет и современное общество: труды V Всероссийской объединенной конференции. Санкт-Петербург, 25-29 ноября 2002 г. СПб.: Изд-во С.-Петерб. ун-та, 2002. с. 97-100.
6. Исмаилова Л. Ю., Косиков С. В. Соотнесение семантик против семантики соотнесений. Технологии информационного общества – Интернет и современное общество: труды VI Всероссийской объединенной конференции. Санкт-Петербург, 3-6 ноября 2003 г. СПб.: Изд-во Филологического факультета СПбГУ, 2003. с. 17-22.
7. Монтегю Р. Прагматика и интенсиональная логика. — в кн. Семантика модальных и интенсиональных логик. Под ред. В. А. Смирнова. - М., Прогресс, 1981.
8. Скотт Д. Советы по модальной логике. — в кн. Семантика модальных и интенсиональных логик. Под ред. В. А. Смирнова. - М., Прогресс, 1981.

Коллективная информатика в решении инфраструктурных проблем

Бельтюков А.П., Маслов С.Г.

Кафедра ТОИ, ФИТ и ВТ, «Удмуртский государственный университет»,
Ижевск, 426034, ул. Университетская, 1, корп.6, офис 312
belt@udsu.ru, msh.sci.it@gmail.com

Аннотация. Выявляется фундаментальный базис информационной технологии решения инфраструктурных проблем организации жизненных процессов, которая обеспечивает устойчивое развитие качества жизни. Рассматривается среда, контекст решения проблемы и сама проблема. В связи с этим вводится понятие «коллективная информатика».

Введение

На современном этапе, по крайней мере, два фактора препятствуют устойчивому развитию качества жизни: во-первых – введение «института элитарности» (есть «элита» и «простые люди»), во-вторых, глубокий системный кризис в обществе. Системный кризис является проявлением тупика в развитии «общества потребления». Оба фактора усиливают друг друга, лишают человека смысла или мотивации в жизни. Фактически человек становится пассивным потребителем, «уничтожителем ресурсов». Важна не только пассивность отдельных индивидов, но и общественных групп в целом. Другая проблема – продвижение приоритета материального над духовным. Чтобы действительно продвигать качество жизни, необходимо творческое проявление личности и общества в целом. Большие надежды возлагались на информационные технологии, но на самом деле, они стали катализатором негативных, разрушающих процессов. Вот поэтому нужно новое позитивное направление в информатике, консолидирующее творческий потенциал Человека на разумное потребление и устойчивое развитие качества жизни. Таким направлением может быть «коллективная информатика», которая развивает коллективный интеллектуальный потенциал для решения сложной инфраструктурных проблем, направленных на устойчивое развитие качества жизни, обеспечивающих симбиоз человека со средой обитания (на идеальном и материальном уровне).

Настоящая работа развивает идеи работ [1,2]

Ключевая проблематика

Под коллективной информатикой далее будем понимать согласованное коллективное поведение в построении идеального информационного объекта, позволяющего управлять материальными и идеальными объектами в жизненном процессе этого коллектива. Можно выделить следующие ключевые проблемы коллективной информатики:

- умение чётко разделить понятия среды, контекста и решаемой проблемы, которое, в частности, предусматривает расслоение контекста для привлечения или создания более эффективных и адекватных средств процесса решения проблемы; среда - совокупные доступные коллективные знания,
- реализация индуктивно-дедуктивного подхода к решению проблемы; здесь индуктивный этап - выявление повторений и аналогий, ассоциаций, обобщение, выявление инвариантов, идеализация, формализации; дедуктивный этап связан с выделением проблемы, цели, новой задачи, деформализацией, детализацией, конкретизацией, логическим выводом,
- организация компьютинга в слоях описания - эффект концентрации, адекватных форм в процессе решения задачи.
- требуется выявление базисов представления информации и их систематизация, метапроцессов, рефлексивности, самоорганизации.

Развитие технологии

Двунаправленные встречные потоки информации индуктивно-дедуктивного характера образуют цикл формирования системы знаний. В этом цикле создаётся адекватный контекст решения проблемы. Развитие этого цикла требует его осмысления на уровне коллективной деятельности. В рамках технологии решения проблемы необходимо выделение внутрисубъектного и отчуждённого знания на уровне индивида и коллектива. В частности, следует решать проблему индивидуального принятие коллективного знания для обеспечения непрерывности и целостности технологического процесса решения проблем. В рамках этого технологического процесса развиваются методы взаимодействия:

- субъект-субъект,
- субъект-машина,
- машина-машина (в частности, саморазвитие);

и более сложные взаимосвязи:

- субъект-машина-субъект
- субъект-машина-машина-субъект (например, социальная сеть, ноосорсинг),
- машина-субъект-машина,
- машина-субъект-субъект-машина.

Концентрация и кристаллизация знаний приводит к образованию кластеров образующих компонент: людей, знаний, отдельных проблем, технологий и т.д.

Заключение

Постановка проблемы, приведенная выше, охватывает не только технологические аспекты человеческой деятельности, но и перестройку всех жизненных процессов профессиональной, культурной и общественной жизни. Повышение качества жизни производится на основе развития творческого потенциала Человека, формирующего устойчивого развитие человека, общества и природы на основе знаний, в противовес принципам общества потребления.

Работа выполнена при финансовой поддержке РФФИ (проект № 11-07-00640-а).

Литература

1. Маслов С.Г. *Системный анализ и управление интеллектуальными ресурсами в ИТ-сфере* //Труды V Международной конференции «Идентификация систем и задачи управления» SICPRO'2006. Москва, 30 января - 2 февраля 2006г. Институт проблем управления им. В. А. Трапезникова РАН - М: ИПУ РАН, 2006, 960-969 с.
2. Славин Б.Б. *Конвергенция архитектур социальной и корпоративной информационной среды человека* // Бизнес-информатика – 2012. – № 2(20). – С. 3–9.

Дескриптивно-конструктивные аспекты е-публикаций

Маслов С.Г.

Кафедра ТОИ, ФИТ и ВТ, «Удмуртский государственный университет»,
Ижевск, 426034, ул. Университетская, 1, корп.6, офис 312
msh.sci.it@gmail.com

Аннотация. Переход к экономике знаний требует новых подходов к капитализации знаний. В докладе обсуждаются некоторые аспекты и проблемы формирования и использования е-публикаций (*ie-pub & ie-fund*), которые коренным образом изменяют как процессы образования, так и процессы профессиональной деятельности. Обсуждаются проблемы представления формы и содержания е-публикаций, а также характер взаимодействия с ними субъекта.

Введение

Печатные издания как основной носитель отчужденного знания теряют свое влияние, уступая место е-публикациям (или е-изданиям). И дело тут не в цене, а в разнообразии и гибкости форм представления актуального знания, в активном характере системы знаний, дефиците времени у современного человека. Поиски лучшей формы изданий и публикаций уже породили: бумажные издания, аппаратные издания (*e-book, e-reader*), онлайн-издания (*books online*), реалистичные издания (*realistic electronic book, 3D book, physical book*), конструктивные электронные издания (*constructive electronic publishing*), умные издания (*smart book*), активные издания (*active book*), «ожившие» издания (*haunted book*), «живые» издания (*live book*) [1-7].

Прорыв в эффективности распространения фундаментальных, передовых и актуальных знаний требует синтеза представлений о читателях, писателях, о процессах прочтения и написания изданий, а также о формах и содержании самих изданий. Новое понимание связано с тем, что е-публикация становится фрагментом единой технологии связывающей идею с ее реализацией в форме доступной для эффективного использования (*ie-pub & ie-fund*).

Ключевая проблематика

Реализация нового понимания е-публикации включает следующие ключевые моменты:

- *точки зрения на контент* (когнитивная, конструктивная, креативная, конативная, коммуникативная, рефлексивная, объективная, субъективная, методологическая, материальная или физическая, идеальная, …);
- *системное описание контента и форм* (морфологическое, функциональное, атрибутивное, генетическое, либернетическое);
- *процессы формализации и деформализации* (идеальные, материальные, абстрактные, конкретные объекты и процессы, инварианты {свойств, правил, уравнений, форм, преобразований, объектов});
- *процессы поиска, навигации, вывода, построения и преобразования контента и управления им*:
    - уровни управления (параметрический, структурный, базисный, рефлексивный);
    - формирование постановок задач;
    - барьеры, ошибки, препятствия, противоречия и методы их преодоления;
- *стратегии и тактики компьютинга;*
- *меры и оценки* (L^rT^s – язык и система, сложностные оценки,…)
- *проблемы коллективного взаимодействия* (проецирование, прочтение, написание, …).

Развитие технологии

Проведенный системный анализ форм изданий и публикаций позволяет выделить, в качестве целевых свойств системы знаний е-публикации, следующие: целостность, технологическую непрерывность, вычислимость, самоорганизуемость, а также симбиоз субъекта (скрытая и осознанная внутрисубъектная система знаний) с внешней системой знаний (в форме ie-pub & ie-fund).

Накопление изданий в библиотеках с последующим поиском нужной информации заменяется новым средством их организации – *фондом* (*ie-fund*), в котором знания сначала упорядочены по следующим атрибутам: *личность – проблема – идея – термин*. Потом сочетание поиска с систематизированными постановками задач, разнонаправленость процесса взаимодействия субъекта с *ie-pub & ie-fund* позволяет получать ответы и решать проблемы, используя наиболее адекватные средства. Основой создания *ie-pub & ie-fund* является активная расслоённая терминологическая система [8-9], которая в отличие от систем онтологий опирается на разнообразие взаимосвязанных форм представления информации. Важным фактором развития технологии е-публикаций является интеграция формализованных и эмоциональных форм представления знаний.

Заключение

Роль е-публикаций безусловно будет возрастать, развиваясь в направлении функционального расширения и систематизации знаний и форм их представления. Важно, что эти формы не просто переходят в активное состояние, а то, что они становятся единицами знаний «вшитыми» в управление и жизненные процессы как на идеальном, так и на материальном уровне. Это, в свою очередь, создает среду не только для повышения качества жизни, но и синтеза образовательных, научно-исследовательских и промышленных процессов, т.е. реализации человеческого потенциала. Однако важно помнить о том, что увлекаясь распространением знаний необходимо помнить о целостности мира, о единстве человека, общества и природы, т.е. разумности жизни.

Работа выполнена при финансовой поддержке РФФИ (проект № 11-07-00640-а).

Литература

1. Eva Siegenthaler, Pascal Wurtz, Rudolf Groner. *Improving the Usability of E-Book Readers*// JUS Vol. 6, Issue 1, November 2010, pp. 25-38.
2. Catherine C. Marshall Collection-Level Analysis Tools for Books Online. http://freedownloadebooks.info/ebooks/collectiono-level-analysis-tools-for-books-online.html (дата обращения: 24.06.2012).
3. Mark T.J. Carden, *E-Books Are Not Books.* http://www.sciweavers.org/publications/e-books-are-not-books (дата обращения: 24.06.2012).
4. Yi-Chun Chu, David Bainbridge, Matt Jones and Ian H. Witten. *Realistic Books: A Bizarre Homage to an Obsolete Medium?* //Proceeding JCDL'04 Proceedings of the 4th ACM/IEEE-CS joint conference on Digital libraries, 78-86.
5. Ilinykh M. S., Maslov S. G. *Constructive Electronic Publishing Based on Notion System and Interactive Dynamic Images*//Knowledge-Based Media Analysis for Self-Adaptive and Agile Multi-Media, Proc. of the European Workshop for the Integration of Knowledge, Semantics and Digital Media Technology, EWIMT 2004, November 25–26, 2004. — London, UK.
6. *Активная система изучения комбинаторной логики в программировании* (Active CloP). http://jurinfor.ru/game/game63.php (дата обращения: 24.06.2012).
7. *Активная система изучения конструкций языков программирования* (Active PLC). http://jurinfor.ru/game/game62.php (дата обращения: 24.06.2012).
8. Маслов С.Г. Бельтюков А.П. *Активная расслоенная терминологическая система*// Устойчивое инновационное развитие: проектирование и управление. - том 7 № 3 (12), 2011 - С.103-113.
9. Maslov, S.G. Beltukov, A.P. *Navigation and synthesis for stratified terminology environment*// Proc. of the 13-th international workshop on Computer Science and Information Technologies (CSIT'2011), Garmisch-Partenkirchen, Germany 2011, UGATU, UFA, pp. 168-171.

Описание процессов в инновационных проектах

Степанова Е.Б.[1,2], Тиренни А.[2]

[1] Кафедра Системного анализа, НИЯУ МИФИ,
Москва, 115409 РФ, Каширское шоссе, 31.
[2] International Research Group «Integrated Systems», Zurich, Switzerland
Elena.Stepanova@mephi.ru

Аннотация. Обсуждаются особенности моделирования процессов в проектах, представляющих новые решения или применение и расширение алгоритмов для использования в новой области деятельности. Акцент ставится на применении Международных стандартов и архитектур моделирования при подготовке заявки в инвестиционные фонды, что актуально на этапе начала деятельности России в ВТО.

Введение

Анализ и экспертиза инновационных проектов подразумевают как работу с моделями и описаниями новых процессов, так и подтверждение самого факта создания заявителями нового решения [1]. В начале работы России в качестве члена ВТО имеются проблемы согласования, или гармонизации, требований национальных и Международных стандартов, языковые сложности, процедурные, на одно из первых мест выходит роль предварительного обсуждения работы на конференциях, публикация в открытых источниках информации. Формат этой деятельности лежит о основе инициатив WIPO по закреплению и подтверждению прав интеллектуальной собственности.

В задачах разработки информационных сред поддержки инновационных процессов выполнение анализа базируется на методологии управления большими системами. Для обеспечения целостности процесса статического и динамического моделирования необходима интеграция результатов различных описаний в рамках единого проекта [2]. Поэтому становление регистра «Industrial and applied mathematics» для отечественных публикаций актуально для предприятий ведущих отраслей.

Стандартизация, ориентированная на моделирование процессов

Применение процессно-ориентированной стандартизации лежит в одном русле с методами описания интегрированных сред со сложной структурой биз-

нес-приложений и контуров, предназначенных для коллективной отработки различных видов задач. При этом имеется зависимость выбора методологии и технологии моделирования процессов от уровня сложности и типа задачи, особености переноса моделей между платформами [3]. Следует отметить, что именно развитие стандарта Менеджмента качества привело к требованиям применять методологию процессно-ориентированного проектирования системы предприятия, и информационной среды поддержки комплекса ключевых процессов соответственно. Интерактивность процесса поддержки комплекса процессов, акцент на актуализацию документирования в системе делает задачу описания жизненного цикла ведущей. Жизненный цикл проекта взаимосвязан с жизненным циклом предприятия и жизненным циклом продукта, на выпуск которого направлена дятельность, а также взаимосвязан с состоянием и развитием информационной среды поддержки процессов предприятия.

Гармонизация, то есть сведение воедино подходов на основании международных стандартов, акцептированных в странах – членах ВТО, и национальных стандартов, которые применялись ранее в этих странах, касается в первую очередь группы документов, используемых крупными компаниями в совместных проектах.

Развитие технологии

Моделирование и визуализация нотаций ключевых процессов особенно существенно в крупных проектах, где математических аппарат описания сложный, используется тензорный анализ, количество ключевых показателей велико, требуются как статические, так и динамические описания, комплексирование различных методов и типов моделей.

Модель и описание жизненного цикла проекта представляют собой обобщающий комплекс фактографической информации по процессам разработки и реализации проекта, и возникает потребность в использовании сред поддержки процесса моделирования. Если в ходе проектирования не использовать программное средство, соответствующее уровню сложности и типу проекта, а диаграммы прорисовывать с помощью не адекватных цели работы графических редакторов, то упущения и ошибки весьма вероятны. Проблема целостности данных остается открытой, несмотря на имеющиеся принципиальные подходы к обеспечению целостности и качества данных в реляционных средах.

Отдельно стоит анализ отклонений и обеспечение возможности моделирования и выработки корректирующих действий по управлению, например, используя адаптивную модель управления. Ошибки могут возникать при трансляции данных из одной системы в другую. Поэтому к числу актуальных относится проблема разработки технологии переноса математической модели, осуществление синхронизации данных между платформами, задействованными в сложной конфигурации регионально распределенной среды. На первое место выступает представление о менеджменте сложного распределенного проекта, в котором занято много специалистов различной квалификации.

Это позволяет снять ряд затруднений, которые в первую очередь связаны с

применением отличающейся от ISO модели жизненного цикла и управления проектом.

Заключение

Гармонизация описаний процессов, выбора архитектуры и нотаций для моделировния, визуализации, являются составляющими цикла подготовки и экспертизы инновационного проекта, в котором в связи со сложностью языка авторы из России - нового члена ВТО, - могут опереться на разработку и реализацию мероприятий по закреплению прав интеллектуальной собственности, стандартизацию. Данный доклад приурочен к выходу монографии в серии «Global Certification in Process Management», в которой редакторы во взаимодействии с авторами различных выпусков представляют ряд проблем, актуальных для разработчиков и менеджеров крупных проектов, которые связанны с работой специалистов различной квалификации в условиях требований процессного менеджмента и сертификации. В первом издании, которое начинает серию, авторы выделили и формализовали некоторые задачи, свяанные с управлением инновационными предприятиями и финансово-промышленными структурами в условиях, когда новые страны вступают в ВТО, и имеются различия между стандартными описаниями проектов разными группами специалистов.

Литература

1. Степанова Е.Б., Урезченко В.М. Проектная и конкурсная инновационная деятельность: от учебного курса до гранта научного фонда. Теория активных систем / Труды международной научно-практической конференции (14 – 16 ноября 2011, г. Москва, Россия). Том 3. Общая редакция – В.Н. Бурков, Д.А. Новиков. – М.: ИПУ РАН, 2011. – 267 с. – С.195-200.
2. Степанова Е.Б., Тимофеев А.В., Максимов Н.В. Разработка и использование методов описания процессов и документопотоков для систем терминального управления документооборотом. Аудит и финансовый анализ. 2012. №4. С. 457-462.
3. Степанова Е.Б., Тиренни А., Чеботарев А.А. Мобильная SAP-платформа для персонала территориально-распределенных проектов. / Сборник научных трудов XI Международной конференции «Развитие единой образовательной информационной среды: на пути к обществу знаний». (25-26 сентября 2012, г. Томск, Россия). – Томск: Изд-во Томского государственного университета, 2012.

Кластеризация на основе семантического анализа

Бохуленков А.С.

Факультет “КиБ”, Кафедра кибернетики(№22), НИЯУ МИФИ
Москва, 115409 РФ, Каширское ш., д. 31
bohulenkov14@gmail.com

Аннотация. С момента появления web в сети с каждым днем становится все больше и больше динамически меняющейся, неупорядоченной информации, хаотическая природа которой затрудняет её поиск и навигацию в среде web. Наиболее распространенными на текущий момент решениями этой проблемы являются полнотекстовой поиск с использованием векторной модели пространства документов, и наиболее распространенные алгоритмы кластеризации.

Введение

С момента появления web в сети с каждым днем становится все больше и больше динамически меняющейся, неупорядоченной информации, хаотическая природа которой затрудняет её поиск и навигацию в среде web. Острую потребность в решении этой проблемы можно увидеть, взглянув на такие компании как google, yandex, и прочие компании, предоставляющие хотя бы частичное решение этих проблем. Сервисы этих компаний пользуются невероятной популярностью, несмотря на то, что уже сейчас они с каждым днем все хуже и хуже справляются с растущим объемом информации. Так же, интересно появление таких проектов как clusty.com, nigma.ru, и подобных, проводящих кластеризацию результатов поиска. Используя простую кластеризацию, данные сервисы многократно повысили свою точность работы и удобность.

Ключевая проблематика

Кластеризация является только одним из возможных решений проблемы динамически меняющегося веба. Наиболее распространенными на текущий момент решениями являются полнотекстовой поиск с использованием векторной модели пространства документов, и наиболее распространенные алгоритмы кластеризации. Такой подход имеет большой ряд недостатков, таких как отсутствие решения проблемы синонимии, отсутствие решения проблемы двусмысленности, отсутствие методов выявления контекстной информации, отсутствие работы с семантикой слов, отсутствие возможности рассмотрения документов с разных точек зрения, работа в пространстве большой размерности. Вследствие отсутствия решений этих недостатков для текущих решений характерны низкая

точность, медленная работа алгоритмов, и как следствие, высокие требования к оборудованию. Так же использование векторного пространства имеет такие недостатки как отсутствие хорошего метода выражения отрицания и отсутствие способа описания логических условий.

Некоторые сравнительно новые решения предоставляют такие процедуры, как латентно-семантическое индексирование, способное снижать размерность векторного пространства, работать с документами на разных языках, выявлять отношения между термами. Но, несмотря на то, что подобные процедуры решают многие из вышеперечисленных проблем, они так же обладают рядом недостатков. К примеру, то же латентно-семантическое индексирование обладает несколькими недостатками:

1. Вычислительные затраты, связанные с сингулярным разложением, на котором построена процедура, весьма значительны.
2. Процедура работает лучше всего на небольших коллекциях документов.

Таким образом, для решения задач требуются либо модификации существующих алгоритмов, либо альтернативные подходы.

Некоторые сведения об алгоритмах кластеризации

Алгоритмы кластеризации разделяют совокупность документов на подмножества. Цель алгоритмов – создание однородных множеств. Кластеризация является наиболее распространенной формой обучения без учителя (в алгоритме не предусмотрено участие эксперта, присваивающего документы классам). Плоская кластеризация порождает совокупность кластеров, не имеющих явных взаимосвязей. Иерархическая – создает иерархию кластеров. Жесткая кластеризация вычисляет жесткое присваивание – каждый документ может быть элементом одного кластера. Мягкая кластеризация – каждый документ в некоторой мере принадлежит каждому кластеру.

Основное предположение, позволяющее применять кластеризацию для информационного поиска, заключается в том, что документы, принадлежащие одному и тому же кластеру, являются примерно одинаково релевантными по отношению к информационным потребностям.

Наиболее известные примеры применения кластеризации – сервисы наподобие Google News, где можно просмотреть все новостные статьи, связанные текущей просматриваемой.

Самая большая проблема в решении задачи кластеризации документов – большая размерность рабочего пространства, что сильно замедляет скорость выполнения алгоритмов. Существует несколько подходов к решению этой проблемы:

1. Использование словаря синонимов. В этом случае строки термов-синонимов сливаются. Проблемы метода – высокая степень изменчивости языка, метод не предоставляет достаточного сокращения размерности.

2. Использование в модели пространства только определенных, считающихся заведомо важными, слов документа. При данном подходе теряется большая часть информации.
3. Отбор наиболее информативных размерностей. В данном случае для каждого измерения вычисляется некоторая величина, характеризующая её ценности для кластеризации. После этого остаются только наиболее ценные измерения. Метод так же страдает от потери большого количества информации о документе.
4. Латентно-семантический анализ. Является одним из видов анализа главных компонент. Суть анализа заключается в том, что в исходном векторном пространстве термов находится небольшое число ортогональных осей, таких, что вдоль этих осей разброс точек, соответствующих документам исходной коллекции, очень велик. Эти оси выбираются в качестве базиса, и на них проецируются все точки исходного векторного пространства. Таким образом, вычисляются координаты точек в новом редуцированном пространстве. Метод обладает большим количеством преимущество, но сильно страдает от низкой скорости выполнения.

Как видно, ни один из способов не предоставляет метода быстрого сокращения размерности пространства до приемлемых размеров без больших потерей информации.

Подробнее об алгоритмах кластеризации см. в [1,2,3].

Некоторые сведения о семантическом анализе

Семантический анализ – процесс создания соответствующих лингвистическим выражениям семантических представлений. Наиболее популярные виды подобных представлений - выражения на языке логики первого порядка, семантические сети и фреймы. На абстрактном уровне все эти представления являются связанными между собой термами, которые могут соответствовать реально существующим объектам, их объектов, либо связям между ними.

Важно то, что эти представления можно рассматривать с разных точек зрения, и, таким образом, использование семантического анализа в задачах информационного поиска и кластеризации уже решает одну из проблем – проблему объективности.

Перечисленные семантические представления обладают рядом важных свойств:

1. Проверяемость – возможность сравнить данные полученные из лингвистического выражения и данные, которыми оперирует анализатор.
2. Однозначность - семантическое преставление лингвистического выражения должно исключать возможность двоякого толкования выражения.
3. Возможность использовать переменные.

4. Возможность строить логические выводы, основываясь на данных семантического представления
5. Выразительность

Для семантического анализа текстовых документов в качестве семантических представлений удобно использовать логику первого порядка.

Спецификация синтаксиса логики первого порядка:

$$
\begin{aligned}
Formula &\rightarrow Atomic\ Formula \\
&\mid Formula\ Connective\ Formula \\
&\mid Quantifier\ Variable, \ldots Formula \\
&\mid \neg Formula \\
&\mid (Formula) \\
Atomic\ Formula &\rightarrow Predicate(Term, \ldots) \\
Term &\rightarrow Function(Term, \ldots) \\
&\mid Constant \\
&\mid Variable \\
Connective &\rightarrow \mid\mid => \\
Quantifier &\rightarrow \forall \mid \exists \mid \\
Constant &\rightarrow Term \\
Variable &\rightarrow x, y, z, \ldots
\end{aligned}
$$

Важным элементом логики первого порядка является лямбда-нотация, предоставляющая возможность абстрагироваться от определенного логического выражения.

Хорошей альтернативой логике первого порядка является дескрипционная логика, представляющая множество языков описания семантики. Дескрипционная логика обладает всеми перечисленными свойствами семантических представлений. Так же она играет важную роль в концепции семантического веба, в котором она должна использоваться при построении онтологий.

При использовании дескрипционной логики подчеркивается представление знаний о категориях, индивидах, принадлежащих этим категориям, и связях между этими индивидами. Множество категорий, предоставляющих знания о конкретной предметной области называется терминологией. Часть знаний, содержащая терминологию называется TBox; часть знаний, содержащая факты об индивидах – ABox. Обычно терминология организована в иерархическую структуру, называемую онтологией.

Подробнее об алгоритмах кластеризации см. в [1].

Предложение решения проблемы

Классическим подходом к решению задачи кластеризации является разбиение текста по термам, проведение их стемминга и построения матрицы “терм-документ”.

Стемминг – процесс нахождения основы слова. Наиболее популярным на данный момент является алгоритм стемминга Портера, отсекающий окончания и суффиксы слов.

Полученная матрица является представлением векторного пространства, в котором строки являются измерениями, а столбцы – точками в пространстве. В реальных приложениях ее размерность получается порядка 40000 термов на 300 документов. При таких размерах все ресурсы уходят на решение проблемы производительности, а задачи повышения точности, и разрешения объективности уходят на второй план.

Возможным решением проблемы может стать использование семантических представлений в качестве измерений пространства. В данном варианте выполнение алгоритма подразумевает выполнение следующих этапов:

1. Разбиение документа на предложения.
   Для решения данной задачи существует множество методов, таких как построение дерева решений, использование методов машинного обучения и другие.
2. Проведение синтаксического анализа.
3. Построение синтаксических структур.
4. Перевод предложений в выражения логики первого порядка.
   Решение данной задачи основано на принципе композиционности. Ключевая идея заключается в том, что семантическое представление предложение может быть построено из представлений его частей. Семантика предложения базируется не только на семантике членов предложения, но так же и на порядке и группировке слов в предложении. Для учета этих зависимостей и был проведен синтаксический анализ.
5. Выявление эквивалентных выражений.
6. Построение матрицы представление – документ.
7. Использование одного из классических алгоритмов кластеризации.

При использовании подобного подхода для построения матрицы, ее итоговая размерность, очевидно, будет намного меньше оригинальной. Пространство меньшей размерности даст выигрыш в скорости исполнения алгоритма, а использование семантических представлений в качестве измерений решает проблемы синонимии и двусмысленности. Так же использование подобной структуры данных дает такие возможности как исполнение запросов в логической форме, и отслеживание времени, к которому относятся документы.

Однако, скорее всего, выполнение пункта 4 предложенного алгоритма будет занимать количество времени сравнимое с временем, занимаемым латентно-семантическим индексированием. Таким образом, от семантического разбора текста имеет смысл отказаться, и рассмотреть решение проблемы с помощью модификации процедуры латентно семантического индексирования.

Подробнее об семантическом и лингвистическом анализе см. в [2].

Предложение решения проблемы на основе латентно-семантического анализа.

В целом, процедура латентно-семантического индексирования решает почти все из перечисленных проблем, в частности, оно способно снижать размерность

матрицы 'терм-документ' до приемлемых размеров, решает проблему синонимии, выявляет семантические связи между термами, но огромное время выполнения алгоритма делает его малоприменимым. Таким образом, решение проблемы со скоростью выполнения предоставит достойное решение многих проблем, связанных с кластеризацией данных.

Предлагаемое решение основано на альтернативном методе проведения процедуры латентно-семантического индексирования. Этот метод основан на расширении симметричного диагонального разложения матриц, называемого сингулярным. Описание разложения:

Пусть для исходной матрицы C имеется матрица U – матриц ,а, имеющая размерность MxM, столбцы которой представляют собой ортогональные собственные векторы матрицы $C\mathrm{C}^T$, а матрица V – матрица, имеющая размерность MxN, столбцы которой являются ортогональными векторами матрицы $C^T C$. Тогда верно следующее утверждение:

Пусть r – ранг матрицы C, имеющей размерность MxN. Тогда существует сингулярное разложение матрицы C = U∑V^T, где:

1) *Собственные значения λ1,…,λr матрицы $C\mathrm{C}^T$совпадают с собственными значениями матрицы $\mathrm{C}^T\mathrm{C}$;*
2) *Пусть σ_i=$\sqrt{\lambda_i}$, где $\lambda_i \geq \lambda_{i+1}$при всех 1≤ i ≤ r-1; тогда матрица ∑ размерности MxN состоит из чисел $\sum_{ii}$=σ_i при всех 1≤ i ≤ r и нулей.*

Числа σ_i называются сингулярными значениями матрицы C.

Для проведения сокращения размерности исходной матрицы до порядка k в матрицах U и V оставляют только k первых столбцов, в матрице ∑ - k первых столбцов и строк. Тогда матрица $M_k = U_k * S_k$ будет содержать в своих строках координаты точек в новом базисе, матрица V^T будет содержать в себе информацию о связи нового редуцированного базиса и старого базиса векторного пространства. Матрица M` $= U_k * S_k * V_k^T$,будет матрицей ранга не выше k, являющейся наилучшей аппроксимацией исходной матрицы M.

Существенным недостатком метода является его высокая вычислительная сложность обусловленная сложностью проведения сингулярного разложения матрицы. Решением этой проблемы может стать реализация алгоритма проведения сингулярного разложения, предназначенного для расчета на кластерах.

Подробнее о латентно семантическом анализе см. в [1,3].

Заключение

В данном методе можно так же использовать альтернативные семантические представления, и это даст соответственные выигрыши. К примеру, использование семантических сетей позволяет использовать весь стек технологий семантического веба – языки RDF, OWL, SPARQL.

Помимо описанного подхода, который является, по сути, гипотезой, можно найти большое количество альтернативных методов работы с документами в web. Существуют методы кластеризации на основе онтологий, семантических

сетей, модификации алгоритма латентно-семантического индексирования, метод главных компонент. Так, или иначе, все эти методы направлены в первую очередь на снижение размерности матрицы "терм-документ", либо на использование альтернативной модели пространства.

Хорошим вариантом альтернативного пространства могла бы стать семантическая сеть, в которой с помощью поиска в графах различных документов ищутся общие подграфы. Но такие алгоритмы имеют очень высокую степень сложности.

Литература

1. Маннинг Кристофер Д., Рагхаван Прабхакар, Шютце Хайнрих. *Введение в информационный поиск.* :Пер. с англ. –М.:ООО"Вильямс", 2011. –528с.
2. Daniel Jurafsky, James H.Martin. *Speech and Language Processing,* 2008. – 988с.
3. Landauer T.K., Foltz P.W., Laham D. *An Introduction to Latent Semantic Analysis. Discourse Processes*, 25, 1998. 259-284

Предметно-ориентированная разработка крупномасштабных программных систем с применением паттернов

Зыков С.В.

Кафедра УРПО, Отделение Программной инженерии, Факультет Бизнес-Информатики НИУ ВШЭ,
Москва, 119435 РФ, Кирпичная ул., д. 33/5, офис 526
szykov@hse.ru,
WWW: http://www.hse.ru/org/persons/3468544

Аннотация. Предложен подход, принципиально позволяющий перейти от заказной разработки ограниченных решений к серийному производству программных систем. Подход основан на применении предметно-ориентированных языков и паттернов, поддержанных семантическими сетями с фреймовой визуализацией. Рассмотрено семейство внедрений подхода в различных отраслях.

Введение

При разработке программных систем для крупных распределенных производственных структур существенные сложности представляет организация "промышленного" производства такого рода систем. Проблемы связаны как с объемом разработки, так и гетерогенным характером структур данных и программной архитектуры. Тем не менее, в целом ряде случаев представляется возможным не только выделить крупные "типовые" компоненты систем, но и организовать их "тиражное", поточное производство с адаптацией "типового" решения к потребностям каждого конкретного заказчика. Объем вновь создаваемого кода при этом составляет порядка 15-20% от общего объема продукта, что приводит к экономии до 50% времени, средств и трудозатрат по сравнению с разработкой с "чистого листа". Такая экономия может быть обеспечена не просто применением паттернов, но их систематическим построением на основе предметно-ориентированного языка типа DSL.

Особенности применяемых паттернов и метаязыка типа DSL

Для разработки программных систем рассматриваемого класса и масштаба предлагается построить формальный язык, описывающий особенности предметной области. Далее, при помощи этого предметно-ориентированного языка строится описание основы будущей программной системы в форме паттернов, или "типовых конструкций", составляющих системную архитектуру и бизнес-логику прикладного программного решения.

При этом важно отметить, что подобное погружение предметно-ориентированного метаязыка в среду проектирования требует применения специализированных визуальных инструментальных средств, интегрирующих возможности фреймов и традиционных диаграмм (в т.ч. стандартов UML и STEP). Создание информационных систем с разработкой метаязыка вполне удовлетворительно может быть представлено на уровне модели посредством семантической сети.

Подобный подход представляется целесообразным, прежде всего, в силу следующих соображений. Первое. Проектировщики используют масштабируемое по архитектуре решение за счет "встроенной" в формализацию возможности построения фреймов с многоуровневой вложенностью. Второе. Предметные аналитики и другие разработчики получают возможность манипулирования масштабируемыми графическими объектами. Третье. Объекты высокой сложности могут быть представлены фреймами, структурно декомпозируемыми на основе логических операций. Четвертое. Динамика интеграции компонентов программных систем рассматриваемого класса и характера моделируется событийными фреймами с учетом причинно-следственных связок.

На основе изложенного подхода получен комплекс решений в форме прикладных программных систем для таких отраслей, как атомная промышленность, нефтегазовый сектор, пассажирские авиаперевозки, сеть розничной торговли, а также для некоторых других областей применения. Реализация предложенного подхода позволила существенно сократить стоимость производства программных систем, ускорить и упорядочить процессы их разработки, снизить возникающие при этом риски, а также в целом ряде случаев перейти от узконаправленных заказных решений к открытым тиражируемым.

Заключение

Показано, что применение подхода, синтезирующего паттерны с предметно-ориентированными языками в условиях тиражной разработки программных систем, обеспечивает существенную экономию сроков и стоимости внедрения. Отмечено, что важную роль при разработке и использовании таких языков и паттернов играют модели и инструментальные средства на основе семантических сетей с фреймовой визуализацией. Приведены примеры успешных внедрений предложенного подхода на предприятиях и в организациях для целого ряда отраслей.

3-я междунар. конференция по аппликативным вычислительным системам (ABC'2012),
Москва, 26-28 ноября 2012 г.
М.: НОУ Институт Актуального образования «ЮрИнфоР-МГУ», 2012

Литература

1. S. Cook, G. Jones, S. Kent, A. C. Wills. *Domain-Specific Development with Visual Studio DSL Tools*. — Pearson Education, Inc, 2008, 524 pp.
2. G. Hohpe, B. Woolf. *Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions.* — Addison Wesley, 2003, 736 pp.
3. Zykov S.V. *Pattern-Based Development of Enterprise Systems: From Conceptual Framework to Series of Implementations*. In: Proc. of the 3rd International Multi-Conference on Complexity, Informatics and Cybernetics (IMCIC 2012), March 25-28, 2012, Orlando, Florida, USA, p.p.189-193

Логический вывод в аппликативной вычислительной системе

Мажирин И.В.

ООО «КИФФ» г. Москва
allo10@yandex.ru

Аннотация. В статье обсуждается представление широко известных систем логического вывода и автоматического доказательства теорем средствами λ-исчисления и комбинаторной логики в различных комбинаторных базисах в рамках разрабатываемой Веб-информационной аппликативной вычислительной системы. Показаны основные отличия аппликативного подхода в реализации таких систем от классических, выполненных на традиционных вычислительных архитектурах. Проект № 10-07-00603-а поддержан РФФИ.

Введение

В последнее время аппликативные вычислительные системы (АВС) заслуженно получают признание в различных сферах деятельности человека, в том числе в промышленности и во всемирной паутине. Было создано и применяется достаточное количество функциональных языков программирования высокого уровня для целей доказательства теорем и логического вывода, от LISP и PROLOG до современных языков ML, HASKELL, F#. Все они реализованы на традиционной вычислительной архитектуре с основной операцией «сложение» и с различной производительностью выполняют поставленные задачи. Общая структура существующих сегодня систем логического вывода выглядит примерно так:

- Интерфейс
- Байт код
- Обычный процессор.

Предлагается формальная система логического вывода для аппликативной вычислительной системы в различных комбинаторных базисах с аппаратной поддержкой на базе современных компьютеров PC в среде Windows.

Общая структура системы:

- Интерфейс
- Комбинаторная абстрактная машина
- Комбинаторный процессор.

Теоретические основания

Комбинаторная логика была введена М. Шейнфинкелем (Moses Shonfinkel) в 1924 году в работе «О строительных блоках математической логики». В работе была показана возможность сведения логики к минимальной совокупности примитивных функций - комбинаторов из некоего базиса и отказаться от связанных переменных. Комбинаторную логику иногда называют бестиповой формой λ-исчисления. Функцией высшего порядка считается функция, аргументом которой и (или) результатом работы которой может быть функция.

Использование функций высших порядков позволило свести логику до языка, содержащего конструктор – аппликацию «•» функции к аргументу и три примитивные функции – комбинаторы K, S, U. В приведенных формулах любая буква может означать как переменную, так и функцию. Используется традиционная запись функций и аргументов для упрощения восприятия.

$Kxy = K•x•y = (K(x))(y) = x$
$Sfgx = S•f•g•x = ((S(f))(g))(x) = (f(x))(g(x))$
$UPQ = U•P•Q = (U(P))(Q) = \forall x.\neg (P(x) \wedge Q(x))$,

где
K – константная функция, для любого x возвращает x,
S – комбинирующая функция, ее результатом для двух функций f и g будет значение применения f(x) к g(x),
U – функция обобщение штриха Шиффера, она применяется к двум предикатам P и Q, ее результатом будет универсальное обобщение отрицания конъюнкции этих предикатов от x.
Оказалось, что комбинаторов K, S, U достаточно для выражения произвольных предикатов первого порядка без использования связанных переменных, которые возникают в логике первого порядка при использовании кванторов общности и существования. Кроме того, при определенных условиях (если x и y имеют логические значения):
- так как аппликация «•» функции к аргументу присутствует между всеми функциями и (или) аргументами то выражение xy = x•y при определенных условиях может получить свойства материальной импликации – элементарного логического вывода со свойствами «если x то y», то есть «•» = «=>» и xy = x•y = x=>y.
- комбинатор K можно считать элементарной дедукцией, так как из двух посылок x и y в результате работы комбинатора остается (возвращается) одна x (сворачивание),
- комбинатор S можно считать элементарной индукцией, так как две исходные функции f и g применяется к одному аргументу x и результатом работы комбинатора является появление двух посылок f(x) и g(x) (ветвление).
Выразительная мощность полученной таким образом аппликативной системы позволяет представлять достаточное количество необходимых объектов в виде комбинаторных термов, в том числе и для доказательства теорем и логического вывода. Так же возможно представление логических констант в виде комбина-

торных термов, которое не является единственным представлением в комбинаторном базисе {K, S}. Например,

«истина» = T = I
«ложь» = F = KI,
где
I = SKK.

Классическая система доказательства теорем и логического вывода

Как известно, за 50 лет развития систем доказательства теорем и логического вывода сложились две основные системы.
Это:
- Система опровержения на основе резолюций.
- Система дедукций на основе правил (прямая, обратная или смешанная).

Появились достаточно точные правила преобразования высказываний на языке Исчисления Предикатов Первого Порядка (ИППП) в так называемую Конъюнктивную Нормальную Форму (КНФ). Рассмотрим общие правила таких преобразований на примере Правильно Построенной Формулы (ППФ) в ИППП:

$$(\forall x)(P(x) \Rightarrow ((\forall y)(P(y) \Rightarrow P(f(x, y)))\wedge\neg (\forall y)(Q(x, y) \Rightarrow P(y)))).$$

Для преобразования указанной ППФ проводятся следующие шаги преобразований.
1. Устранение импликаций (=>)
2. Ограничение области действия связки отрицание (¬) к одной атомарной формуле
3. Разделение переменных
4. Исключение кванторов существования (Сколемизация)
5. Исключение кванторов всеобщности
6. Переименование переменных
7. Преобразование в КНФ – конъюнкцию конечного множества дизъюнкций

В результате преобразований получаем КНФ исходного выражения:

$$(\neg P(x1)\vee\neg P(y)\vee P(f(x1,y)))\wedge(\neg P(x2)\vee Q(x2,g(x2)))\wedge(\neg P(x3)\vee\neg P(g(x3)))$$

Покажем систему опровержения на основе резолюций.
Для простоты изложения приведем правила для предикатов без переменных.
Общее правило вывода с использованием резолюции состоит в следующем:
Пусть имеется два предложения в КНФ:

$$(P1\vee P2\vee \ldots \vee Pn)\wedge(\neg P1\vee Q2\vee \ldots \vee Qm) \rightarrow (P2\vee \ldots \vee Pn)\wedge(Q2\vee \ldots \vee Qm),$$

То есть

$(\neg P1) \wedge P1 \rightarrow nil$,
$P \wedge (\neg P \vee Q) \rightarrow Q$,
где
$(\neg P \vee Q) = (P \Rightarrow Q)$,
$P, Q, P1, \ldots, Pn, Q2, \ldots, Qm$ - предикаты,
nil – пустой дизъюнкт.

В конечном итоге для вышеуказанных целей используется символьная обработка в КНФ дизъюнктов и логических связок с подстановкой и унификацией значений переменных различными программными средствами от Ассемблера и C++ до экзотических или специальных языков программирования на базе традиционной архитектуры Фон-Неймана. Что представляется не очень удобным, так как в процессе вычислений не применяется ни одна арифметическая операция.

Аппликативная система доказательства теорем и логического вывода

Процесс представления в различных комбинаторных базисах доказательства теорем и логического вывода сводится к системе продукций с набором правил и достаточно простой системой управления с возможностью добавления комбинаторных выражений для эвристических и весовых функций с аппаратной поддержкой комбинаторного базиса {K, S}. В результате не тратится программно-аппаратный ресурс на выполнение элементарной операции (аппликации). Комбинаторы базиса {K, S} возможно представить в двоичном коде в виде битовой строки. Это позволяет использовать существующую в настоящее время программно-аппаратную вычислительную среду, построенную на двоичной арифметике.
Один из вариантов представления логических связок в комбинаторном виде:

Отрицание: $\neg x = \neg x = \lambda x.xF$,
Импликация: $x \Rightarrow y = xy = \lambda xy.xy$,
Логическое или: $x \vee y = \vee xy = \lambda xy.xFy$,
Логическое и: $x \wedge y = \wedge xy = \lambda xy.x(yF)F$,
где x, y – переменные
$F = nil = KI$ – комбинаторный терм.

Для простоты изложения приведем правила для предикатов без переменных.
Основное правило резолюции в комбинаторном виде будет выглядеть так:

$P \wedge (\neg P) \rightarrow nil = \wedge P(\neg P) = P(PFF)F \rightarrow F$
или
$(\neg P) \wedge P \rightarrow nil = \wedge(\neg P)P = PF(PF)F \rightarrow F$,
где P - комбинаторный терм.

Как известно, правило логического вывода «модус поненс» выглядит следующим образом:

$P \wedge (P \Rightarrow Q) \rightarrow Q$,
где P, Q - любые формулы

Правило логического вывода «модус поненс» в комбинаторном выражении можно представить следующим образом:

$P \wedge (P \Rightarrow Q) = \wedge P(\Rightarrow PQ) = \wedge P(PQ) = P(PQF)F \rightarrow Q$
или
$P \wedge (P \Rightarrow Q) = P \wedge (\neg P \vee Q) = (P \wedge \neg P) \vee (P \vee Q) = \vee(\wedge P(\neg P))(\wedge PQ) = P(PFF)FF(P(QF)F) \rightarrow Q$,
где P, Q – комбинаторные термы. Оба комбинаторных представления эквивалентны.

Классическая система дедукций на основе правил

В случае системы дедукций на основе правил база знаний (БЗ) делится на две части: факты и правила. Факты являются утвердительными предложениями в форме КНФ, правила выражаются в форме импликаций «если x то y», где x и y имеют логические значения {«истина», «ложь»}. В зависимости от направления логического вывода (прямой или обратный) преобразование БЗ происходит от фактов к целям (прямая система) или от целей к фактам (обратная система). При этом если происходит унификация переменных в сопоставимом факте Fi с антецедентом (консеквентом) правила Ai (Ci), то в БЗ добавляется консеквент (антецедент) сопоставленного правила Ci (Ai). Условием останова таких преобразований базы знаний является достижение заданной цели (прямая система) или заданных фактов (обратная система). Причем, в базе знаний при необходимости можно оставлять замененный на Ai или Ci факт Fi, если необходимо сохранить предысторию процесса логического вывода или использовать алгоритм поиска решения с возвратом (смешанная система).

Прямая дедукция

В общем виде (без переменных) процесс преобразования БЗ при прямой дедукции на основе правил можно представить так:

$\{F1, F2, \ldots, Fi, \ldots, Fn\}$ и $\{A1 \Rightarrow C1, A2 \Rightarrow C2, \ldots, Ai \Rightarrow Ci, \ldots, An \Rightarrow Cn\} \rightarrow$
$\{F1, F2, \ldots, Ci, \ldots, Fn\}$ и $\{A1 \Rightarrow C1, A2 \Rightarrow C2, \ldots, An \Rightarrow Cn\}$.

В свою очередь каждое правило Ai=>Ci может иметь сложный антецедент, состоящий из конъюнктов.

Ai = Ai1^ Ai2^ … ^Aiк

В случае доказательства цели Z вида

Z1^Z2^ … ^Zm=>Z

в классических системах прямой дедукции доказательство происходит, как унификация переменных и прямое символьное сравнение на соответствие фактов и антецедентов правил с элементами конъюнктов целей. При нахождении всех соответствий цель считается доказанной и БЗ пополняется доказанным элементом Z.

Пример.

Пусть имеется БЗ:

{O, M, A, E} и {M^O=>S, M=>W, E=>H}

Имеется цель:

A^W^S^H=>C.

Доказательство.

Все факты в БЗ = <истина>, то есть O, M, A, E = <истина>

1. A – доказано, так как A = <истина> и {A} есть в БЗ
2. {O, M, A, E} и {M^O=>S, M=>W, E=>H} → {O, M, W, A, E} и {M^O=>S, M=>W, E=>H} W – доказано
3. {O, M, W, A, E} и {M^O=>S, M=>W, E=>H} → { O, M, S, W, A, E} и {M^O=>S, M=>W, E=>H} S – доказано
4. {O, M, S, W, A, E} и {M^O=>S, M=>W, E=>H} → {O, M, S, W, A, E, H} и {M^O=>S, M=>W, E=>H} H – доказано
5. {O, M, S, W, A, E, H} и {M^O=>S, M=>W, E=>H} → {O, M, S, W, A, E, H, C} и {M^O=>S, M=>W, E=>H} C – доказано.

Цель C доказана, так как A, W, S, H = <истина> и {A, W, S, H} есть в обновленной БЗ.

Аппликативная прямая дедукция

В комбинаторном представлении процесс преобразования БЗ при прямой дедукции на основе правил можно представить так:

{F1, F2, …, Fi, …, Fn} и {A1C1, A2C2, …, AiCi, …, AnCn} →
{F1, F2, …, Ci, …, Fn} и {A1C1, A2C2, …, AnCn}.

Правило Ai=>Ci в комбинаторном виде: AiCi.

сложный антецедент Ai = ^Ai1(^Ai2(^ … ^Aiк)…).

Цель Z будет выглядеть, как

^Z1(^Z2(^ … ^Zm)…)Z.

В случае комбинаторного представления появляется возможность проверки консеквента каждого правила, находящегося в БЗ, на предмет соответствия конъюнктам цели без прямого символьного сопоставления на соответствие фактов и антецедентов правил с конъюнктами цели. То есть из условия, что все

элементы БЗ должны иметь значение «истина» = I, следует, что имеется возможность проверить на истинность каждое унифицируемое правило в отдельности. То есть справедливо ли AiCi → Ci.

Если Ai = I, то ICi → Ci.

Для правила со сложным антецедентом:

∧Ai1(∧Ai2(∧ … ∧Aiк)…)Ci → Ci.

Если ∧Ai1(∧Ai2(∧ … ∧Aiк)…) = I, то ∧I(∧I(∧ … ∧I)…)Ci → Ci.

В отличие от системы опровержения на основе резолюции, где процесс доказательства происходит до тех пор, пока в качестве результата очередной резолюции не будет получен nil = F, в системе прямой дедукции в комбинаторном представлении при успешном сопоставлении фактов и антецедентов правил со всеми конъюнктами цели БЗ пополняется консеквентом цели, как доказанным элементом БЗ. При этом все антецеденты правил и конъюнкты цели в результате редукции должны быть равны I = T. За счет активной «работы» логических связок «∧» и «=>» в комбинаторном представлении появляется возможность проверять на истинность каждое сопоставимое правило, входящее в БЗ, отдельно. Можно параллельно обрабатывать правила, что должно существенно увеличить производительность системы прямой дедукции. Такое проверяемое правило должно дать в результате редукции комбинаторного терма консеквент, который в свою очередь, должен иметь значение «истинна» = I для сопоставления с конъюнктами цели. В результате, если в БЗ появился консеквент цели Z, то цель считается доказанной. Такой комбинаторный подход в доказательстве цели существенно экономит количество шагов обработки БЗ при получении результата.

Пример.

Пусть имеется БЗ:

{O, M, A, E} и {∧MOS, MW, EH}

Имеется цель:

∧A(∧W(∧SH))C.

Доказательство.

Все факты в БЗ = <истина>, то есть O, M, A, E = <истина> = I

1. A – доказано, так как A = <истина> и {A} есть в БЗ
2. Проверяем консеквент каждого сопоставленного правила, находящегося в БЗ {∧MOS, MW, EH}

∧MOS → ∧IIS → S

MW → IW → W

EH → IH → H

То есть

{O, M, A, E} и {∧MOS, MW, EH} → {O, M, A, E, S, W, H} и {∧MOS, MW, EH}.

Проверяем достижение цели.

∧A(∧W(∧SH))C → ∧I(∧I(∧II))C → C

Обновленная БЗ:

{O, M, A, E, S, W, H, C} и {∧MOS, MW, EH}.

Цель C доказана, так как все конъюнкты, входящие в антецедент цели = I и C является полученным далее не редуцируемым термом.

Есть возможность доказать цель C, в том числе, прямой подстановкой комбинаторных термов для правил в конъюнкты цели, исходя из выражений для консеквентов соответствующих правил.

∧A(∧W(∧SH))C → ∧A(∧(MW)(∧(∧MOS)(EH)))C → ∧A(∧(IW)(∧(∧IIS)(IH)))C → ∧A(∧W(∧SH))C.

Обратная дедукция

В общем виде (без переменных) процесс преобразования БЗ при обратной дедукции на основе правил можно представить так:

{F1, F2, …, Fi, …, Fn} и {A1<=C1, A2<=C2, …, Ai<=Ci, …, An<=Cn} →
{F1, F2, …, Ai, …, Fn} и {A1<=C1, A2<=C2, …, An<=Cn}.

В свою очередь каждое правило Ai<=Ci может иметь сложный антецедент, состоящий из конъюнктов.

Ai = Ai1∧ Ai2∧ … ∧Aiк

В случае доказательства цели Z вида

Z1∧Z2∧ … ∧Zm<=Z

в классических системах обратной дедукции доказательство происходит, как унификация переменных и прямое символьное сравнение на соответствие консеквентов правил с элементами конъюнктов целей. При нахождении всех соответствий антецедентов правил со всеми сопоставленными фактами цель считается доказанной.

Аппликативная обратная дедукция

В комбинаторном представлении процесс преобразования БЗ при обратной дедукции на основе правил можно представить так:

{F1, F2, …, Fi, …, Fn} и {<=A1C1, <=A2C2, …, <=AiCi, …, <=AnCn} →
{F1, F2, …, Ai, …, Fn} и {<=A1C1, <=A2C2, …, <=AnCn},

где <= = K = λxy.x.

Правило Ai<=Ci в комбинаторном виде: <=AiCi,
сложный антецедент Ai = ∧Ai1(∧Ai2(∧ … ∧Aiк)…).
Цель Z будет выглядеть, как

<= (∧Z1(∧Z2(∧ … ∧Zm)…)Z.

Пример.
Пусть имеется БЗ:

{O, M, A, E} и {M∧O<=S, M<=W, E<=H}

Имеется цель:

A∧W∧S∧H<=C.

В комбинаторном виде БЗ:

{O, M, A, E} и {<=(∧MO)S, <=MW, <=EH}

В комбинаторном виде цель:

<=(∧A(∧W(∧SH)))C.

Доказательство.

Необходимо доказать факты {A, W, S, H}

1. A – доказано, так как A = <истина> и {A} есть в БЗ
2. Проверяем антецедент каждого сопоставленного правила, находящегося в БЗ {<=(∧MO)S, <=MW, <=EH}

<=(∧MO)S → K(∧MO)S → ∧MO → ∧II → I, доказано S

<=MW → KMW →M → I, доказано W

<=EH → KEH → E → I, доказано H

То есть

{O, M, A, E} и {<=(∧MO)S, <=MW, <=EH} → {O, M, A, E, S, W, H, C} и {<=(∧MO)S, <=MW, <=EH}

{A, W, S, H} входит в БЗ. Цель доказана.

Проверяем достижение цели.

<=(∧A(∧W(∧SH)))C → K(∧I(∧I(∧II)))C → ∧I(∧I(∧II)) → I.

Как видно из примера, в процессе доказательства цели в аппликативной системе обратной дедукции также возможна независимая и параллельная обработка сопоставленных правил, как и в системе прямой дедукции, что существенно экономит количество шагов обработки БЗ для получения результата.

Аппликативная и традиционная системы доказательства теорем и логического вывода

Аппликативная система доказательства теорем и логического вывода имеет ряд существенных отличий от традиционной:

1. В аппликативной системе возможно распараллеливание процесса логического вывода в рамках каждого отдельно взятого предложения Pi в составе КНФ, так как все Pi, входящие в КНФ, должны быть истинными одновременно. То есть если
   КНФ = P1∧P2∧ … ∧Pn = T, то проверку на истинность каждой Pi можно проводить отдельно вне зависимости от других предложений. То есть можно одновременно проверять на истинность все Pi.
   P1 = T
   P2 = T
   …
   Pn = T
2. Так как все объекты, в том числе, логические связки представлены в избранном комбинаторном базисе, то при редукции комбинаторных термов теряется различие между объектами для самого процесса ре-

дукции, выполняемого КАМ (Комбинаторной Абстрактной Машиной). Логические связки в этом случае становятся активными и комбинаторные выражения для последних редуцируются вместе с остальными термами. Таким образом, можно получить значение k-местной Pi. То есть в традиционной системе программирования логическая операция ∧ или ∨ будет всегда двухместной и выполняется одной командой. К примеру, если необходимо получить значение A1∨A2∨A3∨… ∨An, то необходимо сначала вычислить значение A1∨A2 (одна команда логического сравнения), далее результат сравнить с A3 (еще одна команда логического сравнения) и так далее до Ak. В случае аппликативной вычислительной системы есть возможность вычислять k-местные выражения, содержащие указанные логические операции в процессе редукции одного терма. В этом случае:

A1∨A2∨A3∨… ∨Ak = ((A1∨A2)∨A3)∨… ∨)Ak = (∨(∨A1A2)A3)…)Ak = (λxy.xFy (λxy.xFyA1A2)A3)…)Ak = (A1FA2)FA3)…)FAk – один терм.

3. При представлении логических связок в виде комбинаторных выражений, указанных выше, импликация «=>» эквивалентна элементарной операции КАМ - аппликации. То есть

A1=>A2 = =>A1A2 = λxy.xyA1A2 = A1A2.

Таким образом, получается, что все комбинаторные термы одновременно связаны между собой не только аппликацией, но и операцией импликации «=>». Это должно давать существенное повышение производительности при выполнении прямой и обратной дедукций, так как в традиционных системах программирования команда, выполняющая операцию импликации «=>», не предусмотрена и она выражается через известные логические связки:

A1=>A2 = ┐A1∨A2,

что предполагает дополнительные машинные операции для выполнения заданного алгоритма.

Веб среда и аппликативная система доказательства теорем и логического вывода

Как известно, описанные выше системы логического вывода и доказательства теорем широко применяются в различных вычислительных системах в веб-среде, где необходимо автоматическое принятие решений, обработка семантической информации, поиск информации и т. д. Эффективность применяемых сегодня вычислительных систем, подключенных к веб, напрямую зависит от существенных факторов.

1. Идеология построения вычислительной системы.

Работа любой вычислительной системы, как и пользователя, с веб средой похожа на выполнение функции «запрос (R) – ответ (P)», где запрос обрабатывается с помощью некоторого функционального инструментария f, взаимодействующего как с пользователем, так и со средой. То есть, P = f(R). Из этого следует, что аппликативный или функциональный подход в идеологии построения вы-

числительных систем может быть наиболее естественным и эффективным. Представление логического вывода и доказательства теорем в функциональной комбинаторной среде вписывается в рамки такой идеологии и представляется наиболее эффективным и логичным для веб-среды.

2. Производительность программно-аппаратной среды.

Сегодня все вычислительные системы базируются на классических программно-аппаратных вычислительных средах типа Фон-Неймана. Возможности таких сред по поддержке аппликативных вычислительных систем существенно ограничены, так как они построены для выполнения в основном арифметических, а не функциональных задач. В конечном счете, таким системам приходится решать задачи веб с помощью того, что есть в распоряжении, что является не очень эффективным. Именно поэтому предложена аппаратная поддержка аппликативной комбинаторной среды с комбинаторным базисом {K, S}, соответствующим бинарному представлению данных в битовом представлении {0, 1} в современных программно-аппаратных средах, интегрируемая в классическую архитектуру для повышения производительности вычислений.

Заключение

Как видно из двух систем доказательства теорем и логического вывода существенным отличием аппликативной вычислительной системы является активное участие в процессе доказательства логических связок ∨, ∧, ┐, =>, представленных в комбинаторном виде. В, практически, любом языке программирования мы можем определить одной командой вычисление логического значения применения логической связки только к двум аргументам - логическим значениям, результатом выполнения которой также будет логическое значение. В случае аппликативной вычислительной системы логические связки выражены в том же комбинаторном базисе, что и аргументы и количество логических связок и аргументов может быть более двух, что приводит к унификации выполнения логических операций и существенному повышению производительности вычислений. Таким образом, различные логические операции над аргументами сводятся к одной операции аппликации вместо существующего на сегодня битового сложения логических аргументов. В более сложных и объемных дизъюнктах в предложениях КНФ, чем показанных выше, можно использовать алгоритм перебора и сравнения предикатов, входящих в каждый из дизъюнктов. Комбинаторные выражения возможно выполнять (редуцировать) с помощью комбинаторной абстрактной машины. Более подробно ее работа описана в [4-6]. Для комбинаторной абстрактной машины возможна аппаратная поддержка и сопряжение с существующей традиционной вычислительной архитектурой Фон-Неймана.

Литература

1. Moses Shönfinkel *Über die Bausteine der mathematischen Logik.* // Mathematische Annalen 92. - 1924. - P. 305—316.
2. Nils J. Nilsson *Principles of Artificial Intelligence*. SRI International Tioga Publishing Co, California, USA. - 1980. – 376 p.
3. В.Э. Вольфенгаген, В. Я. Яцук *Аппликативные вычислительные системы и концептуальный метод проектирования систем знаний* / Под ред. профессора Л. А. Майбороды - МО СССР, 1987. – 255 с.
4. В.Э. Вольфенгаген *Комбинаторная логика в программировании Вычисление с объектами в примерах и задачах.* Издание 3-е дополненное и переработанное - Москва: Институт ЮрИнфор-МГУ, 2008. – 383 с.
5. Mazhirin I.V. *Data processing by combinators.* // Proc. of 12th International Workshop on Computer Science and Information Technologies (CSIT'2010). - USATU, Ufa, Russia. - 2010. - Vol. 2. - P. 149-152.
6. Mazhirin I.V. *Combinatory Abstract Machine.* // Proc. of 4th Russian-German Workshop Innovation Information Technologies: Theory and Practice. - USATU, Ufa, Russia. - 2011. - P. 33-35.
7. I.V. Mazhirin *Applicative programming environment by combinators.* // Proc. of 13th International Workshop on Computer Science and Information Technologies (CSIT'2011). - USATU, Ufa, Russia. - 2011. - Vol. 1. - P. 33-35.
8. Mazhirin I.V. *Applicative information system for the web environment. //* MEDES'11: International Conference on Management of Emergent Digital Eco-Systems Proceedings (MEDES'11) November 21-24, 2011 San-Francisco, USA. - 2011. - P. 143-144.
9. Мажирин И.В. *Аппликативная система логического вывода*. // Труды XVII Байкальской Всероссийская конференции «Информационные и математические технологии в науке и управлении» Часть III.-Иркутск: ИСЭМ СО РАН, 2012. – 285 с. – 2012. – С. 75-81.
10. Mazhirin I.V. *Applicative inference system by combinators*. // Proceedings of 14th International Workshop on Computer Science and Information Technologies (CSIT'2012). - USATU, Ufa, Russia. - 2012. - vol. 1. - P. 54-57.

Система описания предметной области на основе семантических конструкций: интерактивная визуализация с использованием силового подхода

Навроцкий В.В.

Институт "ЮрИнфоР-МГУ",
Москва, 119435 РФ, М. Пироговская ул., д. 5, офис 37
vyacheslav.navrotsky@gmail.com

Аннотация. Для удобства работы с сетевым языком системы описания предметной области разработан интерфейс конечного пользователя, использующий интерактивную визуализацию графов прикладных и поддерживающих конструкций. В качестве подхода к автоматическому построению изображения графа выбран силовой подход (магнитно-пружинная модель).

Введение

Семантический метод [1] (специализация метода концептуального моделирования предметной области (ПО) [2]) заключается в построении:

-- языка, конструкции которого предназначены для описания ПО;

-- формально заданной области значений, элементы которой могут быть осмыслены в терминах ПО;

-- отображения означивания (интерпретации), позволяющей соотносить значения конструкциям языка; средств настройки интерпретации, обеспечивающей получение означиваний, соответствующих конкретной задаче или классу задач.

Предложенные средства поддержки были реализованы в виде системы Lux-AZ, обеспечивающей описание ПО на основе семантических конструкций с использованием среды программирования Delphi в качестве поддерживающей системы программирования [3]. Семантические конструкции Lux-AZ, следуя [4], включают понятия, или концепты, и фреймы – простые и составные. Имеется механизм агрегирования семантических конструкций в наборы (модули) с установлением отношений между ними, использующий поддерживающие служебные семантические конструкции (модульная организация) [5]. Было принято решение снабдить систему Lux-AZ графическим интерфейсом конечного пользователя для моделирования ПО, позволяющего работать с визуальными представлениями структур объектов.

Понятие *визуализации информации* (information visualization) весьма широко. Таблицы, графы, карты и даже текст, статические и динамические, служат тому,

чтобы ответить на вопрос, найти отношения, показать лучше то, что будет хуже восприниматься, будучи представленных в других в формах. В более узком смысле, данный термин применяется к визуальному представлению нечисловой информации, такой как файлы и строки кода в программах, библиотеки и библиографические БД, сети отношений в Интернет и т.п. В структурном анализе разработано большое число выразительных средств для проектирования, в том числе графических. В качестве абстрактной структуры для моделирования информации о сложных концептуальных структурах часто применяется граф [6]. Графы используются для представления информации, которая может быть представлена как объекты и связи между ними, т.е. применимы и для визуализации семантических конструкций и связей между их наборами (модульной организации БЗ).

В качестве примера использующей графовое представление визуализации в системах представления знаний, использующих фреймовую модель, можно привести виджет типа GraphWidget – альтернативу формам для создания и заполнения экземпляров классов в редакторе Protégé-Frames [7].

Основные положения

Для систем визуализации графов была предложен [8] и уточнен [9] *основной порядок визуализации* (basic visualization pipeline), состоящий из следующих шагов:

1) Моделирование. Приложение может быть программой, БД, структурой данных, протоколом передачи данных, протоколом безопасности и т.п. Данные об отношениях между сущностями в приложении обычно моделируются в виде графа. На этом этапе граф представляет собой исключительно комбинацию вершин и ребер и не имеет геометрических атрибутов, под которыми понимаются координаты, цвет, размер, форма. Процесс получения графа для приложения называется шагом моделирования.

2) Определение расположения (layout). Граф, созданные на шаге моделирования, обычно изображается в виде вершин, представленных некоторыми геометрическими фигурами, возможно, с текстом, и ребрами, представленными линиями между этими фигурами. Процесс добавления дополнительных геометрических атрибутов к графу для создания изображения называется шагом определения расположения.

3) Уточнение расположения. В интерактивных системах постоянно происходят изменения графа пользователем или программой. Эти изменения могут испортить расположение. Например, добавление новой вершины может привести к её наложению на уже существующую, добав-

ление нового ребра – к пересечению ребер. Поэтому требуется уточнение расположения графа. Процесс изменения геометрических атрибутов графа для приведения изображения в желаемый вид называется шагом уточнения расположения.

К понятиям, характеризующим изображение графа, относят способ изображения, эстетические критерии и ограничения.

Под *способом изображения* (drawing convention) графа понимается основное правило, которому изображение должно удовлетворять. В числе широко используемых способов изображения графа полигональное, прямолинейное, ортогональное, сеточное, восходящее (нисходящее).

Алгоритмы автоматической визуализации графов позволяют получить диаграмму, представляющую структуру графа. Цель процесса размещения в том, чтобы отобразить информацию, содержащуюся в структуре графа, в форме, которая сделает её проще для чтения, понимания и использования, для чего должны, насколько это возможно, выполняться определенные *эстетические критерии* (aesthetics), определяющие параметры изображения.

В то время как способы изображения и эстетические критерии относятся к целому графу (к изображению графа целиком), на определенные подграфы (части изображения графа) могут быть наложены *ограничения* (constraints).

Исходя из особенностей входных данных, требований к изображению и к быстродействию системы, выбирается подход к построению изображения графа: «топология-форма-метрика», иерархический, на основе видимости, силовой или «разделяй и властвуй».

Выбор силового подхода

Силовые алгоритмы являются интуитивными методами для создания прямолинейных изображений графов. Они весьма популярны из-за того, что их базовые версии просты для понимания и программной реализации.

В *силовом подходе* (force-based approach) моделируется система физических сил, задаваемая для исходного графа, которая приходит к конфигурации (т.е. расположению вершин графа на плоскости), который соответствует локальный минимум энергии (положение равновесия).

Данный подход включает две основные составляющие:

- силовую модель (например, мы можем назначить считать, что вершинам соответствуют положительно заряженные частицы, отталкиваемые друг от друга по закону Кулона, а ребрам – пружины, подчиняющиеся закону Гука);
- технику для нахождения конфигурации (взаимного расположения вершин), соответствующей локальному минимум энергии.

Эксперименты с силовыми методами [10, 11 и др.] показали, что они дают хорошие изображения некоторых известных графов, часто дают высоко симметричные изображения и имеют тенденцию к равномерному распределению вершин.

Дополнительно базовые версии могут быть снабжены различными ограничениями, например, могут быть применены специальные силы для того, чтобы расположить вершины в пределах некоторой области или на некоторой кривой.

В качестве метода размещения изображения графа семантической конструкции и модульной организации в системе Lux-AZ выберем силовой метод:

- для визуализации применяемого в Lux-AZ сетевого языка нет сложившейся практики использования конкретного способа изображения, например, ортогонального или сеточного, каковые строятся при использовании других подходов, а [12] указывает в качестве одного из важных эстетических критериев уменьшение числа изгибов ребер;
- радиальная симметричность, к которой стремится изображение графа, получаемое с помощью силового метода, подчеркивает центральное положение предиката в простом фрейме;
- силовой подход несложен в реализации по сравнению с другими методами, в частности, он не требует отдельного шага планаризации и решения сопутствующих задач оптимизации;
- силовая модель имеет большое количество настраиваемых параметров, что делает силовой подход гибким [13];
- силовой подход может быть расширен (как будет показано ниже) таким образом, чтобы визуально выявить иерархию в изображаемых структурах, т.е. может быть применен и для визуализации модульной организации и отношений иерархии между семантическими конструкциями.

Эстетическими критериями, обычно принимаемыми для силовых методов, являются:

1) единообразная длина ребер;
2) минимизация пересечения ребер;
3) выявление симметричности;
4) равномерно распределение вершин;
5) размещение в заданных границы (возможно, специфической формы).

От строгого соблюдения критерия 5 решено отказаться:

- необходимость задания специфических форм области, в которой должно быть размещено изображение графа, для визуализации семантического моделирования, не является очевидным;
- т.к. желательно, чтобы вершины были не точечными, а имели бы достаточную площадь для размещения текстовой информации, а также различались геометрической формой, то при попытке размещения изображения графа с достаточно большим числом вершин на небольшой площади высока вероятность перекрытия геометрических фигур друг другом.

При небольшой площади области размещения изображения графа избежать перекрытия вершин настройкой параметров силовой модели не удается, при этом увеличивается время, за которое будет достигнуто положение равновесия. Силовой алгоритм может быть модифицирован, например, добавлением *силового сканирования* (force-scan) [14], что, однако, довольно сложно в реализации и увеличивает время работы алгоритма притом, что необходимость вписать изображение графа в небольшую область не является критической.

Существуют различные варианты силового подхода, например:

– электрические силы отталкивания между вершинами, соединенными пружинами, с различным выбором функций [10, 15];
– совместное использование евклидовой метрики и кратчайшего пути между двумя вершинами [11];
– использование энергетической модели, метода имитации отжига для нахождения минимума энергии [16, 17].

Эмпирический анализ различных вариантов силового подхода [18], показал, что в целом все они дают приблизительно одинаковые по качеству размещения изображения графа результаты.

В [19] предлагается дополнительный эстетический критерий для силового метода, а именно:

6) соответствие заданной ориентации.

Выполнение этого критерия с той или иной точностью, позволяет в рамках силового метода наглядно изображать иерархические структуры в некотором приближении к иерархическому подходу, подходу «разделяй и властвуй» для деревьев, не отказываясь при этом от изображения структур, в которых отсутствует иерархичность, и не реализуя для визуализации последних отдельного метода.

Метод, описанный в [19], расширяет *пружинную модель* (spring model) из [10, 15] до *магнитно-пружинной модели* (magnetic-spring model) и достаточно прост в реализации: система одинаково заряженных части, соединенных пружинами, некоторые из которых намагничены, помещается в магнитное поле, по направлению которого разворачиваются намагниченные пружины, обеспечивая ориентацию соответствующих им ребер.

Магнитно-пружинная модель

Воспользуемся следующей пружинной силовой моделью [1]:

– каждой вершине графа $u \in V$ соответствует частица имеющая заряд q_u, причем $\forall u, v \in V \frac{q_u}{|q_u|} = \frac{q_v}{|q_v|}$;
– каждому ребру $(u, v) \in E$ соответствует пружина жесткостью k_{uv}.

Тогда на вершину $u \in V$ действует сила

$$F(u) = \sum_{(u,v)\in E} F_{attr}(u, v) + \sum_{(u,v)\in V\times V} F_{rep}(u, v), \qquad (1)$$

где

$F_{rep}(u, v)$ - сила отталкивания, с которой воздействует на u любая другая вершина $v \in V$, следует закону Кулона;

$F_{attr}(u, v)$ – сила притяжения, с которой воздействует на u пружина, соответствующая ребру $(u, v) \in E$, следует закону Гука.

Пусть $d(p,q)$ – евклидова метрика между точками p и q и $p_v = (x_v, y_v)$ - положение вершины v. Тогда для компоненты x силы $F(u)$ из (1) получаем

$$F_x(u) = \sum_{(u,v)\in E} k_{uv}^{(1)}(d(p_u,p_v) - \ell_{uv})\frac{x_u - x_v}{d(p_u,p_v)} + \sum_{(u,v)\in V\times V} \frac{k_{uv}^{(2)}}{\left(d(p_u,p_v)\right)^2}\frac{x_u - x_v}{d(p_u,p_v)},$$

где

ℓ_{uv} – естественная (с нулевой энергией) длина пружины между вершинами u и v;

$k_{uv}^{(1)} = k_{uv}$ – жесткость пружины между вершинами u и v;

$k_{uv}^{(2)}$ – коэффициент (кулоновской силы) отталкивания между вершинами u и v.

Указанные параметры не зависят от координат вершин.

Поскольку вершины не являются точными, чтобы снизить вероятность наложения геометрических фигур, их представляющих, необходимо, чтобы

$$k_{uv}^2 \sim S(u)$$
$$k_{uv}^2 \sim S(v)$$

где $S(u)$ и $S(v)$ – площади геометрических фигур, представляющих вершины u и v в изображении графа. При этом нежелательно, чтобы всё изображение графа в совокупности увеличивалось прямо пропорционально росту размеров вершин или близко к этому.

В силу этого возьмем

$$k_{uv}^{(2)} = q_u q_v$$

и

$$q_u = q' \log_2(xy),$$

где

$q' > 0$ – константа, определенная опытным путем;

$x > 0$, $y > 0$ – соответственно длина и высота прямоугольника, в который вписана геометрическая фигура, представляющая вершину u в изображении графа.

Приведенная силовая модель непосредственно служит для удовлетворения двух эстетических критериев:

- сила притяжения между двумя смежными вершинами u и v обеспечивает примерное равенство расстояния между ними желаемому, т.е. задаваемому параметру ℓ_{uv};
- сила отталкивания обеспечивает такое расположение, при котором любые две вершины не будут слишком близко друг к другу.

Иные эстетические критериев выполняются неявным образом. Так, как говорилось выше, изображение графа, построенного с помощью силового метода, имеет тенденцию быть симметричным.

Приведем модификацию описанных силовой модели и алгоритма так, чтобы могут выполняться заявленный эстетический критерий, касающийся ориентации дуг в выбранном направлении.

Разделим ребра E графа на намагниченные E_m и ненамагниченные E_{nm}:

$$E = E_m \cup E_{nm}$$

В силовой модели намагниченным ребрам соответствуют намагниченные пружины S_m, служащие для управления ориентацией ребер, а ненамагниченным ребрам – обычные пружины S_{nm}. В свою очередь намагниченные пружины разделяются на *одно-* и *двунаправленные*.

При выборе типа пружины руководствуемся следующим:

- дугам, как правило, соответствуют однонаправленные намагниченные пружины;
- ребрам, как правило, соответствуют ненамагниченные пружины или же двунаправленные намагниченные пружины.

Поместим модель в магнитное поле. Можно выделить основные (параллельное, радиальное, концентрическое) и составные (ортогональное, полярно-концентрическое) магнитные поля, которые могут быть получены из основных.

Пусть $b(x, y)$ – характеристика магнитного поля в точке (x, y) и $\vec{m}(x, y)$ – направление магнитного поля в ней, тогда положим, что магнитная индукция в данной точке составляет

$$\vec{B}(x, y) = b(x, y)\vec{m}(x, y)$$

Считая поля равномерными, принимаем в качестве b некоторую константу:

$$b(x, y) = b'$$

Для основных полей можно, очевидно, получаем:

- для параллельного:

$$\vec{m} = (0,1) \qquad \text{«север»}$$
$$\vec{m} = (-1,0) \qquad \text{«запад»}$$
$$\vec{m} = (0,-1) \qquad \text{«юг»}$$
$$\vec{m} = (1,0) \qquad \text{«восток»}$$

(прочие направления получаются сложением указанных выше)

- для радиального:

$$\vec{m} = \frac{x, y}{|(x, y)|}$$
$$B(0,0) = 0$$

- для концентрического:

$$\vec{m} = \frac{(y, -x)}{|(x, y)|} \qquad \text{«по часовой стрелке»}$$
$$\vec{m} = \frac{(-y, x)}{|(x, y)|} \qquad \text{«против часовой стрелки»}$$
$$B(0,0) = 0$$

Силу, с которой действует магнитное поле, положим равной

$$F_{mag} = c_{uv} b d^{\alpha} \theta^{\beta}, \tag{2}$$

где

$c_{uv} > 0$, $\alpha > 0$ и $\beta > 0$ – параметры для настройки модели;

d – расстояние между двумя вершинами, соединяемыми данной дугой;

$-\pi < \theta < \pi$ – угол между направлением поля и направлением однонаправленной намагниченной пружины. В случае двунаправленной намагниченной пружины существует два угла $\theta_1 > 0$ и $\theta_2 < 0$, поэтому берем

$$\theta = \begin{cases} \theta_1, & |\theta_1| = \min\{|\theta_1|, |\theta_2|\} \\ \theta_2, & |\theta_2| = \min\{|\theta_1|, |\theta_2|\} \end{cases}$$

Хотя магнитное поле воздействует с силой F_{mag} на каждую намагниченную пружину, будем вычислять F_{mag} как две силы, равные по модулю и обратные по направлению, действующие на пару вершин, соединяемых ребром, которому соответствует данная намагниченная вершина.

С учетом магнитных сил получаем итоговую силу $\vec{F}(u)$, действующую на вершину $u \in V$:

$$\vec{F}(u) = \sum_{(u,v)\in V\times V} \overrightarrow{F_{rep}}(u,v) + \sum_{(u,v)\in E} \overrightarrow{F_{attr}}(u,v) + \sum_{(u,v)\in E_m} \overrightarrow{F_{mag}}(u,v)$$

Параметры c_{uv}, α и β в (2) подобраны опытным путем:

$$c_{uv} \cong \frac{k_{uv}^{(1)}}{5},$$
$$\alpha = 1,$$
$$\beta = 1.$$

Пусть частица, соответствующая вершине $u \in V$ имеет массу m_u и скорость $\overrightarrow{\mathcal{V}_u}$. В начальный момент времени все частицы неподвижны. Считаем, что под воздействием силы $\vec{F}$ u сообщена скорость

$$\overrightarrow{\mathcal{V}_u'} = \left(\overrightarrow{\mathcal{V}_u} + \vec{F}t\right)d, \tag{3}$$

где

t – шаг времени;

$0 < d < 1$ – коэффициент затухания,

и u смещается в точку

$$p_u' = p_u + t\overrightarrow{\mathcal{V}_u'}$$

Значения t и d в (3) подобраны опытным путем.

В начальный момент времени общая кинетическая энергия системы

$$U = \sum_{u\in V} U_u,$$

где U_u – кинетическая энергия вершины u

$$U_u = m_u \mathcal{V}_u^2,$$

равна нулю.

В результате действия сил кинетическая энергия системы принимает значение $U_i > 0$.

Реализация

В работе используется сторонний freeware компонент построения графов TSimpleGraph 2.80 – визуальный компонент, который обеспечивает простой пользовательский интерфейс для рисования графов. TSimpleGraph является на

данный момент единственным развивающимся средством визуализации графов, написанным на Delphi и соответственно могущим быть интегрированным в систему Lux-AZ без использования дополнительных технологий и средств.

Рассмотрим реализованную процедуру построения изображения графа.

Шаг инициализации. Здесь предварительно вводятся (в виде записей, record) заряженные (с одинаковым знаком) частицы и соединяющие их пружины, представляющие соответственно вершины и ребра размещаемого графа.

В начальный момент времени скорость частицы равна нулю.

Положение частицы устанавливается случайным образом (любые две частицы не должны иметь одинаковые координаты), если для данного графа не был ранее запущен алгоритм размещения. В противном случае берутся координаты размещенных ранее вершин (что соответствует шагу уточнения изображения).

Множитель в зависимости заряда от площади, шаг времени для вычисления скорости и положения частицы, коэффициент затухания, масса частицы были определены экспериментально.

Шаг размещения. Для представлений вершин и ребер, полученных на шаге инициализации запускается силовой алгоритм, определенный выше.

Шаг сдвига. В результате работы силового алгоритма координаты отдельных частиц могут принять:

- отрицательные значения;
- значения значительно больше 0.

Начало координат в рабочем поле исходного компонента TSimpleGraphModel находятся в верхнем левом углу. При попытке непосредственно отрисовать соответствующие вершины, может иметь место ситуация, когда соответственно:

- отрицательные значения координат будут заменены на 0 и произойдет наложение друг на друга отдельных вершин;
- весь граф смещен вправо и/или вниз, при этом сверху и слева останется неиспользованное место.

Поэтому для всех вершины должны быть предварительно смещены на (Δ_x, Δ_y), где

$$\Delta_x = -\min_{\forall i, x_i - \frac{w_i}{2} < 0} \left(x_i - \frac{w_i}{2}\right) - \min_{\forall i, x_i - \frac{w_i}{2} > 0} \left(x_i - \frac{w_i}{2}\right),$$

$$\Delta_y = -\min_{\forall i, y_i - \frac{h_i}{2} < 0} \left(y_i - \frac{h_i}{2}\right) - \min_{\forall i, y_i - \frac{h_i}{2} > 0} \left(y_i - \frac{h_i}{2}\right),$$

(x_i, y_i) – координаты частицы,
w_i и h_i – ширина и высота прямоугольника, в который вписана геометрическая фигура, представляющая вершину.

В результате данного преобразования граф сохранит взаимное расположение вершин, полученное в результате работы силового алгоритма, и будет смещен в левый верхний угол экрана.

На **шаге отрисовки** объектам типа TGraphNode (вершины графа) присваиваются координаты частиц, полученные на шаге сдвига, и далее производится отрисовка геометрических фигур, соответствующих вершинам и ребрам, на новых местах.

Для визуализации правильных структур объектов (семантических конструкций различного типа, модулей) предлагается графическое представление на сети, в котором каждому из описанных элементов ставится в соответствие графический объект: вершина или ребро графа определенного вида. Внешний вид графического объекта определяется его графическими атрибутами. Имея в виду, что при разработке графической нотации желательно иметь представление, которое (а) можно настраивать для удобства пользователя и (б) которое даже при сделанной указанной настройке будет читаемо другим пользователем, знакомым с принципами данной нотации, графические атрибуты разделяются на редактируемые и нередактируемые. Для задания стиля (внешнего вида) вершин и рёбер графа используются служебные семантические конструкции и механизм вычислений расширений последних, что следует принципу однородности среды реализации, принятому в системе.

Компонент интерактивной визуализации содержит ряд сервисных процедур, таких как динамического пропорционального изменения размеров отдельной вершины, определение типа дуги по инцидентным вершинам и отслеживание целостности структуры, скрытие и отображение подграфов с учетом семантики и др.

Заключение

Силовой подход является гибким в том смысле, что, изменяя параметры модели, можно регулировать вклад каждой компоненты и сил, действующих на определенный тип вершин, таким образом, настраивая получаемое изображение. Недостатком является скорость (каждая итерация занимает $O(|V|^2 + |E|)$ времени, где $|V|$ – число вершин в графе и $|E|$ – число ребер), но для графов с числом вершин не более 50, т.е. для большинства случаев, возможных в системе, это ограничение не является критическим. С более сложными графами также можно работать путем последовательной декомпозиции и последующей визуализации частей. Т.к. опытная эксплуатация показала, что удовлетворительные изображения для рассматриваемых структур объектов могут быть получены до достижения локального минимума энергии, используется комбинация нескольких критериев останова, определяемых исходя из числа вершин в графе.

Литература

1. Исмаилова Л.Ю. *Семейство языков категорного типа для обеспечения средств управления семантикой* / Л.Ю. Исмаилова, С.В. Косиков // Материалы международной конфренции «Электронная культура. Информационные технологии будущего». – Астрахань, 2009.

2. Вольфенгаген В. Э. *Методы и средства вычислений с объектами. Аппликативные вычислительные системы* / В. Э. Вольфенгаген. – М.: АО "Центр ЮрИнфоР", 2004. – 789 с.
3. Зайцев А.Е. *Lux-AZ – система описания предметной области на основе семантических конструкций* / Зайцев А.Е., Л.Ю. Исмаилова, С.В. Косиков // III Всероссийская научная конференция с международным участием "Технологии информатизации профессиональной деятельности". Сборник научных трудов. - Ижевск: УдГУ, 2011.
4. Вольфенгаген В.Э. *Система представления знаний с использованием семантических сетей* / В.Э. Вольфенгаген, О.В. Воскресенская, Ю.Г. Горбанев // Вопросы кибернетики. Интеллектуальные банки данных. М.: АН СССР, 1979. – с. 49-69.
5. Навроцкий В.В. *Применение подгонки концептуальных зависимостей и библиотек связанных баз знаний при моделировании предметных областей* / В.В. Навроцкий // Всероссийская конференция «Информационные технологии в образовании XXI века». Всероссийская молодежная научная конференция «Информационные технологии в образовательном процессе исследовательского университета». Сборник научных трудов. – М.: НИЯУ МИФИ, 2011. – с. 52-56.
6. Battista G. Di. *Graph Drawing: Algorithms for the Visualization of Graphs* / G. Di Battista, P. Eades, R. Tamassia, I.G. Tollis. – Prentice-Hall, 1999. – 397 p.
7. *Protege-Frames User's Guide* [Электронный ресурс]. – Электрон. дан. – Stanford Center for Biomedical Informatics Research, 2009. – Режим доступа: http://protegewiki.stanford.edu/wiki/PrF_UG, свободный. – Загл. с экрана.
8. Kamada T. *On Visualization of Abstract Objects and Relations.* PhD thesis / T. Kamada. – Department of Information Science, University of Tokyo, 1988.
9. Lin T. *A General Schema for Diagrammatic User Interfaces.* PhD thesis / T. Lin. – University of Newcastle, 1993.
10. Fruchterman T. *Graph Drawing by Force-directed Placement* / T. Fruchterman, E.Reindold // Software. – Pract. Exp., 21(11): 1129-1164, 1991.
11. Kamada T. *An Algorithm for Drawing General Undirected Graphs* / T. Kamada, S. Kawai. - Inform. Process. Lett., 31:7-15, 1989.
12. Purchase H. *Which Aesthetics Has the Greatest Effect on Human Understanding?* / H. Purchase // Proc. 5th Int. Symp. on Graph Drawing. – 1998. – pp. 248-261.
13. Пупырев С.Н. *Визуализация динамических графов для анализа сложных сетей* / С.Н. Пупырев, А.В. Тихонов // Модел. и анализ информ.систем. – 2010. – Т. 17, №1. – С. 117-135.
14. Misue K. *Layout Adjustment and the Mental Map* / K. Misue, P. Eades. W. Lai, K. Sugiyama // Journal of Visual Languages and Computing, 6. – 1995. – pp. 183-210.
15. Eades P. *A Heuristic for Graph Drawing* / P. Eades // Congressus Numerantium, 42: 149-160, 1984.
16. Davidson R. *Drawing Graphics Nicely Using Simulated Annealing* / R. Davidson, D. Harel // ACM Trans. Graph., 15, no. 4. – 1996. – pp. 301-331.
17. Tunkelang D. *An Aesthetic Layout Algorithm for Undirected Graphs. Master’s Thesis* / D. Tunkelang. – MIT, 1992.

18. Brandenburg F. J. *An Experimental Comparison of Force-Directed and Randomized Graph Drawing Algorithms* / F. J. Brandenburg, M. Himsolt, C. Rohrer // Graph Drawing (Proc. GD'95), vol. 1027 of Lecture Notes Comput. Sci. – Springer-Verlag, 1996. – pp. 76-87.
19. Sugiyama K. *A Simple and Unified Method for Drawing Graphs: Magnetic-Spring Algorithm* / K. Sugiyama, K. Misue // Graph Drawing (Proc. GD'94), vol. 894 of Lecture Notes Comput. Sci., Springer-Verlag, 1995. – pp. 364-375.

Применение искусственных нейронных сетей для распознавания рукописных символов*

Парубец В.В., Девятых Д.В., Берестнева О.Г.

Кафедра ПМ, Институт кибернетики НИ ТПУ,
634004, Томск, ул. Советская, 84/3
vparubets.mail@gmail.com

Аннотация. Описано применение искусственных нейронных сетей в задаче распознавания образов. Исследовано влияние использования пороговых значений нейронов и различных функций активации на точность обучения и распознавания. Показано ускорение обучения сети с применением графических процессоров с поддержкой технологии CUDA

Введение

Теория распознавания образов – раздел кибернетики, который включает теоретические изыскания и методы по классификации и идентификации объектов, процессов и сигналов, которые могут быть охарактеризованы различными свойствами и особенностями. Создание искусственной системы распознавания означает теоретический и технический вызов. Необходимость в подобном распознавании возникает в различных областях – от военных и охранных систем до захвата различных аналоговых сигналов. Одной из нерешённых проблем в данной области является распознавание рукописного текста.

Распознавание образов формально определено, как сопоставление входного образа с ранее определённым классом. Для того чтобы нейронная сеть была способна решать данную задачу, необходимо обучить сеть последовательностью входных образов, сопоставленных с категориями, к которым они принадлежат. По завершению обучения нейронная сеть распознаёт не применявшиеся ранее входные сигналы, описанные всё теми же категориями.

Постановка задачи

Задачей данного исследования является определение различных факторов, влияющих на нейронную сеть: 1) коэффициент скорости обучения, 2) типы активационных функций, 3) архитектура сети, 4) применение технологии CUDA для ускорения обучения.

Описание технологии CUDA

Особенностью оборудования, поддерживающего технологию CUDA (Compute Unified Device Architecture – унифицированная архитектура вычислительного устройства), является возможность обеспечивать на порядок большую (по сравнению с кластерами) пропускную способность при работе с памятью [4]. В графических ускорителях NVIDIA, начиная с восьмой серии, реализована архитектура параллельных вычислений CUDA, которая предоставляет специализированный программный интерфейс для не графических вычислений.

Логически графический процессор с поддержкой CUDA можно рассматривать как набор многоядерных процессоров. Основными вычислительными блоками таких видеочипов являются мультипроцессоры, которые состоят из восьми ядер, нескольких тысяч 32-битных регистров, 16 Кбайт общей памяти, текстурного и константного кэшей.

До официального появления технологии CUDA проводились эксперименты по использованию графических карт настольных систем для реализации потоковых вычислений. Так, с помощью графических программно-аппаратных интерфейсов и представления данных в качестве массивов текстур, удалось добиться троекратного увеличения производительности в экспериментах. Стоит отметить также и одно из ключевых достоинств технологии CUDA – отсутствие необходимости в разработке программ следовать графическим «метафорам» – типам данных и принципам построения вычислений, характерным исключительно для обработки вершин и пикселей при построении кадра.

Реализация нейронной сети с сочетанием CUDA и OpenMP и параллелизма центральных процессоров на уровне языка программирования и программно-аппаратного интерфейса (API) создает еще один уровень прироста производительности. Литература, касающееся не только CUDA, но и предыдущие аппаратные реализации генетических алгоритмов и нейронных сетей демонстрируют ускорение между 2× и 15× [4–6].

Основным источником информации по применению данной технологии может послужить портал разработчика компании NVIDIA, где можно найти как наиболее свежую версию программных библиотек, так и основную информацию о непосредственном внедрении графических процессоров для общих расчетов в своём проекте.

Несмотря на большую информативность портала компании NVIDIA, непосредственные оценки производительности и некоторые тонкости применения графических процессоров для задач численного интегрирования, опирались на работы [4–6].

Таким образом, применение описанных выше алгоритмов в связке друг с другом имеет право на существование и возможность развития.

Входные данные

Объектом исследования в работе являются 70000 оцифрованных изображений рукописных арабских цифр. Эти данные являются частью базы данных MNIST(Maryland National Institute of Standard and Technologies), созданной Американским институтом стандартов. Данные поделены на обучающую и тестирующую выборки, в соотношении 6 к 1.

Каждый из 70000 образов представляет из себя матрицу 28×28 элементов. Каждый элемент матрицы содержит в себе значения цвета пикселя(0 – черный, 255 – белый). Каждому образу присвоена метка, вектор, состоящий из 10 элементов, все элементы которого принимают значение 1 либо 0 (в зависимости от функции активации), элемент, принимающий значение 1, определяет класс, которому относится образ.

Вариации в построении нейронных сетей

Для решения задачи распознавания рукописных символов выбран многослойный перцептрон. Данный класс сетей обучался методом обратного распространения ошибки.

Для исследования влияния параметров нейронной сети было создано 4 экземпляра сетей:

1) Данная нейронная сеть состояла из нейронов, не использующих пороговые значения. Нейроны выходного слоя, не обладали активационной функцией, и каждый нейрон выходного слоя лишь суммировал значения от нейронов скрытого слоя, имеющих с ним синаптические связи.
2) Нейроны скрытого слоя данной сети, включали в себя такой элемент как пороговое значение, что оказало сильное влияние на значения выходов нейронов скрытого слоя, положительно повлияло на результаты тестирования нейронной сети.
3) Нейроны скрытого слоя при создании выходного сигнала использовали пороговое значение, а нейроны скрытого слоя обладали активационной функцией, идентичной активационной функции нейронов скрытого слоя.
4) Нейронная сеть №4 имеет те же параметры, что и сеть №2, но для обучения сети приняты меры для распараллеливания вычислений с помощью CUDA

Все нейронные сети имеют 758 входных, 600 скрытых и 10 выходных нейронов.

Параметр скорость обучения

В результате выбора наиболее часто встречающегося коэффициента скорости обучения нейронной сети, а именно, 0.1. Ни одна из нейронных сетей не дос-

тигла приемлемой точности даже на стадии обучения, более того, при каждом шаге обучения среднеквадратичная ошибка увеличивалась. Аналагочиная ситуация произошла при скорости обучения равной 0.01.

Параметры скорости обучения равные 0.001, 0.0001 и меньше, позволили процессу обучения приблизить результаты обучения к критерию останова, однако, не смогли этого сделать из за ограничения на количество эпох обучения.

Наиболее правильным оказался выбор, сделанный в пользу параметра скорости обучения, изменяющегося со временем. На 1-ой эпохе обучения этот параметр принимал значение равное 0.001, и с каждой эпохой обучения уменьшался на 0.00005, после 18 эпох обучения, изменение скорости обучения прекращалось, однако обучение продолжалось еще 5 эпох. Данный параметр скорости обучения позволил нейронным сетям достичь критерия останова равному 0.1, и не превысить количество эпох обучение, равное 30.

Активационная функция

В качестве активационных функций, для каждой сети были использованы логистическая и функция гиперболического тангенса. Для функции гиперболического тангенса константа обе константы принимали одинаковые значения.

Параметр крутизны сигмоиды, принимающий значение меньше 0.5 приводил к тому, нейронные сети не достигали критерия останова, и обучение сети останавливалось по достижении максимального количества эпох обучения.

Параметр крутизны сигмоиды принимавший значения от 5 и выше, приводил к тому, что симгоида вырождалась в функцию, близкой функции еденичного скачка, что отрицательно сказывалось на результатах обучения.

Точность

Таблица 1 показывает точность, достигнутую при тестировании нейронных сетей в зависимости от функций активации и наличия пороговых значений сети.

Таблица 1

Точность, достигнутая при тестировании сети

Функция активации	$\frac{1}{1+e^{-v_j(n)}}$	$\tanh\left(v_j(n)\right)$	$\frac{1}{1+e^{-3v_j(n)}}$	$3\tanh\left(3v_j(n)\right)$
Сеть №	точность			
1	**54,4%**	**51,9%**	**68,7%**	**48,1%**
2	**84,2%**	**83,3%**	**84,5%**	**81,3%**
3	**93,4%**	**90,8%**	**92,7%**	**85,3%**

Наилучший результат в тестировании показала сеть №3 (нейроны скрытого слоя при создании выходного сигнала использовали пороговое значение, а нейроны скрытого слоя обладали активационной функцией, идентичной активационной функции нейронов скрытого слоя). Применение сигмоидальной активационной функции позволило достичь точность 93,4% при распознавании рукописных символов.

Применение технологии CUDA

Данные результаты достигнуты с нейронной сетью со следующими параметрами:

1) сигмоидальная функция активации,
2) количество входных нейронов – 758, скрытых – 600, выходных – 10,
3) параметр крутизны сигмоиды – 1.

Таблица 2

Скорость обучения

Реализация алгоритма обучения	**Время обучения (ч:мм)**	**ускорение**
ЦПУ, 1 поток	6:43	-
CUDA	0:55	7,2x

Несмотря на существенную разницу в назначении алгоритмов, исполняемых на GPU, можно выделить набор общих инструкций, которых следует придерживаться при составлении программы для GPU:

1. Выделение набора инструкций исполняемых одним экземпляром ядра.
2. Выделение набора данных, общего для блока ядер и загрузка его в разделяемую память.
3. Если объемы данных на шаге 2 слишком велики или набор инструкций выделенных на шаге 1 предполагает слишком интенсивные вычисления задача декомпозируется на более простые задачи.
4. Производятся вычисления над подгруженными ранее данными.
5. Результаты вычисления копируются назад, в глобальную память.

Так же стоит особое внимание уделять размерностям блоков и сетки. Так, если ядра содержат большое количество инструкций - стоит уменьшить размерность блоков. Это позволит более эффективно использовать регистровую память GPU.

При сравнительно малом количестве потоков большую роль играют задержки, связанные с передачей данных с хоста на видеокарту, что существенно снижает итоговую производительность. Увеличение количества вычислительных потоков и, соответственно, объёма данных, обрабатываемых одновременно, позволяет избежать данное «бутылочное горлышко» технологии CUDA.

Заключение

Оценивая влияние архитектуры искусственной нейронной сети и параметров функции активации в задаче распознавания рукописных цифр, можно заключить, что наиболее оптимальной с точки зрения точности является применение многослойного перцептрона с сигмоидальной активационной функцией и применением порогового значения при создании выходного сигнала.

Применение мер по распараллеливанию вычислений на этапе обучения сети с использованием графических процессоров позволяет добиться существенного (в 7 раз) ускорения.

Литература

1. Галушкин А.И. О методике решения задач в нейросетевом логическом базисе // Нейрокомпьютер.- 2006.- №2.- С.49-70.
2. Мызников А.В., Россиев Д.А., Лохман В.Ф. Нейросетевая экспертная система для оптимизации лечения облитерирующего тромбангиита и прогнозирования его непосредственных исходов // Ангиология и сосудистая хирургия.- 1995.- №2.- С.100.
3. Горбань А.Н., Россиев Д.А. Нейронные сети на персональном компьютере.- Новосибирск: Наука.- 1996.- 276с.
4. Harris M. Mapping Computational Concepts to GPUs. // GPU Gems 2.-Addison Wesley.- 2006.- С.493-508.
5. J. D. Hall, N. A. Carr, and J. C. Hart. Cache and Bandwidth Aware Matrix Multiplication on the GPU. Technical Report. -UIUCDCS-R-2003-2328, URL: http://graphics.cs.uiuc.edu/~jch/papers/UIUCDCS-R-2003-2328.pdf
6. Horn D. Stream Reduction Operations for GPGPU Applications. // GPU Gems 2. - Addison Wesley.- 2006.- С. 573-589
7. Хайкин С, Нейронные сети полный курс, 2-е изд., испр.: пер. с англ. – М.:ООО «И.Д. Вильямс», 2006. 1104 с.
8. Лазарев И.А., Русаков В.И., Ткачева Т.Н. Иммунопатологические изменения у больных язвенной болезнью желудка и двенадцатиперстной кишки и их иммунокоррекция // 4 Всесоюзный Съезд гастроэнтерологов.: Тез. докл. - Москва, Ленинград, 1990.- Т.1.- С.702-703.
9. А.Н.Горбань, В.Л.Дунин-Барковский, А.Н.Кардин «Нейроинформатика» Отв. Ред. Новиков Е.А., РАН, Сиб. Отд., Институт выч. Моделирования – Новосибирск: Наука, 1998.

Построение авторских ограничений на редактирование конструкций практикума в сервисе групповой разработки системы тестирования

Парфёнова И.А.

Кафедра ПКИ и ИТ, Институт ``ЮрИнфоР-МГУ",
Москва, 119435 РФ, М. Пироговская ул., д. 5, офис 37
irina.a.parfenova@gmail.com,

Аннотация. Основная цель создания сервиса групповой разработки системы тестирования заключается в повышении эффективности сопровождения практикумов группой независимых экспертов. Заявленная цель достигается, в том числе, за счет задания авторских ограничений на редактирование конструкций практикумов, которые ранее не применялись в системах сопровождения систем тестирования и системах смежных классов.

Введение

Повышение качества тестирования обучаемых требует создания специализированных средств подготовки и сопровождения практикумов группой независимых экспертов. В силу противоречивости взглядов экспертов в предметных областях (ПО), особый интерес видится в предоставлении возможности настройки авторских ограничений на изменение созданных ими конструкций практикума (КП). Исследование возможностей систем тестирования и систем смежных классов (СДО, CMS, автоматизации документооборота, контроля версий) показали, что они не предоставляют специализированных механизмов управления авторскими ограничениями. Под КП будем понимать названия вопросов, их текст, комментарии к ним, альтернативы ответов, комментарии к ним и их стоимость.

Ключевая проблематика

В процессе создания сервиса групповой разработки системы тестирования (СГРСТ) была предложена совокупность теоретических методов, описывающих взаимодействие экспертов, учитывающая задания экспертам функционально-тематических ролей и построения авторских ограничений для работы на уровне КП в темах практикумов, в разделах терминологического словаря, с метаданными практикумов.

Экспертам, создавшим в теме практикума вопрос, предлагается механизм задания ограничений на использование и редактирование КП, принадлежащих их вопросам другими зарегистрированным в СГРСТ экспертами. Разработаны и реализованы две модели построения авторских ограничений: простая и расширенная.

В простой модели эксперту, создавшему в практикуме вопрос, предлагается либо разрешить редактирование данного вопроса с сохранением его авторства, либо разрешить редактирование вопроса с удалением его авторства, либо запретить редактирование вопроса. Под вопросом в данном контексте понимается набор всех КП, ему принадлежащих. Также эксперт может зафиксировать максимальное количество допустимых альтернатив ответа на вопрос, разрешить экспертам устанавливать ограничение на время ответа и добавлять к вопросу отметку «основного».

В расширенной модели эксперту предлагается построить формулу, связывающую право на действие с экспертами (ФПД). Таким образом, ставится и решается задача задания и разбора ФПД, в том числе рекурсивной. Рассмотрим пример ФПД:

&(Редактировать ответ, Редактировать комментарий к вопросу, Редактировать название вопроса, {_Статус[1][Эксперт-волонтер]})
(Редактировать комментарий к ответу нет, Редактировать комментарий к ответу да, Добавить ответ, {#(_Роль_в_теме[2][Семейное право; Корректор])(_Опыт>3[1][Семейное право])}).

Эксперт, построивший данную ФПД, разрешает редактировать три определенных КП экспертам, зарегистрированным в СГРСТ со статусом волонтера, а также разрешает выполнять другие три действия в рамках его вопроса экспертам, обладающим функциональной ролью «корректор» в теме практикума «семейное право», или экспертам, имеющим опыт работы в теме «семейное право» более трех лет. При разработке расширенной модели применялась логика исчисления предикатов первого порядка в семантике Тарского и фрагменты алгебры фреймов. Строится алфавит языка задания ФПД, содержащий предикатные символы, описывающие характеристику экспертов, символы действия, логические связки (и, или, не), служебные символы и знаки операций. При построении ФПД используется префиксная форма записи. Запишем формальную грамматику, используя расширенную форму Бекуса-Наура:

Формула = & «(» Формула «)» «(» Формула «)» | Выражение.
Выражение = Список действий { Характеристики экспертов }
Список действий = Действие, Список действий | Действие,
Характеристики экспертов = Связка (Предикат) (Предикат) | Предикат
Предикат = Имя предиката [Операция Число] «[» Арность «]» «[» Значение { «,» Значение} «]».
Значение = «Семейное право» | «Корректор» | «Эксперт-волонтер» | …
Связка = «&» | «#» | «n».
Имя предиката = «_Статус» | «Роль_в_теме» | «_Опыт» | …
Операция = «>» | «<» | «=» | «!=».
Число = «0» | «1» | «-1» | «2» | «-2» | …
Арность = «1» | «2» | «3».

Далее строится семантика описывающая множество ФПД. Задается набор предикатов, с каждым из которых связывается определенная характеристика экспертов. Для примера рассмотрим семантическую интерпретацию двухместного предиката _Роль_в_теме[2][Семейное право; Корректор] на заданных конкретных значениях. На первом аргументном месте находится конкретная тема практикума, на втором функциональная роль. Рассматриваемый предикат принимает истинное значение, в случае если эксперт обладает ролью «Корректор» в теме практикума «Семейное право». Логические связки интерпретируются стандартным образом в двузначной логике.

Информация о правах на создание / редактирование КП, об экспертах, внесших изменения, будем называть метаданными практикума. Организуем метаданные практикума в семантическую сеть. Семантическая сеть – модель ПО, имеющая вид ориентированного графа, вершины которого соответствуют объектам ПО, а дуги задают отношения между ними[5]. Таким образом, семантическая сеть отражает семантику ПО в виде понятий и отношений между ними. На Рис.1 представлен фрагмент семантической сети метаданных практикума.

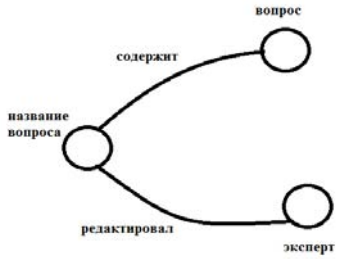


Рис.1. Фрагмент семантической сети хранения метаданных практикума.

Выделим способы хранения метаданных практикума, организованных в семантическую сеть:

1. БД (ссылки между Таблицами БД – дуги семантической сети, строки в Таблицах БД – вершины семантической сети);
2. XML (вершины и дуги семантической сети – атрибуты XML документа).

Критерии выбора способа хранения метаданных:

1. обеспечение совместимости практикумов в различных системах;
2. степень сложности реализации механизмов интерпретации данных;
3. обеспечение целостности метаданных.

Основываясь на хранении метаданных в виде семантической сети, в компоненте задания авторских ограничений СГРСТ реализуется функция, определяющая наличие права на конкретное действие у эксперта. Функция двух аргументов возвращает значение истины, если право на действие доступно эксперту, или значение лжи в ином случае.

Задается структура хранения предикатов в файле формата XML: имен, диапазонов значений, сравнимости, информации по их обработке и описания.

Фрагмент XML:

<characteristic>

<name>_Роль_в_теме</name>
<compare>нет</compare>
<ranges> <range>Тема</range> <range>Роль</range> </ranges>
<function>them_expert</function>
<parameters>
<result>Эксперт обладает функциональной ролью (1) в теме (2)</result>
</characteristic>

Использование приведенной структуры хранения предикатов позволяет реализовать механизмы задания предикатам произвольных имен и динамического определения новых предикатов нулевой арности, что делает интерфейс пользователя более настраиваемым и дружелюбным. Произвольные имена задаются предикатом путем создания копии XML-файла, хранящего данные о предикатах, для каждого пользователя СГРСТ. Полученные копии отличаются именами предикатов, расположенных в узле «name». Новый предикат может быть объявлен пользователем за счет объединения имеющихся предикатов логическими связками. Например, предикат _Не_волонтер может быть задан как n(_Статус[1][Эксперт-волонтер]). На данном этапе разработки семантически проинтерпретированы 7 предикатов и определены семнадцать действий над КП.

При процессе генерации выходных практикумов учитываются передаваемые экспертами юридические правомочия на воспроизведение, копирование и публичный показ созданных ими в СГРСТ вопросов. Юридический контроль работы экспертов может быть осуществлен на основании задаваемых в СГРСТ лицензионных соглашений.

Заключение

Решена задача построения авторских ограничений на редактирование КП в СГРСТ. Исследование возможностей систем сопровождения тестирования и систем смежных классов: СДО, CMS, автоматизации документооборота, контроля версий с созданной Автором ранее версией СГРСТ позволили выделить пути расширения СГРСТ. В отличие от разработанного СГРСТ описанные системы содержат ряд частных решений по сопровождению практикумов, но не содержат специализированных механизмов групповой поддержки практикумов, позволяющих задавать экспертами права на редактирование КП, принадлежащих им вопросов. В процессе создания СГРСТ была предложена совокупность теоретических методов, описывающих взаимодействие экспертов, учитывающая задания экспертам функционально-тематических ролей для работы на уровне КП в темах практикумов, а также механизм задания ограничений на использование и редактирование КП их вопросов другими экспертами.

Используя различные подходы к формализации поддержки групповой разработки практикумов, управляемой правами, разработаны специализированные механизмы сопровождения. Расширенная модель, связывающая право на действие с экспертами, разработана с применением логики исчисления предикатов первого порядка в семантике Тарского и фрагментов алгебры фреймов. Заданы синтаксис и семантика предикатов (проинтерпретированы семь предикатов и

семнадцать прав на действия), построена формальная грамматика разбора ФПД, в том числе рекурсивных. Предложена структура хранения предикатов, позволяющая экспертам задавать им произвольные имена и динамически определять новые предикаты нулевой арности.

При выполнении работы были зафиксированы требования к семантической сети хранения метаданных практикумов, способами хранения выбраны: БД и XML-документ. Определены требования к структуре хранения метаданных, реализованы процедуры доступа / модификации метаданных и их семантического контроля. Для организации интерпретации программного кода использован сервер Apache, функциональные возможности серверной компоненты СГРСТ реализованы на языке PHP, для хранения данных использована база данных MySQL, в которой задано сорок отношений. Пользовательские интерфейсы реализованы на языке HTML и JavaScript.

Подготовлена коммерческая версия СГРСТ, комплект поставки которой содержит инсталлятор, инструкцию по установке, справочную информацию, эксплуатационную документацию по ГОСТ 19.101-77. СГРСТ зарегистрирован в ФГУП НТЦ "Информрегистр" с номер государственной регистрации: 0321201224.

Литература

1. Исмаилова Л.Ю., С.В. Косиков. *Ситуационно обусловленное оценивание для управления семантикой в обучающих системах по юриспруденции*. – Прикаспийский журнал: управление и высокие технологии №2(14) 2011. – с. 68-73.
2. Исмаилова Л.Ю., Косиков С.В. *Информационные системы и технологии*. Гл. 2. Монография. Красноярск: Научно-инновационный центр, 2011. -156 с.
3. Вольфенгаген В.Э. *Методы и средства вычислений с объектами. Аппликативные вычислительные системы*. М.: Институт «ЮрИнфоР-МГУ», 2004. - 778 с.
4. Вольфенгаген В.Э. *Конструкции языков программирования. Приёмы описания*. – М.: АО «Центр ЮрИнфоР», 2001.
5. Вольфенгаген В.Э. *Аппликативные вычислительные технологии. Готовые решения для инженера, преподавателя, аспиранта, студента*. М.: АО "Центр ЮрИнфоР", 2009. –64 с.
6. Ахо А.В. *Компиляторы: принципы, технологии и инструментарий*, 2-е изд. / А.В. Ахо и др.; пер. с англ. – М. : ООО «И.Д. Вильямс», 2008. – 1184 с. : ил. – Парал. тит. англ.
7. Тарский А. *Введение в логику и методологию дедуктивных наук* / А. Тарский; пер. с англ. – М.: Гос. изд-во иностранной литературы, 1948. – 327 с.
8. Кревский И.Г. *Формальные языки, грамматики и основы построения трансляторов*: Учебное пособие / И.Г. Кревский, М. Н. Селиверстов, К. В. Григорьева. Под ред. А. М. Бершадского – Пенза: Изд-во Пенз. гос. ун-та, 2002. – 124 с.

Двухуровневая концептуализация предметных областей средствами дескрипционной логики

Рословцев В.В. Шапкин П.А.

Кафедра №22 «Кибернетика», НИЯУ МИФИ,
Москва, 115409, Каширское ш., 31
vladiros@gmail.com p.shapkin@gmail.com

Аннотация. Во многих предметных областях возникают задачи, требующие многоуровневой классификации, не только объекты данных подвергаются классификации, но и метаданные, в частности, сами классы объектов, требуют категоризации. Стандартных возможностей дескрипционной логики, однако, недостаточно для решения такой задачи; в частности, такой возможности не предоставляет и каркас Jena. Предлагается решение этой проблемы с помощью выделения каждого уровня категоризации в отдельную модель, а связи соотнесения между уровнями устанавливать посредством аннотаций.

Введение

Особенностью многих предметных областей, например, предметной области ERP-систем, являются то, что необходимо не только классифицировать объекты предметной области, но и сами классы (и свойства) объектов. Причем этот процесс в некоторых случаях может быть рекурсивным: категории классов объектов могут повергаться категоризации, а эти категории, в свою очередь, снова классифицироваться, и т.д.

Такие задачи возникают в связи необходимостью формализованного описания закономерностей, встречающихся в структуре предметных областей.К числу таких закономерностей относятся возникающие иерархии объектов и их классов, характер связей между объектами, и др.

Возможностей дескрипционной логики [4] недостаточно для непосредственного моделирования таких многоуровневых классификационных схем непротиворечивым образом. В частности, такой возможности не предоставляет и каркас Jena. Для преодоления этой трудности, будем использовать двухуровневую схему концептуализации[1]: на первом уровне в роли объекта классификации будут выступать объекты предметной области, а на втором – классы объектов. Далее встает задача установления соотнесения между этими уровнями, а именно: соотнесения классов первого уровня с индивидами, представляющимиэти классы на втором уровне.

Двухуровневая схема концептуализации предметных областей

В процессе концептуализации предметной области решаются следующие задачи:

- выявляется состав классов объектов предметно области;
- для каждого класса объектов выявляется множество характеристик – базовых свойств, присущих объектам этого класса;
- выявляются связи между классами объектов.

Оставляя в стороне вопрос идентификации классов, остановимся подробнее на том, что собой представляют характеристики. Под характеристикой мы понимаем свойство, областью значения которого является один из заранее фиксированного множества базовых доменов.

На некотором этапе выделяется набор базовых, атомарных доменов, которые на практике привязываются к базовым машинным типам данных – целым и вещественным числам различной размерности, булев тип, строковый, и другие. Подобный перечень задает множество базовых типов в каждом языке программирования. Стоит отметить, что набор таких типов оказывается более или менее общепринятым, что, по крайней мере, в принципе, существенно облегчает организацию интероперабельности [2] между программами, написанными на различных языках программирования.

Это же обстоятельство создает основу и для обеспечения интероперабельности на семантическом уровне, т.е. на уровне моделей предметной области: основу описания каждого класса составляет перечисление присущих ему характеристик. В таком случае, соотнесение между двумя различными концептуальными моделями предметной области (КМПО) требует установления следующих соответствий:

1. между отдельными классами объектов;
2. для соответствующих классов – между их характеристиками;
3. связей между объектами;
4. в общем случае могут возникать пограничные ситуации, когда свойство, представляющее собой характеристику некоторого класса в одной КМПО, соотносится со связью между объектами в другой КМ той же предметной области.

Установление таких соотнесений требуется в тех случаях, когда необходимо обеспечить совместное функционирование двух или более (информационных) систем, каждая из которых оперирует собственной, отличной от прочих схемой концептуализации. Для двух фиксированных схем, соотнесение между ними может быть определено посредством жесткого указания преобразования для объектов каждого типа. Однако этот подход неэффективен, в частности, если:

- хотя бы одна из схем подвержена (относительно частым) изменениям, например, вследствие изменений в протекающих в предметной области (информационных) процессах или/и их моделях в соответствующей программной системе;
- велико количество отдельных преобразований, которые необходимо указать, вследствие большого количества объектов в предметной области или/и (относительно) большого количества взаимодействующих систем.

В таком случае классы объектов предметной области и связи между ними, в свою очередь, необходимо классифицировать и оснащать системой метаданных. В частности, относительно свойств необходимо указывать, являются ли они характеристиками или связями, а для связей указывать кардинальность. В результате, помимо собственно концептуальной модели предметной области, которую будем называть моделью первого уровня, требуется также построить «модель модели» (которую будем называть моделью второго уровня). Если в модели первого уровня понятиями являются классы объектов предметной области, то в модели второго уровня центральных понятий два – Класс и Свойство. Возникает также иерархия, служащая для классификации понятий первого уровня.

Поскольку принцип организации модели второго уровня, в целом, тот же, что и у модели первого уровня, эффективным оказывается использование тех же средств концептуализации.Проблема заключается в том, что эти две модели должны быть разнесены, во избежание противоречий, а между ними приходится устанавливать связь.

Применение техник аппликативных вычислительных систем для этих – одно из хороших решений; однако, по тем или инымпричинам, проектировщик может быть ограничен в своем выборе, и ему приходит оставаться в рамках классической дескрипционной логики. В этом случае к числу стандартных инструментов относится язык OWL [6], среда концептуального моделирования Protégé[5], каркас классов Jena, позволяющий использовать аппарат дескрипционной логики в программных системах.

Конструкции дескрипционной логики в Jena

В дескрипционной логике основными понятиями являются концепты, роли, представляющие собой бинарные отношения на индивидах, и дескрипции, представляющие собой предложения дескрипционной логики, описывающие составные роли и концепты. В Jena концептуальные модели записываются в терминах классов, свойств и индивидов:

- Класс (OntClass) – понятие; класс объектов предметной области. Концепты могут быть как простые, так и производные – объединения/пересечения других классов, или же концепты, созданные посредством ограничения – некоторого необходимого (достаточного, необходимого и достаточного) условия, налагаемого на индивиды.
- Свойство (OntProperty) – способ выражения когнитивного представления об объектах соответствующего класса – области определения свойства (т.е. множество возможных индивидов, к которым это свойство применимо). Свойства в Jena бывают нескольких видов:
    - характеристики(Datatypeproperty), о которых речь уже шла выше;
    - объектные свойства (ObjectProperty), описывающие связи между объектами предметной области, в том числе и такие, как отношение «часть-целое», дочерне-родительские связи;

- аннотации(Annotationproperty) – особый вид свойств, не являющийся частью формальной модели и служащий для вспомогательных целей;
- Индивид (Individual) – собственно, конкретные (действительные?) объекты в предметной области.

Все перечисленные элементы называются ресурсами и наделяются идентификаторами в форме URI, которые должны быть уникальны, по крайней мере, в пределах одной модели – логически связанной совокупности всех рассматриваемых ресурсов. Отметим, что выразительных возможностей дескрипционной логики, поддерживаемой Jena, достаточно для задания, в частности, кардинальностей связей, что позволяет различать связи «один-к-одному», «один-ко-многим», «много-ко-многим». Это также инструмент, позволяющий фиксировать, например, является то или иной свойство «обязательным», т.е. может ли его значение быть «пустым» или нет.

В то же время, дескрипционная логика, сама по себе, недостаточна эффективна для решения задачи описания вычислительной схемы свойств. Дело в том, что некоторые свойства объектов могут простыми – в том смысле, что это просто данные, которые некоторым образом хранятся. Но некоторые свойства могут быть вычислимыми, причем алгоритм вычисления может быть нетривиальным и даже основываться на связях объекта с другими объектами. Дескрипционная логика не позволяет эффективно решать эту задачу.

Некоторым решением этой проблемы является погружение дескрипционной логики в аппликативную вычислительную среду[3].

Соотнесение различных уровней концептуализации

Особенностью описываемого подхода является то, что оба уровня описываются одни и теми же средствами, т.е. как классификация объектов ПОна первом уровне, так и категоризация классов на втором – осуществляется средствами дескрипционной логики. Это означает, что сущность, представляющая собой класс с точки зрения первого уровня, переходит в разряд экземпляров на втором. То же происходит и со свойствами: они становятся экземплярами соответствующих классов.

Требуется, таким образом, установить связь между классами и свойствами и представляющими их на вышестоящем уровне индивидами. Эту задачу можно разделить на две подзадачи:

- установить, к какому «метаклассу» относится рассматриваемая интенсиональная конструкция первого уровня;
- установить, каким конкретным индивидом этого «метакласса» данный конкретный класс будет представлен.

На практике данные задачи можно решать следующим образом. В Jena предусмотрена особая категория свойств – аннотации (annotationproperty); аннотации могут быть приписаны к классу, свойству или индивиду и служат для связывания с тем или иным элементом модели каких-либо метаданных. Поскольку аннотации, с точки зрения Jena, являются своего рода свойствами, на аннотаци-

ях также определено отношение вложенности. Роль аннотаций аналогична роли атрибутов в таких языках программирования, как Java или C#.

Таким образом, для каждого класса из второго уровня («метакласса») в первом уровне необходимо предусмотреть аннотацию; связь между аннотациями и «метаклассами» удобно осуществлять за счет приписывания идентификаторов – URI, совпадающих с точностью до некоторого префикса, специфичного для модели каждого уровня. В результате, с одной стороны, соотнесение аннотаций с «метаклассами», с одной стороны, оказывается алгоритмически почти тривиальным, а с другой – уникальность идентификации различных сущностей сохраняет глобальный характер, предотвращая возможные парадоксы. Необходимо отметить, что если моделирование на обоих классификационных уровнях можно осуществлять одними и теми же средствами, то соотнесение между этими уровнями осуществляется внешними, по отношению к дескрипционной логике, средствами.

Заключение

В настоящей статье был кратко рассмотрен подход к организации многоуровневой концептуализации средствами дескрипционной логики за счет ее оснащения дополнительными, внешними средствами. В случае, когда использования возможностей десприпционной логики в программной системе обеспечивается средством, подобным каркасу классов Jena, предлагается использовать механизм аннотаций для установления связи между моделями разных уровней.

Ряд вопросов, связанных с рассматриваемой проблематикой, но оставленных нами без рассмотрения, включает организацию собственно процесса взаимодействия двух (или более) систем, что требует принятия ряда инженерных решений.

Литература

1. Вольфенгаген, В.Э. Аппликативные вычислительные системы и концептуальный метод проектирования систем знаний / В. Вольфенгаген, В. Яцук // М: МО СССР. – 1987.
2. Брюхов Д. Интероперабельные информационные системы: архитектуры и технологии / Д. Брюхов, В. Задорожный, Л. Калиниченко и др. // СУБД. – 1995. – Т. 4.
3. Шапкин, П. А. Вычисления с концептами в аппликативном языкепрограммирования / П. А. Шапкин // Аппликативные вычислительные системы: Труды конференции по аппликативным вычислительным системам (АВС’2010) / Под ред. В. Э. Вольфенгаген. – М.:НОУ Институт Актуального образования «ЮрИнфоР-МГУ», 2010. – С. 205–213.
4. The Description Logic Handbook: Theory, Implementation, and Applications / F. Baader, D. Calvanese, D. McGuinness et al. – Cambridge University Press, 2007. – 622 pp.

5. The Protégé OWL plugin: An open development environment for semantic web applications / H. Knublauch, R. Fergerson, N. Noy, M. Musen // Lecture notes in computer science. – 2004. – Pp. 229–243.
6. OWL, язык веб-онтологий. Руководство. Рекомендация W3C 10 февраля 2004. http://sherdim.rsu.ru/pts/semantic_web/REC-owl-guide-20040210_ru.html.

Аппликативные методы декомпозиции вычислительных процессов

Рословцев В.В. Шумский Л.Д.

Кафедра №22 «Кибернетика», НИЯУ МИФИ,
Москва, 115409, Каширское ш., 31
vladiros@gmail.com shumskyld@gmail.com

Аннотация. В настоящей работе анализируется техника декомпозиции вычислений на основе понятия комбинатора – в системах со связыванием переменных и в системах, позволяющих программировать без переменных. Далее рассматривается техника декомпозиции вычислений на основе структурной декомпозиции объектов – как вычислителей, так и объектов-данных. В основе такой структурной декомпозиции лежит обобщенная модель аппликативных объектов.

Введение

В настоящее время, в условиях, когда: постоянно растет сложность протекающих в предметных областях информационных процессов и связанных с ними вычислений; экспоненциально растут объемы обрабатываемых данных, причем возрастает доля метаданных, а также их сложность; многие предметные области характеризуются высокой степенью изменчивости, а значит, требуется оперативно вносить изменения в модель предметной области и протекающих в ней процессов – в таких условиях ключом к обеспечению эффективности (информационных) систем оказывается декомпозиция (и повторное использование) вычислительных процессов.

При этом возникает две относительно независимые проблемы:

- формулировка (вычислительной) задачи, причем, предпочтительно, с введением минимально возможного количества новых элементов, а это требует декомпозиции задачи на более простые составляющие и использование уже ранее определенных задач;
- определение, по крайней мере, одного метода решения каждой задачи, и предпочтительным оказывается максимальное использование уже существующих наработок, что снова требует композиционного подхода, при котором новые решения строятся в форме композиции ранее созданных вычислительных компонент.

Решение первой из указанных проблем обеспечивает концептуальную целостность программной системы и минимальность понятийного каркаса предметной области, а также высокий аналитический потенциал ее модели. Решение второй проблемы приводит к гибким архитектурам открытого типа, которые, с

одной стороны, обеспечивают относительно высокую степень повторного использования программного кода, а с другой – качественно упрощают расширяемость и адаптивность программной системы.

Попытки решения, по крайней мере, второй из указанных проблем проявляются в современных тенденциях к использованию архитектур сервис-ориентированного типа (см. например [1]). Это включает не только разработку приложений на основе «классических» веб-сервисов, которые, как правило, обслуживаются веб-серверами и взаимодействуют по SOAP или другим высокоуровневым протоколам, но и ситуации, когда управление жизненным циклом сервисов и взаимодействие между ними обеспечивается иными средствами. Работой сервиса может управлять специальный функционал операционной системы – например, службы в операционных системах семейства MicrosoftWindows, – или/и специализированные приложения – например, консольные, обеспечивающие пользовательский интерфейс в виде командной строки (CLI – Command-LineInterface). Сам сервис при этом, с точки зрения операционной системы, может быть некоторым дочерним процессом. Взаимодействие сервисов может обеспечиваться самыми различными способами, зависящими от специфики конкретной программной системы: от «локальной» шины типа DBus до тех или иных надстроек над стеком протоколов TCP/IP – как стандартизованных, так и разработанных специально «под задачу».

Композиционный подход к организации вычислений обеспечивает аппликативный компьютинг, суть которого – в рассмотрении вычисления как относительно самостоятельного блока, построенного композиционным образом из уже имеющихся блоков вычислений, причем все переменные в каждом блоке связаны, а сами блоки – замкнуты [2]. В настоящей работе анализируется техника декомпозиции вычислений на основе понятия комбинатора – вλ-исчислениях, в которых формализуется связывание переменных и подстановка,а также в системах, позволяющих программировать без переменных[2,3]. Далее рассматривается техника декомпозиции вычислений на основе структурной декомпозиции объектов – как вычислителей, так и объектов-данных. В основе такой структурной декомпозиции лежит обобщенная модель аппликативных объектов[5,6], нацеленная, в частности, на формальное исследование способов построения более сложных объектов из более простых.

Анализ аппликативных техник декомпозиции вычислений

В аппликативных вычислительных системах (АВС) объекты представляют собой функциональные сущности и:

- основной способ построения составных объектов – операция аппликации, или применения, одного объекта к другому, что содержательно понимается как применение объекта-функции к объекту-аргументу;
- объекты способны, в зависимости от контекста, выступать как в роли функции, так и в роли аргумента, причем допускается самоприменимость объектов;

- число аргументных мест (арность) объектов-функций потенциально не фиксировано заранее и проявляется постепенно в процессе взаимодействия с другими объектами.

К числу АВС относятся системы, основанные на комбинаторной логике (КЛ) или λ-исчислении. Класс объектов АВС строится индукцией по структуре объектов, где в качестве базовых блоков выступают атомы (в общем случае – переменные и атомарные объекты). Объекты АВС, по своей структуре, отвечают, по крайней мере, следующим требованиям:

1. (базис индукции)
   a. предполагается наличие бесконечного множества переменных и некоторого (возможно, пустого) множества атомарных объектов;
   b. переменные и атомарные объекты называются атомами; атомы суть объекты;
2. (шаг индукции) если A и B – некоторые объекты, то (AB) – объект.

С точки зрения организации и декомпозиции вычислительных процессов одним из ключевых оказывается понятие комбинатора – объекта, не содержащего свободных переменных. Переменная называется свободной в некотором объекте, если она имеет хотя бы одно свободное вхождение в этом объекте. Понятие свободного вхождения переменной в объект определяется индукцией по построению объекта.

В качестве примера будем рассматриватьвычисление Fумножения комплексных чисел на некоторое фиксированное комплексное число $a + ib$. Будем считать, что $\mathbb{C}$–конструктор комплексных чисел, так что если x и y – некоторые вещественные числа, то $c = \mathbb{C}xy$ – комплексное число $x + yi$, причем $\mathrm{Re}\, c = x$ и $\mathrm{Im}\, c = y$. Будем использовать следующую схему декомпозиции вычислительного процесса F:

$$F = g \circ \langle f_1 \circ p_1, f_2 \circ p_2 \rangle,$$

где f_1 и f_2 – некоторые «подпроцессы», а p_1 и p_2 – некоторые «проекции», задача которых – преобразовать входные данные для процесса F к виду, пригодному для обработки, соответственно, процессами f_1 и f_2. Преобразование входных данных может, в частности, включать извлечение из них некоторой части с последующей дополнительной обработкой. Компонент g выполняет некоторую агрегацию частичных результатов, полученных от f_1 и f_2, в некоторый объект, который и будет считать результатом F.Цепочка равенств, приведенная ниже, демонстрирует, как будет проходить вычисление с использованием указанной схемы декомпозиции:

$$Fx = (g \circ \langle f_1 \circ p_1, f_2 \circ p_2 \rangle)x = g(\langle f_1 \circ p_1, f_2 \circ p_2 \rangle x) = g[(f_1 \circ p_1)x, (f_1 \circ p_1)x] = g[f_1(p_1 x), f_2(p_2 x)]$$

Комбинаторная логика. В комбинаторной логике фиксируется некоторые число атомов, в том числе несколько, образующих базис, а единственной метаоперацией является аппликация. Объекты формируют индуктивный класс, который классически определяется следующим образом:

1. (базисиндукции)
   a. предполагается наличие бесконечного множества переменных;
   b. предполагается наличие конечного или бесконечного множества т.н. атомарных объектов, в том числе объектов **K** и **S**;

c. переменные и атомарные объекты называются атомами; атомы суть объекты;

2. (шаг индукции) если A и B – некоторые объекты, то(AB) – объект.

Понятие свободного вхождения переменной (например, переменной x) в некоторый объект(A) комбинаторной логики определяется индукцией по структуре объекта. Выражение «переменная xвходит свободно в объект X» кратко будем записывать в виде «$x \in \mathrm{FV}(X)$»:

1. $A = x$, тогда $x \in \mathrm{FV}(x)$ (любая переменная входит свободно в саму себя);
2. $A = (MN)$, тогда $x \in \mathrm{FV}(MN) \Leftrightarrow x \in \mathrm{FV}(M) \vee x \in \mathrm{FV}(N)$.

В КЛ комбинатором оказывается объект, составленный без использования переменных (т.е. только из атомарных объектов). КЛ предоставляет средства описания вычислительных процессов в константном виде, т.е. используя лишь атомарные объекты и их композицию. Существенным вопросом, таким образом, оказывается выбор атомарных вычислительных процессов. Этот выбор, с одной стороны, существенно зависит от специфики предметной области; с другой стороны, существуют вычислительные процессы, инвариантные относительно предметной области, и хорошими кандидатами на роль таких фундаментальных процессов являются, в частности, объекты **K** и **S** [3].

Для того чтобы продемонстрировать, как в комбинаторной логике осуществляется рассматриваемая схема декомпозиции, необходимо представить модели объектов $\cdot\circ\cdot$ (композиция) и $\langle\cdot,\cdot\rangle$ (т.н. спаривание) – а также объектов: $[\cdot,\cdot]$ (пара, точнее конструктор пары), π_1 (первая проекция пары) и π_2 (вторая проекция пары). Воспользуемся для этого результатами из [MCBO]:

$$f \circ g = Bfg \qquad [a,b] = Dab = BC(CI)ab$$

$$\langle f,g\rangle = BS(BD)fg \qquad \pi_1 = CIK \qquad \pi_2 = CI(KI)$$

Особенность использования комбинаторной логики состоит в том, что все вычислительные процессы, в конечном итоге, необходимо описать с помощью некоторого количества заранее фиксированных базисных комбинаторов. Построения такого описания – задача, вообще говоря, нетривиальная, т.к. выбранный набор комбинаторов, в предположении, что он полон и такое разложение вообще возможно, может оказаться малопригодным для выражения с его помощью задач того или иного класса: выражения будут слишком громоздкими. С другой стороны, набор базисных комбинаторов фиксируется заранее, а количество входящих в него комбинаторов относительно невелико, что позволяет выполнять реализацию, либо поддержку, этих комбинаторов на аппаратном уровне. Это приводит к резкому повышению эффективности вычислений.

Таким образом, применение комбинаторных техник может быть удачным для предметно-ориентированных решений, когда предварительная фиксация специализированных комбинаторов может обеспечить как лаконичное представление вычислительных процессов, так и высокую эффективность самих вычислений.

λ-исчисление. В λ-исчислении наличие предопределенных атомарных объектов не является обязательным: все объекты могут быть синтезированы средствами самого λ-исчисления. Индуктивный класс термов задается следующим образом:

1. (базисиндукции)
   a. предполагается наличие бесконечного множества переменных;
   b. предполагается наличие конечного или бесконечного множества т.н. атомарных объектов;
   c. переменные и атомарные объекты называются атомами; атомы суть объекты;
2. (шаг индукции)
   a. если A и B– некоторые объекты, то(AB)–объект
   b. если A– объект, а x– некоторая переменная, то$(\lambda x.A)$–объект.

λ-исчисление, таким образом, не предполагает наличие каких-то предопределенных,исходных объектов, таких, как **K** или **S**. Вместо этого предоставляется добавочное средство комбинирования объектов – операция функциональной абстракции, которая связывает одну переменную в объекте. λ-исчисление позволяет при конструировании вычислительных блоков явно указывать число и тип (в системах с типами) формальных параметров и формализовать передачу фактических аргументов – посредством операции подстановки. Замкнутость блоков достигается связыванием с помощью метаоперации функциональной абстракции всех переменных, использованных при конструировании блока. В λ-исчислении понятие свободного вхождения некоторой переменной x в (некоторый) объектA определяется следующим образом:

1. $A = x$, тогда $x \in \mathrm{FV}(x)$ (любая переменная входит свободно в саму себя);
2. $A = (MN)$, тогда $x \in \mathrm{FV}(MN) \Leftrightarrow x \in \mathrm{FV}(M) \vee x \in \mathrm{FV}(N)$;
3. $A = (\lambda y.M)$, причем $x \neq y$, тогда $x \in \mathrm{FV}(\lambda y.M) \Leftrightarrow x \in \mathrm{FV}(M)$.

Относительно исходных объектов, если они есть, обычно принимается неявное соглашение, что они не содержат свободных переменных.

Аналогично тому, как это было проделано для комбинаторной логики, необходимо предъявить модели используемых объектов в λ-исчислении:

$$f \circ g = (\lambda xyz.x(yz))fg \qquad [a,b] = Dab = (\lambda xyr.rxy)ab$$

$$\langle f,g\rangle = (\lambda xyz.D(xz)(yz))fg \qquad \pi_1 = \lambda x.x(\lambda xy.x) \qquad \pi_2 = \lambda x.x(\lambda xy.y)$$

При использовании λ-исчисления для организации вычислений потенциал метода последовательной декомпозиции может использоваться наиболее полно. Выделяемые на каждом этапе подпроцессы могут подвергаться дальнейшей декомпозиции, а обнаруженные повторяющиеся или однотипные фрагменты могут быть обобщены и выделены в отдельные объекты, с последующим многократным их использованием. В результате такой последовательной декомпозиции необходимые примитивы *синтезируются*. Недостаток применяемого подхода заключается в более трудоемком процессе вычисления из-за интенсивности выполнения операции подстановки. Кроме того, при полном отказе от использования исходных объектов процесс собственно описания вычислительных процессов может оказаться чрезмерно трудоемким.

Отсутствие жесткой привязки к заранее фиксированному набору примитивов приводит к нескольким возможным последствиям:

- возникает принципиальная возможность формировать необходимый базис объектов по необходимости, в т. ч. «на лету»;

- в результате последовательной декомпозиции могут быть сформулированы требования, исходя из которых можно уже подбирать набор целевых исходных комбинаторов;
- при наличии нескольких заранее заготовленных наборов исходных комбинаторов, можно выбирать один из них – наиболее подходящий.

Применение λ-исчисления для декомпозиции вычислений наиболее перспективно в ситуациях, где наибольшее значение имеет универсальность, возможность адаптации к широкому кругу задач.

Обобщенная модель аппликативных объектов

Принцип, по которому строятся объекты АВС, достаточно прост и моделируется процессом построения решетки потоковых диаграмм, причем в роли потоковых диаграмм выступают комбинаторы. Общая такова: выбирается некоторое, пустое или непустое, конечное или бесконечное, множество исходных объектов, которые затем, с помощью заранее фиксированного конечного набора «конструкторов» комбинируются в более сложные объекты. Эти относительно сложные объекты, наравне с исходными, могут использоваться для построения еще более сложных объектов, и т.д.

Исходные объекты. Исходные объекты, как правило, являются частью формулировки конкретных АВС. В программировании в роли исходных объектов выступают стандартные библиотеки, различные каркасы и т.п. Назначение исходных объектов – обеспечить достаточный базис для создания конструкций определенного класса.

Среди исходных объектов некоторые могут быть сложными по своей структуре, а некоторые – могут и не поддаваться дальнейшей декомпозиции: либо потому, что это по какой-либо причине невозможно, либо потому, что это нежелательно или нецелесообразно (например, потому, что нарушает логическую целостность системы, см. также главу 9, в частности, раздел 9.1, в [4]). Такие неделимые объекты называются первичными (см. [3]). Прочие исходные объекты, не являющиеся первичными, будем называть вторичными. Подобно остальным сложным объектам (т.е. не входящим в состав в исходных и называемых поэтому производными), вторичные объекты строятся из первичных (и других исходных объектов) с помощью ранее упомянутых конструкторов. Включение тех или иных неатомарных объектов в состав исходных является вопросом обеспечения удобства использования проектируемой АВС, а не гарантией ее каких-либо фундаментальных свойств, и способствует, в частности, лаконичности производных объектов.

Способы композиции объектов. Операция аппликации является основным способом построения более сложных объектов из более простых, и эта операция присутствует во всех АВС, по определению. Но при формулировании конкретной АВС могут вводиться и другие способы комбинирования объектов, например, операция функциональной абстракции. По причинам, изложенным в работе [5,6], такие операции мы будем называть термообразующими функциями (ТОФ). Существенным является следующее обстоятельство: из всех ТОФ толь-

ко использование операции аппликации может приводить к возникновению новых редексов в составном объекте.

Пусть a и b – некоторые объекты, а F – некоторая термообразующая функция (отличная от аппликации). Тогда составной объект Fab не будет содержать иных редексов, кроме тех, что содержались в объектах a и b.

Обобщенная аппликативная предструктура. Модель АСВ удобно представлять аппликативной структурой, которая, в свою очередь, классически записывается в виде пары следующего вида[3,4]:

$$(\boldsymbol{O}, \varepsilon),$$

где $\boldsymbol{O}$– множество объектов АСВ, ε ··– операция аппликации. Приведенная запись соответствует случаю среды без типов. В случае среды с типами, аппликативная структура принимает вид:

$$(\{\boldsymbol{O}^{\sigma}\}, \{\varepsilon_{\sigma\tau}\}),$$

где$\{\boldsymbol{O}^{\sigma}\}$и $\{\varepsilon_{\sigma\tau}\}$– семейства с параметрами σ и τ. $\boldsymbol{O}^{\sigma}$– совокупность объектов среды, принадлежащих типу σ. При этом $\boldsymbol{O} = \sum_{\sigma} \boldsymbol{O}^{\sigma}$. Символу $\varepsilon_{\sigma\tau}$ будем приписывать тип $\boldsymbol{O}^{\sigma} \times \boldsymbol{O}^{\tau} \to \boldsymbol{O}$.

Относительно того, какой тип приписывать ТОФ, в частности, аппликации, отметим следующее. Во-первых, базовых возможностей две: либо это будет «функциональный» тип вида $T_1 \to \cdots \to T_p \to T_{p+1}$, либо «операторный», вида $T_1 \times \ldots \times T_p \to T_{p+1}$. Второй вариант отражает, например, то обстоятельство, что «частично построенные» объекты, как правило, не рассматриваются. С другой стороны, первый вариант связан с более мягкими требованиями к выразительным возможностям метатеории: в АВС отсутствует изначальное понятие о декартовом произведении и кортежах, так что «функциональный» подход, в некотором смысле, более «естественный». С точки зрения настоящей статьи выбор того или иного варианта непринципиален.

Поскольку все объекты АВС строятся из первичных с помощью фиксированного набора ТОФ, для того, чтобы задать набор объектов АСВ, достаточно перечислить исходные объекты и набор ТОФ. Условимся обозначать такие наборы исходных объектов символом $\boldsymbol{D}$ (быть может, с индексами). Набор термообразующих функций договоримся обозначать $\boldsymbol{F}$(быть может, с индексами). Поскольку речь идет об аппликативных средах, наличие операции аппликации подразумевается по умолчанию. Тогда аппликативнаяпредструктураАВС представляется парой следующего вида[5,6]:

$$(\boldsymbol{D}, \boldsymbol{F})$$

Одно из преимуществ данной модели состоит в том, что она достаточно хорошо описывает, как аппликативные объекты моделируются средствами языков программирования. При использовании объектно-ориентированного подхода строятся иерархии классов, в которых выделяются классы атомарных объектов; кроме того, как правило, каждой ТОФ сопоставляется класс объектов, построенных с ее помощью. В функциональных языках основной метод заключается в использовании типов-сумм, когда каждой ТОФ соответствует некоторый конструктор.

Техника декомпозиции вычислений на основе обобщенной модели аппликативных объектов

В основе хорошо построенной декомпозиции вычислений лежит идея разделения ответственностей, которая, в данном случае, выражается в том, что каждый вычислительный процесс «обрабатывает» только объекты определенного типа или определенной категории. Каждый поступивший на «обработку» данному процессу объект подвергается декомпозиции на составляющие его части, и каждая такая часть передается в соответствующий специализированный процесс – который может быть как самостоятельной сущностью, так и структурной составляющей данного процесса. Частичные результаты, полученные этими «подпроцессами», подвергаются некоторой агрегации, в частности, они могут «упаковываться» с помощью некоторой ТОФ в результирующий объект.

Пусть$\mathcal{F}: T_1 \to \cdots \to T_m \to T_{m+1}$ – некоторая ТОФ. Для каждой такой ТОФ определим семейство объектов $\pi_{\mathcal{F}}^{i}: T_{m+1} \to T_i$ (естественно, $i \in [1, m]$), так что:

$$\pi_{\mathcal{F}}^{i}(\mathcal{F} a_1 \dots a_m) = a_i,$$

для любых $a_1: T_1, \dots, a_m: T_m$.Рассмотрим объект $\mathbb{H}^n$ – «оператор»каррирования, такой что:

$$\mathbb{H}^n f[x_1, \dots x_n] = f x_1 \dots x_n$$

Например, для $n = 2$:

$$\mathbb{H} = C\big(BS(CB\pi_1)\big)\pi_2 \qquad\qquad \mathbb{H} = \lambda f p. f(\pi_1 p)(\pi_2 p)$$

Приведем достаточно общую схему вычислительного процесса, осуществляющего преобразование объектов вида $F a_1 \dots a_m$в объекты вида $G b_1 \dots b_l$:

$$F = \mathbb{H}^l \mathcal{G} \circ \langle g_1, \dots, g_l \rangle \circ \langle f_1 \circ \pi_{\mathcal{F}}^{1}, \dots, f_m \circ \pi_{\mathcal{F}}^{m} \rangle$$

В этой схеме f_i – подпроцессы, осуществляющие покомпонентное преобразование конституент объекта $F a_1 \dots a_m$, а g_k отвечают за вычисление отдельных компонент результирующего объекта $G b_1 \dots b_l$, причем в данной схеме отражен тот факт, что каждый b_k зависит либо от всех $f_i a_i$, либо от некоторых из них, либо ни от каких – в зависимости от вида соответствующей g_k. Существенно, что при такой схеме декомпозиции каждая из компонент b_k вычисляется независимо от остальных. Преимущества такой схемы очевидны:

- каждый из элементов $f_1, \dots, f_m, g_1, \dots g_l$может быть как внутренним компонентом F, так и внешней, самостоятельной сущностью;
- каждый из элементов $f_1, \dots, f_m, g_1, \dots g_l$ может быть заменен на «аналогичный» (например, такой же по типу) ему независимо от остальных;
- более того, схемы декомпозиции можно рассматривать «сами по себе», как объекты вида

$$\lambda g_1 \dots g_l f_1 \dots f_m. \mathcal{G} \circ \langle g_1, \dots, g_l \rangle \circ \langle f_1 \circ \pi_{\mathcal{F}}^{1}, \dots, f_m \circ \pi_{\mathcal{F}}^{m} \rangle$$

- сама схема декомпозиции может подменяться без обязательной замены/модификации объектов $f_1, \dots, f_m, g_1, \dots g_l$;
- схемы такого вида хорошо подходят для различных архитектур приложений сервис-ориентированного типа.

В качестве иллюстрации, покажем, как с помощью такой схемы записать упомянутый в начале статьи процесс вычисления умножения комплексных чисел на некоторое фиксированное комплексное число $a + ib$:

$$\mathbb{HC} \circ \langle \mathbb{H}(\lambda xy.\, a \cdot x - b \cdot y), \mathbb{H}(\lambda xy.\, a \cdot y + b \cdot x) \rangle \circ \langle \mathrm{Re}, \mathrm{Im} \rangle$$

Заключение

В работе рассмотрены схемы вычислительных процессов, обеспечивающие основу для проектирования сложных вычислительных сред, в которых:

- существенно обеспечение распределения вычислений между несколькими физическими или логическими вычислителями;
- требуется снизить сложность расширения/модификации функционала системы.

В целом, это открытые архитектуры, подразумевающие слабую связность компонент.

Один из существенных вопросов, оставшихся за рамками тематики настоящей работы, – это семантика собственно распределения вычислительных блоков и объектов данных (исходных, а также промежуточных и конечных результатов вычислений) между отдельными вычислительными узлами. Эти вопросы исследуются, например, в работе [7], в которой разрабатывается сопряжение λ-исчисления и модельной логики с типами.

Еще одна интересная проблема связана учетом доступности вычислительных мощностей: каждый элемент вычислительной сети, вообще говоря, имеет конечную «пропускную способность». Это включает в себя два вопроса: во-первых, о наличии достаточных ресурсов для решения той или иной задачи (в частности, в рамках некоторого временного интервала); во-вторых, – оптимизация использования вычислительных ресурсов, распределение нагрузки между узлами.

Литература

1. Богданов А.В., Станкова Е.Н., Мареев В.В. Сервис-ориентированная архитектура: новые возможности в свете развития GRID-технологий. / Всероссийский конкурсный отбор обзорно-аналитических статей по приоритетному направлению "Информационно-телекоммуникационные системы", 2008. – 32 с.
2. Вольфенгаген В.Э. Комбинаторная логика, или настойчивое приглашение к компьютингу // Предисловие к 3-му изданию книги «Комбинаторная логика в программировании». – 3-е изд., дополн. ипереpаб. – М.:АО «Центр ЮрИнфоР», 2008.— X+384 с.
3. Вольфенгаген В.Э. Кварки, атомы, молекулы компьютинга // Аппликативные вычислительные технологии. Готовы решения для инженера, преподавателя, аспиранта, студента / Под ред. к.т.н. Л.Ю. Исмаиловой. – М.: ЗАО «ЮрИнфоР®», 2009. –сс. 6-44.

4. Вольфенгаген В.Э. Методы и средства вычислений с объектами. Аппликативные вычислительные системы. – М.: АО «Центр ЮрИнфоР», 2004. – XVI+789 с.
5. Рословцев В.В. Структура и метод формирования аппликативной вычислительной среды. // Сборник научных трудов SWorld. Материалы международной научно-практической конференции «Современные проблемы и пути из решения в науке, транспорте, производстве и образовании ‘2011». – Выпуск 4, том 5. – Одесса: Черноморье, 2011. – С. 14-22.
6. Roslovtsev V.V., Luchin A.E. Concept of Higher-Order Applicative Computational Environment.// Proceedings of the 11th international workshop on computer science and information technologies CSIT'2009, Crete, Greece, 2009. – pp. 48-53.
7. T. Murphy, K. Crary, R. Harper, F. Pfenning. A Symmetric Modal Lambda Calculus for Distributed Computing. Computer Science Department, 2004. Paper 465. http://repository .cmu.edu/compsci/465

Конечно-элементное моделирование кварцекристаллических нановесов на основе микрокамертонных резонаторов

Матисон Н.Л., Симонов В.Н.

Кафедра Физики прочности,
Национальный исследовательский ядерный университет «МИФИ»,
Москва, 115419 РФ, Каширское ш.31
simonov.valer@yandex.ru,

Аннотация. Приводятся результаты конечно-элементного моделирования микрокамертонного резонатора, покрытого наноразмерным покрытием с исследуемыми механическими и физико-химическими свойствами.

Для разработки математической модели микрокамертонного резонатора был использован метод конечных элементов и возможности проведения модального анализа в программном комплексе конечно-элементного анализа «ANSYS».

В качестве конечного элемента для разбиения построенной по рабочим чертежам трёхмерной модели использовался конечный элемент SOLID186. Этот элемент представляет собой гексаэдр. Он может принимать вырожденные призматическую и тетраэдральную формы. Во всех формах элемент обладает 20 узлами и 60 степенями свободы, имеет функцию формы второго порядка (квадратичную) с линейным распределением производных характеристик (напряжения, деформации), допускает постановку с учётом физической и (или) геометрической нелинейностей.

В качестве механических свойств материалов использованы матрица упругих констант кристаллического кварца, а для моделирования свойств плёнок – значения модуля Юнга и плотности в пределах, типичных для большинства технических пленок (полимерных, металлических и пр.): модуль Юнга в пределах от 0.001 до 100 ГПа, плотность от 0.01 до 10 г/см^3. Сетка модели содержит 79872 элемента и 364934 узла с 1094802 степенями свободы, гексаэдральная, с корректным соотношением сторон элемента. На элементах покрытия (толщина покрытия 2 мкм) разбиение построено, исходя из требования наличия на каждом из структурных составляющих конструкции не менее двух элементов в направлении наименьшего размера.

На *рис. 1* в качестве иллюстрации изображена изометрическая модель суммарных перемещений при изгибных колебаниях исследуемого микрокамертонного резонатора.

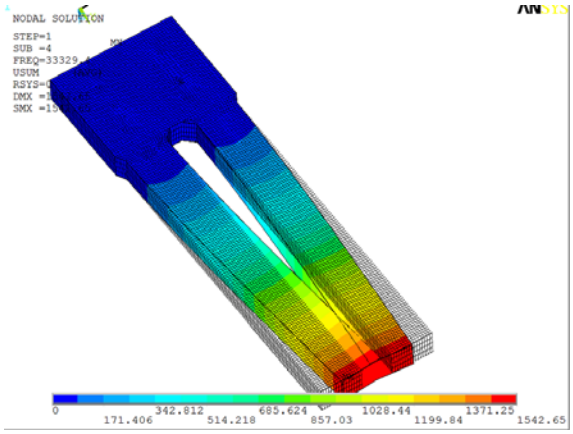


Рис.1. Изометрическая модель суммарных перемещений при изгибных колебаниях исследуемого микрокамертонного резонатора.

Для исследования зависимости чувствительности резонанса микрокамертонного резонатора к изменению модуля Юнга и плотности плёнки проводилось моделирование влияния указанных свойств плёнки на собственную частоту изгибных колебаний микрокамертонного резонатора. Так как производить напыление нанообъекта по причине объемной формы микрокамертонного резонатора можно с трех сторон, было проведено моделирование влияния на резонансную частоту не только изменений механических свойств плёнки, но также расположения плёнки. В связи с этим были рассмотрены 4 варианта нанесения плёнки на ветви микрокамертонного резонатора: на лицевые поверхности, на боковые поверхности, на торцевые поверхности, на все поверхности одновременно. Эти варианты проиллюстрированы на графиках *рис. 2* для зависимости частоты резонатора от модуля Юнга покрытия.

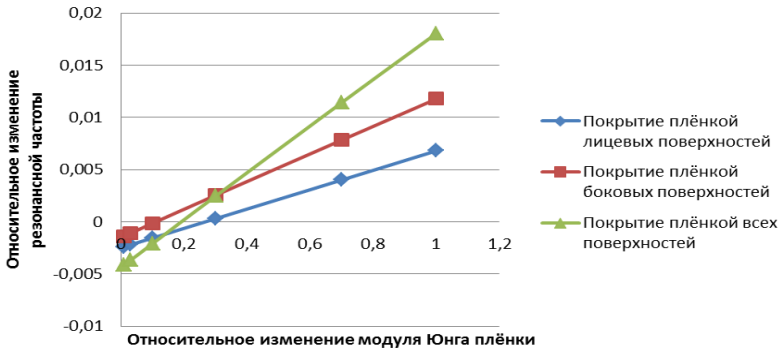


Рис.2, а. Относительные изменения частоты в зависимости от относительного изменения модуля Юнга пленки

По результатам проведённого моделирования установлено, что, во-первых, зависимость относительного изменения частоты dF_s от плотности плёнки ρ' и модуля Юнга плёнки E' является линейной в широких пределах отношения толщины пленки h' к толщине микрокамертона H, во-вторых, изменение частоты под влиянием пленки, нанесенной на все поверхности одновременно, является алгебраической суммой изменений частот под влиянием пленки, нанесенной на боковые, лицевые и торцевые поверхности, в-третьих, изменение частоты может быть приведено к выражению:

$$dF_S = (h'/H)\, K_E(b,H)(E'/E) - K_\rho(b,H,L)(m'/M) \qquad (1)$$

где $K_E(b,H)$ и $K_\rho(b,H,L)$ - коэффициенты, зависящие от ширины b, толщины H, длины L струн микрокамертона и от того, на какие поверхности микрокамертона нанесена пленка, M – масса единицы площади микрокамертона. Из-за ограничений объема статьи нет возможности привести коэффициенты K_E и K_ρ для всех вариантов нанесения пленки.

Supporting problem-oriented conceptual dependencies in the web-application

Slobodskoy E.G.

Institute for Contemporary Education "JurInfoR-MGU",
Moscow, 119435 RF, M. Pirogovskaya ul., 5, office 37
jeneksl@gmail.com

Abstract. Developing web-application that estimate facts from specific domains required considering different opinion logic. For solving this purpose author suggested to develop methods for implementing web-application that would use formalization models with different opinion logic. It is shown that using specialized frame oriented tools gives some advantages. In this article are presented some ways of web-application building. It is shown that service-oriented architecture could be realized as web-services exchanging XML-messages. Handling of XML-messages realized by using SAX-parser. It gives a possibility of dynamical managing user's interface. The library of classes containing description of parser methods and keeping XML-handling results was made. The interfaces are combined by basic interface constructions: label, text field, table, tree, list, button. This list of basic interface constructions could be changed by the overriding method of defining it's elements. Java-applet technology has chosen as a technology for developing web-application. Server component will be developed via PHP.

Introduction

Formalization of subject domains is still one of the fundamental problem of the computer science. There are several subject domains, which are especially difficult to formalize. Some of them have next specific properties: (1) objects has the opportunity to assign properties to another objects, (2) this properties can conflict to each other and dynamically change. In this paper has considered the following subject domain - «a swindling under the Criminal Code of the Russian Federation and similar situations».

The attempt to formalize chosen subject domain and to realized web-application, which supports frame constructions based formalizations, is made in this paper. One of the main functions of the projected system is data entering. This data is interpreted as extension of some frame, so it is necessary to take into consideration user properties:

- user's platform characteristic;
- user's individual characteristics.

Accordingly, one of the main problem of the developing such system is providing an ability to manage user's interface dynamically.

Implementation of web-system

Service-oriented architecture could be implementing thru wide range of technologies: REST, RPC, CORBA, DCOM, SOAP. We decided to choose technologies corresponding to the criteria: necessity in application server, and amount of data transmitted. In this paper the decision to use RESTful web-services is made. The client of web-service is implemented thru Java-applet technology.

Methods proposed

To providing an ability to manage user's interface dynamically is proposed to implement message generator and message handler. These messages contain graphical user interface information. So the generated and handled messages should be specified. The set of base graphical constructions is defined. In this paper this set consists of label, text field, button, table, tree, list and panel. All of these constructions are part of Java Swing library. These constructions can be classified by the way of use in the web-system:

- Output elements: label, table, tree, list,
- Input elements: text field, table, tree, list, button.
- Elements that used to combined other elements: panel.

System work tied with generating, transferring and handling potentially critical amount of data. Execution of this operation can lead to appearing the OS message «application is not response». Time-consuming operations are executed in separate threads to provide friendly graphical user interface.

The main purpose of the implemented web-system is to support frame constructions. Thus, it is necessary to provide input/output ways for frame intension and extension. In this paper decision to expand the set of graphical construction by the combining of base graphical constructions has made. As a result graphical construction «frame» is obtained. «Frame» is a combination of panel, label and of the following elements: table, tree, and list.

Generated messages

XML-message contains an information about graphical user interface. In this paper the example of right message with appropriate comments are shown.

```
<JPanel NAME=" JLABEL=" LAYOUT=" AXIS=" LAF=">
    <JPanel NAME=" LAYOUT=" AXIS=" LAF=">
        <Frame NAME=" TYPE=" FNAME=">
        </Frame>
```

```
</JPanel>
<JPanel NAME=" LAYOUT=" ROWS=" COLUMNS=" LAF=">
        <JButton NAME=" JLABEL=" IMAGE=" ALIGN=" HANDLER="/>
        <JButton NAME=" JLABEL=" IMAGE=" ALIGN=" HANDLER="/>
        <JButton NAME=" JLABEL=" IMAGE=" ALIGN=" HANDLER="/>
</JPanel>
</JPanel>;
```

where:

Table 1: mapping tags and GUI-components

JPanel	javax.swing.JPanel
JLabel	javax.swing.JLabel
JButton	javax.swing.JButton
JTextField	javax.swing.JTextField
Frame	myFrames.Frame

Parameters's description:

Table 2: parameters's description in the XML-message

Tag	Parameter	Description
JPanel	NAME	The name of this panel
	LAYOUT	The name of layout
	AXIS	horizontal/vertical sequence of the component's position
	ROWS	The count of rows
	COLUMNS	The count of columns
	LAF	Name of the graphical style Look&Feel
JLabel	NAME	The name of this label
	JLABEL	The displayed text on this label
JButton	NAME	The name of this label
	JLABEL	The displayed text on this button
	IMAGE	The displayed image on this button
	HANDLER	The name of executable function
JTextField	NAME	The name of this text field
	JLABEL	The displayed text on this text field

Demonstration

Formalizations of subject domains similar to swindling are used as demonstration. This formalizations are obtained by using type theory [1], [2].

Architecture

The common architecture of client-server application is shown in Fig. 1 below.

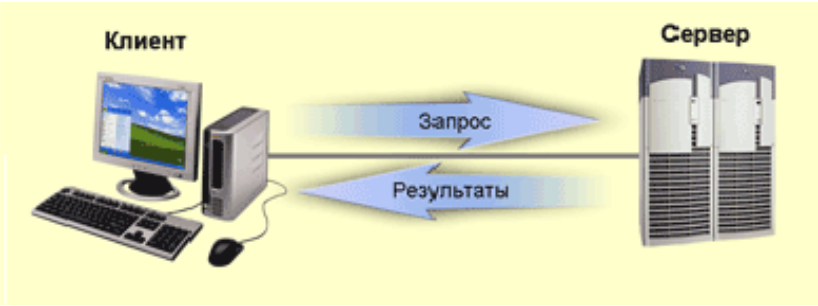


Figure 1: The architecture of client-server application

Web-service

The web-service is implemented in PHP language and contains four modules:

- formalism – contains extensional and intensional frame characteristics;
- eftest – describes income messages handling algorithm;
- nucleus – describes the rules of frame executions;
- work_with_files – describes outcome messages transferring to client side and writing data to database.

Client of the web-service

The client of web-service is implemented thru Java-applet technology and contains three modules:

- Frames - contains classes descriptions, these classes implement frame language and expands standard graphical constructions from java library «swing». The design pattern «strategy» [3] is used to abstract frame behavior from its way to visualize.

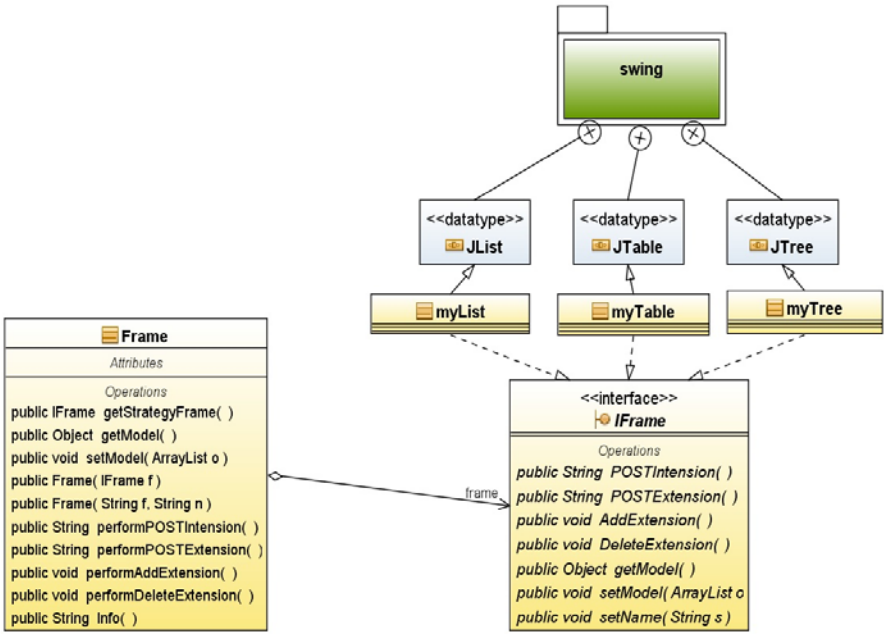


Figure 2: UML-diagram, frames library, classes description

- Functions - contains classes descriptions, this classes implement different system functions, such as data transferring and graphical style applying.
- Applet - contains classes descriptions, this classes implement Java-applet and XML-parser. XML-parser is used not only for handling messages, it is used like a storage of graphical constructions. When the content of this storage has changed, applet is changed too. This behavior is reached by using design pattern - «observer» [3].

The class javax.swing.SwingWorker<T, V> is used to execute time-consuming operations in the separate threads.

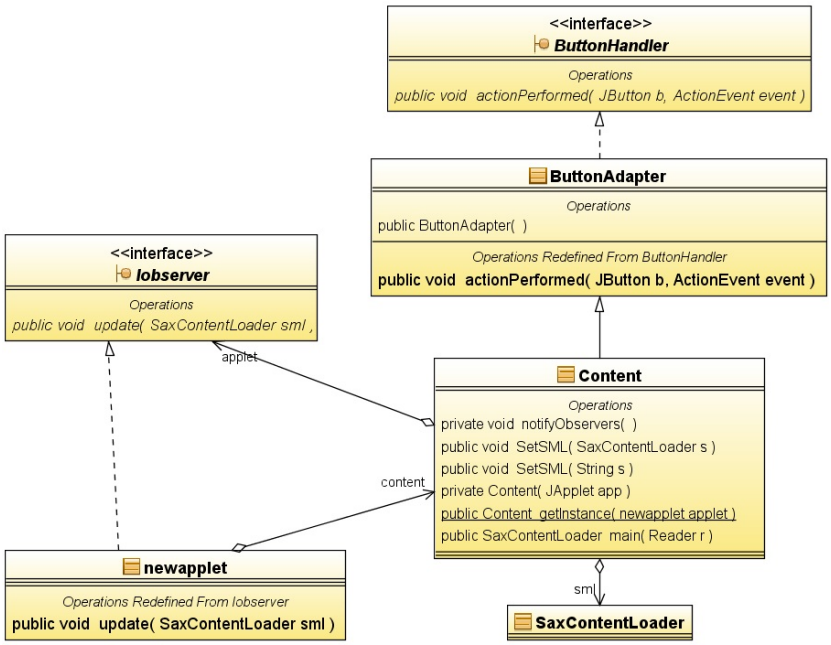


Figure 3: UML-diagramma, applet library, classes description

Conclusion

Оценивая ключевые перспективы и возможности получения прорывных технологических решений в области информационных технологий нужно учесть еще, как минимум, три направления развития РМД. Специального развернутого обсуждения и пристального внимания заслуживают реляционные методы проектирования информационных систем в целом и реляционных систем, в частности.

References

1. Вольфенгаген В.Э. *Методы и средства вычислений с объектами. Аппликативные вычислительные системы*. - М. : JurInfoR Ltd., АО "Центр ЮрИнфоР", 2004. – XVI+789 с.
2. Вольфенгаген В.Э. Рабочие материалы семинара ФОИТ / В. Э. Вольфенгаген. – 2010.
3. Фримен Э., ФрименЭ., Сьерра К., Бейтс Б. *Паттерны проектирования*. – СПб.: Питер, 2011. - 656 с.

4. Исмаилова Л.Ю., С.В. Косиков. *Ситуационно обусловленное оценивание для управления семантикой в обучающих системах по юриспруденции*. – Прикаспийский журнал: управление и высокие технологии №2(14) 2011. – с. 68-73.
5. Белнап Н., Стил Т. *Логика вопросов и ответов* / пер. с англ. - М.: изд-во «Прогресс», 1981. - 288 с.
6. Тарский А. *Введение в логику и методологию дедуктивных наук* / А. Тарский; пер. с англ. – М.: Гос. изд-во иностранной литературы, 1948. – 327 с.

Применение клеточных автоматов в мультиагентном моделировании транспортных систем

Барзиков К.В., Фомичева О.Е.

Кафедра АСУ, МГГУ,
Москва, 119991 РФ, Ленинский просп., д. 6
barzikov@gmail.com,
WWW: http://www.asu.msmu.ru

Аннотация. При рассмотрении вопросов безопасности и целесообразности внесения изменений в транспортные системы в условиях современного крупного города возникает, в числе прочих, задача изучения движения групп людей с учетом препятствий. Очевидно, такое изучение целесообразно проводить путем построения и исследования моделей. В рамках диссертационного проектирования аспирантом Барзиковым К. В. предложен метод моделирования транспортных систем с использованием клеточных автоматов.

Введение

Наиболее универсальными методами моделирования являются методы системной динамики, дискретно-событийное и многоагентное моделирование. В отличие от этих и других подходов к моделированию, предлагаемый в данном исследовании гибридный метод моделирования транспортных систем, пассажиропотоков, пересадочных узлов и систем распределенного обслуживания с использованием клеточных автоматов позволит с высокой степенью точности моделировать пассажиропотоки в транспортных системах, а также решать сходные задачи. Преимуществом данного метода является также наглядное графическое представление как исходных данных, так и результатов моделирования.

Ключевая проблематика

Математическое моделирование движения пассажиропотоков представляет собой проблему, родственную определенному классу задач газодинамики. Для решения некоторых задач газодинамики оказалось возможным применить модели класса клеточных автоматов. Поэтому интересно рассмотреть возможность построения моделей движения групп людей, перемещающихся целена-

правленно при наличии препятствий их движению, на основе клеточных автоматов. В [3] было показано, что дифференциальные уравнения в частных производных, возникающие при построении моделей такого движения, являются существенно нелинейными и не могут быть эффективно решены традиционными методами, например, при помощи разностных схем. В связи с этим в [3] предлагалась модель на основе клеточного автомата, позволявшая описать некоторые характерные ситуации, возникающие при таком движении.

Особенности применения клеточных автоматов

В предлагаемом методе мультиагентного моделирования отдельно взятый агент (пассажир или какой-либо сходный с ним объект), а также пассивные элементы транспортной системы (стены, препятствия, входы и выходы, зоны посадки-высадки с транспортных средств) представлены в виде состояний элементов клеточного автомата. В системе присутствуют источники агентов, которые генерируют в своей окрестности агентов с заранее заданными пользователем при построении модели параметрами. Агенты в процессе моделирования передвигаются от источников агентов к приемникам, где выводятся из системы или переносятся в другое модельное пространство (например, в другую область моделируемого транспортного объекта). На каждом шаге агент стремится занять ту клетку из своей окрестности, которой соответствует минимальный потенциал приводного скалярного поля. Приводные скалярные поля формируются до начала процесса моделирования приводным клеточным автоматом. При этом агент не может переместиться в клетку, соответствующую препятствию или занятую на данном шаге другим агентом. Для обеспечения имитации работы различных элементов транспортной системы (движение поездов, задержка при проходе через пункты контроля), а также для накопления статистических данных предусмотрены вспомогательные алгоритмы. В процессе моделирования предлагаемый метод позволяет отследить такие факторы, как плотность и скорость пассажиропотока в транспортной системе, нагрузка на элементы транспортной системы, пропускная способность, а также обеспечивает непосредственное визуальное наблюдение за процессом моделирования при выводе клеточного поля на экран компьютера в виде растрового изображения.

Заключение

Авторами статьи предложен новый метод моделирования пассажиропотоков в транспортных системах. Данный метод обладает некоторой универсальностью, что позволит моделировать не только пассажиропотоки, но и сходные по своим свойствам с ними объекты, т.е. различные системы массового обслуживания. Применение клеточных автоматов в моделировании широко представлено в зарубежной научной литературе и является относительно новым и перспективным подходом. Дальнейшее развитие метода позволит построить систему моделирования транспортных систем с формализацией моделей на графическом языке моделирования.

Литература

1. Малинецкий Г.Г., М. Е. Степанцов, Применение клеточных автоматов для моделирования движения группы людей, Журнал вычислительной математики и математической физики, 44:11 (2004), 2094–2098
2. Степанцов М. Е. Математическая модель направленного движения группы людей, Матем. моделирование, 16:3 (2004), 43–49
3. Малинецкий Г.Г., Степанцов М.Е. Моделирование динамики движения толпы при помощи клеточных автоматов с окрестностью Марголуса // Известия Высших учебных заведений. Прикладная нелинейная динамика, 1997, том 5, №5, С. 75-79.
4. Степанцов М.Е. Моделирование движения группы людей на основе решеточного газа с нелокальными взаимодействиями. Известия высших учебных заведений. Прикладная нелинейная динамика, 1999. № 5. С. 44—46
5. Тоффоли Т., Марголус Н. Машины клеточных автоматов. М: Мир, 1991.
6. Agent-based modeling: Methods and techniques for simulating human systems. Eric Bonabeau. Proceedings of the National Academy of Sciences of the USA, 2002, vol. 99 No Suppl. 3, p.7280-7287, doi 10.1073

Формирование приводных скалярных полей при мультиагентном моделировании пассажиропотоков

Барзиков К.В., Фомичева О.Е.

Кафедра АСУ, МГГУ,
Москва, 119991 РФ, Ленинский просп., д.6
barzikov@gmail.com,
WWW: http://www.asu.msmu.ru

Аннотация. В рамках разработки методов мультиагентного моделирования пассажиропотоков с использованием клеточных автоматов перед авторами возникла необходимость формирования в виртуальном пространстве моделирования приводного скалярного поля, минимумы которого соответствуют той области, куда стремятся агенты, имитирующие движение пассажиров в транспортной системе. Данное исследование было проведено в рамках диссертационного проектирования и его результатом стала разработка специализированного двухмерного клеточного автомата, формирующего данное поле.

Введение

Разработка системы мультиагентного моделирования пассажиропотоков с использованием клеточных автоматов в транспортных системах, являясь конечной целью диссертационного проектирования авторов, предусматривает имитацию движения пассажиров в виде движения виртуальных агентов на поле клеточного автомата. Агенты порождаются с заранее заданной частотой источниками агентов, и, огибая виртуальные препятствия, имитирующие реальную конструкцию того или иного транспортного объекта (пересадочного узла, станции и пр.), достигают приемников агентов, после чего выводятся из системы. При этом системой моделирования накапливается статистика, на основании которой можно судить о загруженности объекта и оптимальности режима его работы. Один из вспомогательных алгоритмов данного подхода представлен в настоящей статье.

Ключевая проблематика

Для осуществления процессов моделирования транспортных систем на основе мультиагентного подхода необходимо в модельном пространстве, внутри которого движутся отдельные агенты-участники движения (пассажиры или транспортные средства) предусмотреть механизм, позволяющий каждому отдельному агенту находить наиболее оптимальный путь движения к тому объекту транспортной системы, которого ему необходимо достичь. При моделировании с использованием клеточных автоматов [2, 3] наиболее удобным представляется сформировать в модельном пространстве скалярное поле, силовые линии которого направлены от границ пространства до той или иной цели, расположенной в минимуме (или максимуме) данного поля. Также должна быть предусмотрена возможность формировать поле с учетом обхода препятствий, находящихся в модельном пространстве и соответствующих реальным препятствиям в моделируемой системе (стены, колонны, конструктивные элементы станций, переходов, транспортных узлов).

Принцип работы приводного клеточного автомата

Приводной клеточный автомат служит для создания двухмерного скалярного поля [4] в произвольной матрице P, при помощи которого агенты движутся к тому или иному приемнику наикратчайшим путем. Скалярное поле, задается в виде значений элементов некоторой матрицы P, так, что $P_{i,j} = f(i,j)$, где $f(i,j)$ – функция поля. При условии полного отсутствия препятствий и наличии точечного приемника агентов в начале координат функция поля близка к уравнению Евклидовой метрики для плоскости $f(i,j) \approx \sqrt{i^2 + j^2}$ и аппроксимируется клеточным автоматом. Матрица M содержит информацию о препятствиях Pr, которые должны огибаться силовыми линиями формируемого поля.

При формировании скалярного поля элементы приводной матрицы P могут быть рассмотрены, как элементы множества: $p \in P \equiv \{0,1,\dots,k,\dots n\}$, изменение состояний элементов происходит в дискретном времени в соответствии с правилом перехода $p_{i,j}(t+1) = \varphi\big(p_{k,l}(t)\big|p_{k,l}(t) \in N\big)$, где N – окрестность Мура, которая может быть записана, как $N_M^1(i,j) = \{p_{k,l}||i-k| \le 1, |j-l| \le 1\}$, φ – функция перехода, которую можно сформулировать следующим образом:

Если $p_{i,j} = \mu$, где μ – некоторое вспомогательное значение, соответствующее точке модельного пространства, подлежащей обработке на данном шаге, тогда $p_{i,j} = t\ \forall p_{i,j} \in P, p'_{i,j} = \mu\ \forall p'_{i,j} \in P'$, где P' –

вспомогательная матрица с размерностью той же, что и P. Затем Если $p'_{i,j} = \mu$, тогда $\forall p_{k,l} \in N_M^1(i,j)$ таких, что $m_{k,l} \neq Pr, m_{k,l} \in M\ p_{k,l} = \mu, p'_{i,j} = 0$. Вместо окрестности Мура допускается использовать также окрестность Фон Неймана.

Заключение

Рассмотренный в данной работе клеточный автомат успешно применяется в рамках разработки метода моделирования пассажиропотока в транспортных системах. Однако, стоит отметить, что применение клеточных автоматов в моделировании является достаточно перспективным методом. После некоторой доработки применение описанного выше клеточного автомата возможно также при решении задач моделирования электронно-оптических систем электровакуумных приборов и других сходных задач.

Литература

1. Тоффоли Т., Марголус Н. Машины клеточных автоматов. М: Мир, 1991.
2. Фомичева О.Е., Барзиков К.В. Анализ и оптимизация транспортных систем с использованием клеточных автоматов. Доклад. Первая Российская конференция "Системный анализ и семиотическое моделирование SASM-2011", секция No. 4, доклад No. 8 25 февраля 2011г.
3. Фомичева О.Е., Барзиков К.В. Анализ и оптимизация транспортных систем при помощи клеточных автоматов. Доклад. Неделя Горняка-2011, семинар No. 15
4. Гольдфайн И.А. Векторный анализ и теория поля. М.: Наука, 1968.

Математические модели автоматического реферирования текста

Цветков А.И.

Кафедра «Кибернетика», НИЯУ «МИФИ»,
115409, г. Москва, Каширское ш., д. 31
aitsvetkov@mephi.ru

Аннотация. Лавинообразный рост объема публикаций и информации, доступной из электронных источников, делает особенно актуальной задачу об автоматическом реферировании текста. За последние пять десятилетий исследователями было разработано множество подходов к решению этой задачи для различных парадигм и областей применения. В данной статье приводится обзор основных математических моделей, используемых для построения и оценки систем автореферирования.

Введение

Электронная информация играет все большую роль во всех сферах жизни современного общества. В последние годы объем доступной текстовой информации возрос настолько, что возникает угроза обесценивания этой информации в связи с трудностями поиска необходимых сведений среди множества доступных текстов. Развитие ресурсов интернета многократно усугубило проблему информационной перегрузки. В этой ситуации особенно актуальными становятся методы автоматизации реферирования текстовой информации, то есть методы получения сжатого представления текстовых документов – рефератов (аннотаций).

Постановка проблемы автоматического реферирования текста и соответственно попытки ее решения с использованием различных подходов предпринимались многими исследователями. В работе [26] приводится следующее определение реферата: текст, произведенный из одного или нескольких текстов, содержащий важную информацию из оригинала, длиной не более половины оригинала, и обычно значительно более короткий. История применения вычислительной техники для автореферирования насчитывает уже более 50 лет [5, 21]. За эти годы выработаны многочисленные подходы к решению данной проблемы, которые достаточно четко подразделяются на два направления:

- *квазиреферирование*, основанное на выделении из первичных документов с помощью определенных формальных признаков наиболее информативных фраз (фрагментов), совокупность которых образует некоторый экстракт;

• *краткое изложение*, основанное на выделении из текстов наиболее существенной информации в понимаемой машиной форме и порождении из нее новых текстов (рефератов), содержательно обобщающих первичные документы.

В данной статье вначале приводится обзор методов реферирования одного текста с акцентом на методы квазиреферирования. Затем приводится обзор методов реферирования наборов документов с акцентом на методы краткого изложения. В заключение описываются методики и стандарты оценки систем реферирования, от которых во многом зависят будущие результаты в данной области.

Автоматическое реферирование одного документа

В первых работах в области автоматического реферирования научных текстов были предложены следующие парадигмы для выделения наиболее информативных фрагментов текста:

• методы, основанные на использовании статистических параметров различных элементов текста (слов, предложений...) прежде всего по *частоте встречаемости* [21]; в результате ранжирования лексики в том или ином документе они определяют слова с высоким рангом и их сочетаемость в различных фразах и по этим показателям оценивают информативность данных фраз;

• *позиционные* методы, опирающиеся на предположение о том, что информативность предложения находится в зависимости от его позиции в тексте документа [5];

• *индикаторные* методы, основанные на функциональной идентификации фраз первичного документа с помощью индексации их специальными словами-маркерами, индикаторами и коннекторами, образующими лексический аппарат данного способа экстрагирования [10].

В работах [14, 27] описан метод, использующий меру TF-IDF (произведение множителей *term frequency* и *inverse document frequency*) для оценки информативности (весов) слов. В основе метода лежит гипотеза о том, что "вес некоторого слова пропорционален частоте появления этого слова в документе, и обратно пропорционален частоте употребления слова в других документах коллекции":

$$\mathrm{TF} = \frac{n_i}{\sum_k n_k}, \qquad \mathrm{IDF} = \log \frac{|D|}{|\{d_i \supset t_i\}|}, \qquad (1, 2)$$

где n_i – число вхождений слова в документ, в знаменателе TF – общее число слов в данном документе, $|D|$ – количество документов в корпусе, $|(d_i \supset t_i)|$ – количество документов, в которых встречается t_i (при $n_i \neq 0$).

К классическим методам реферирования с обучением относится метод, использующий *наивный байесовский классификатор* [16]. Пусть s – предложение из реферата S, а $F_1, \ldots, F_k$ – набор независимых признаков из [5, 10] с добавлением *длины предложения* и *наличия слов из прописных букв*. Для составления реферата осуществляется выбор n предложений с наибольшим значением ранга:

$$P(s \in \mathcal{S} \mid F_1, F_2, ..F_k) = \frac{\prod_{i=1}^{k} P(F_i \mid s \in \mathcal{S}) \cdot P(s \in \mathcal{S})}{\prod_{i=1}^{k} P(F_i)} \quad (3)$$

Также в различное время были опробованы методы, использующие для выделения наиболее значимых предложений *деревья решений* [17], *скрытые марковские модели* [8], *логлинейные модели* [25], *нейронные сети* [28].

Часть методов экстрагирования опирается на более сложные лингвистические модели. Один из таких методов описан в работе [3]. В ней вводится понятие *лексической цепочки* – последовательности лексически связанных слов в тексте, охватывающей короткие (для соседних слов) или длинные (между предложениями текста) расстояния. Авторы пользуются определением *лексической связности* – множества способов ссылки элементов текста на встреченные в тексте ранее (повторение, синонимия, совстречаемость) [13]. Описанный подход состоит в разбиении текста на элементы, нахождении лексических цепочек, и выборе сильных лексических цепочек для выделения наиболее значимых предложений.

В работе [22] излагается метод квазиреферирования, использующий в своей основе теорию риторической структуры текста. Согласно данной теории, любой текст может быть представлен в виде графа, узлами которого являются *дискурсивные единицы*. Узлы графа связаны *функциональными отношениями*, указывающими на то, что каждая единица дискурса существует не сама по себе, а добавляется автором к некоторой другой для достижения определенной цели. Функциональные отношения, как правило, являются асимметричными: более значимый их компонент называется *ядром*, менее значимый – *сателлитом*. После построения риторической структуры текста, реферат может быть составлен путем обхода значимых компонент графа. На рис. 1 представлено дерево дискурса, задающее порядок выбора предложений: 2 > 8 > 3, 10 > 1, 4, 5, 7, 9 > 6.

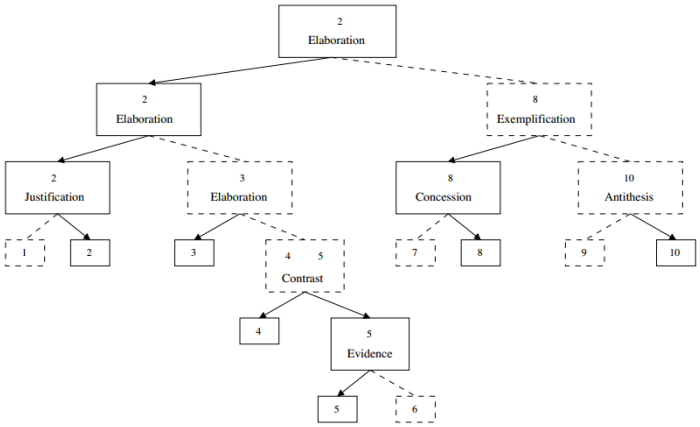


Рис. 1. Дерево дискурса из работы [22]. Вершины пронумерованы позициями предложений в исходном тексте. Ядерные компоненты имеют сплошную обводку, сателлиты – пунктирную.

Теория РСТ также используется для решения задачи автореферирования в отечественной работе [2].

Автоматическое реферирование набора документов

Начиная со второй половины 1990-х гг. получили распространение методы реферирования набора документов, результатом применения которых является обзорный реферат, содержащий основные положения из всех документов набора. Одной из первых систем подобного рода является система SUMMONS, описанная в работе [23]. Данная система состоит из двух основных компонент: *планировщика содержания*, который выделяет информацию для включения в реферат из заранее подготовленной базы данных на основе шаблонов, и *лингвистического генератора*, преобразующего выделенную информацию в грамматически верный и связный текст. Планировщик содержания, используемый в системе, реализован на основе набора эвристических правил.

Лингвистический генератор FUF/SURGE [12], использованный авторами, реализован на основе *функциональной унификационной грамматики* [15]. Данный формализм предполагает равноправное представление структур составляющих и правил анализа как ограничений. С содержательной точки зрения, правила ФУГ описывают соответствие между поверхностным синтаксическим представлением фразы и структурой семантических отношений. Например, на рис. 2 изображено правило анализа предложения со сказуемым в активном залоге.

$$\begin{bmatrix} \text{Category} = \text{S} \\ \text{Patterns} = (\ldots\ \text{Predicator Direct - Object} \ldots) \\ \text{Subject} = \text{Actor} = \left[\text{Category} = \text{NP}\right] \\ \text{Direct - Object} = \text{Affected} = \left[\text{Category} = \text{NP}\right] \\ \text{Predicator} = \begin{bmatrix} \text{Category} = \text{Verb} \\ \text{Transitivity} = \text{Material - Process} \end{bmatrix} \\ \text{Voice} = \text{Active} \end{bmatrix}$$

Рис. 2. Правило анализа предложения со сказуемым в активном залоге.

FUF представляет собой реализацию ФУГ, расширенную *типизацией* [11].

Для применения разработанных в [23] методов в более широком спектре областей, когда ручное составление шаблонов извлечения информации неприемлемо, авторы работ [4, 24] предложили следующий подход: вначале в исходном тексте выделяется множество *тем* (обычно соответствующих параграфам текста) путем решения задачи *кластеризации* элементов текста по статистическим, позиционным и лексическим признакам. Затем система переходит к этапу *синтеза информации*, на котором предложения внутри каждой темы разбираются статистическим анализатором [7], производится слияние предикатно-аргументных структур и определение функциональных ролей. Наконец, полученное дерево зависимостей элементов текста обходится в поисках *глагольных фраз*, которые передаются в лингвистический генератор.

В работе [6] представлен метод *максимальной граничной значимости*, позволяющей объединить в одной метрике *релевантность запросу* и меру *новизны информации*. Пусть Q – запрос или профиль пользователя, R – ранжированный список элементов текста, полученный из поисковой системы. Рассмотрим инкрементальную процедуру, на каждом шаге добавляющую элемент текста к

множеству S. Для каждого элемента текста $D_i \in R \backslash S$ из множества еще не рассмотренных элементов вычислим граничную значимость:

$$\mathrm{MR}(D_i) := \lambda \mathrm{Sim}_1(D_i, Q) - (1-\lambda) \max_{D_j \in S} \mathrm{Sim}_2(D_i, D_j), \quad (4)$$

где $\lambda \in [0, 1]$ – параметр, определяющий значимость релевантности относительно избыточности, а $\mathrm{Sim}_1(x,y) = \mathrm{Sim}_2(x,y) = \frac{\langle x,y \rangle}{\|x\| \cdot \|y\|}$. Одним из привлекательных свойств метода является построение рефератов, специфичных для заданной темы.

Методики оценки систем реферирования

Задача оценки качества получаемых рефератов представляет сложность по той причине, что не существует идеального реферата для заданного текста или группы текстов. Мнения экспертов по поводу критериев оценки значительно расходятся, а в работах по автоматическому реферированию используется широкий спектр трудно сопоставимых методик оценки результатов, что еще больше осложняет задачу [9]. Ниже приведен обзор методов, задающих стандарт в данной области.

В работе [18] приводится сравнение нескольких методов ручной и автоматической оценки рефератов. Также рассматривается процедура оценки рефератов, использованная на конференции Document Understanding Conference 2001. Процедура заключалась в ручной оценке информационного покрытия эталонных рефератов результатами автоматического реферирования по следующей шкале: полное покрытие (4), значительное покрытие (3), некоторое покрытие (2) и слабое покрытие (1). Также экспертами оценивались грамматическая правильность и связность текстов. Затем рассчитывалась *взвешенная полнота* (*weighted recall*) для порогового значения $t = 1,\ldots,4$:

$$\frac{\text{Число элементов текста, оцененных } t \text{ или выше}}{\text{Общее число элементов текста в реферате}} \quad (5)$$

Полученные от разных экспертов значения оценки информационного покрытия для одних и тех же рефератов различались на 18% для одного документа и на 7,6% для набора документов, что говорит о низкой степени согласованности оценок. В задаче квазиреферирования оценки различались на 40% для одного документа и 29% для набора документов. Для преодоления этой проблемы были предложены методы автоматического оценивания рефератов.

Один из таких методов опирается на расчет суммарного совпадения N-грамм:

$$\mathrm{NAMS} = a_1 \cdot \mathrm{NAM}_1 + a_2 \cdot \mathrm{NAM}_2 + a_3 \cdot \mathrm{NAM}_3 + a_4 \cdot \mathrm{NAM}_4, \quad (6)$$

где NAM_n – коэффициент совпадения N-грамм (последовательностей из N слов):

$$\frac{\text{Число N-грамм, совпадающих между данным рефератом и эталоном}}{\text{Общее число N-грамм в реферате}} \quad (7)$$

Наибольшее соответствие экспертным оценкам получила модель с $a_2 = 2/3$ и $a_1 = 1/3$.

В работе [18] предложен набор метрик для автоматической оценки рефератов ROUGE (Recall-Oriented Understudy for Gisting Evaluation – оценка смысловой

полноты). Пусть $R = \{r_1, \ldots, r_m\}$ – множество эталонных рефератов, s – оцениваемый реферат, а $\Phi_n(d)$ – бинарный вектор N-грамм, содержащихся в документе d. Тогда метрика ROUGE вычисляется по следующей формуле:

$$\text{ROUGE-N}(s) = \frac{\sum_{r \in R} \langle \Phi_n(r), \Phi_n(s) \rangle}{\sum_{r \in R} \langle \Phi_n(r), \Phi_n(r) \rangle}, \tag{8}$$

где $\langle ., . \rangle$ – операция скалярного произведения векторов.

Еще одна ROUGE-метрика принимает во внимание *длиннейшую совпадающую последовательность* слов в предложениях.

$$\text{ROUGE-L}(s) = \frac{(1+\beta^2) R_{\text{LCS}} P_{\text{LCS}}}{R_{\text{LCS}} + \beta^2 P_{\text{LCS}}},$$
$$R_{\text{LCS}}(s) = \frac{\sum_{i=1}^{u} LCS(r_i, s)}{\sum_{i=1}^{u} |r_i|}, \; P_{\text{LCS}}(s) = \frac{\sum_{i=1}^{u} LCS(r_i, s)}{|s|}, \tag{9, 10, 11}$$

где $|x|$ – длина предложения x, $LCS(x, y)$ – максимальная длина совпадающей последовательности из слов предложений x и y, а β – параметр, определяющий точность и полноту соответствия.

Существует адаптация метрики ROUGE для русского языка, предложенная в работе [1].

В работе [20] предлагается теоретико-информационный подход к автоматической оценке рефератов. Основной идеей этого метода является расчет меры расхождения двух вероятностных распределений, конкретно – *расстояния Дженсена-Шеннона* между распределением, полученным из оцениваемого реферата, и распределением, полученным из множества эталонов. Пусть распределение с параметром θ_R генерирует эталонные рефераты документов из множества $D = \{d_1, \ldots, d_n\}$. По аналогии, оцениваемая система производит рефераты по распределению с параметром θ_A. Хорошей системой реферирования будет та, для которой значение θ_A близко к значению θ_R. Одна из подходящих для оценки этого расстояния теоретико-информационных метрик – *KL*-метрика:

$$KL(p^{\theta_A} || p^{\theta_R}) = \sum_{i=1}^{m} p_i^{\theta_A} \log \frac{p_i^{\theta_A}}{p_i^{\theta_R}}. \tag{12}$$

Однако, *KL*-метрика не ограничена и несимметрична, поэтому авторы предлагают использовать метрику Дженсена-Шеннона:

$$\begin{aligned} JS(p^{\theta_A} || p^{\theta_R}) &= \frac{1}{2} KL(p^{\theta_A} || r) + \frac{1}{2} KL(p^{\theta_R} || r) = \\ &= H(r) - \frac{1}{2} H(p^{\theta_A}) - \frac{1}{2} H(p^{\theta_A}), \end{aligned} \tag{13}$$

где $r = \frac{1}{2} p^{\theta_A} + \frac{1}{2} p^{\theta_R}$ – среднее распределение.

Заключение

Рост объема информации, доступной из электронных источников, создает потребность в эффективных и точных системах автоматического реферирования. Несмотря на то, что исследования в этой области ведутся уже более пятидесяти лет, проблема составления рефератов становится все более актуальной. Исследователями были испробованы подходы, основанные как на выделении наибо-

лее информативных фрагментов текста, так и на генерации краткого изложения реферируемых документов.

В данной работе рассматриваются математические модели, используемые для построения систем автоматического реферирования. Приводится обзор методов реферирования одного текста с акцентом на методы квазиреферирования. Затем приводится обзор методов реферирования наборов документов с акцентом на методы краткого изложения.

В заключение описываются методики и стандарты оценки систем реферирования. Сложность задачи составления эталонных рефератов усугубляется значительными расхождениями в экспертных оценках качества рефератов. В такой ситуации особые надежды возлагаются на автоматические методики оценки рефератов и метрики оценки, принятые научным сообществом в данной области.

Литература

1. Тарасов С.Д. Исследование и оптимизация параметров алгоритма Manifold Ranking на основе метрики автоматической оценки качества обзорного реферирования ROUGE-RUS. // *RCDL'2009*. – 2009. – С. 86–93.
2. Тревогда С.А. *Методы и алгоритмы автоматического реферирования текста на основе анализа функциональных отношений: диссертация кандидата технических наук: 05.13.01*. – Санкт-Петербург, 2009. – 157 с.
3. Barzilay R., Elhadad M. Using lexical chains for text summarization // *Proceedings ISTS'97*. – 1997.
4. Barzilay R., McKeown K., and Elhadad M. Information fusion in the context of multi-document summarization // *Proceedings of ACL '99*. – 1999.
5. Baxendale P. Machine-made index for technical literature – an experiment // *IBM Journal of Research Development*. – 1958. – № 2 (4). – С. 354–361.
6. Carbonell J., Goldstein J. The use of MMR, diversity-based reranking for reordering documents and producing summaries. // *Proceedings of SIGIR '98*. – 1998. – С. 335–336.
7. Collins M. *Head-Driven Statistical Models for Natural Language Parsing. PhD thesis*. – University of Pennsylvania, 1999. – 276 с.
8. Conroy J. M., O'leary D. P. Text summarization via hidden markov models // *Proceedings of SIGIR '01*. – 2001. – С. 406–407.
9. Das D., Martins A.F.T. *A Survey on Automatic Text Summarization*. – Pittsburgh: Carnegie Mellon University, 2007. – 31 с.
10. Edmundson H. P. New methods in automatic extracting // *Journal of the ACM*. – 1969. – № 16 (2). – С. 264–285.
11. Elhadad M. Types in functional unification grammars // *Proceedings of the 28th Annual Meeting of the Association for Computational Linguistics*. – 1990. – С. 157–164.
12. Elhadad M., Robin J. *An Overview of SURGE: A reusable comprehensive syntactic realization component*. – Beer Sheva: Ben Gurion University, 1996. – 10 с.
13. Halliday M.A.K., Hasan R. *Cohesion In English*. – London: Longman, 1976.

14. Jones K.S. A statistical interpretation of term specificity and its application in retrieval // *Journal of Documentation*. – 1972. – № 28 (1). – С. 11–21.
15. Kay M. Functional grammar // *Proceedings of the 5th Annual Meeting of the Berkeley Linguistic Society*. – 1979. – С. 142–158.
16. Kupiec J., Pedersen J., Chen F. A trainable document summarizer // *Proceedings SIGIR '95*. – 1995. – С. 68–73.
17. Lin C.–Y. Training a selection function for extraction // *Proceedings of CIKM '99*. – 1999. – С. 55–62.
18. Lin C.-Y., Hovy E. Manual and automatic evaluation of summaries // *Proceedings of the ACL-02 Workshop on Automatic Summarization*. – 2002. – С. 45–51.
19. Lin C.-Y. Rouge: A package for automatic evaluation of summaries // *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*. – 2004. – С. 74–81.
20. Lin C.-Y., Cao G., Gao J., and Nie J.-Y. An information-theoretic approach to automatic evaluation of summaries // *Proceedings of HLT-NAACL '06*. – 2006. – С. 463–470.
21. Luhn H. The automatic creation of literature abstracts // *IBM Journal of Research and Development*. – 1958. – № 2 (2). – С. 159–165.
22. Marcu D. Improving summarization through rhetorical parsing tuning // *Proceedings of The Sixth Workshop on Very Large Corpora*, Montreal, Canada. – 1998. – С. 206–215.
23. McKeown K. R., Radev D. R. Generating summaries of multiple news articles // *Proceedings of SIGIR '95*. – 1995. – С. 74–82.
24. McKeown K., Klavans J., Hatzivassiloglou V., Barzilay R., and Eskin E. Towards multidocument summarization by reformulation: Progress and prospects // *AAAI/IAAI*. – 1999. – С. 453–460.
25. Osborne M. Using maximum entropy for sentence extraction // *Proceedings of the ACL'02 Workshop on Automatic Summarization*. – С. 1–8.
26. Radev D. R., Hovy E., McKeown K. (2002). Introduction to the special issue on summarization // *Computational Linguistics*. – № 28(4). – С. 399–408.
27. Salton G., Buckley C. Term-weighting approaches in automatic text retrieval // *Information Processing and Management*. – 1988. – № 24 (5). – С. 513–523.
28. Svore K., Vanderwende L., Burges C. Enhancing single-document summarization by combining RankNet and third-party sources // *Proceedings of the EMNLP–CoNLL*. – 2007. – С. 448–457.

Разработка интеграционной шины для среды распределения вычислительных процессов

Рословцев В.В. Шумский Л.Д.

Кафедра №22 «Кибернетика», НИЯУ МИФИ,
Москва, 115409, Каширское ш., 31

vladiros@gmail.com shumskyld@gmail.com

Аннотация. В данной работе представляется подход к созданию интеграционной шины для распределенных вычислительных сетей, которая может быть использована для создания масштабируемых, отказоустойчивых сетей, способных решать широкий класс задач. Для разработанной интеграционной шины описывается алгоритм ее работы, ее основные технические и концептуальные характеристики. Отдельно описывается возможное расширение распределенной вычислительной сети с помощью внешней интеграционной шины, которая может служить для выполнения специфического анализа решаемых задач, а также упрощения осуществления контроля и управления несколькими сетями.

Введение

В настоящее время, когда объем данных и, особенно, метаданных экспоненциально растет, требования к системам, оперирующих ими все более усложняются, а также существенно усложняются информационные процессы, как никогда остро встает вопрос об эффективности используемых информационных систем. Для создания систем, наилучшим образом отвечающих этим требованиям, используется подход к построению систем, использующих распределенные вычисления. Современные системы распределенного типа, как правило, имеют сервис-ориентированный тип. Системы данного типа составлены из набора сервисов, каждый из которых предназначен для решения одной или нескольких задач, для решения которых используется вычислительная сеть. Для взаимодействия этих сервисов обеспечивается самыми разнообразными средствами, начиная от крайне высокоуровневых, таких как передача данных по протоколу SOAP, взаимодействие с использованием некоторой локальной шины, такой как D-Bus [1] или CORBA [2], и, наконец, наиболее производительное взаимодействие – с использованием общей, физической, памяти.

Современные системы распределенных вычислений имеют два существенных недостатка: во-первых, обмен данными между сервисами, как правило, крайне усложнен и не выделен в отдельный логический блок. Это существенно усложняет масштабирование таких систем и, особенно, масштабирование по новым вычислительным алгоритмам и новым способам обмен данными. Во-

вторых, вопрос разбивки задачи на подзадачи, для обсчета их с помощью сервисов каждый раз решается в отдельном порядке, без достаточной формализации, что усложняет повторное использование данных сервисов и применения их для новых, но аналогичных задач.

Для решения данных проблем предлагается

1. Использование общего концептуального подхода к решению задачи декомпозиции данных и задач;
2. Использование разработанной интеграционной шины, которая позволяет существенно упростить масштабирование распределенной вычислительной сети по вычислительным мощностям, используемым алгоритмам и способам передачи данных.

Описываемая шина должна решать два класса задач интеграции.

Первая группа задач – интеграция вычислительных узлов сети между собой. В эту группу входят задачи формирования пакетов данных оптимальным образом и контроль их целостности, определение маршрутов передачи пакетов данных между узлами, обеспечение отказоустойчивости сети, разрешение конфликтов доступа, т.е. ситуаций, когда необходимо получить данные с узла не находящегося в открытом контуре. Приоритетными метриками для оценки качества решения задач из этой группы являются скорость выполнения задач шиной, количество ошибок при передаче данных и количество устраненных ошибок.

Вторая группа задач – интеграция различных вычислительных сетей. В эту группу входят задачи, связанные с анализом вычислительных процессов в сети и анализом данных и архитектуры.

Дальнейшее изложение построено по следующему плану: первый раздел содержит некоторый теоретический базис, описывающий разработанные средства и правила их функционирования. Второй раздел содержит подробное описание работы интеграционной шины в вычислительной сети. Третий раздел содержит описание возможной надстройки, т.н. внесетевой интеграционной шины, которая может быть использована для увеличения эффективности работы сети и упрощения решения части технических задач.

Теоретические основы интеграции в вычислительных сетях и декомпозиции вычислений

Наиболее подходящей формальной моделью для описания декомпозиции вычислительных процессов являются аппликативные вычислительные системы. Например, на основе типизированного λ-исчисления был разработан фундаментальный язык, позволяющий описывать как распределенный код, так и распределенные ресурсы [3]. В таких системах объекты представляют собой функциональные сущности, основной способ построения новых объектов – операция аппликации одного объекта к другому, причем объекты способны, в зависимости от контекста, выступать как в роли функции, так и в роли аргумента.

Декомпозиция вычислительных процессов с помощью некоторой аппликативной вычислительной системы, например, с помощью λ-выражений может

быть построена следующим образом. В качестве первого шага необходимо определить набор термов, выражающих то или иное базовое вычислительное действие. Набор этих термов определит базис некоторого подмножества λ-выражений, термы, лежащие в этом подмножестве мы будем называть вычислительными термами. Каждый вычислительный терм должен быть апплицирован к некоторому терму, выражающему набор данных. Термы, выражающие наборы данных, также могут быть разложены в некотором базисе. Задача декомпозиции вычислительного процесса, в таком случае, сводится к разложению терма, соответствующего процесса и нахождение соответствующего разложения терма, соответствующего блоку данных. Формально, данное преобразование может быть записано следующим образом:

$$FD = \langle f_1 \circ \ldots \circ f_n \rangle \mathcal{D}D$$

Где $\mathcal{D}D = \langle d_1 \circ \ldots \circ d_n \rangle$ – соответствующая декомпозиция данных

В основе любой интеграционной шины лежит интеграционный процесс. С формальной точки зрения, интеграционный процесс на вычислительной шине распределенной вычислительной сети – это рекурсивный процесс в реальном времени, оперирующий только информацией, как элементами контекста, без ограничений по времени на начало выполнения действий. Этот процесс может быть представлен, как следующая композиция действий:

$$\mathcal{P} = < \mathcal{I} \circ \mathcal{A}_1 \circ \ldots \circ \mathcal{A}_n \circ \mathcal{O} >$$

Интеграционный процесс отличается от обычного информационного процесса тем, что, во-первых, контекст выполнения данного процесса заполняется только после выполнения первого действия (не существует предварительно объявленных параметров) и во время выполнения процесса в контексте не может появляться новых объектов, а только различные представления уже существовавших. Во-вторых – интеграционный процесс должен состоять из последовательности действий, первым действием которой является действие типа $\mathcal{I}$ – получение данных из внешнего источника в контекст процесса и завершаться действие типа $\mathcal{O}$ – передачей данных во внешнюю систему.

Интеграционный процесс, выполняемый на вычислительной шине, отличается от обычного интеграционного процесса тем, что обязательно имеет рекурсивную природу [4]. В терминах алгебры процессов данный процесс может быть записан как

$$D^*E \vee C = D(E^*D) \vee E \vee C$$

В данной записи: D – действие декомпозиции данных, E – действие вычисления блока данных, C – действие композиции блока данных, * означает рекурсивное замыкание процесса Клини, ∨ означает последовательное выполнение действий. Данный процесс может быть описан как рекурсивное разбиение данных, согласно некоторому решающему алгоритму, результатом которого может быть новая декомпозиция, или непосредственное вычисление блока данных. После каждого вычисления, собранные результаты должны подвергаться композиции. Диаграмма такого процесса представлена на рисунке 1.

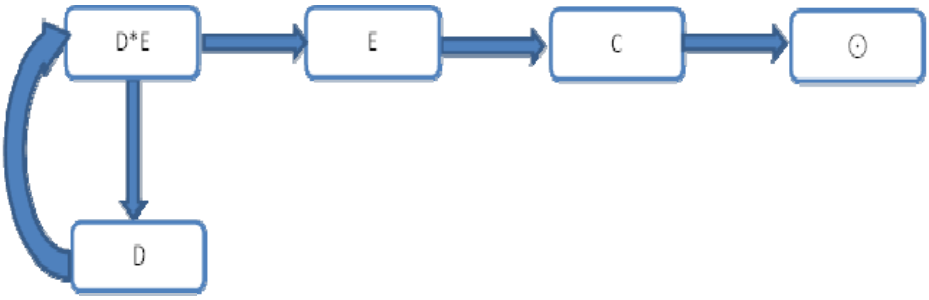


Рисунок 1 Диаграмма процесса

Подробное описание интеграционного процесса, выполняемого на шине, приводится в разделе 2.

Описание внутрисетевой интеграционной шины

Интеграционная шина в распределенной вычислительной сети (РВС) это узел, обеспечивающий передачу данных и обеспечивающий необходимую прозрачность для вычислительных узлов [5]. Информация на шине представляется на двух уровнях – на хранимом уровне, в виде сообщений с присвоенными идентификаторами и заголовками и на транспортном уровне, для пересылки данных между узлами. Шина должна обладать доступом к производительному хранилищу данных для того, чтобы хранить текущие выполняемые задачи, а также результаты выполненных задач для обеспечения адаптивного характера работы. Доступ к данному хранилищу должен быть предоставлен внесетевым интеграционным шинам. Т.к. основной метрикой работы транспортного узла является время, которое затрачивается на выполнение действий интеграционного процесса, то информация не представляется на концептуальном уровне, а на хранимом представляется в максимально простом виде только для задач статистики и обеспечения отказоустойчивости сети.

Особенностью внутрисетевой шины является жесткая регламентация всех выполняемых операций и формата передаваемых данных. Т.к. для данного узла необходимо сократить объем передаваемых данных, то передается бинарный массив данных, фиксированной длинны, содержащий необходимый набор идентификаторов типа GUID или URI, в соответствии с [6]. Первый элемент массива всегда ID пакета, второй – ключ типа пакета, а третий ID текущей задачи. Остальные идентификаторы определяются типом пакета. Все используемые справочники, такие как тип пакета, тип решаемой задачи и т.д., а так же идентификаторы узлов в пределах сети должны быть одинаковы на всех шинах, которые находятся в одном контуре. Обеспечение соответствия справочников между сетями может осуществляться на внесетевых интеграционных шинах.

Обмен данными между шиной и прочими узлами должен осуществляться с помощью адаптеров. Адаптер, используемый для обмена данными с узлом, выбирается такой, чтобы поддерживать максимальную скорость передачи и отказоустойчивость обмена данными. В порядке убывания должны использо-

ваться следующие адаптеры: общая память, межпроцессорное взаимодействие, взаимодействие по локальной сети, HTTP адаптер, прочие виды адаптеров. Все передачи данных должны осуществляться в асинхронном режиме. Для каждого передаваемого типа сообщений и для каждого адаптера должно быть определено максимальное время ожидания ответа. Если узел не отвечает за указанное время, то связь с ним считается нарушенной, и он отключается от сети до устранения ошибки. Включен обратно в сеть, он может быть только в ручном режиме.

Шина должна поддерживать выполнение процесса, состоящего из следующих действий.

- **Получение задачи** – к внутрисетевой шине могут быть подключены клиентские модули, т.е. модули, которые могут пересылать в вычислительную сеть сформированные задачи. Задачей, с точки зрения интеграции, является набор из ID источника запроса, тип решаемой задачи, ID ресурсных узлов в сети и ID ресурсов на этих узлах, идентификатор ресурса имеет тип URI. С его помощью, на вычислительных узлах, могут быть сформированы идентификаторы также типа URI на некоторые логически выделенные элементы ресурса. Задачи на интеграционную шину могут также отправлять вычислительные узлы, после выполнения декомпозиции. Получение задачи является начальным действием процесса вычисления. Получив задачу, шина переводит ее в хранимый вид, чтобы контролировать, в дальнейшем, процесс вычисления. Для каждой задачи должны быть сохранены основные ее параметры и параметры решения, а именно – характеристики данных, полученные с ресурсного узла, вычислительный узел, произведший декомпозицию задачи, результат декомпозиции и время решения задачи;
- **Анализ полученной задачи** – задачи, зарегистрированные на шине, должны быть подвергнуты анализу для того, чтобы обеспечить большую устойчивость системы и обеспечить более правильный подбор алгоритма решения. Анализ производится адаптивно. В расчет принимаются – характеристики данных, получаемые от ресурсного узла, узлы, которые способны решать данную задачу каким-либо алгоритмом и предыдущие запуски такой задачи с данными, а также данные получаемые из внесетевых адаптивных шин. Задача анализа, во-первых, вычислить такую группу узлов, что результат адаптивной функции будет оптимальным, а, во-вторых, получить оценку времени выполнения задачи. Шина обменивается сообщениями со всеми ресурсными узлами, указанными в сообщении клиента и со всеми вычислительными узлами. При передаче сообщения на вычислительный узел к пакету добавляются идентификаторы всех опрошенных интеграционных шин, включая собственный идентификатор. Это необходимо для предотвращения возникновений зацикливания. Для анализа используется справочники алгоритмов решения и характеристик ресурса, хранящиеся на шине. Результатом анализа является идентификатор используемого алгоритма, группа свободных вычислительных узлов, поддерживающих выбранный алгоритм и наиболее производительный узел из группы, определенный на основании предыдущих запусков. Если в результате анализа выясняется, что в сети нет ресурсов для решения задачи, то производится оценка – сколько необходимо ожидать, источнику задачи передается соответствующее сообще-

ния, а задача ставится в очередь с отметкой о времени ожидания. Через указанное время производится повторный ее анализ;

- **Передача запроса на декомпозицию задачи** – выбранному, наиболее производительному узлу, передается запрос на декомпозицию. Пакет данных не содержит никаких дополнительных идентификаторов. Если выбранный узел оказывается недоступным, шина повторно выполняет анализ. Передав запрос, шина передает сообщение клиенту-источнику задачи, что обработка начата и сохранят у себя запись, содержащую ID задачи и ID клиента. После получения сообщения-результата с соответствующим идентификатором задачи шина переправляет его на указанный ID источника. На запрос декомпозиции вычислительный узел отвечает пакетом данных, содержащим ID задачи, код выполненной декомпозиции (декомпозиция на задачи, задания или отсутствие декомпозиции) и список идентификаторов подзадач/ заданий. Сами подзадачи/ задания передаются независимо;
- **Получение задания** – шина может получать задания только от вычислительных узлов. Задание отличается от задачи тем, что не требует определения решающего алгоритма. Пакет данных задания содержит следующие поля – ID задания, ID набора заданий, номер решающего алгоритма и идентификаторы ресурсного узла и ресурса на этом узле. Перед получением заданий шина должна получить запрос на открытие пула заданий, состоящий из ID вычислительного узла, который запрашивает пул, ID пула и количество заданий, которое в нем должно быть. После получения задания шина должна проверить есть ли открытый пул заданий с переданным идентификатором, если такой пул найден, то шина кладет задание в пул с отметкой «не обработано». Если задание с переданным ID уже есть в пуле, то шина игнорирует полученное задание. В фоновом режиме шина должна обрабатывать открытые пулы данных. Если находится пул с заданием в статусе «не обработано», то шина пытается найти свободный узел, поддерживающий алгоритм решения задания. Если узел находится, то шина меняет статус задания в пуле на «в обработке», делает отметку о начале обработки и ID выбранного узла, и передает задание вычислительному узлу. Если шина находит задание в статусе «в обработке», время обработки которого больше, чем допустимое время, с точки зрения решающего алгоритма, то шина передает решающему узлу сообщение об отмене обработки, отмечает этот узел как недоступный и меняет статус задания на «не обработано». Если шина находит пул заданий, содержащий только задания в статусе «обработано», и количество этих заданий равно ожидаемому количеству для этого пула, то она закрывает пул и отправляет результаты на соответствующий вычислительный узел;
- **Получение запроса на предоставление данных** – т.к. вычислительный узел может не иметь доступа к ресурсному узлу, на котором находится ресурс, необходимый для вычислений, то шина должна обеспечивать такой доступ. Для этого должен быть выделен ресурсный узел, к которому имеют доступ все вычислительные узлы, на который, по запросу, используя отдельно выделенный канал, шина может перекладывать ресурсы с других ресурсных узлов. После использования этого промежуточного ресурса он должен быть удален. Карта доступов и полномочий ресурсных и вычислительных узлов должна также храниться на шине и использоваться при анализе задач;

- **Получение результата выполнения задания** – после получения результата выполнения задания шина находит это задание, меняет его статус на «обработано». Кроме того она получает характеристики блока данных, связанных с заданием с ресурсного узла и сохраняет время выполнения узлом задания с полученными характеристиками. Эта информация будет использоваться для анализа производительности узла и эффективности алгоритма в зависимости от характеристик данных.

Преимуществами использования интеграционной шины, обладающей описанными характеристиками, в распределенных вычислительных сетях являются: Поддержка различных алгоритмов решения задач одного типа, легкая масштабируемость сети, высокая эффективность используемых вычислительных узлов и методов решения, отказоустойчивость сети, разрешение вопросов доступов и полномочий.

Описание внесетевой интеграционной шины

Внесетевая интеграционная шина – это узел, который может быть подключен к одной или нескольким распределенным вычислительным сетям для увеличения качества их работы путем сбора дополнительной статистики и осуществления дополнительного контроля. Этот узел не является собственно частью сети и выполняет лишь вспомогательные функции. Использование такой шины необходимо, например, если части одного и того же вычислительного процесса должны выполняться в нескольких независимых сетях, между которыми должен передаваться только финальный результат [7].

Задачей этого узла является осуществление концептуальной оценки решаемых задач на основании статистики, полученной из хранилищ данных сетевых интеграционных шин. Каждая задач представляется в виде индивидного объекта некоторой концептуальной модели, определенной на данной шине. С помощью логического вывода для определенной концептуальной модели внесетевая интеграционная шина может определять скрытые свойства некоторых задач и передавать эти данные на сетевые интеграционные шины. С помощью такого анализа могут быть определены существенные и несущественные для каждой из задач характеристики ресурсов, эффективность каждого из алгоритмов, вычислительная мощность узлов, загруженность каждого из алгоритмов, загруженность вычислительной сети в целом и прочая важная статистическая информация. Результаты работы внесетевой шины могут быть транслированы на сетевые шины в ручном или автоматическом режиме. К одной вычислительной сети может быть подключено несколько внесетевых шин, использующих различные концептуальные модели для получения разносторонних статистических материалов. С помощью этого метода могут, также, быть получены оценки эффективности использования различных статистических алгоритмов и концептуальных моделей.

На основании построенной статистики внесетевая шина может оптимизировать деревья связей внутри сети таким образом, чтобы загруженность обмена сообщениями между различными ее участками была минимальна. Эта особен-

ность внесетевых шин может быть использована для подключения новых узлов в сеть. Для этого узел прописывается на внесетевой шине, она проверяет его вычислительную мощность, используя тестовые данные из хранилищ шин, и подключает его в сеть таким образом, чтобы усилить некоторый вычислительный контур, упростить решение часто встречающихся задач и упростить транспорт данных. Аналогично, внесетевая шина может подключать клиенты к сетевым шинам таким образом, чтобы наиболее часто встречаемые на клиенте задачи попадали в контур, который может решать их наилучшим образом с минимальными транспортными расходами.

Следующей решаемой на данной шине задачей является приведение в соответствие всех справочников и таблиц адресов и реестра алгоритмов каждой из подключенных сетей. Это позволяет проще объединять различные сети в одну. Достаточно только соединить их вначале через общую внесетевую шину, а после того как внесетевая шина приведет их соответствие соединить непосредственно. Таким образом, внесетевая шина может играть роль дополнительного элемента управления сетями, предоставляя единую точку управления всеми реестрами и справочниками.

Еще одной функцией контроля, которая может осуществляться на интеграционной шине данного типа является преобразование ресурсов. Различные ресурсные узлы могут предоставлять ресурсы одного типа в разных представлениях, предлагать различные интерфейсы и возвращать различные характеристики. Для решения этих проблем предлагается реализовывать на внесетевой шине полную концептуальную модель каждого типа ресурса и прописывать или аналитически выводить правила преобразования интерфейсов и характеристик одного представления ресурса к этому, полному, виду. Таким образом, для анализа интеграционная шина будет использовать не урезанное представление, которое может выдать ресурсный узел, а более полное и аналитически правильное. Это позволит внесетевой шине собирать и транслировать более полную статистику, чем та, которую может собирать сетевая шина.

Использование дополнительной, внесетевой в распределенных вычислительных сетях позволяет существенно увеличить качество использования сети, предоставляет более полную и наглядную статистику ее работы и добавляет дополнительный и централизованный узел управления вычислительными сетями. Позволяет проще выполнять настройку сетей, упрощает их масштабирование и развертывание.

Заключение

В данной работе приводится описание и алгоритм работы интеграционной шины для распределенной вычислительной сети, описываются базовые концепции интеграции с использованием такой шины, требования, которым она должна удовлетворять и технические характеристики передаваемых данных, ее особенности в части анализа и характера работы. Дано подробное описание интеграционного процесса, который должен выполняться с ее помощью.

Кроме того, в данной статье описывается возможное расширение распределенной вычислительной сети с помощью внешней интеграционной шины, которая может служить для выполнения специфического анализа решаемых задач, а также упрощения осуществления контроля и управления несколькими сетями.

В рамках дальнейших работ по данной задаче планируется более полная формализация композиции и декомпозиции вычислительных процессов, более полная формализация интеграционных процессов, анализ адаптивных статистических алгоритмов и прототипирование интеграционной шины и вычислительных компонентов и анализ работы

Литература

1. Robert Love Get on the d-bus // Linux Journal. - 2005. - №130.
2. Steve Vinoski CORBA: Integrating Diverse Applications Within Distributed Heterogeneous Environments // IEEE Communications Magazine. 1997. - С. 46 - 55.
3. Tom Murphy, Karl Crary, Robert Harper, Frank Pfenning A Symmetric Modal Lambda Calculus for Distributed Computing // - Pittsburgh, PA: Computer Science Department Carnegie Mellon University, 2004.
4. Jan A. Bergstra, Wan Fokkink, Alban Ponse Process Algebra with Recursive Operations. - Amsterdam, The Netherlands: Elsevier Science, 2001.
5. Andrew S.Tanenbaum, Maarten Van Steen Distributed Systems. Principles and Paradigms. - 2-е изд. - Amsterdam, Netherland: Pearson Prentice Hall, 2007.
6. Uniform Resource Identifier (URI): Generic Syntax // STD 66 (2005) URL: http://tools.ietf.org/html/rfc3986
7. Wilhelm Hasselbring Information System Integration // Communications Of The ACM. - 2000. - №6. - С. 33-38.

Temporal lambda calculus: application of typed lambda calculus for constructing interval temporal logic

Shumsky L.D.

Department of Cybernetics and Information Security,
National Research Nuclear University "MEPhI"
Moscow, 115409 RF, Kashirskoye Sh., 31
shumskyld@gmail.com

Abstract. This paper describes model of interval temporal logic based on typed lambda calculus. The main of aim of this paper is constructing formal typed lambda calculus based model which allows reasoning about dynamic properties of application domain. Temporal logic was chosen as start point for such model. Developed model provides tools for describing both static entities like objects, their properties and actions and basic temporal objects – states, events. Model also allows description of complex processes. Model allows limited reasoning about dynamic characteristics of processes with a help of causality rules.

Keywords: Typed Lambda Calculus, Temporal Logic, States, Events, Processes.

Introduction

This work describes interval temporal logic model based on typed lambda calculus. The importance of creation such model is justified with the fact that temporal logic is one of the most powerful tools which allows reasoning about plans, events and processes. Due to that fact temporal logic is one of the best solutions for describing dynamic application domain. In spite of importance of such representations modern temporal logics like described in (McDermont, 1982), (Lamport, 1994), (Venema, 2001), (James F. Allen, 1994), have range of crucial disadvantages, which makes their understanding, usage and implementations more complicated. The main reason of these disadvantages of that logic is enclosed in its definitions and axioms. It is the first order predicate logic in different variants of logic, so all its rules and theorems are in the forms of special predicates. Theses predicates should be connected with existing model of application domain. Final formulas, which should describe behavior of real objects, are occurred to be cumbrous and even redundant. Existing kinds of logics can't solve the classical "frame problem" and even stress it. The frame problem is the challenge of representing the effects of action without having to represent explicitly a large number of obvious non-effects (Dennett, 1987). More general, it is the problem of separating relevant features of current situation. Adding to the representation model of application domain temporal dimension raises question of determining actions or objects in every moment of time. There are a few already existing ways of

solving frame problem like (McCarthy, 1986), (McCarthy, 1986), however being implemented they complicate models and don't solve problem totally in temporal case. Cause of listed and some others disadvantages of systems, based on classical temporal logic, they are not wide spread in real practice.

Suggested way of construction temporal logic on the base of typed lambda calculus allows getting out of considerable part of mentioned problems. At first, lambda terms properties simplify adding temporal dimension to existing model of representation of application domain. Such extension will cost only a determination of some new constants and rules in the developed model. At second, extensions of β and η rules of term equivalence allow decreasing frame problem influence. Term equality in such model could be set by reducing them to one expression. Finally, developed model could be easily extended for supporting new kinds of actions or objects by adding corresponding constants.

Source models

Temporal logic model for formal representation of dynamic application domains is the first source model for our work. The typical concepts for some applied tasks are concepts of time continuation, time evaluations, and changes in time and so on. For constructing model, which allows obtaining reliable results, we should proceed from natural requirements to temporal objects. There are three kinds of temporal objects in natural languages (Jean-Pierre Desclés, 1993): – a state, an event and a process. A state is characterized in natural languages by the absence of changes, i.e. all observed parts of state are homogeneous. Saying some phrase describing state speaker means neither beginning nor ending of this state. Event is characterized by explicit changes in continuous background. Event is true in closed interval. It has explicit beginning and ending. The last of mentioned temporal objects is a process. Process is temporal situation, which is not homogeneous. Every process is characterized by its start state, set of its possible sates and rule of changing states. Such linguistic process is always true in half open interval, i.e. we know when it starts but we don't know exactly when it will be finished. These three kinds of temporal objects are called natural.

We will use interval temporal logic described in (James F. Allen, 1994)with extensions for describing causality relationships between states and events described in (McDermont, 1982) as initial point for developing our model. Formal model provided by these papers suits well for declared targets. At first, posed models are quite expressive. They allow describing essential classes of phenomena, constructing formal conclusions about events, plans and actions and describing different kinds of relationships between temporal objects. At second, these models are natural enough, because they provide tools for describing all natural temporal objects, so they can be easily used for describing real application domains. Constructions of our temporal logic, starting from determination of time and finishing with complex relationships between states and events, will be provided in later sections in the form of lambda calculus terms.

Temporal lambda calculus is mentioned in number of works (Henk Barendregt, 2008), (Barendregt, 1991). There are different kinds of typisation, which are used for

describing different kinds of dependences between terms and types. For performing settled targets λC kind of lambda calculus suits the best way. This typisation called calculus of constructions allows describing terms dependent of terms, terms dependent of types, types dependent of terms and types dependent of types. In this model types are considered to be terms of special type *. For type constant * next rules are considered: if A is a type then A: *, *: □, * → *: □, (*, *): □. Types dependent of types or terms are written with help of type abstraction Π.

We will add special object "context" written Γ to maintain all objects of model. Context will be used as special storage where objects contained with their types. Adding objects $o_1, \ldots, o_n$to context Γ is written as Γ, $(o_1, \ldots, o_n)$, deleting this objects is written as Γ \ $(o_1, \ldots, o_n)$.

Designed model

1 The model of time

This section is devoted to model of time description, which will be used later. We will use interval model of time in which time is represented as solid line where different continuous intervals could be selected. In this work time is considerate to be continuous and non-branching. Continuality of time means that smaller subinterval exists for every interval of time. Time is non-branching when future and past are determined and unique.

In this paper will be developed the approach showed in (Davies, 1995). Also there will be added next atomic types:

- Type τ will stand for type of time points;
- Type α will stand for real numbers;
- Type β will be used to describe extended Boolean values – 1 corresponds to truth, 0 to false and # will mean that expression neither true nor false;
- Type ν with single value Null: ν will stand for error. If any expression has type ν then this expression is incorrect in model.

Then we will add basic time operators ⊙, δ and Π after adding required basic types. Operator δ returns date of time point – natural number corresponding to this point. Every time point is identified uniquely by its date. Operator ⊙ proceeds backward transformation – it returns time point with date equal to the passed number. Finally, operator Π takes time point t_1 and real number a and returns continuous segment of time line with beginning at point t_1 and finish at point t_2 with date equal date of t_1 shifted on a. These operators follow next inference rules:

$$\frac{\Gamma \vdash \mathrm{T} : \tau}{\Gamma \vdash \delta T : \alpha}; \frac{\Gamma \vdash \mathrm{A} : \alpha}{\Gamma \vdash \odot A : \tau}; \frac{\Gamma \vdash A : \alpha, T \vdash \tau}{\Gamma \vdash \Pi A T : (\tau, \tau)} \quad (1)$$

Our model of time will be determined with a help of these operators and axioms of time. We should make remark concerning Π operator before listing axioms. Ift_1: τandt_2: τare time points and $\delta t_1 < \delta t_2$ then expression $\Pi(\delta t_2)t_1$ is equal to$\Pi(-\delta t_1)t_2$ and

stands for time interval from t_1 to t_2 whereas expression $\Pi(\delta t_1)t_2$ has meaning of another interval from t_2 to some point t_3 with date $\delta t_3 = \delta t_2 + \delta t_1$.

Now axioms of model could be listed.

1. Axiom of continuations. Every time interval has a smaller subinterval: $\forall I_1: (\tau, \tau)\ \exists \epsilon > 0: \alpha |\ \Pi\big(-(\delta(LI_1) + \epsilon)\big)(RI_1): (\tau,\ \tau),\ where\ R(X, Y) = X, L(X, Y) = Y$
2. Axiom of smoothness. Any two time intervals could be joined in one bigger interval if they meet in at least one point:$\forall I_1: (\tau, \tau), I_2: (\tau, \tau), \exists t_1: \tau |\ t_1 \in I_1 \& t_1 \in I_2 \rightarrow \Pi\big(-\delta(LI_1)\big)(RI_2): (\tau, \tau)$;
3. Axiom of infinity. For each time interval exists its successor and predecessor $\forall T: \tau\ \exists A_1: \alpha, A_2: \alpha\ |\ A_1 < \delta T < A_2 \& \Pi A_1 T: (\tau, \tau) \& \Pi A_2 T: (\tau, \tau)$;
4. Axiom of period's equality. If period i is immediately followed by periods j and k and period l is immediately followed by period j then l is followed by k: $\forall I_1, I_2, I_3, I_4: (\tau, \tau)\ MI_1I_3 \&\ MI_1I_4 \&\ MI_2I_3 \rightarrow MI_2I_4, where\ MIJ = (RI = LJ)$. Second definition of equality is written by next expression: $\forall I_1, I_2, I_3, I_4: (\tau, \tau)\ MI_1I_3 \&\ MI_1I_4 \&\ MI_3I_2 \& MI_4I_2 \rightarrow\ I_3 =_{\beta\eta} I_4$. This expression states that if two periods are situated between two other periods then they are equal;
5. Axiom of time line singularity. For two pairs of successive time intervals exist three cases of disposition. These pairs could meet in one time point or in points shifted on m from each other:

$$\forall I_1, I_2, I_3, I_4: (\tau, \tau)\ MI_1I_2 \&\ MI_3I_4 \rightarrow MI_1I_4 \wedge\ (\exists I_5: (\tau, \tau) |\ MI_3I_5 \& MI_5I_2) \wedge\ (\exists I_5: (\tau, \tau) |\ MI_1I_5 \& MI_5I_4)$$

Theses axioms declare model of time which we will use later. This model allows declaring complex relations and operators for time intervals like relations of following and chaining. They will be inferred in later sections as needed.

2 Actions and objects

In this section there will be represented objects and actions. Behaviour of these objects in time will be considered later. Object in this paper will be thought as some term typed with atomic type Θ. Every term with type Θ → γ i.e. every function, which can be applied to object will be called property of that object. Applicability of property to object is defined by next term A: (Θ → γ) → Θ → β.The final type of property with type γ of object Θ isΠT: Θ, P: Θ →γ. IF(APT)γν, where term IF: Πa: β, θ_1:*, θ_2:*:a = 1 θ_1, a ≠ 1 θ_2 – is a conditional term. This final type shows that property which suits object's type may be not evaluated for it. In other words the result of application of property term to object term is γ typed term if this property is applicable for this object. Type of property will be written as $\mathcal{P}_\gamma$ later. Although objects and properties are not temporal objects we should state that in system, which changing in time, they could have different "versions", i.e. the same object could exist in different variations dependent on moment of time in which we examining that object. Taking these considerations into account general type of objects should be changed from Θ toΠt: τ, Γ.IF(Γ, (θ, t))Θν = $\mathcal{O}_\Theta$. For further work with this object its correct version should be received. If version could not be received then type of result of entire evaluation with this object will be an error. Terms of applicability and properties do not

depend on point of time, they depend only on version of object, so they are evaluated after version is received. All versions of objects are stored in context Γ.

It should be reminded that one the main aim of developing model is planning, i.e. determining result of sequence of actions in every moment of time, so notion of actions is crucial for our model. Despite action is not temporal object, it closely relates with the notion of time. At first, execution of every action requires time as resource, because no actions executes momentary. At second, actions will require its preconditions to be hold in this model and preconditions could be changed in time, so they will be represented with terms typed τ → β. Finally, some actions executing together could change same objects by blocking or not blocking them. In this case it should be traced than the required objects are available and what changes occurred last. Taking all this requirements into account we propose next definition of action. Action is a term whose application to real number, to set of preconditions and to set of objects means that in sometime these objects will be changed in some way if all preconditions are hold. Type of action changing objects $\mathcal{O}_{\Theta 1} \ldots \mathcal{O}_{\Theta n}$ by its definition is:

$$\mathcal{A} = \Pi\Theta_1 : * \ldots \Theta_n : *.\alpha \to (\tau \to \beta)^m \to (\Theta_1 \ldots \Theta_n) \to \big(\beta, (\Theta_1 \ldots \Theta_n)\big) \qquad (2)$$

Execution action A with duration a at the point of time t, using objects $o_1, \ldots, o_m, o_{m+1}, \ldots o_n$ = O if objects 1 … n and objects m+1… n if preconditions $p_1, \ldots, p_n$ = P are hold is ruled by statement (3).

$$\frac{\Gamma, \mathrm{t}: \tau, (o_1, t) \ldots, (o_m, \mathrm{t}), \ldots, (o_{\mathrm{m}+1}, \mathrm{t}) \ \ldots (o_n, \mathrm{t}) \vdash (p_1 t \& \ldots \& p_n t) = 1}{\big(\forall i \in (t, t+a) \forall j \in (1, m) \to \Gamma \backslash (o_j, i)\big)\big(\Gamma, Tail(AtPO)\big)(Head(AtPO) = 1)} \qquad (3)$$

The result of execution AtPO is a set of versions of objects from O in time t+a and the boolean value which shows if the action was executed successfully or not. We can see from the rule that versions of blocked objects are removed from the context. It means that every other action, whichrequiresthatobjectscouldnotbeexecutedbecauseitwouldnotbeabletoreceive them. On the other hand, non-blocked objects are available and they could be changed by other actions, but actual version of these objects will be set by the action which finishes last.

Not every action could be executed in model. For example, if some precondition of action is not hold at the start of this action or some required object is blocked, then action execution fails. Failed action will not change context and head of such execution will be equal to 0 or # depending on kind of occurred error.

3 State and temporal reduction

The first temporal object "state" will be considered in this section. State is temporal analogue to property. Property does not depend on time, it depends only on objects version whereas state is the statement that value of some property of some objectis equal to the specified value in some moment of time.

Type of temporal objects is T → (τ, τ) → β, where T is template which defines specific kind of temporal object. T is equal to S = Θ →$\mathcal{P}_\gamma$→ γ for states. Type con-

structor T → (τ, τ) → β is called standard temporal notation. Type of state is written $\mathcal{T}_S$ in this notation.

If $S_1$is an elementary state then expression $S_1oP_\alpha3(1, 2) = 1$ means that object o is in state S_1 in time interval (1,2), i.e. value of property P_α of object o is equal to 3 in time interval (1,2). Complex state could be constructed as combination of elementary ones if corresponding properties are independent. Otherwise, we can construct complex property of object and construct elementary state for that property. It should be mentioned that state defines its object, property and value uniquely. In previous example state S_1may be applied only to object o, its property P_α and value 3. For other objects, properties or values another state should be used.

Defined states meet all criteria to be natural, namely:

- State is homogeneous. If semantic valence of state is known in some interval then it does not change in any time point inside this interval. This could be proved with a help of axioms of time structure;
- State is completely defined by its non-temporal parameters. This means that state is pearly temporal object.

For the convenience of working with temporal objects special reduction rules will be introduced. In classical βη lambda calculus there are two reduction rules:

β rule

$$(\lambda x.M)N =_\beta [x := N]M$$

and η rule

$$\lambda x.Xx =_\eta X.$$

Both β and η rules can be used as weak equality.

Temporal extension of reduction will allow using reduction for comparing temporal terms in time. Temporal reduction of term T_1 to term T_2 in point of time t is defined by next inference rule.

$$\frac{\Gamma \vdash T_1 : \mathcal{T}_T, T_2 : \mathcal{T}_T, \mathrm{T}T_1\mathrm{t} = \mathrm{T}T_2 t, T_1 \to_{\beta\eta} T_2}{T_1 \to_{\beta\eta}^{t} T_2} \quad (4)$$

This rule uses truth operator T. Being applied to temporal object this operator returns semantic valence of this object. This operator has general type $\mathcal{T}_T$→τ→β. We will use extension T_I: $\mathcal{T}_T$→ (τ, τ) →β of T operator for stating semantic valence of temporal object in time interval. It should be mentioned that temporal reduction can be applied only to temporal objects by its definition. Equality of non-temporal objects does not depend on time of consideration so temporal reduction degenerates to usual βη reduction.

Temporal reduction should consider defined structure of time. This means that no contradictions should exist – if some time expression is correct then it should be possible to make reduction by this expression. We have introduced reduction by time point. Inference rule (5) introduces reduction by time interval.

$$\frac{T_1 \to_{\beta\eta}^{t_1} T_2, T_1 \to_{\beta\eta}^{t_2} T_2, \delta t_1 < \delta t_2, T_I T_1 \Pi(-\delta t_1) t_2 = T_I T_2 \Pi(-\delta t_1) t_2}{T_1 \to_{\beta\eta}^{\Pi(-\delta t_1) t_2} T_2} \quad (5)$$

Later for time interval reduction will be used shortcut as shown in expression (6).

$$T_1 \rightarrow_{\beta\eta}^{\Pi(-\delta t_1)t_2} T_2 \coloneqq T_1 \rightarrow_{(t_1,t_2)} T_2 \tag{6}$$

This notation is much more compact and clear. Next inference rule adds to the model support of times smoothness.

$$\frac{T_1 \rightarrow_{(t_1,t_2)} T_2, T_1 \rightarrow_{(t_2,t_3)} T_2}{T_1 \rightarrow_{(t_1,t_3)} T_2} \tag{7}$$

Rules, which add support of other axioms, can be added by analogy. Presented machinery of temporal reduction is very useful for reasoning about properties of different temporal objects.

4 Events and causality

This section is devoted to second temporal object "event". Event is temporal analogue of action. Connection between events and actions is established as follows – events E_1 and E_2 occurred in the moments of time t_1 and$t_2$if there is action A which started in the moment t_1 and finished in the moment $\delta t_2 = \delta t_1 + a$ where a is duration of A. In standard notation type of event is $\mathcal{T}_A$, where A = $\mathcal{A}\rightarrow\beta$, i.e. full type of event is $\mathcal{A}\rightarrow\beta\rightarrow(\tau, \tau)\rightarrow\beta$

If E_1 is even then expression $E_1A1(1,2) = 1$ means that event E_1 has occurred and action A successfully (with output 1) finished in time interval (1,2). Expression $E_2A\#(1,2) = 1$ means that event E_2 has occurred and action A started in time interval (1,2).

After defining events and states we will define causality relationships between them. As start point we will use definitions and agreements defined in (McDermont, 1982). Developed framework of causality relationships can be a powerful tool for obtaining new statements about temporal changes of application domain.

Causality relationships can be set among two events or among event and state. Relationship between events will be considered at first as simple one. In this case occurring (or not) of the first event is followed by occurrence (or not) of second the second one. This relationship is defined by term $EC\colon \mathcal{T}_E \rightarrow \mathcal{T}_E \rightarrow (\tau \rightarrow \tau) \rightarrow \alpha \rightarrow \beta$. .

Value 1 of term EC means that second event should occur if first has occurred. Value 0 of term EC means that second event should not occur if first has occurred. Value # of term EC means that there is no relationship between events. Reasoning with help causality relationship between two events is defined by next inference rule.

$$\frac{\Gamma, E_1A_11(t_1,t_2) = 1, ECE_1E_2(t_3,t_4)a_1 = \{0,1\}}{\Gamma \vdash E_2A_2\left(\Pi(a_1+\delta t_4)\left(\odot(a_1+\delta t_3)\right)\right) = \{0,1\}} \tag{8}$$

For defining causality relationship between state and event we should define notion of persistence of state first. Notion of persistence is follows from next premises. At first, we should make assumptions about future states of objects. At second, we can never be sure about truth of reasoned values of states, i.e. we could not take into account all possible changes. So, persistence of state will be represented by term PR:Πb: $\beta.\mathcal{T}_E\rightarrow(\tau, \tau)\rightarrow$IFbβv. Using this term if we want to claim that object o is in state S_1

in time interval (t_1, t_2) then we use next expression $PRS_1(t_1, t_2)1$, and $S_1oP_\gamma\gamma(t_1, t_2)$ is equal to 1. As soon as time interval is in past we could uniquely define term $S_1oP_\gamma\gamma(t_1, t_2)$ and expression for persistence is$PRS_1(t_1, t_2)(S_1oP_\gamma\gamma(t_1, t_2))$. In that case if object was not in predicted state then type of persistence changes to error type. Developed technique allows planning and predicting occurrences of events and values of states in future and control planning errors.

With help of defined notion of persistence we could define causality relationship between state and event. If there is causality relationship between event E and state S and there is occurrence of event E then state S in a persistence since some time point. This relationship is defined by term $EC{:}\mathcal{T}_E \rightarrow \mathcal{T}_E \rightarrow (\tau \rightarrow \tau) \rightarrow \alpha \rightarrow \beta$. Value 1 of term PC means occurrence of event is followed by persistence of state. Value # of term PC means that there is no relationship between event and fact. Value 0 is not allowed for term PC. Reasoning with a help of causality relationships between event and fact is defined by next inference rule.

$$\frac{\Gamma, E_1A_11(t_1,t_2) = 1, PCE_1S_1(t_3,t_4)a_1 = 1}{\Gamma \vdash PRS_1\Big(\Pi(a_1 + \delta t_4)\big(\odot(a_1 + \delta t_3)\big)\Big)1} \tag{9}$$

It should be noted that rules of causality in this model can fail. Fail of rules means that they should be corrected under new conditions.

5 Processes

One of the main aims of the developing theory is description and maintaining of processes, so notion of process took important place. In our theory process is composed by next elements:

1. **Initial and runtime contexts**. Initial context contains all initial values and objects of model. In this context all objects, which exist before process starts and values of all properties of contained objects are known. Also, initial context contains all actions, available for process. For each action its preconditions, used objects and duration is also known. If required objects do not exist in initial context – they can be identified by some properties. For example, action A_1 changes value of object with value of property P_1 equal to 2 instead of action A_1changes value of object to_1. Initial context contains all resources which are required by process to start. As soon as process starts, new runtime context creates and content of initial context copies to runtime, so initial context does not change while process executes. Content of runtime context is called processes output if process is finished. In general, one object can be contained in initial contexts of two or more processes. This case raises question of division resources among processes;

2. **Storage of temporal objects**. Execution of process in time is controlled by registration of temporal objects – states and events. This allows abstracting from real process and controlling process on higher level. Registration the result of every action during processes run is not efficient way of control, because process could contain a lot of actions and result of a single one is not important for model at all. Moreover, control of each object and action strikes productivity. Sates and objects do not contain in initial or runtime context due to they do not relate straight to the model, so they is

contained in special storage. This storage also contains causality relationships defined in model;

3. **Abstract process execution machine**. This abstract mechanism carries out next functions. At first, it contains information about possible time intervals and of processes start, all required conditions and initial action. When all conditions and time intervals and conditions met this machine starts process by executing initial action. During processes runtime this mechanism provides control of time and executes actions when their start time occurs. The machine controls temporal objects and causality rules and informs external systems about them. For example, if persistence of some state fails then the machine produces outside special signal and general systems received that signal starts new process. General system, which controls all processes resources and receives its outside signals is called processes owner. Finally, the abstract machine stares mask of expected final state of process and conditions of processes completion. The machine stops process if its runtime context suits final mask or at least one of completion conditions holds.

Developed construction of process provides quite high level of independence of described processes and allows defining and controlling numbers of processes in one general system. Moreover, this notation allows determining process in their natural kind, i.e. process is true in half open interval. We know its start point and, in general case, we do not know when it will be finished.

Conclusions

This paper describes the model of interval temporal logic based on typed lambda-calculus. The main of aim of this paper is to construct formal typed lambda calculus based model which allows reasoning about dynamic properties of application domain. Temporal logic was chosen as start point for such model.

Developed model provides tools for describing both static entities like objects, their properties and actions and basic temporal objects – states, events. Model also allows describing complex processes; model allows high level control and research of described processes. Special emphasis is paid to resources of processes. All considered objects both temporal and static is described in lambda terms. This uniform notation noticeably simplifies model. Developed notation is close to natural determinations of these objects.

Main advantages of developed model are its homogeneity, easiness of modification of either some components or the whole model and model's expressivity. Model allows limited reasoning about dynamic characteristics of processes with a help of causality rules.

For this model next improvements are proposed. At first, reasoning in model will be improved. Model should allow finding out temporal regularities and patterns. At second, model will be extended by more precise description of processes. More attention will be paid to concurrent and parallel processes. Questions of resource division will be revised with special care.

References

1. Barendregt, H., 1991. Introduction to generalized type systems.April.pp. 125-154.
2. Davies, R., 1995. A Temporal Logic Approach to Binding Time Analysis.*BRICS Report Series,* October.
3. Dennett, D., 1987. Cognitive Wheels: The Frame Problem in Artificial Intelligence.
4. Henk Barendregt, W. D. R. S., 2008. *Typed Lambda Calculus (Draft).*Nijmegen and Pittsburgh.
5. James F. Allen, G. F., 1994. *Actions and Events in Interval Temporal Logic,* Rochester: Computer Science Dept. University of Rochester.
6. Jean-Pierre Desclés, Z. G., 1993. Is the notion of process necessary.
7. Lamport, L., 1994. The Temporal Logic of Actions.*ACM Transactions on Programming Languages and Systems*, Issue 79.
8. McCarthy, J., 1986. Applications of Circumscription to Formalizing Common Sense Knowledge, Stanford: Computer Science Department Stanford University.
9. McCarthy, J., 1986. Circumscription -- a Form of Nonmonotonic Reasoning, Stanford: Computer Science Department Stanford University.
10. McDermont, D., 1982. A Temporal Logic for Reasoning About Processes and Plans..COGNITIVE SCIENCE 6, pp. 101-155.
11. Venema, Y., 2001. Temporal Logic. B: The Blackwell Guide to Philosophical Logic. :Blackwell Publishers.

Level based information theory

Shumsky L.D.

Department of Cybernetics and Information Security,
National Research Nuclear University "MEPhI"
Moscow, 115409 RF, Kashirskoye Sh., 31
shumskyld@gmail.com

Abstract. The main aim of our work is to develop sketch of general information theory, which should provide tools for description information, its sources, methods and rules of its processing. Key idea of this theory is dividing information on some levels corresponding to its functions and features. We expect that it will be possible to solve problems of information exchange, storage and processing in much more elegant and efficient way using developed theory.

Introduction

Although it could be heard that we are living in "World of information" and information is really widespread concept, there are a lot of opinions how this concept should be treated. This drawback could not be omitted easily. At first, role of information in different fields has been grooving steadily since the beginning of last century. At second, information connects different parts of sciences and even connects different sciences (Paul Davies, 2010). The same models and methods of information processing are successfully used in math, computer science, economic, biology, psychology and philosophy.

Some elements of information theory like coding theory, noised information passing channels and entropy counting can be taken into account when designing artificial neural nets. It should be mentioned that received results are corresponding well with results received for natural neural nets (Fyfe, 2000). In physics, concept of information is going to change the paradigm at all. Classically, connection among math, information and physic is described as math → physic → information. From this point view, math relations are the most basic aspects of existence. Observing physical world is the expression of those math relations. Information is thought to be the derived concept showing only concrete state of matter. As far as quantum mechanics is researched better new paradigm is arises. This relates to the fact that in quantum world there are no concrete events and only probability distribution exists. Concerning to the new paradigm, information is the source of existence and relation receives next view: information → law of physics → matter (Wheeler J.A., 1998).

Those examples show one side of information – information as a data. Whenever we are talking about information which is created, stored or processed by human, other models are used and other sciences are involved. For example, different semantic representation models are widely used in cognitive psychology (Neisser, 1967)

where they are used to represent humans mind, memory, speech and mechanisms of problem solving. We are able to name lots of different examples of information revealing in different spheres, and we can conclude that notion of information is extremely important.

The main aim of our work is to develop sketch of general information theory, which should provide tools for description information, its sources, methods and rules of its processing. We expect that it will be possible to solve problems of information exchange, storage and processing in much more elegant and efficient way using such theory.

Information processes and definition of information

There are lots of different definitions of information, but none of them is general enough for fitting our objects. So we have to define information in this paper as follows. Information is an aggregate of all data from different levels fixed in some system, where every data element is produced by this system or is received from other system. We are able to make next conclusions from this definition.

At first, data could have different levels. The lowest level of information represents raw data – these data are independent, registered with help of some interactions properties of some objects generated by their states. Raw data are data of level 0. Other levels of data represent data about lower level data i.e. data of level n are metadata for level n-1. For example, data of level 1 describes some settings for received raw data. Then data of level 2 describes migration of system's settings. The idea of separating data by levels is one of the core ideas for our work. At second, data are tied with some system. We will not consider data, which is not tied to observed system. More to say, data are not only tied with system, they are fixed in system. It means that we cannot have any piece of data which is tied with no system or which is tied with two or more systems. This restriction provides some kind of abstraction. Finally, although some systems are able to produce data, any piece of data has its' source. It is not possible for data to appear from nothing.

Next important notion for our theory is the notion process. This concept is important because aim of every information system is to maintain some processes. These processes define representation of information inside of the system, rules of its processing and allow to measure quantity of information. More to say, often system should support different processes, which defines different representation of the same pieces of information. For example, if system stores and translate output of some sensor then it may read data in binary format store it as relations in database and send them as XML. Translation of data from one format to another requires huge effort so representing information in the most suitable for each action view will simplify the system and make it more transparent.

Corresponding to the classical processes theories process is a graph, which nodes are states of system and edges are some actions (Fokkink). Each action in process requires some resources to be executed, i.e. action changes some properties of system. Information process will be defined next way. Information process is a process which is executed in information system and all resources of which are pieces of information

or information services like channels and storages. All actions of information processes are able to change information and or use information services without changing. Also, execution of action can require time. For example, process "read data from channel 1, validate them using scheme S and send to channel 2" is an information process. This process has three actions. The first one requires data of level one in channel 1, the second one requires result of the first action and data of level 1 and the last one requires the output of the second.

We can identify three kinds of actions depending on types of data and services they require. Specific view of information is preferred or each kind of action. So information can be presented on three different levels in our model. All of these levels are connected to each other and information could be moved from level to level using only standard mechanisms of the model. In other words, we offer model of information view of which depends on actions which requires this information to be executed. These structures allow providing information in the most suitable form for each, which we want to execute.

The first type of actions requires channel services, data of level 0 and fixed number of data of level 1. Actions of that sort are transmissions actions. These actions are used to read data or to send them outside. First level data are used by those actions as transmissions settings. They describes format of sending data, characteristics of receiving or sending systems, parameters of channel services and other settings. Amount of those data is fixed, it does not depend on amount of data which we want to send or read. Transport level of information corresponds to this kind of actions.

The second type actions requires storage services, data of level 0, number of data of level 1 and fixed number of data of level 2. Actions of that sort are storage actions. They are mostly used to store data in long perspective and provide access to stored data. Storable level of information corresponds to this kind of actions. As an example, we can provide (Langel, 2009), (Codd, Futher normalization of the data base relational model, 1971), (Codd, Relational completeness of data base sublanguage, 1972). First level data are used as metadata for stored data. They describe data types and domains and other features. For example, those data describes encoding of text values and possible values for enumerated data. Amount of these data depends linearly on amount of raw data, because countable number of features can be introduced. Data of level 2 are general scheme which describes how data is stored. In analogy with Codd's relational model, these data are scheme of relationships. Amount of these data do not depend on amount of stored data.

The last kind of actions is data processing actions. Actions of this class are hard to define in general. These actions do not use any services, but they are able to require any amount of levels of data. The result of these actions is new data of different levels or some changes in existing data. For these actions we will provide conceptual level of information. Information on conceptual level is connected with conceptual scheme of application domain. Ontology, written using description logic (Baader, Calvanes, & McGuiness, 2003) is an example of information on conceptual level.

Levels of information

This section describes main features of each of presented information levels. For each level we will describe main function of this level, specific data and services for this level and rules of transfer of information from this level to other. Information level defines representation of raw data, specific data of higher levels and actions which requires information on that level.

The first considered level is the transport level. The main aim of this level is to establish and maintain connection with outside information system and data sources. There is no special interpretation or information processing n on this level. This level is distinguished from other for next reasons. At first, distinguishing of this level is justified by the fact that all tasks of information exchange are quite similar. At second, transmission of information is a quite specific task, which is often not connected with main tasks of the system. Finally, transmission uses specific settings and services which are not used by other actions and processes.

Information from transport level can be passed to other levels using next channels. Channel to storable level is used to parse and store input structured information. This channel is used when structure of information is known. Channel to conceptual level is used to determine structure of non-structured input information or to recover some specific features. It should be noted that after receiving information cannot stay on transport level.

Information could be received by transport level from other systems or from raw information sources. The example of raw information source (or data source another words) is some physical mechanism which is sending sequence of symbols, which belongs to some set called source's alphabet (Ash, 1965). More generally, data source is a source which sends non structured data of level 0 and, may be, fixed amount of data of level 1. Such raw data often should be processed and structured before saving, so preferred channel for these data is the channel to conceptual level. In contrast, information sources send structured information and preferred channel for these data is a channel to storable level.

Summing up, information on transport level is adapted for maintaining transmission tasks. Raw data are represented in the most suitable for transmission view, level contains data of level 1 concerning rules of message routing and mapping. Actions which are executed on this level use outside channel services, raw data, settings to determine final sending or receiving format and settings to determine senders or receivers.

The second considered level is the storable level. This level provides representation of information which helps storing information and provides optimal access to stored information. The second level aim is to validate information with help of data types, data domains and conceptual schemes. Validation on this level serves the purpose of defining how information should be stored in the most efficient way. If some information is invalid it should not be saved and determining this information optimizes work of storable level. This level is distinguished from others because storing some information for long period and providing access to it is quite common task and often requires translation to some special view. These tasks are often solved in non-optimal way and using special storablelevel can increase system's efficiency.

Information from storable level can be passed to other levels using next channels. Channel to transport level is used to send some data from system. Channel to conceptual level is used to evaluate some features of data. Transmission into these channels is performed in the form of inner structures. Key metadata for this level are data types, domains and relationship scheme. Under domain will mean directed partly ordered set, on which basis could be determined (Samson Abramsky, 1994). Generally, domains define all possible values of some field. Data types in our model are named objects which contain domain, rules of validation and rules of conversion.

The last considered level is conceptual level of information. This level provides representation of information which simplifies some specific tasks of information processing. For simplifying these tasks this level provides number of different high level data for each piece of raw data. This special data are the main reason of distinguishing this level. Although amount of special metadata simplifies execution of some actions, they could seriously strike efficiency of the whole system. Conceptual level plays role of filter and aggregator for the information of the system. Information is filtered by questions of the system. Questions of the system are possible aggregates of data which system should provide. Questions are a mechanism which lets to evaluate amount of information. Amount of information is a measure of uncertainty in answers to system's questions. All information which do not increase amount of information of the system is filtered. Aggregation and structuration of information is performed with a help of conceptual schemes.

Information from conceptual level can be passed to other levels using next channels. Channel to transport level is used to send some new settings for outside systems. This channel is also used to send some messages produced directly by conceptual level, for example, investigated structure of the input raw data or messages which has complex structure. Channel to storable level is used to send some new high level data or to send evaluated values.

Conclusions

This paper describes example of building general information theory, which allows describing data, metadata and information processes in complex. This theory is based on the idea that different information processes requires different representation of the same information and translation of this information within the framework of general theory can be much more efficient than performing this translation externally. Having analyzed possible actions we offer usage of three-level structure of information. We distinguish transport, storable and conceptual levels of information. For each level we describe main function of this level, specific data and services for this level and rules of transfer of information from this level to other.

This approach may be vital for constructing information systems, which should solve different tasks or if support of this processes should be added for existing systems.

It is planned next further developments of this theory. At first, more formal analysis and representation of information processes and actions should be made. Corres-

ponding to that analysis more formal description of information and data levels will be made. At second, each level of information will be described more carefully. It will be specified underlying models and levels features. Finally, for this theory is proposed the software realization which let testing it in real world conditions.

References

1. Ash, R. B. (1965). Information Theory. Dover Publications, Inc.
2. Baader, F., Calvanes, D., & McGuiness, D. L. (2003). The Description Logic Handbook: Theory, Implementation, and Applications. Cambridge: Cambridge University Press.
3. Codd, E. F. (31 August 1971 г.). Futher normalization of the data base relational model. IBM Research Laboratory.
4. Codd, E. F. (6 March 1972 г.). Relational completeness of data base sublanguage. IBM Research Laboratory.
5. Fokkink, W. Process Algebra: An Algepraic Theory of Concurrency. Amsterdam.
6. Fyfe, C. (2000). Artificial Neural Networks and Information Theory (vol. 1.2). The University of Paisley.
7. Langel, J. (2009). Logic and Information. A Unifying Approach to Semantic Information Theory. Freiburg: UniPrint.
8. Neisser, U. (1967). Cognitive psychology. New York: Meredith.
9. Paul Davies, N. H. (Ред.). (2010). Information and the Nature of Reality. From Physics to Metaphysics. New York: Cambridge University Press.
10. Samson Abramsky, A. J. (1994). Domain Theory. B D. M. Abramsky, Handbook of Logic in Computer Science (T. 3). Oxford: Clarendon Press.
11. Wheeler J.A., F. K. (1998). It from bit. B W. J.A., Geons, Black Holes, and Quantum Foam: A Life in Physics. New York: Norton.

Компьютинг и его принципы: направление, деятельность, научная дисциплина*

Вольфенгаген В.Э.

НИЯУ МИФИ / Группа компаний “ЮрИнфоР”, Москва, 119435 РФ
vew@jmsuice.msk.ru,
WWW: http://www.wolfengagen.mephi.ru

Аннотация В настоящее время компьютинг признается одной из естественных наук. Информационные процессы и вычисления были обнаружены в глубинных структурах многих естественно-научных областей. Вычисление там присутствовало задолго до появления первых компьютеров, но существенный сдвиг в понимании этого произошел только в последнем десятилетии. До этого момента молчаливо предполагалось, как само собой разумеющееся, что компьютинг является наукой об искусственном. Теперь же многими учеными усматривается, что информационные процессы происходят и в природе.

Ключевые слова: компьютинг, вычисление, информационные процессы.

Введение

В целом сегодняшний взгляд на *математику* соответствует ее пониманию как логической науки, которая четко структурирована и правильно обоснована. Эти аспекты понимания связаны с представлениями о *формализации* и *строгости* при изложении своих идей, в частности, результатов выполненного субъектом научного исследования. Под *компьютерными науками* (computer science) или *информатикой* обычно понимают разновидность *инженерии*, которая направлена на построение или конструирование по большей части не физических, а абстрактных сущностей, которые соответствуют заранее поставленным целям или требованиями.

Как можно заметить, и математика, и компьютерные науки в качестве предметной области имеют абстракции, в случае математики – это ‘идеи’ субъекта, которые абстрактны, а в случае компьютерных наук – это ‘абстрактные сущности’, которые, конечно, идеальны. При выполнении исследования в той или иной области усилия направляются на порождение новых абстрактных сущностей, установление иных пониманий уже известных сущностей, открытие новых взаимоотношений между ними, либо переосмысление уже известных связей и зависимостей, их адаптация к новым условиям. Это дает возможность получить некоторое представление о *предмете* и *методе* математики и компьютерных наук.

---

* Посвящается 70-летию Национального исследовательского ядерного университета «МИФИ»

1 Компьютинг как естественно-научная дисциплина

В настоящее время компьютинг признается одной из естественных наук (см. P. Denning, [3]). Информационные процессы и вычисления были обнаружены в глубинных структурах многих естественно-научных областей. Вычисление там присутствовало задолго до появления первых компьютеров, но существенный сдвиг в понимании этого произошел только в последнем десятилетии. До этого момента молчаливо предполагалось, как само собой разумеющееся, что компьютинг является наукой об искусственном. Теперь же многими учеными усматривается, что информационные процессы происходят и в природе.

1.1 Революция в понимании компьютинга

Если речь вести о революционных изменениях в понимании роли и места компьютинга, то это заняло длительный период времени. Различают три этапа, во время которых развивались соответственно *инструментальные средства* (начиная с 1940 г.), *методы* (начиная с 1980 г.) и *фундаментальные процессы* (начиная с 2000 г.).

На первом этапе появились электронные цифровые компьютеры, а компьютинг рассматривался как средство, обслуживающее решение уравнений, расшифровку кодов, анализ данных, управление бизнес-процессов и моделирование. Пользуясь компьютингом как инструментальным средством, удавалось выполнять прежде недоступный для вычислений анализ. Кроме того, компьютинг способствовал развитию целого ряда важных технологий, среди которых атомная энергетика, авиастроение, кораблестроение, синтез препаратов, строительство и предсказание погоды.

На втором этапе компьютинг стал незаменимым во многих областях. Он эволюционировал *от* использования имеющегося знания *до* получения нового знания. Заметим, что нобелевский лауреат по физике Кен Вильсон среди первых сказал, что вычисление на компьютере превратилось в третью опору науки, помимо прежних двух – теории и эксперимента. Так возникло представление о *вычислительной науке*, в рамках которой научный результат получался с существенным использованием компьютера. Развитие технологии вскоре привело к возникновению высокопроизводительных вычислений.

На третьем этапе у вычислений проявилась важная особенность. В различных областях знания были выявлены информационные процессы, развивающиеся в глубинных структурах материи. В частности, биология стала рассматриваться как информационная наука. “Механика” жизни кодируется цифровым образом, а выход отдельных систем считывается целыми последовательностями “головок”. Результаты по биологии не ограничиваются стенами лаборатории, а возникают сразу во многих направлениях. Это, прежде всего, относится к структуре ДНК, а биологи и компьютерщики стали выполнять совместные исследования.

Биология не является исключением. Например, в физике считается, что квантовые волны несут информацию, порождающую физические эффекты. Значительное развитие получили квантовые вычисления и квантовая криптография.

Успехи компьютинга привели даже к декларированию нового типа науки, согласно посылкам которой книга природы написана на языке компьютинга, и это вместо известного принципа, что книга природы написана на языке математики, восходящего к Галилею (см. S. Wolfram, [4]).

Экономическим системам также присущи информационные потоки. В менеджменте рабочие потоки и социальные сети рассматриваются как фундаментальные информационные процессы, встречающиеся во всех организациях. Художники и писатели стали повсеместно использовать вычисления, начиная от анализа и заканчивая созданием новых произведений. При анализе Web обнаружены новые формы социального поведения, а также способы компьютинга, при которых Web в целом рассматривается как испытательный стенд. Артефакты компьютинга стали даже культовыми, тому есть много свидетельств, например, iPod, eBay, Wikipedia, Google, Playstation, Xbox, Wii и многое другое. Участились случаи, когда даже политиками выполняется сложный социальный анализ данных и организуется блоггинг. Неоднократно отмечалось, что вычислительные понятия глубоко укоренились и в повседневном мышлении, а вычисление как таковое стало повсеместным.

Во многих областях применение компьютинга оказывается новым, а применение представления об информации – нет. Во многих областях именно информация стала ключевым понятием. Во многом это произошло благодаря кибернетике, возникшей в 1948 г. из работ Н. Винера. Кибернетика определялась как "наука о взаимодействии и управлении как в машинах, так и в живых организмах". Как теперь считается, кибернетика не смогла выжить как самостоятельная наука, поскольку немногие из членов научного сообщества разделяли точку зрения Н. Винера, согласно которой кибернетика является чем-то особенным по сравнению с другими науками.

Принятие идеи, что компьютинг – наука, происходило не сразу. Например, известна позиция крупного физика Р. Фейнмана, который видел отличие компьютерных наук от физики в том, что они, как и математика, не являются настоящей наукой. Он считал, что компьютерные науки ближе всего к инженерии, когда что-либо изготавливается, а не сводится к установлению неких абстракций с последующим оперированием ими. Это представление более или менее соответствовало состоянию дел на уровень начала 1980-х гг. Буквально через одно-два десятилетия физики стали уже считать, что информационные процессы имеются в самой природе, а компьютеры способствуют их выявлению и изучению. Это был уже принципиальный сдвиг в понимании сути компьютинга. До тех пор долгое время информационные процессы понимались как результат человеческих усилий, то есть искусственный феномен, порожденный также изготовленными людьми же компьютерами.

Но теперь прежнее понимание компьютерных наук как изучение всего, связанного с компьютерами и ими вызванного оказывается устаревшим. В настоящее время компьютинг стал пониматься как изучение естественных и искусственных информационных процессов. К сфере компьютинга относят компьютерные науки, компьютерную инженерию, программную инженерию, информационные технологии и науки, а также информационные системы.

1.2 Основные принципы

В середине 1990-х гг. стало складываться впечатление, что компьютинг достиг такой фазы своего развития, когда стало возможным выделить его *фундаментальные принципы* ([1], [2]), которые сводились к следующему:

– выявление глубинной структуры вычисления и причин ее наличия во многих областях знания;
– выявление общих принципов, присущих технологиям и способствующих упрощению, новым открытиям и инновациям;
– выработка общего языка, пользуясь которым можно вести междисциплинарное обсуждение вычислений;
– стимулирование новых подходов к обучению компьютингу и его изучению, а также стимулирование интереса к компьютингу среди молодежи.

Фундаментальные вопросы, порожденные выделенной системой принципов, сводились к следующему:

– что представляет собой *информация*;
– что представляет собой *вычисление*;
– каким образом вычисление расширяет область познания;
– каким образом вычисление ограничивает область познаваемого.

Подобному знаменитому вопросу биологии: “Что есть жизнь?”, – перечисленные вопросы возникали во всякой новой ситуации, породив современную систему так называемых “великих принципов” компьютинга (см. [5]). Их установление оказывается гораздо более трудной задачей, чем это можно себе вообразить. Прежде всего, как оказалось, *не сложилось* серьезного сообщества, которое было бы заинтересовано в обсуждении этих принципов. Кроме того, неясно, как сформулировать некоторые из наиболее глубинных принципов. Первые попытки их перечисления приводили просто к варианту переупорядочивания известных классификаций ACM. Со временем удалось добиться большего и к настоящему времени возникли формулировки семи – не обязательно независимых, – групп принципов, которые проявляют себя во множестве технологий:

– вычисление, его смысл и границы;
– коммуникация как надежная передача данных;
– координация как кооперация распределенных в сети сущностей;
– переформирование как хранение и поиск информации;
– автоматизация и установление ее границ;
– оценка как предсказание производительности и планирование объема;
– проектирование как построение надежных программных систем.

Выбор этих категорий обусловлен тем, что ими покрываются основные функции вычислительных систем. Хотя число новых технологий и принципов растет, но число их категорий, вероятно, останется неизменным на протяжении длительного времени. Эти категории позволяют единым образом взглянуть на совокупное знание о компьютинге, не дробя его в зависимости от проявлений в приложениях. Каждая из категорий обеспечивает относительно самостоятельный круг вопросов, открывая перспективу для исследования. То же можно сказать о

различных комбинациях из этих категорий. Вместе с тем одну и ту же сущность можно рассматривать в аспектах различных категорий. Например, протоколы Internet часто рассматривают с позиций коммуникативности, а также как средство координации или средство переформирования. Как оказалось, большинство из известных компьютерных технологий опираются на принципы сразу всех семи категорий. Тем самым подкрепляется уверенность в том, что разумно интерпретируя установленные принципы, можно увидеть общность различных технологий.

Компьютинг находится в непрерывном взаимодействии с прочими областями знания. Из них можно не только узнать больше о самом компьютинге, но и отыскать лучшие пути познания мира. Эти взаимосвязи не так просто укладываются в рамки известных определений, которые по большей части сводят компьютинг к исполнению алгоритма на компьютере. Но установленные семь принципов серьезно упрощают эту задачу, поскольку компьютинг понимается как *последовательность представлений*, причем и каждый переход от представления к представлению управляется представлением же. Таким образом, абстракция *представления* занимает центральное место в объяснительной системе компьютинга. В соответствии с таким пониманием оказывается, что ДНК может вычислять, то есть сам компьютер – это всего лишь инструмент, а вычисление имеет статус принципа.

Приводимая далее систематизация иллюстрирует принципы из отдельно взятой категории, а также дает примеры из компьютинга и иных областей (см. [5]).

Нерешаемость (вычисление).

Круг задач. Более трех тысяч задач из науки, инженерии и коммерции требуют существенно больше вычислений, причем даже для задач небольшой размерности, чем это можно сделать за обозримое время.

Общие примеры. Отыскание оптимальных решений. Задача коммивояжера. Задача упаковки рюкзака. Задача упаковки корзины. Задача покрытия плоскости.

Прикладные примеры. Доставка посылок. Перевозка на платформах. Маршрутизация такси. Маршрутизация самолетов. Планирование производства.

Сжатие (коммуникация).

Круг задач. Представления данных и алгоритмов можно подвергнуть существенному сжатию, причем впоследствии можно восстановить наиболее значимую информацию.

Общие примеры. Сжатие звука (MP3, MP4, ACC), изображений (JPEG, GIF), файлов (Zip). Преобразование Фурье.

Прикладные примеры. Код Морзе.

Выбор (координация).

Круг задач. Принцип неопределенности: невозможно однозначно выбрать одну из нескольких альтернатив за ограниченный отрезок времени.

Общие примеры. Аппаратура, безотказно реагирующая на прерывания. Взаимные исключения. Блокирования.

Прикладные примеры. Управление трафиком. Телефонные и сетевые маршрутизаторы. Цепочки ДНК. Свобода воли в психологии.

Локальность (переформирование).

Круг задач. Вычисления кластеризуют свои действия повторного вызова информации в иерархически агрегированные области пространства и времени для расширенных периодов.

Общие примеры. Виртуальная память. Аппаратное кеширование. Web-кеширование. Структуры взаимосвязи в параллельных машинах.

Прикладные примеры. Функциональные кластеры клеток мозга. Почти декомпозируемые экономические системы. Равновесные состояния в биологии.

Поиск (автоматизация).

Круг задач. Отыскание образа или конфигурации среди очень большого пространства возможностей.

Общие примеры. Генетические алгоритмы. Эволюционный компьютинг. Использование алгоритма границ и ветвей. Градиентный поиск.

Прикладные примеры. Генетическое эволюционирование. Передача генов потомкам.

Узкие места (оценивание).

Круг задач. Законы вынужденного протекания: в любой сети пропускная способность произвольного узла рассчитывается как произведение пропускной способности сети и числа посещений в расчете на одно задание для узла.

Общие примеры. Насыщение и эффекты горла бутылки в сетях связи.

Прикладные примеры. Быстрое распространение городских пробок. Конструирование поточных линий на производстве.

Иерархическое агрегирование (проектирование).

Круг задач. Бóльшие сущности компонуются из множества меньших сущностей.

Общие примеры. Уровни ОС и сетевого программного обеспечения. Модульность. Абстракция.

Прикладные примеры. Ступенчатое масштабирование в астрономии и физике. Функциональные органы в биологии. Фракталы.

1.3 Направления развития компьютинга

Компьютинг продолжает развиваться, устанавливаются его новые принципы, в то время как прежние перестают использоваться. Примером нового принципа является свободно масштабируемая структура сетевого соединения, и примером устаревшего – руководство по логике цепей электровакуумных ламп. Для мониторинга развития компьютинга и отыскания среди технологий и прикладных областей связей, которые вытекают из сформулированных принципов, на сайте, где они зафиксированы, поддерживается специальная библиотека. Она представляет собой разрастающуюся подборку материалов, инструментальные средства и процесс редактирования, направленные на поддержку изучения, преподавания, применения и организацию перекрестных ссылок технологий и принципов (см. [5]).

Среди тенденций компьютинга все больше проявляют себя игры. Их продвигает не только индустрия видеоигр. Кроме нее, высокую заинтересованность проявляют деловые круги и военные организации, которые широко применяют

игры-симуляторы виртуальной реальности. На их основе создаются эффективные тренинги, закрепляющие те или иные квалификационные навыки. Во многих университетах мира бакалавриат и магистратура строятся на основе использования (деловых) игр. Это возможное проявление довольно глубокой тенденции.

Распространение игр объяснимо в рамках категории *координирования*, поскольку именно игра представляет собой модель правил взаимодействия, управляющих сложными адаптивными социо-техническими системами. Интерпретируя игру правилами координирования, получаем, по-видимому, ее наиболее общую модель, в рамках которой получают объяснение все частные случаи координирования. Как известно, игры подразделяются на конечные и бесконечные. В конечную игру обычно вступают, имея целевую установку выигрыша, а в бесконечную – поддержания игры как таковой.

Такая конечная игра является иным проявлением представления о замкнутом, или *завершаемом* вычислении. Поскольку нынешние интересы смещаются в направлении от замкнутых к *открытым* вычислениям, то, возможно, помимо собственного желания, попадаем в область бесконечных игр.

К примеру, теоретические компьютерные науки развиваются в направлении от замкнутых вычислений к интерактивным вычислениям. Как оказалось, значительный объем информации скрыт от непосредственной сети Web и содержится в базах данных, а извлекается посредством интерфейсов, которые не сводятся к запросам поисковых механизмов. По некоторым оценкам даже оказывается, что информации, которая в сети доступна непосредственному поиску – не более 1%, а остальная информация скрыта в базах данных. Социологи и политологи склонны рассматривать Web-пространство как игру, в которой новые стратегии могут привести к изменению само́й игры, когда доступной поиску станет бо́льшая доля хранимых данных.

Разрастающиеся сообщества, среди которых особенно на виду eBay, Web, Google, iTunes, Wikipedia, Blogosphere, Amazon.com, превращаются в исследовательские лаборатории, где отрабатываются различные инновации, строятся социальные сети и т.п. Таким образом, Web и Internet, которые по своей природе являются бесконечными играми, предоставляют новые области для развития идей компьютинга. К настоящему времени в этой связи сложилось представление о Web-науке (см. [9]). К примеру, по законам статистики, свободно масштабируемые сети предоставляют такие структуры, которые создаются в Web самими людьми в результате их деятельности, причем получают оправдание многие стратегии избыточности, поиска социальной организации сетей и обнаружения знания. Эти особенности послужили даже основой для появления термина "человеческие вычисления". В качестве своего побочного продукта, игры приводят к полезным результатам, для которых не было заранее известных алгоритмов.

1.4 Обучение и изучение

Представления о принципах, в соответствии с которыми вычисление рассматривается как самостоятельная сущность и которые не ограничиваются компьютерами, но применимы ко всем областям, все больше проникают в сферу образо-

вания. Это приводит к инновациям в преподавании компьютинга и стимулирует интерес студентов к этой дисциплине. Известны различные примеры обучения детей идеям компьютинга, когда не возникает необходимости в использовании самих компьютеров (см. [6]).

Революция в компьютинге влияет на всех нас. Как выяснилось, информация и вычисление для многих областей являются фундаментальными процессами. Компьютинг больше не рассматривается в узком смысле как наука исключительно об искусственном, а трактуется расширительно, а его предметом служат информационные процессы вообще, вне зависимости от того, вызваны они естественными или искусственными причинами. Компьютинг продолжает развиваться, при этом обнаруживаются его проявления во все новых и новых областях, а в этом смысле различия между отдельными областями представляются все меньше и меньше. Интерес к играм в компьютинге не случаен, поскольку именно игры могут рассматриваться как модели больших и сложных адаптивных систем, которые работают без остановки. Таким образом, компьютинг представляет собой бесконечно продолжающуюся игру. Развивающаяся революция поможет преодолеть кризисную ситуацию, сложившуюся в образовании, а также сломать почти укоренившееся представление о завершении развития компьютерных наук. Этот кризис способствует проникновению в сферу образования различных инноваций, что в свою очередь способствует формированию ярких личностей.

2 Передача знаний

Обычная схема передачи знаний – от старшего поколения к младшему, – перестает работать, как только дело касается информационных технологий. Оба субъекта – передающий знания и получающий их, – находятся в “информационном поле”, которое быстро трансформируется как в качественном, так и в количественном отношении. Понятийная система и ее структура не успевают стабилизироваться настолько, чтобы речь можно было вести о постепенном наращивании мастерства владения тонкими деталями профессии. Возникают новые понятия, которые надо усвоить, они обрастают структурой зависимостей, требующего быстрого осмысления. Сложившиеся понятия меняют свое содержание и/или интерпретацию, имеющиеся между ними структурные зависимости утрачивают свой прежний смысл и приобретают новый.

Темп изменений настолько высок, что и передающий, и получающий знания оказываются в положении ученика, которому требуется быстро усвоить урок, преподносимый реальностью, а преимущество оказывается не столько у того, кто глубже проникает в суть дела, а чаще у того, кто быстрее отбрасывает прежние представления и принимает новые представления, проявляя готовность именно к переменам. Сама динамика изменения понятий становится предметом анализа, а устоявшиеся понятийные структуры существуют на правах инвариантов.

Константность понятий утрачивает свое доминирующее положение, ей на смену приходит изменчивость понятий. Выражение динамики понятия требует не только адекватного языка, на котором это можно сделать, но и доведения это-

го языка до массы потребителей – тех, кто приобретает знания. Этот новый язык находится в стадии становления, но уже начинает претендовать на ту универсальную роль, которую до недавних пор играла математика в ее классическом понимании.

Для выражения этой идеи приходится применять некоторый запас метаматематических средств.

Константное понятие можно было бы выразить, постулируя

$$(\eta) \qquad \lambda x.Xx = X, \text{ где } x \notin X.$$

Еще лучше было бы записать выражение "константной функции"

$$(\lambda x.c)a \stackrel{\beta}{=} c$$

для $x \notin c$ и произвольного a. Записывая

$$(\lambda y.yx)F \stackrel{\beta}{=} Fx,$$

получаем выражение "переменной функции". Тогда по правилу объемности (ξ) записываем

$$\frac{(\lambda y.yx)F \stackrel{\beta}{=} Fx}{\lambda x.(\lambda y.yx)F \stackrel{\beta}{=} \lambda x.Fx},$$

что вновь дает формулировку постулата экстенсиональности (η).

Правда, не все сочтут приведенной чисто синтаксическое рассмотрение удовлетворительным. Выражение же идеи 'быть концептом', то есть 'быть понятием' обычно требует привлечения средств семантического моделирования и *соотнесений*.

Теория константных концептов. Построим набросок теории константных концептов, как того требуют семантические соображения.

Поскольку для концепта C выполняется

$$\|\lambda x.Cx\| : \text{ среда } \to (\text{ индивид } \to \{true, false\}),$$

то

$$\begin{aligned} \|\lambda x.Cx\|\rho d &= \Lambda(\varepsilon \circ \langle \|C\|, \|x\| \rangle)\rho d \\ &= \varepsilon[\|C\|[\rho, d], \|x\|[\rho, d]] \\ &= \varepsilon[C, d] \\ &= C(d). \end{aligned}$$

На языке множеств это значит

$$\|\lambda x.Cx\|\rho \subseteq C \in [C],$$

а для соотнесений $\rho \in I$ на языке типов получается

$$\|\lambda x.Cx\| : I \to C \to [\,] \text{ или } \|\lambda x.Cx\| : I \to [C].$$

Отсюда получаем, что

$$\begin{aligned}\|\lambda x.Cx\| &= \Lambda\|Cx\| : I \to C \to [\,] \\ &= \|Cx\| : (I \times C) \to [\,] \\ &= \|Cx\| : [I \times C].\end{aligned}$$

Теперь, поскольку $\lambda x.Cx = C$, можно записать

$$\|\lambda x.Cx\| = \|C\| : I \to C \to [\,].$$

Теория переменных концептов. Для получения теории переменных концептов воспользуемся результатами, полученными для константных концептов с тем отличием, что C будем ассоциировать с параметром, который обозначим через T. Эта параметризация ведет к множествам

$$H_T(I) = \{h|I \to T\}.$$

Только что рассмотренные соотношения перепишем с использованием T. На языке множеств это значит

$$\|\lambda x.Cx\|\rho \subseteq T \in [T],$$

а для соотнесений $\rho \in I$ на языке типов получается

$$\|\lambda x.Cx\| : I \to T \to [\,] \text{ или } \|\lambda x.Cx\| : I \to [T].$$

Отсюда получаем, что

$$\begin{aligned}\|\lambda x.Cx\| &= \Lambda\|Cx\| : I \to T \to [\,] \\ &= \|Cx\| : (I \times T) \to [\,] \\ &= \|Cx\| : [I \times T].\end{aligned}$$

Теперь, поскольку $\lambda x.Cx = C$, можно записать

$$\|\lambda x.Cx\| = \|C\| : I \to T \to [\,].$$

Заключение

Целью настоящей работы было вызвать интерес ко всему кругу вопросов, сопровождающих возникновение, становление и развитие компьютинга. Дальнейшее вхождение в проблематику можно построить и осуществить по-разному.

Для профессионального изучения специальной математики, включая строение среды вычислений, аппликативные вычислительные системы, семантику языков программирования могут понадобиться источники [13], [14], [15], [16], [17], [18].

Современное введение в разработку специальной математики, способной выполнять непосредственный учет динамики предметной области, а также взаимодействие информационных процессов можно изучить по источникам [19], [20], [21], [22], [23], [24], [25], [26], [27], [28], [29].

Благодарности

Работа является обобщением ряда результатов проектов 10-07-00417-а, 11-07-00305-а, 12-07-00554-а, частично поддержанных грантами РФФИ.

Список литературы

1. Denning P. J. *Great principles of computing.* – Commun. ACM, Vol. 46, №11, 2003. – pp. 15-20.
2. Denning P. J. *Is computer science science?* – Commun. ACM, Vol. 48, №4, 2005. – pp. 27-31.
3. Denning P. J. *Computing is a natural science.* – Commun. ACM, Vol. 50, №7, 2007. – pp. 13–18.
   DOI http://doi.acm.org/10.1145/1272516.1272529
4. Wolfram S. *A New Kind of Science.* – Wolfram Media, Inc., May 14, 2002. ISBN 1-57955-008-8
5. *Great Principles of Computing.* – http://cs.gmu.edu/cne/pjd/GP.
6. Bell T., Whitten I., Powell M. *Computer science unplugged*.
   http://www.unplugged.canterbury.ac.nz.
7. Friedman T. L. *The World Is Flat: A Brief History of the Twenty-first Century.* – Farrar, Straus and Giroux, 2005. – 488 p.
   Русский перевод: Фридман Томас *Плоский мир. Краткая история XXI века.* М.: АСТ, 2006. – 608 с.
8. Shneiderman B. *Web science: a provocative invitation to computer science.* – Commun. ACM, Vol. 50, №6, 2007. – pp. 25-27.
   DOI http://doi.acm.org/10.1145/1247001.1247022
9. Berners-Lee T., Hall W., Hendler J., O'Hara K., Shadbolt N., and Weitzner D. *A framework for Web science.* – Foundations and Trends in Web Science, Vol. 1, Issue 1, 2006. – pp. 1-130.
   http://www.nowpublishers.com/web/
10. King W., Marks P., McCoy S. *The most important issues in knowledge management.* – Commun. ACM, Vol. 45, №9, 2002. – pp. 93-97.
11. Marks P., Polak P., McCoy S., Galletta D. *Sharing knowledge.* – Commun. ACM, Vol. 51, №2, 2008. – pp. 60-65.
    DOI http://doi.acm.org/10.1145/1314215.1314226
12. Shih E., Kraemer K. L., and Dedrick J. *IT diffusion in developing countries.* – Commun. ACM Vol. 51, №2, Feb. 2008. – pp. 43-48.
    DOI http://doi.acm.org/10.1145/1314215.1314223
13. Backus J. *Can programming be liberated from the von Neumann style? A functional style and its algebra of programs.* – Communications of the ACM, Vol. 21, 1978. – pp. 613-641.
14. Barendregt H., Bunder M., Dekkers W. *Systems of Illative Combinatory Logic Complete for First-Order Propositional and Predicate Calculus.* – J. Symbolic Logic Volume 58, Issue 3, 1993. – pp. 769-788.
    http://projecteuclid.org/DPubS?service=UI&version=1.0&verb=Display&handle=euclid.jsl/1183744297
15. de Bruijn N. G. *Lambda calculus notation with nameless dummies, a tool for automatic formula manipulation, with application to the Church-Rosser theorem.* – Indagationes Mathematicae, Vol. 34, 1972. – pp. 381-392

16. Hindley J. R., Lercher B., Seldin J. P. *Introduction to Combinatory Logic.* – London: Cambridge University Press, 1972.
17. Landin P. J. *The next 700 programming languages.* – Communications of the ACM, Vol. 9, 1966. – pp. 157-166.
18. Whitehead A. N., Russell B. *Principia Mathematica* (3 vols). – Cambridge University Press, 1910.
19. Wolfengagen V. E. *Computational Model for Data/Metadata Objects with Transition Effects.* – Proceedings of the 5th International Workshop on Computer Science and Information Technologies CSIT'2003. – Vol. 1, Ufa, Russia, 2003
20. Wolfengagen V. E. *Environment with State Transitions forWeb Information Systems: Case Study.* – Proceedings of the 6th International Workshop on Computer Science and Information Technologies CSIT'2004. – Vol. 1, Budapest, Hungary, 2004
21. Wolfengagen V. E. *Environment with State Transitions for Web Information Systems: Case Study.* – Proceedings of The Sixth International Conference on Information Integration and Web Based Applications & Services (iiWAS2004), 27-29 September 2004, Jakarta, Indonesia, 2004. http://www.iiwas.org/Proceedings/iiWAS2004.pdf
22. Wolfengagen V. E. *Applicative computing. Its quarks, atoms and molecules.* – L. Yu. Isnmailova (Ed.) – Moscow: "Center JurInfoR", 2010. – 62 p.
23. Вольфенгаген В. Э. *Конструкции языков программирования. Приемы описания.* – М.: АО ≪Центр ЮрИнфоР≫, 2001. – 276 с.
Издание поддержано грантом РФФИ, проект 01-01-14068-д.
24. Вольфенгаген В. Э. *Методы и средства вычислений с объектами. Аппликативные вычислительные системы*. – М.: JurInfoR Ltd., АО "Центр ЮрИнфоР", 2004. – xvi+789 с.
Издание поддержано грантом РФФИ, проект 03-01-14055-д.
⋆ ⋆ ⋆ Книга отмечена дипломом Фонда развития отечественного образования на конкурсе 2005 г.
25. Вольфенгаген В. Э. *Логика. Конспект лекций: техника рассуждений.*– М.: АО "Центр ЮрИнфоР", 2001. – 137 с.; 2-е изд., дополн. и перераб. – М: АО "Центр ЮрИнфоР", 2004. – 229 с.
⋆ ⋆ ⋆ Книга стала победителем в номинации "Лучшее учебное издание по точным наукам" на III Общероссийском конкурсе учебных изданий для высших учебных заведений "Университетская книга 2006"
26. Вольфенгаген В. Э. *Аппликативный компьютинг: попытки установить природу вычислений*. – Сборник научно-популярных статей – победителей конкурса РФФИ 2006 года. Выпуск 10. – Под ред. чл.-корр. РАН В.И. Конова. – М.: Октопус/Природа, 2007. – с. 445-459 http://www.rfbr.ru/default.asp?doc_id=28319
27. Вольфенгаген В. Э. *Комбинаторы: объекты, помогающие понять строение компьютинга*. – Сборник научно-популярных статей – победителей конкурса РФФИ 2007 года. Выпуск 11 / Под редакцией чл.-корр. РАН В.И. Конова. – М.: Издательство "Октопус", 2008. – с. 365-378. http://www.rfbr.ru/default.asp?doc_id=29492
28. Вольфенгаген В. Э. *Объекты в программировании: тенденции аппликативного подхода к вычислениям.* / Всероссийский конкурсный отбор обзорно-аналитических статей по приоритетному направлению "Информационно-телекоммуникационные системы", 2008. – 36 с. http://www.ict.edu.ru/lib/index.php?id_res=5641
29. Вольфенгаген В. Э. *Аппликативные вычислительные технологии. Готовые решения для инженера, преподавателя, аспиранта, студента*. / Под. ред. к.т.н. Л.Ю. Исмаиловой. – М.: ЗАО ЮрИнфоР, 2009. 64 с.

Указатель авторов

Fomitcheva O.E. 19

Kulyanitsa A.L. 19

Shumsky L.D. 267, 277
Slobodskoy E.G. 237

Zykov S.V. 8

Балакирев А.А. 33
Барзиков К.В. 244, 247
Баронов О.Р. 79
Бельтюков А.П. 164
Берестнева О.Г. 207
Бохуленков А.С. 173
Брежнева А.Н. 61

Вольфенгаген В.Э. 1, 283
Воронков Э.Н. 58

Девятых Д.В. 207

Егармин К.Н. 58

Журавлева О.О. 45, 55

Зыков С.В. 180

Исмаилова Л.Ю. 45, 136, 150

Климаков А.В. 33
Ковтун И.И. 76
Косиков С.В. 136, 150
Куляница А.Л. 40

Мажирин И.В. 183
Мамедов Н.В. 33
Маслов С.Г. 164, 167
Матисон Н.Л. 234

Навроцкий В.В. 195
Невский А.Ю. 65, 70
Непейвода Н.Н. 103
Нестерович С.А. 82

Парубец В.В. 207
Парфёнова И.А. 213
Петров А.Е. 129

Рословцев В.В. 218, 224, 258

Севальнев В.В. 55
Сидорова Е.Б. 101
Симонов В.Н. 234
Сметанина О.Н.
Степанов А.Л. 33
Степанова Е.Б. 170

Ташани Н.А. 88, 94
Тиренни А. 170
Томакова Р.А. 61

Фомичева О.Е. 40, 244, 247,

Цветков А.И. 250
Цициашвили Г.Ш. 131

Шабунин А.Л. 134
Шабунин Л.В. 101, 134
Шапкин П.А. 218
Шумский Л.Д. 224, 258

Юсупова Н.И. 24

Ясинецкий С.П. 24

Приложение. Видеокурс “Комбинаторная логика” (13 лекций)

Автор: проф. Вольфенгаген В.Э.

Содержание видеокурса
(см. URL http://www.jurinfor.ru/educ/dvdcl.php)

Часть I. Аппликативный компьютинг

1. Лямбда-конверсии • Запись функции с использованием оператора функциональной абстракции. • Строение формальной системы лямбда-конверсий. • Объекты. • Интерпретация объектов лямбда-исчисления. • Свободные и связанные переменные. • Нормальные формы. • Редукция. Конверсии. • Подстановка. • Постулаты формальной системы лямбда-конверсий. • Метаоператоры аппликации и абстракции.

2. Комбинаторы • Выражение общих законов комбинаторами. • Конверсии и строение формальной системы комбинаторной логики. • Синтез объекта с заданными свойствами. • Представление абстракции. • Интерпретация объектов комбинаторной логики. • Парадоксальный комбинатор Карри Y. • Неподвижная точка. Теорема о неподвижной точке.

3. Базисы • Метод погружения. • Базис I, K, S и алгоритм разложения в базисе. • Свойство базисности. • Базис I, B, C, S и алгоритм разложения в базисе.

4. Нумералы • Представление нумералов. • Свойства нумералов. • Функция “следование за”. • Вычисления с нумералами. • Вычисления с неподвижной точкой.

5. Встроенные системы • Погружение и встроенные вычислительные системы. • Осуществление погружения. Большой пример. • Содержательная идея, предформализация, формализация. • Аксиоматизация встроенной системы. • Пример формулировки теоремы о погружении. • Доказательство теоремы о погружении разбором случаев. • Концепт-теория и индивид-теория.

Часть II. Компьютинг в декартово замкнутой категории (д.з.к.)

6. Вычисления в категории • Представление о категориальной абстрактной машине (КАМ). Вычисление значения в д.з.к. • Оценивающее отображение. Среда. Примеры вычисления значения выражений. • Коллизии переменных. Устранение коллизий. Кодирование по Дебрейну. Числа Дебрейна. • Вычисление значения в д.з.к. с учетом кодирования по Дебрейну. •

Комбинаторный “клей”. • Формулировки теорий вычисления и обоснование их свойств.

7. Значение выражений: теория вычислений в категории • Значение выражений и техника вычисления значений. • Среда вычисления значений и ее представление. • Теория вычисления значений по Дебрейну. • Синтаксическая теория вычислений. • Различные формы вычисления значения выражений.

8. Значение выражений: способы вычисления в категории • Способы вычислений. • Исполнение скомпилированного кода. • Применение сборки кода. • Сравнение способов вычислений.

9. Конструирование в категории абстрактной машины (АМ) • Представление о конструировании абстрактной машины. • Работа абстрактной машины. Состояния. • Цикл работы абстрактной машины. • Примеры вычислений. Компилирование кода и его исполнение.

10. Цикл работы абстрактной машины • Описание всевозможных переходов состояний. • Пример вычисление значения 2-местного предиката. • Компилирование кода. • Исполнение кода.

11. Оптимизация вычислений в категории • Вычисление на абстрактной машине свертывания по постулату (β). • Компилирование кода и его эквивалентные преобразования. • Экономии кодирования и обоснование вычисления β-свертывания. • Принцип оптимизации (Beta) и его вывод. • Оптимизация и экономия на примере вычисления значения 2-местного оператора.

12. Расширение и реализация абстрактной машины • Неподвижная точка в вычислениях и инструкция ветвления. • Кодогенерация для выражения с неподвижной точкой. • Рекурсивная модификация среды (р.м.с.). • Пример вычисления с р.м.с. • Большой пример. Выполнение кодогенерации с оптимизацией.

13. Исполнение кода с рекурсивной модификацией среды на абстрактной машине • Большой пример. Вспомогательные обозначения для упрощения кода. • Большой пример. Исполнение кода с р.м.с. • Анализ цикличности в вычислениях. Параметры цикла.