
Научно-исследовательский институт
«Центрпрограммсистем»

Программные продукты и системы

МЕЖДУНАРОДНЫЙ НАУЧНО-ПРАКТИЧЕСКИЙ ЖУРНАЛ

2016, том 29, № 4
(год издания двадцать девятый)

Главный редактор
С.В. ЕМЕЛЬЯНОВ, академик РАН

Тверь

SOFTWARE & SYSTEMS

(PROGRAMMNYE PRODUKTY I SISTEMY)

International research and practice journal

2016, vol. 29, no. 4

Editor-in-Chief
S.V. EMEL'YANOV, Academician of the Russian Academy of Sciences

Tver
Russian Federation

Research Institute CENTERPROGRAMSYSTEM

© ПРОГРАММНЫЕ ПРОДУКТЫ И СИСТЕМЫ

Международный научно-практический журнал

2016, т. 29, № 4,
DOI: 10.15827/0236-235X.116

Главный редактор

С.В. ЕМЕЛЬЯНОВ,
академик РАН (г. Москва, Россия)

Научные редакторы:

Н.А. СЕМЕНОВ, д.т.н., профессор ТвГТУ
(г. Тверь, Россия)
В.Н. РЕШЕТНИКОВ, д.ф.-м.н., профессор, МАИ
(г. Москва, Россия)
А.Н. СОТНИКОВ, д.ф.-м.н., профессор МСЦ РАН
(г. Москва, Россия)

Издатель НИИ «Центрпрограммсистем»
(г. Тверь, Россия)

Учредители: МНИИПУ (г. Москва, Россия),
Главная редакция международного журнала
«Проблемы теории и практики управления» (г. Москва, Россия),
Закрытое акционерное общество
«Научно-исследовательский институт
«Центрпрограммсистем» (г. Тверь, Россия)
Журнал зарегистрирован в Комитете Российской Федерации
по печати 26 июня 1995 г.

Регистрационное свидетельство № 013831

Подписной индекс в каталоге
Агентства «Роспечать» **70799**

ISSN 0236-235X (печатн.)

ISSN 2311-2735 (онлайн)

МЕЖДУНАРОДНАЯ РЕДАКЦИОННАЯ КОЛЛЕГИЯ

Семенов Н.А. – д.т.н., профессор Тверского государственного технического университета, заместитель главного редактора (г. Тверь, Россия)
Решетников В.Н. – д.ф.-м.н., профессор Московского авиационного института (национального исследовательского университета), заместитель главного редактора (г. Москва, Россия)
Арефьев И.Б. – д.т.н., профессор Морской академии Польши (г. Щецин, Польша)
Афанасьев А.П. – д.ф.-м.н., профессор Московского физико-технического института (технического университета), заведующий Центром распределенных вычислений Института проблем передачи информации РАН (г. Москва, Россия)
Баламетов А.Б. – д.т.н., профессор Азербайджанского научно-исследовательского и проектно-изыскательского института энергетики (г. Баку, Азербайджан)
Батыршин И.З. – д.т.н., профессор Мексиканского института нефти (г. Мехико, Мексика)
Вагин В.Н. – д.т.н., профессор Московского энергетического института (технического университета) (г. Москва, Россия)
Голенков В.В. – д.т.н., профессор Белорусского государственного университета информатики и радиоэлектроники (г. Минск, Беларусь)
Еремеев А.П. – д.т.н., профессор Московского энергетического института (технического университета) (г. Москва, Россия)
Котов А.С. – кандидат наук, ассистент профессора университета Уэйна (штат Мичиган) (г. Детройт, США)
Кузнецов О.П. – д.т.н., профессор Института проблем управления РАН (г. Москва, Россия)
Курейчик В.М. – д.т.н., профессор Инженерно-технологической академии Южного федерального университета (г. Таганрог, Россия)
Лисецкий Ю.М. – к.т.н., генеральный директор «S&T Ukraine» (г. Киев, Украина)
Мамросенко К.А. – к.т.н., доцент Московского авиационного института (национального исследовательского университета), руководитель Центра визуализации и спутниковых информационных технологий НИИСИ РАН (г. Москва, Россия)
Мейер Б. – доктор наук, профессор, заведующий кафедрой Высшей политехнической школы – ЕТН (г. Цюрих, Швейцария)
Нузен Тхань Нгу – д.ф.-м.н., профессор, проректор Ханойского открытого университета (г. Ханой, Вьетнам)
Николов Р.В. – доктор наук, профессор Университета библиотековедения и информационных технологий Софии (г. София, Болгария)
Осипов Г.С. – д.ф.-м.н., профессор, заместитель директора Института системного анализа РАН (г. Москва, Россия)
Палюх Б.В. – д.т.н., профессор Тверского государственного технического университета (г. Тверь, Россия)
Рахманов А.А. – д.т.н., профессор, заместитель генерального директора Концерна «РТИ Системы» (г. Москва, Россия)
Серов В.С. – д.ф.-м.н., профессор Университета прикладных наук Оулу (г. Оулу, Финляндия)
Сотников А.Н. – д.ф.-м.н., профессор, Межведомственный суперкомпьютерный центр РАН (г. Москва, Россия)
Сулейманов Д.Ш. – академик АН Республики Татарстан, д.т.н., профессор Казанского государственного технического университета (г. Казань, Республика Татарстан, Россия)
Тарасов В.Б. – к.т.н., доцент Московского государственного технического университета им. Н.Э. Баумана (г. Москва, Россия)
Таратухин В.В. – доктор философии, управляющий директор Европейского исследовательского центра в области информационных систем (ERCIS) Вестфальского университета им. Вильгельма (г. Мюнстер, Германия)
Хорошевский В.Ф. – д.т.н., профессор Московского физико-технического института (технического университета) (г. Москва, Россия)
Язенин А.В. – д.ф.-м.н., профессор Тверского государственного университета (г. Тверь, Россия)

АССОЦИИРОВАННЫЕ ЧЛЕНЫ РЕДАКЦИИ

Московский энергетический институт (технический университет), г. Москва, Россия
Технологический институт Южного федерального университета, г. Таганрог, Россия
Тверской государственный технический университет, г. Тверь, Россия
Научно-исследовательский институт «Центрпрограммсистем», г. Тверь, Россия

АДРЕС РЕДАКЦИИ

Россия, 170024, г. Тверь, пр. 50 лет Октября, 3а
Телефон (482-2) 39-91-49
Факс (482-2) 39-91-00
E-mail: red@cps.tver.ru
www.swsys.ru

Подписано в печать 23.11.2016 г.
Отпечатано ООО ИПП «Фактор и К»
Россия, 170028, г. Тверь, ул. Лукина, д. 4, стр. 1
Выпускается один раз в квартал. Общее количество выпусков 116.
Год издания двадцать девятый. Формат 60×84 1/8. Объем 196 стр.
Заказ № 58. Тираж 1000 экз. Цена 257,40 руб.

Автор статьи отвечает за подбор, оригинальность и точность приводимого фактического материала.
Авторские гонорары не выплачиваются. При перепечатке материалов ссылка на журнал обязательна.

**© SOFTWARE & SYSTEMS
(PROGRAMMNYE PRODUKTY I SISTEMY)**

International research and practice journal

2016, vol. 29, no. 6,
DOI: 10.15827/0236-235X.116

Editor-in-chief

S.V. Emelyanov, Academician of the Russian Academy of Sciences
(Moscow, Russian Federation)

Science editors:

N.A. Semenov, Dr.Sc. (Engineering), Professor TvSTU
(Tver, Russian Federation)

V.N. Reshetnikov, Dr.Sc. (Physics and Mathematics), Professor MAI
(Moscow, Russian Federation)

A.N. Sotnikov, Dr.Sc. (Physics and Mathematics), Professor JSCC RAS
(Moscow, Russian Federation)

**Publisher Research Institute
CENTERPROGRAMSYSTEM
(Tver, Russian Federation)**

The Founders: International Scientific
and Research Institute for Management Issues
(Moscow, Russian Federation),
the Chief Editorial Board
of International Magazine "Theoretical and practical
issues of management" (Moscow, Russian Federation),
Research Institute CENTERPROGRAMSYSTEM
(Tver, Russian Federation)
The magazine is on record
in Russian committee
on press 26th of June 1995
Registration certificate № 013831
ISSN 0236-235X (print)
ISSN 2311-2735 (online)

INTERNATIONAL EDITORIAL BOARD

Semenov N.A. – Dr.Sc. (Engineering), Professor of Tver State Technical University, Deputy Editor-in-Chief
(Tver, Russian Federation)

Reshetnikov V.N. – Dr.Sc. (Physics and Mathematics), Professor of Moscow Aviation Institute (National Research University),
Deputy Editor-in-Chief (Moscow, Russian Federation)

Arefev I.B. – Dr.Sc. (Engineering), Professor of Poland Szczecin Maritime Academy (Szczecin, Poland)

Afanasyev A.P. – Dr.Sc. (Physics and Mathematics), Professor of Moscow Institute of Physics and Technology,
Head of Centre for Distributed Computing of Institute for Information Transmission Problems (Moscow, Russian Federation)

Balametov A.B. – Azerbaijan Scientific-Research & Design-Prospecting Power Engineering Institute (Baku, Azerbaijan)

Batyrshin I.Z. – Dr.Sc. (Engineering), Professor of Mexican Petroleum Institute (Mexico City, Mexico)

Vagin V.N. – Dr.Sc. (Engineering), Professor of Moscow Power Engineering Institute (Technical University)
(Moscow, Russian Federation)

Golenkov V.V. – Dr.Sc. (Engineering), Professor of Belarusian State University of Informatics and Radioelectronics
(Minsk, Republic of Belarus)

Eremeev A.P. – Dr.Sc. (Engineering), Professor of Moscow Power Engineering Institute (Technical University)
(Moscow, Russian Federation)

Kotov A.S. – Ph.D. (Computer Science), Assistant Professor, Wayne State University (Detroit, MI, USA)

Kuznetsov O.P. – Dr.Sc. (Engineering), Professor of the Institute of Control Sciences of the Russian Academy of Sciences
(Moscow, Russian Federation)

Kureichik V.M. – Dr.Sc. (Engineering), Professor of Academy of Engineering and Technology Southern Federal University
(Taganrog, Russian Federation)

Lisetskiy Yu.M. – Ph.D.Tech.Sc., CEO of "S&T Ukraine" (Kiev, Ukraine)

Mamrosenko K.A. – Ph.D. (Engineering), Associate Professor of Moscow Aviation Institute (National Research University),
Head of Center of Visualization and Satellite Information Technologies SRISA RAS (Moscow, Russian Federation)

Meyer B. – Dr.Sc., Professor, Head of Department in Swiss Federal Institute of Technology in Zurich, ETH
(Zurich, Switzerland)

Nguyen Thanh Nghi – Dr.Sc. (Physics and Mathematics), Professor, Vice-Principal of Hanoi Open University (Hanoi, Vietnam)

Nikolov R.V. – Full Professor of the University of Library Studies and Information Technology (Sofia, Bulgaria)

Osipov G.S. – Dr.Sc. (Physics and Mathematics), Professor, Deputy of the Principal of Institute of Systems Analysis
of the Russian Academy of Sciences (Moscow, Russian Federation)

Palyukh B.V. – Dr.Sc. (Engineering), Professor of Tver State Technical University (Tver, Russian Federation)

Rakhmanov A.A. – Dr.Sc. (Engineering), Professor, Deputy of the CEO of Concern "RTI Systems"
(Moscow, Russian Federation)

Serov V.S. – Dr.Sc. (Physics and Mathematics), Professor of the Oulu University of Applied Sciences (Oulu, Finland)

Sotnikov A.N. – Dr.Sc. (Physics and Mathematics), Professor, Joint Supercomputer Center of the Russian Academy
of Sciences (Moscow, Russian Federation)

Suleimanov D.Sh. – Academician of TAS, Dr.Sc. (Engineering), Professor of Kazan State Technical University
(Kazan, Republic of Tatarstan, Russian Federation)

Tarassov V.B. – Ph.D. (Engineering), Associate Professor of Bauman Moscow State Technical University
(Moscow, Russian Federation)

Taratoukhine V.V. – Ph.D. (Engineering), Dr.Ph., Managing Director of the Competence Centre ERP and ERCIS Lab
Russia of the ERCIS (Muenster, Germany)

Khoroshevsky V.F. – Dr.Sc. (Engineering), Professor of Moscow Institute of Physics and Technology
(Moscow, Russian Federation)

Yazenin A.V. – Dr.Sc. (Physics and Mathematics), Professor of Tver State University (Tver, Russian Federation)

ASSOCIATED EDITORIAL BOARD MEMBERS

Moscow Power Engineering Institute (Technical University), Moscow, Russian Federation

Technology Institute at Southern Federal University, Taganrog, Russian Federation

Tver State Technical University, Tver, Russian Federation

Research Institute CENTERPROGRAMSYSTEM, Tver, Russian Federation

EDITORIAL OFFICE ADDRESS

50 let Oktyabrya Ave. 3a, Tver, 170024, Russian Federation

Phone: (482-2) 39-91-49 Fax: (482-2) 39-91-00

E-mail: red@cps.tver.ru

www.swsys.ru

Passed for printing 23.11.2016

Printed in printing-office "Faktor i K"

Lukina St. 4/1, Tver, 170028, Russian Federation

Published quarterly. 29th year of publication

Format 60×84 1/8. Circulation 1000 copies

Prod. order № 58. Wordage 196 pages. Price 257,40 rub.

Вниманию авторов

Международный журнал «Программные продукты и системы» публикует материалы научного и научно-практического характера по новым информационным технологиям, результаты академических и отраслевых исследований в области использования средств вычислительной техники. Практикуются выпуски тематических номеров по искусственному интеллекту, системам автоматизированного проектирования, по технологиям разработки программных средств и системам защиты, а также специализированные выпуски, посвященные научным исследованиям и разработкам отдельных вузов, НИИ, научных организаций.

Решением Президиума Высшей аттестационной комиссии (ВАК) Министерства образования и науки РФ международный журнал «Программные продукты и системы» внесен в Перечень ведущих рецензируемых научных журналов и изданий, в которых должны быть опубликованы основные научные результаты диссертаций на соискание ученых степеней кандидата и доктора наук.

Информация об опубликованных статьях по установленной форме регулярно предоставляется в систему Российского индекса научного цитирования (РИНЦ), в CrossRef и в другие базы и электронные библиотеки.

Условия публикации

К рассмотрению принимаются ранее нигде не опубликованные материалы, соответствующие тематике журнала (специализация 05.13.XX – Информатика, вычислительная техника и управление) и отвечающие редакционным требованиям.

Работа представляется в электронном виде в формате Word. При обилии сложных формул обязательно наличие статьи и в формате PDF. Формулы должны быть набраны в редакторе формул Word (Microsoft Equation или MathType). Объем статьи вместе с иллюстрациями – не менее 10 000 знаков. Диаграммы, схемы, графики должны быть доступными для редактирования (Word, Visio, Excel). Все иллюстрации для полиграфического воспроизведения представляются в черно-белом варианте. Цветные, тонированные, отсканированные, не подлежащие редактированию средствами Word рисунки и экранные формы следует присылать в хорошем качестве для их дополнительного размещения на сайте журнала в макете статьи с доступом по ссылке. (Публикация материалов с использованием гипертекста, графики, аудио-, видео-, программных средств и др. возможна в электронном издании «Программные продукты, системы и алгоритмы», сайт www.swsys-web.ru.) Заголовок должен быть информативным; сокращения, а также терминологию узкой тематики желательно в нем не использовать. Количество авторов на одну статью – не более 4, количество статей одного автора в номере, включая соавторство, – не более 2. Список литературы, наличие которого обязательно, должен включать не менее 10 пунктов.

Необходимы также содержательная структурированная аннотация (не менее 200 слов), ключевые слова (7–10) и индекс УДК. Название статьи, аннотация и ключевые слова должны быть переведены на английский язык (машинный перевод недопустим), а фамилии авторов, названия и юридические адреса организаций (если нет официального перевода), пристатейные списки литературы – транслитерированы по стандарту BGN/PCGN.

Вместе со статьей следует прислать сопроводительное письмо-рекомендацию в произвольной форме, экспертное заключение, лицензионное соглашение, а также сведения об авторах: фамилия, имя, отчество, название и юридический адрес организации, должность, ученые степень и звание (если есть), контактный телефон, электронный адрес, почтовый адрес для отправки бесплатного авторского экземпляра журнала.

Порядок рецензирования

Все статьи, поступающие в редакцию (соответствующие тематике и оформленные согласно требованиям к публикации), подлежат обязательному рецензированию в течение месяца с момента поступления.

В редакции есть устоявшийся коллектив рецензентов, среди которых члены международной редколлегии журнала, эксперты из числа крупных специалистов в области информатики и вычислительной техники ведущих вузов страны, а также ученые и специалисты НИИ «Центрпрограммсистем» (г. Тверь).

Рецензирование проводится конфиденциально. Автору статьи предоставляется возможность ознакомиться с текстом рецензии. При необходимости статья отправляется на доработку.

Рецензии обсуждаются на заседаниях рабочей группы, состоящей из членов научного совета журнала. Заседания проводятся раз в месяц в НИИ «Центрпрограммсистем» (г. Тверь), где принимается решение о целесообразности публикации статьи.

Статьи, одобренные редакционным советом, публикуются бесплатно в течение года с момента одобрения, а отправленные на доработку – с момента поступления после устранения замечаний.

Редакция международного журнала «Программные продукты и системы» в своей работе руководствуется сводом правил Кодекса этики научных публикаций, разработанным и утвержденным Комитетом по этике научных публикаций.

УДК 004.4: [658.011.56+338.534]:334.726
DOI: 10.15827/0236-235X.116.005-014

Дата подачи статьи: 10.02.16
2016. Т. 29. № 4. С. 5–14

ПРОБЛЕМЫ И ПЕРСПЕКТИВЫ СОЗДАНИЯ APS ДЛЯ МУЛЬТИНАЦИОНАЛЬНЫХ КОМПАНИЙ

А.А. Сухобоков, к.т.н., artem.sukhobokov@yandex.ru

(Компания «САП СНГ», Космодамианская наб., 52/2, г. Москва, 115054, Россия);

А.В. Сухобоков, предприниматель, andrey.sukhobokov@optimalmngmnt.com

(Компания «Оптимальное Управление», ул. Фестивальная, 22, корп. 6, г. Москва, 125581, Россия);

Д.С. Лахвич, аспирант, dmitry.lakhvich@optimalmngmnt.com;

И.В. Тихонов, аспирант, ilya.tikhonov@optimalmngmnt.com

(Московский государственный технический университет им. Н.Э. Баумана,
ул. 2-я Бауманская, 5, г. Москва, 105005, Россия)

Приведена классическая функциональная архитектура систем синхронного планирования (APS). Показано, что применяемые в настоящее время на этапе сводного планирования математические модели и методы линейного математического программирования не позволяют решить ключевую для мультинациональных компаний задачу оптимизации прибыли компании в целом. Рассмотрена история развития математических моделей и методов оптимизации для решения задачи оптимизации прибыли после налогообложения в мультинациональных компаниях. Развитие математических моделей показало, что для оптимизации внутренних цепочек поставок больших мультинациональных компаний необходимо решать задачи оптимизации очень большой размерности, содержащие десятки миллиардов столбцов и строк с билинейными или трилинейными ограничениями. Рассмотрены подходы к распараллеливанию численных алгоритмов решения таких задач. В качестве наиболее перспективного выделен подход, предполагающий использование платформ Big Data для решения задач большой размерности. Определена дополнительная функциональность, которой должны обладать модуль проектирования цепочек поставок, а также модуль планирования спроса, чтобы APS мог эффективно использоваться в мультинациональных компаниях. Определена дополнительная функциональность, которой должны обладать модули управления финансовыми цепочками и модули контроллинга смежной ERP-системы, чтобы APS мог эффективно использоваться в мультинациональных компаниях. Показано, что перспективный APS для мультинациональных компаний должен разрабатываться на основе комплекса из двух технологических платформ – In-Memory и Big Data. Представлена функциональная архитектура APS для мультинациональных компаний в контексте корпоративной системы в целом. Дополнительно на один уровень глубже детализирована архитектура модуля сводного планирования APS, обеспечивающего решение оптимизационной задачи как для одного, так и для конечного числа периодов времени.

Ключевые слова: APS, мультинациональная компания, внутрифирменные цепочки поставок, трансфертные цены, товаропотоки, задача билинейной оптимизации, распараллеливание алгоритмов оптимизации, сводное планирование, архитектура APS, платформы Big Data.

Основным средством оптимизации цепочек поставок являются APS (Advanced Planning System или, в другой трактовке, Advanced Planning & Scheduling systems), поставляемые в составе систем ERP II, SCM-систем или в виде отдельных продуктов. Одна из наиболее уважаемых организаций в области методологии управления предприятиями APICS (American Production and Inventory Control Society) дает определение APS [1], согласно которому эти системы обладают следующим набором свойств:

- осуществляют анализ и планирование логистики и производства на краткосрочные, среднесрочные и долгосрочные периоды;
- используют развитые математические алгоритмы или логику для оптимизации или моделирования при решении следующих задач: составление расписаний загрузки ограниченных мощностей, подбор поставщиков, планирование инвестиций, планирование загрузки ресурсов, прогнозирование, управление потребностями и пр.;
- одновременно учитывают ряд ограничений и бизнес-правил, позволяющих обеспечить планирование и составление расписаний в реальном вре-

мени, поддержку принятия решений, возможности ATP (available-to-promise) и CTP (capable-to-promise);

- часто генерируют и оценивают множество сценариев, один из которых менеджмент затем использует как «официальный план»;

- включают пять основных компонентов: планирование потребностей, планирование производства, составление расписаний производства, планирование дистрибуции, планирование транспортировки.

В основополагающей монографии [2], которая расширяется и пополняется при каждом очередном издании, приводится обобщенная архитектура APS-системы [3], показанная на рисунке 1.

Теоретически APS могут оптимизировать затраты, прибыль, NPV, NCF, риски, гибкость и устойчивость цепочки поставок, уровень обслуживания клиентов [4]. Однако в реально внедренных системах в подавляющем большинстве случаев осуществляется минимизация затрат на закупку, хранение, транспортировку и производство продукции [5]. В [6] показан редкий случай – внедрение APS в компании SSAB Oxelösund, где в каче-



стве целевого показателя максимизируется прибыль. Максимум прибыли достигается за счет используемого метода планирования сроков выполнения работ по заказам клиентов: чем прибыльнее заказ, тем быстрее он исполняется и меньше стоит в очередях, ожидая выполнения других заказов.

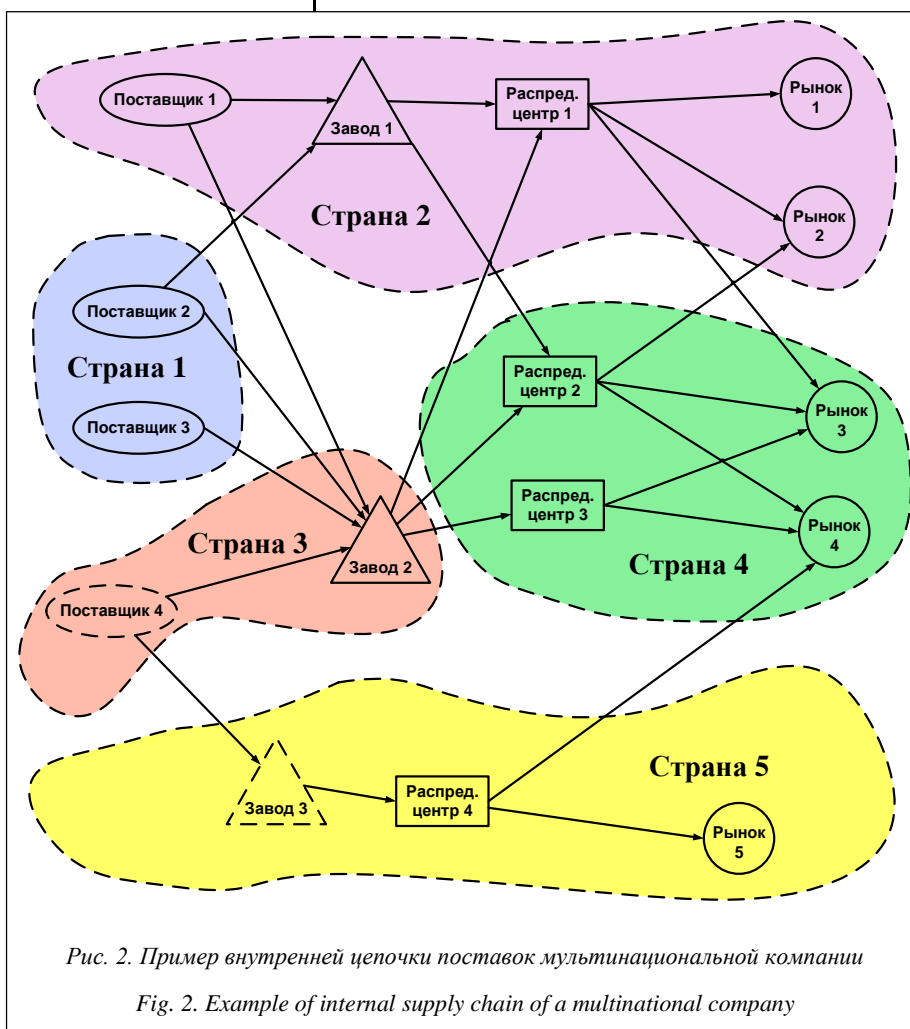
В подавляющем большинстве APS, когда используются строгие методы оптимизации и математические модели, применяются методы линейного программирования и смешанного линейного целочисленного программирования [7]. Существует множество задач, в которых целесообразно применять нелинейные модели и методы нелинейного и смешанного нелинейного целочисленного программирования, но их использование ограничено большой сложностью расчетов и ограниченностью используемых вычислительных ресурсов [7]. Уменьшение сложности используемых математических моделей, использование линейных моделей означают, что оптимизация в этих случаях выполняется по одной группе параметров, например, по объему грузоперевозок, загрузке транспортных мощностей, длине транспортных маршру-

тов. При этом отсутствуют возможности дополнительной оптимизации за счет варьирования цен, следовательно, имеющиеся возможности оптимизации финансовых показателей являются ограниченными.

Методы нелинейного программирования и смешанного нелинейного целочисленного программирования сейчас в основном применяются только в специализированных решениях для непрерывного

производства (в APS для предприятий нефтеперерабатывающей, химической и других подобных отраслей) [8–10].

В мультинациональных компаниях процесс оптимизации цепочек поставок имеет существенные особенности. Для их понимания на рисунке 2 приведен пример структуры такой компании.



После того как с помощью одного из APS строится логистическая схема, при которой достигается минимум затрат на производство и транспортировку в мультинациональной компании, производится финансовая оптимизация найденной схемы. Для этого используются пакет Global Tax Planning [11] от THOMPSON REUTERS, входящий в состав линейки решений ONESOURCE, другие подобные пакеты или внутренние разработки компаний большой четверки (Deloitte, EY, KPMG, PwC). Суть решаемой задачи состоит в том, чтобы, учитывая особенности налогообложения дочерних компаний, работающих в разных странах, и ограничения, накладываемые налоговыми органами этих стран, максимизировать общую прибыль после налогообложения мультинациональной компании в целом. Оптимизация достигается за счет варьирования трансфертных цен (цен на товары между дочерними компаниями) и перераспределения затрат на транспортировку. Для формирования математической модели используются данные по различным видам постоянных и переменных затрат в дочерних компаниях, ставки множества налогов и пошлин в каждой из стран. Решаемая задача, как и большинство задач оптимизации логистики, является задачей линейного программирования, то есть решается последовательность двух линейных задач.

На самом деле такое последовательное применение линейных моделей не позволяет найти истинный максимум прибыли. Он практически никогда не лежит в той области, где достигается минимум затрат. Чтобы найти этот максимум, надо решить общую оптимизационную задачу, в которой одновременно варьируются планы производства, методы транспортировки и трансфертные цены. В тех случаях, когда задача оптимизации логистики формулируется как задача линейного программирования, общая задача оптимизации прибыли мультинациональной компании будет билинейной – в ее ограничениях будут присутствовать произведение цен и товаропотоков между дочерними компаниями. Если же задача оптимизации логистики формулируется как билинейная задача в силу того, что при разных режимах работы оборудования из одного и того же сырья можно получить разное соотношение конечных продуктов (процессы в химии и нефтепереработке), общая задача оптимизации прибыли мультинациональной компании будет трилинейной.

Математические модели и алгоритмы для оптимизации внутренних цепочек поставок мультинациональных компаний

Эволюция развития моделей и алгоритмов. Впервые задача совместной оптимизации цепочек поставок мультинациональной компании за счет варьирования товаропотоков и трансфертных цен была математически сформулирована в 1976 г. в

работе [12]. Для ее решения предложен рекуррентный эвристический алгоритм поиска максимума прибыли, который на каждом шаге изменял набор трансфертных цен и решал линейную задачу поиска оптимальных товаропотоков. Процесс прекращался при остановке роста целевого значения – суммарной общей прибыли компании после налогообложения. Предложенная модель имела ряд ограничений:

- предполагалось, что мультинациональная компания имеет центральный распределительный центр, из которого все продукты транспортируются в дочерние компании;
- в модели отсутствовала возможность определить необходимое количество сырья и комплектующих, используемых при производстве конечных продуктов (Bill of Materials – BOM);
- не учитывалась возможность распределения транспортных расходов: они всегда относились на счет получателя.

В 1989 г. в работе [13] была предложена нелинейная модель смешанного целочисленного динамического программирования, в которой максимизировалась прибыль мультинациональной компании после уплаты налогов. Нелинейность была обусловлена использованием в ней трансфертных цен и переменных, отражающих принятые решения о распределении накладных расходов по заводам. Трансфертная цена каждого продукта формировалась из надбавки и суммы затрат на производство продукции, плюс затраты на транспортировку, включая пошлины. Такой подход является недостатком этой модели, так как в большинстве случаев пошлины зависят от цены перемещаемых товаров, а в работе они рассматриваются как часть стоимости доставки за единицу товара. В качестве алгоритма решения задачи предложен иерархический процесс, на каждой итерации которого решается смешанная задача линейного целочисленного программирования для поиска оптимальных товаропотоков и ищутся оптимальные значения надбавок. Процесс повторяется, пока не будет найден локальный оптимум. Поскольку надбавки не имеют каких-либо ограничений, результирующее решение может приводить к сильному уменьшению налоговых выплат, что, естественно, неприемлемо для налоговых органов.

На базе модели, представленной в [13], в [14] была предложена многопродуктовая модель для одного периода, с помощью которой формализована задача распределения ресурсов для производителей персональных компьютеров. Это первая модель, в которой были введены бинарные переменные для распределения транспортных расходов между поставщиком и потребителем.

В 2001 г. Carlos J. Vidal и Marc Goetschalckx предложили модель управления внутренними цепочками поставок мультинациональной компании, в которой реализована возможность поставки сы-

рья от внешних глобальных поставщиков, учитывается стоимость запасов, предусмотрена возможность распределения транспортных расходов между участниками цепочки поставок [15]. Также предусмотрены различные варианты формирования трансфертных цен (для стран, где по законодательству трансфертные цены формируются исходя из стоимости товаров по контрактам CIF, и для стран, где по законодательству трансфертные цены формируются исходя из стоимости товаров по контрактам FOB). Модель сформулирована как билинейная задача оптимизации с линейной целевой функцией. Для решения задачи авторы предложили эвристику, основная суть которой сводится к решению задачи методом разделения переменных. На каждой итерации сначала фиксируются трансфертные цены и решается задача оптимизации товаропотоков. Потом для найденных значений товаропотоков ищутся оптимальные трансфертные цены. После конечного числа повторений процесс сходится к локальному оптимуму. Поскольку качество локального оптимума сильно зависит от начальной точки, авторы исследовали семь различных вариантов задания начальной точки.

Позднее, в 2013 г., эти же авторы в [16] предложили эвристический алгоритм для решения задачи, сформулированной в [15], суть которого в том, что сначала точным методом решается линейная задача, представляющая собой упрощение первоначальной билинейной задачи, а затем из найденной точки иницируется поиск точного решения билинейной задачи методом ветвей и границ. Предложенный подход обеспечивает сходимость решения к глобальному оптимуму с любой требуемой точностью.

В 2008 г. Tan Miller и Renato de Matta предложили билинейную модель для максимизации глобальной прибыли мультинациональной компании, которая позволяет определить оптимальную стратегию производства для каждого завода, а также оптимальные планы закупок и дистрибуции в сочетании со значениями трансфертных цен [17]. Модель учитывает налоги и курсы обмена валют в каждой стране. Как и в [13], трансфертные цены представляют собой суммы затрат и надбавок, но на размер надбавок наложены ограничения. Особенность этой модели в том, что единицей планирования является страна, а не отдельный завод или центр дистрибуции. Для стран с большими расстояниями, таких как Россия, Китай, США, Канада, учет стоимости транспортировки без привязки к фактическому местоположению завода приводит к очень большим погрешностям. В связи с отсутствием эффективных методов решения билинейных задач большой размерности, чтобы применять модель на практике, авторы предложили несколько допущений и упростили разработанную модель до линейной.

Эти же авторы в 2015 г. предложили решать сформулированную билинейную задачу методом декомпозиции Бендерса, не изменив ключевую особенность детализации своей модели до уровня страны [18].

В 2008 г. S. Perron, P. Hansen, S. Le Digabel и N. Mladenovic предложили в [19] новую формулировку математической модели, представленной в [15]. Они сформулировали модель только для стран, где по законодательству цены формируются исходя из стоимости товаров по контрактам CIF, но предложенный ими подход позволяет примерно на 65 % сократить число произведений переменных в ограничениях, что дает возможность существенно сократить время решения и работать с моделями большей размерности. Также в работе была предложена линейная модель, являющаяся упрощением билинейной. Для решения билинейной задачи в работе были протестированы четыре метода:

- эвристика, предложенная в [15];
- обычный метод локального поиска с чередующимися окрестностями, традиционно применяемый для решения билинейных задач оптимизации (метаэвристика VNS);
- метод локального поиска с чередующимися окрестностями, специально адаптированный для решения задачи оптимизации внутренних цепочек поставок мультинациональных компаний (адаптированная метаэвристика VNS);
- метод ветвей и границ, позволяющий найти строгое решение билинейной оптимизационной задачи.

По результатам тестирования на примерах лучшим оказался специализированный метод локального поиска с чередующимися окрестностями, а точный сильно уступает эвристическим по размерности задач, которые могут быть решены с его помощью.

Представленная в [19] математическая модель была доработана в работе [20] с учетом присутствующих в российской экономике НДС и экспортных пошлин.

В работе [21] высказана идея подхода к построению гибкой математической модели для оптимизации внутренних цепочек поставок, позволяющей учитывать любое необходимое число переделов сырья, возможность продажи и закупки полуфабрикатов у сторонних компаний, наличие произвольного числа эшелонов распределительных центров.

Параллельные алгоритмы. Билинейные задачи, которые требуется решать для оптимизации внутренних цепочек поставок мультинациональных компаний, имеют очень большие размерности. В [21] приведен пример, когда для представления модели необходима матрица, имеющая более 5×10^{10} столбцов и примерно 2×10^{10} строк. Естественным способом решения задач больших размерностей является распараллеливание процесса

решения. Практически для всех рассмотренных ранее в этом разделе эвристических и строгих алгоритмов существуют их параллельные версии. При распараллеливании используются два основных подхода.

Подход 1. Параллельно рассчитывается много вариантов одной и той же задачи с разными исходными данными, а потом результаты этих расчетов сравниваются и выбирается лучший. В качестве примера можно привести распараллеливание метаэвристики поиска с чередующимися окрестностями для решения задач непрерывной и дискретной оптимизации VNS [22], которую в общем виде можно описать так:

- определение последовательности размеров окрестности и начальной точки;
- циклический поиск локального оптимума, начиная с первого заданного размера окрестности и заданной начальной точки; если в результате локального поиска на очередном шаге будет найдено новое лучшее значение оптимума, использовать на следующем шаге найденную точку оптимума в качестве начальной, в противном случае перейти к поиску на следующем размере окрестностей.

Для повышения эффективности поиска глобального оптимума с помощью чередующихся окрестностей было разработано несколько вариантов [22] параллельного алгоритма VNS (PVNS). Наиболее эффективный из них заключался в наращивании числа решений, выбираемых в текущей окрестности, и в параллельном выполнении локального поиска для каждого из них.

Подход 2. Основная задача преобразовывается и декомпозируется на несколько задач меньшей размерности, которые могут решаться параллельно, а их результаты потом объединяются и позволяют найти решение основной задачи. Такой подход используется как при решении задач линейного программирования методами декомпозиции Бендерса или Данцига–Вульфа [23], так и при решении целочисленных задач параллельными методами ветвей и границ или секущих плоскостей [24].

Однако все перечисленные алгоритмы имеют некоторые общие характерные особенности: они ориентированы на распараллеливание не более чем на несколько десятков (иногда сотен) параллельных подпроцессов и требуют интенсивного обмена данными между отдельными подпроцессами в процессе решения задачи. Примером проявления этих свойств является ситуация с распараллеливанием алгоритмов линейной оптимизации в компании SAP [25]. Компания к 2004 г. испытывала потребность в переходе на параллельные алгоритмы решения задач целочисленного линейного программирования из-за имеющей место тенденции роста размерности при оптимизации цепочек поставок. В течение трех лет она проводила исследования по реализации параллельных алгоритмов с помощью

декомпозиции Данцига–Вульфа (выделение блоков в исходных матрицах и параллельное решение задачи для отдельных блоков на разных узлах кластера). Результаты показали, что при увеличении числа параллельно обрабатываемых блоков основной матрицы более, чем до 25–35, дальнейшего увеличения размерности решаемой задачи и скорости решения задачи не происходит.

Появление программных платформ для работы с большими данными и соответствующих вычислительных кластеров предъявляет новые требования к параллельным алгоритмам оптимизации [26]. Программные и аппаратные средства Big Data в своем современном состоянии ориентированы на совершенно другие характеристики кластеров. Число параллельно работающих узлов может составлять десятки и сотни тысяч, но связи между узлами не способны обеспечить интенсивный обмен информацией в процессе обработки данных. Применение кластеров Big Data позволило на порядок снизить стоимость обработки информации по сравнению с суперкомпьютерами, но для эффективного использования таких ресурсов требуются принципиально иные алгоритмы. В [27] предлагается один из вариантов решения билинейной задачи оптимизации на платформе Big Data.

Обсуждение архитектуры будущих APS для оптимизации внутренних цепочек поставок мультинациональных компаний

В первых работах, рассматривавших роль и место системы оптимизации внутренних цепочек поставок мультинациональных компаний в общей корпоративной системе управления [20], предлагалось реализовать отдельный такой модуль и интегрировать его с ERP-системой. В этом случае этот модуль как бы играл роль APS. Однако в процессе обсуждения проблематики потенциальные клиенты выразили заинтересованность не только в решении среднесрочной задачи максимизации прибыли на один период 12–18 месяцев за счет одновременного варьирования планов производства, товаропотоков и трансфертных цен. Не менее важной они посчитали и задачу более детального одновременного планирования производства, товаропотоков и трансфертных цен на несколько (конечное число) периодов. Такая оптимизация позволит учесть прогнозы изменения спроса конечных пользователей, рыночных цен, курсов валют, а также время производства товаров на имеющихся заводах, скорость их транспортировки по различным маршрутам и время обработки в распределительных центрах. Суммарная длительность всех периодов такого варианта планирования может быть несколько меньшей или равной длительности укрупненного среднесрочного планирования на один период.

Как же конкретно необходимо расширить функциональность создаваемых APS и ERP, чтобы успешно использовать методы одновременной оптимизации логистики и трансфертных цен? Все модули APS для мультинациональной компании (в том числе модуль планирования закупок, модуль планирования транспорта, модуль ATP и др.) должны быть реализованы с учетом мультинационального характера оптимизируемых цепочек поставок. Например, при планировании транспорта надо учитывать, что возможно взимание таможенных и других сборов не только за перемещаемые через границу товары, но и за пересечение границы транспортным средством (автомобилем).

Приведем самые важные ключевые требования к APS и другим интегрированным с ним продуктам в составе корпоративной системы мультинациональной компании.

1. В модуле проектирования логистических сетей должна быть предусмотрена возможность описания государств и экономических зон с отличающейся системой налогообложения. В этих описаниях должны быть отражены все виды взимаемых налогов, пошлин и сборов, а также методы расчета их величин. Объекты логистической цепочки (в том числе внешние поставщики и потребители) должны присваиваться одному из описанных государств или выделенной в нем экономической зоне. В связи с тем, что финансовый модуль является одним из центральных компонентов ERP-системы, будет логично, если все необходимые мастер-данные по государствам, экономическим зонам и методам налогообложения будут вводиться там, а в APS будут поступать через интеграционный интерфейс.

2. Функциональность модуля планирования спроса APS должна быть шире, чем просто сбор, прогноз и предоставление данных о предполагаемом спросе. Если продукты производства продаются конечным пользователям по ценам в локальных национальных валютах разных государств, то спрос необходимо прогнозировать в этих национальных валютах. Прогноз динамики курсов валют должен вводиться в управление финансовыми цепочками и быть доступным для разных приложений, в том числе для APS, чтобы оптимизировать показатели мультинациональной компании в целом в валюте корпоративного учета. Еще одним расширением функциональности модуля планирования спроса должны стать функции сбора, прогнозирования и предоставления модулю сводного планирования данных об эластичности спроса – его изменении в зависимости от изменения цен для конечных потребителей. Это позволит планировать объемы производства и сбыта, обеспечивающие максимальную прибыль.

3. В [28] в качестве одного из направлений будущего развития APS указана тесная интеграция APS со средствами контроллинга ERP-системы.

Для APS, обеспечивающего одновременную оптимизацию логистики и трансфертных цен в мультинациональных компаниях, актуальность решения этой проблемы резко повышается. Для определения допустимых значений трансфертных цен методом Cost plus в мультинациональной компании необходима единая система учета затрат, охватывающая все дочерние компании в разных странах, в которой будут учитываться различные нюансы процессов закупки, производства и дистрибуции, в том числе стоимость различных видов приобретаемого сырья и комплектующих, услуги контрагентов, наличие консигнационных и специальных запасов, использование периодических заказов и так далее. В существующих ERP-системах такая система контроллинга сейчас отсутствует. Единственным известным авторам решением такого типа является разрабатываемый компанией IM&C пакет GCP [29, 30], расширяющий возможности SAP ERP. Но он имеет очень ограниченное число внедрений, и компания IM&C расширяет его функциональность, чтобы реализовать оптимизацию трансфертных цен без оптимизации товаропотоков. Для успешной реализации и последующего распространения APS, осуществляющего одновременную оптимизацию трансфертных цен и товаропотоков, необходимо наличие таких компонентов в составе стандартных средств контроллинга ERP-системы.

4. Модули управления финансовыми цепочками обеспечивают корпоративное бюджетирование, управление финансовыми потоками, получение кредитов, взаимодействие с банками, операции на валютных и товарно-сырьевых биржах, на рынках ценных бумаг и деривативов, хеджирование рисков и другие процессы, необходимые для непрерывного функционирования компании. В случае мультинациональной компании взаимное влияние цепочек поставок и финансовых цепочек заметно усиливается. Ставки по кредитам, курсы валют и другие условия влияют на объемы затрат, которые учитываются при планировании цепочек поставок. И, наоборот, реализация тех или иных хозяйственных операций в разных странах влияет на финансовые цепочки. Недостаток синхронизации финансового и бизнес-планирования в APS отмечался еще в [31], однако в существующих системах он так и не устранен. В APS для мультинациональных компаний должен быть реализован механизм синхронизации, позволяющий инициировать перепланирование финансовых цепочек при появлении изменений в цепочках поставок и, наоборот, перепланировать цепочки поставок при появлении изменений в финансовых цепочках.

5. Использование высокопараллельных численных методов на базе технологических платформ Big Data для решения задачи одновременной оптимизации планов производства, товаропотоков и трансфертных цен требует одновременного приме-

нения нескольких технологических платформ, на базе которых будет построен APS для мультинациональных компаний.

В составе модуля сводного планирования такого APS необходимо выделить следующие блоки:

- два блока формирования и работы с математическими моделями (один – для модели с одним периодом времени, другой – с конечным числом периодов времени);
- два блока управления процессом решения оптимизационной задачи (по одному для каждой модели);
- блок численных расчетов.

Блок формирования математической модели в [21] предложено разрабатывать на базе In-Memory платформы SAP HANA. Выбор платформы In-Memory действительно целесообразен, поскольку модель содержит достаточно большой объем данных. Ее надо сформировать за какое-то разумное время, а процесс формирования предпо-

лагает получение большей части данных по интеграционным интерфейсам из ERP-системы мелкими порциями с помощью множества отдельных запросов. Рассчитанный в [21] максимальный объем данных модели составляет 3,2 ЕВ. Эта оценка завышена на несколько порядков из-за того, что для больших моделей использован коэффициент заполненности разреженных матриц, характерный для малых моделей. Тем не менее объем больших моделей может измеряться в терабайтах. Использование платформы In-Memory позволит относительно быстро сформировать данные такого объема. Если в перспективе (как, например, планирует SAP) ERP и APS будут полностью перенесены на платформу In-Memory, оба блока формирования математических моделей для APS мультинациональных компаний не будут требовать реализации на отдельной платформе.

Блоки управления процессом решения оптимизационной задачи осуществляют процесс загрузки

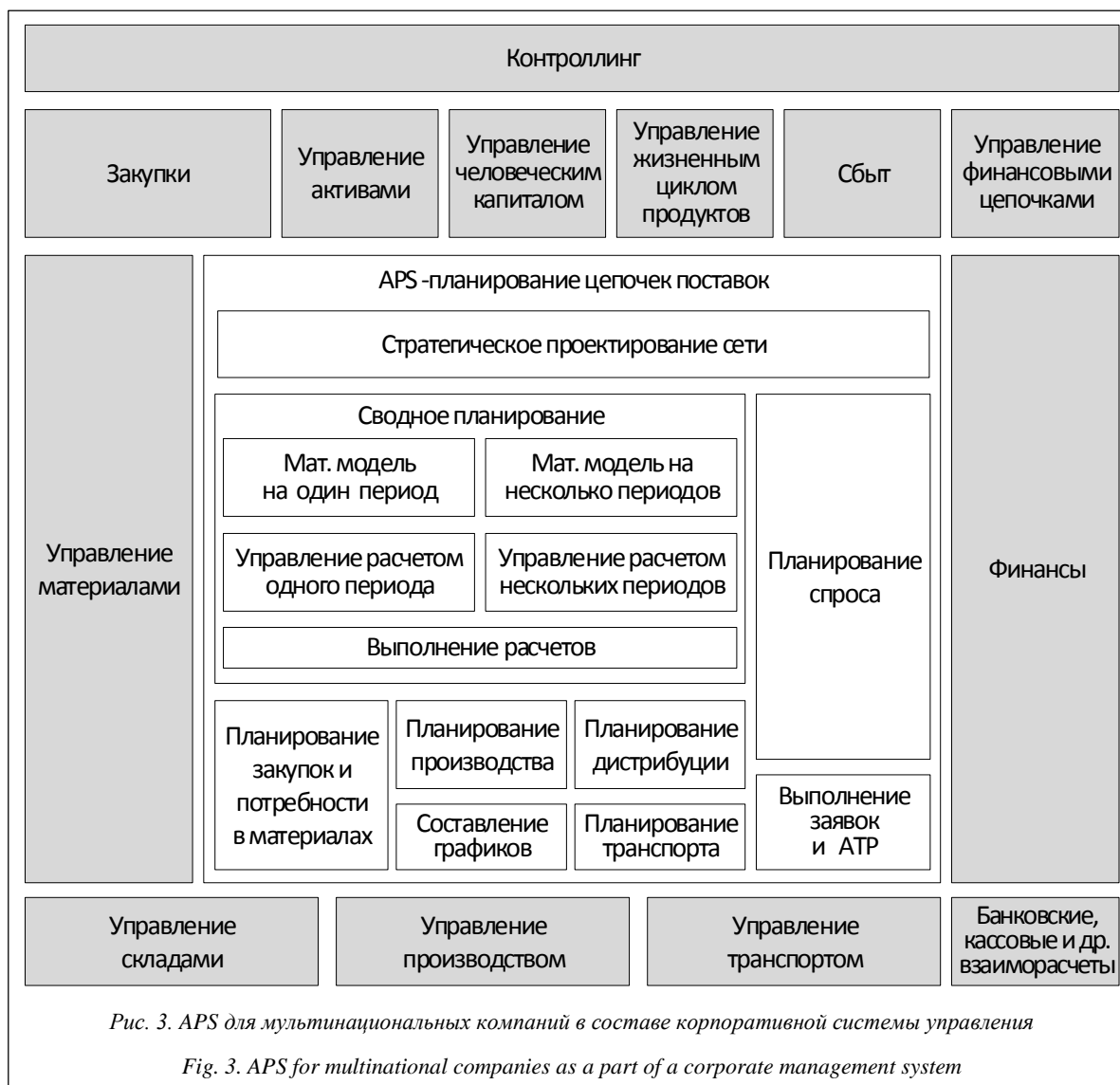


Рис. 3. APS для мультинациональных компаний в составе корпоративной системы управления

Fig. 3. APS for multinational companies as a part of a corporate management system

в Hadoop или Spark одной или последовательности из нескольких математических моделей, интерпретируют результаты выполнения каждого отдельного расчета на платформе Big Data и обеспечивают возврат окончательных итоговых результатов в корпоративную систему управления.

Для компаний, имеющих сложную внутреннюю логистическую структуру, может потребоваться использовать в блоке численных расчетов кластер, содержащий несколько десятков тысяч серверов. Это сразу заставляет при проектировании такого блока ориентироваться на использование облачной инфраструктуры, предоставляемой внешними провайдерами. В менее сложных случаях облако может быть гибридным или частным: часть серверов используемого кластера (или все) могут быть взяты из собственной ИТ-инфраструктуры компании.

Архитектура APS для мультинациональной компании характеризуется большим числом информационных связей между APS и другими модулями корпоративной системы [20]. Чтобы не перегружать концептуальное представление архитектуры, авторы воспользовались минималистским подходом, примененным в [3]. Это позволило представить APS для мультинациональной компании в контексте корпоративной системы в целом и одновременно детализировать модуль сводного планирования (рис. 3). Модули вне APS, структура которых не раскрыта, выделены на рисунке серым цветом.

При проектировании будущих APS для мультинациональных компаний должны учитываться как конкретные требования 1–5, представленные выше, так и требования по функциональному взаимодействию с каждым из смежных модулей корпоративной системы, представленных на рисунке 3. Такой комплексный подход позволит разработать APS, способные существенно повысить эффективность работы мультинациональных компаний.

Число мультинациональных компаний в мире постепенно увеличивается за счет того, что все больше средних и даже мелких компаний из разных стран выходят на зарубежные рынки, создают дочерние компании в других государствах, чтобы теснее взаимодействовать со своими поставщиками и потребителями. Эта тенденция, а также связанное с ней постепенное увеличение роли мультинациональных компаний в глобальной экономике являются сильным стимулом для практической реализации сформулированных требований и воплощения предложенных решений.

Литература

1. Blackstone J.H. Jr. (Ed.), APICS Dictionary. 13th ed., APICS Publ., 2011, 164 p.
2. Supply Chain Management and Advanced Planning. Concepts, Models, Software, and Case Studies. 5th ed. H. Stadtler, C. Kilger, H. Meyr (eds.). Springer, Heidelberg, NY, Dordrecht, London, 2015, 572 p.

3. Meyr H., Wagner M., Rohde J. Structure of advanced planning systems. Supply Chain Management and Advanced Planning. Concepts, Models, Software, and Case Studies. 5th ed. H. Stadtler, C. Kilger, H. Meyr (eds.). Springer, Heidelberg, NY, Dordrecht, London, 2015, pp. 99–106.
4. Fleischmann B., Koberstein A. Strategic network design. Supply Chain Management and Advanced Planning. Concepts, Models, Software, and Case Studies. 5th ed. H. Stadtler, C. Kilger, H. Meyr (eds.). Springer, Heidelberg, NY, Dordrecht, London, 2015, pp. 107–123.
5. Ivert L.K. Use of Advanced Planning and Scheduling (APS) systems to support manufacturing planning and control processes. PhD thesis. Chalmers Univ. of Technology, Göteborg, Sweden, 2012, 111 p. URL: <http://publications.lib.chalmers.se/records/fulltext/162616/162616.pdf> (дата обращения: 27.01.2016).
6. Jonsson P., Kjellsdotter L., Rudberg M. Applying advanced planning systems for supply chain planning: three case studies. Int. Jour. of Physical Distribution & Logistics Management. 2007, vol. 37, iss. 10, pp. 816–834.
7. Kallrath J., Maindl T.I. Real Optimization with SAP® APO. Springer, Berlin, Heidelberg, NY, 2006, 340 p.
8. Lasschuit W., Thijssen N. Supporting supply chain planning and scheduling decisions in the oil and chemical industry. Comp. and Chem. Eng. 2004, no. 28, pp. 863–870. URL: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.470.6859&rep=rep1&type=pdf> (дата обращения: 27.01.2016).
9. Al-Qahtani K.Y., Elkamel A. Planning and Integration of Refinery and Petrochemical Operations. Wiley-VCH Verlag GmbH & Co, Germany, 2010, 209 p.
10. Lebreton B., Meyr H., Rosiè H., Seipl C., Wetterauer U. Architecture of selected APS. Supply Chain Management and Advanced Planning. Concepts, Models, Software, and Case Studies. 5th ed. H. Stadtler, C. Kilger, H. Meyr (eds.). Springer, Heidelberg, NY, Dordrecht, London, 2015, pp. 341–361.
11. Global Tax Planning. 2013, 2 p. URL: <https://tax.thomsonreuters.com/wp-content/pdf/global-tax/GlobalTaxPlanning.pdf> (дата обращения: 27.01.2016).
12. Nieckels L. Transfer Pricing in Multinational Firms: A Heuristic Programming Approach and a Case Study. NY, John Wiley Publ., 1976, 202 p.
13. Cohen M.A., Fisher M., Jaikumar R. International manufacturing and distribution networks: a normative model framework. Managing International Manufacturing. K. Ferdows (Ed.). North-Holland, Amsterdam, 1989, pp. 67–93.
14. Cohen M.A., Lee H.L. Resource deployment analysis of global manufacturing and distribution networks. Jour. of Manufacturing Operations Management. 1989, no. 2, pp. 81–104.
15. Vidal C.J., Goetschalckx M. A global supply chain model with transfer pricing and transportation cost allocation. European Jour. of Operational Research. 2001, no. 129, pp. 134–158.
16. Vidal Holguin C.J., Goetschalckx M. A solution to the transfer pricing problem by successively reducing the duality gap. Heuristica. 2008, vol. 15, pp. 107–131.
17. Miller T., de Matta R. A global supply chain profit maximization and transfer pricing model. Jour. of Business Logistics. 2008, vol. 1, no. 29, pp. 175–199.
18. de Matta R., Miller T. Formation of a strategic manufacturing and distribution network with transfer prices. European Jour. of Operational Research. 2015, vol. 241, iss. 2, pp. 435–448.
19. Perron S., Hansen P., Le Digabel S., Mladenovic N. Transfer Pricing in a Global Supply Chain. GERAD, February 17, 2008, 31 p.
20. Сухобоков А.А. Исследование и разработка моделей и архитектуры средств контроллинга для межрегиональных предприятий в составе систем класса ERP II: дисс. ... канд. техн. наук, М.: Изд-во МГТУ им. Баумана, 2009. 196 с.
21. Tikhonov I. Using Big Data for the optimization of internal supply chains of multinational companies. Cloud of Science. 2015, vol. 2, no. 3, pp. 419–429. URL: https://cloudofscience.ru/sites/default/files/pdf/CoS_2_419.pdf (дата обращения: 27.01.2016).
22. Кочетов Ю.А., Младенович Н., Хансен П. Локальный поиск с чередующимися окрестностями // Дискретный анализ и исследование операций. 2003. Т. 10. № 1. С. 11–43.
23. Кривоножко В.Е. Развитие конечных методов решения

задач оптимизации. Декомпозиционный подход: дисс. ... д.ф.-м.н. М.: ИСА РАН, 1996. 273 с.

24. Хохлюк В.И. Параллельные алгоритмы целочисленной оптимизации. 2007. 140 с. URL: <http://math.nsc.ru/LBRT/u1/hohl/book.pdf> (дата обращения: 27.01.2016).

25. Brown H., Schumm C. Parallelization of SAP SCM optimizer using grid computing. Proc. Conf. Supply Network and Logistics Management. SAP Univ. Publ., St. Leon-Rot, 2007.

26. Сухобоков А.А., Лахвич Д.С. Влияние инструментария Big Data на развитие научных дисциплин, связанных с моделированием // Наука и образование. 2015. № 3. С. 207–240. URL: <http://technomag.bmstu.ru/doc/761354.html> (дата обращения: 27.01.2016).

27. Lakhvich D. Supply chains and transfer prices optimization using Apache Hadoop infrastructure and IBM ILOG CPLEX solver. Cloud of Science. 2015, vol. 2, no. 3, pp. 430–436. URL: https://cloudofscience.ru/sites/default/files/pdf/CoS_2_430.pdf (дата обращения: 27.01.2016).

28. Stadtler H., Kilger C., Meyr H. Conclusions and Outlook. Supply Chain Management and Advanced Planning. Concepts,

Models, Software, and Case Studies. 5th ed.: H. Stadtler, C. Kilger, H. Meyr (eds). Springer, Heidelberg, NY, Dordrecht, London, 2015, pp. 501–509.

29. Wurm F. Planning and Controlling Consolidated Earnings based on Consolidated Cost of Goods Manufactured. Controlling, iss. 5, June 2005, Version 1.0-15.11.05, IM&C GmbH, 19 p. URL: http://www.im-c.net/fileadmin/imc/daten/Financial_Management_GCP/artikel_original_EN.pdf (дата обращения: 27.01.2016).

30. Presentation GCP Engine for Corporate Controlling. IM&C, 2011. 47 p. URL: http://www.im-c.net/fileadmin/imc/daten/Financial_Management_GCP/Supply_Value_Chain_EN.pdf (дата обращения: 27.01.2016).

31. Gruat-La-Forme F.A., Botta-Genoulaz V., Campagne J.-P., Millet P.-A. Advanced planning and scheduling system: an overview of gaps and potential sample solutions. Proc. Int. Conf. on Industrial Engineering and Systems Management. 2005, Marrakech, Morocco, 2005, pp. 683–695. URL: <https://halshs.archives-ouvertes.fr/hal-00395107/document> (дата обращения: 27.01.2016).

Software & Systems

DOI: 10.15827/0236-235X.116.005-014

Received 10.02.16

2016, vol. 29, no. 4, pp. 5–14

APS FOR MULTINATIONAL COMPANIES: PROBLEMS AND PROSPECTS OF DEVELOPMENT

A.A. Sukhobokov¹, Ph.D. (Engineering), artem.sukhobokov@yandex.ru

A.V. Sukhobokov², Cofounder, andrey.sukhobokov@optimalmngmnt.com

D.S. Lakhvich³, Postgraduate Student, dmitry.lakhvich@optimalmngmnt.com

I.V. Tikhonov³, Postgraduate Student, ilya.tikhonov@optimalmngmnt.com

¹ SAP CIS, Kosmodamianskaya Emb. 52/2, Moscow, 115054, Russian Federation

² Optimal Management, LLC, Festivalnaya St. 22/6, Moscow, 125581, Russian Federation

³ Bauman Moscow State Technical University, 2nd Baumanskaya St. 5, Moscow, 105005, Russian Federation

Abstract. The article presents the common architecture of the Advanced Planning System (APS). It shows that mathematical models and methods of linear programming, which are now applied for the master planning, are not able to solve a key problem of profit optimization for multinational companies. The paper also considers the evolution of mathematic models and methods used for optimizing profit after the tax in multinational companies. In current situation in order to optimize the supply chain of big multinational companies it is necessary to solve a problem of great dimensionality, because data matrices contain dozens of billions columns and lines and have bilinear and trilinear constraints. Approaches to parallelization of numeral algorithms for this problem are also under consideration. The main stress is on a most prospective approach which uses Big Data platforms for solving great dimensionality problems. Hereafter, the article defines the additional functionality of a supply chain design module and a demand planning module, so that APS could be of use in multinational companies. The authors also define the additional functionality of financial supply chain management modules and of neighboring ERP-system's controlling modules. It is shown that prospective APS for multinational companies should be developed on the basis of a complex of two integrated technology platforms: the In-Memory platform and the Big Data platform. The functional architecture of APS for multinational companies is presented in the context of the general enterprise management system as a whole. In addition, the architecture of APS master planning module is detailed one level deeper. It solves optimization problems for one time period, as well as for a finite number of time periods.

Keywords: APS, multinational company, intercompany Supply Chains, transfer prices, goods traffic, bilinear optimization problem, parallelization of optimization algorithms, Master Planning, APS architecture, Big Data platforms.

References

1. Blackstone J.H. Jr. (Ed.), *APICS Dictionary*. 13th ed., APICS Publ., 2011, 164 p.
2. *Supply Chain Management and Advanced Planning. Concepts, Models, Software, and Case Studies*. 5th ed. H. Stadtler, C. Kilger, H. Meyr (eds.). Springer, Heidelberg, NY, Dordrecht, London, 2015, 572 p.
3. Meyr H., Wagner M., Rohde J. Structure of advanced planning systems. *Supply Chain Management and Advanced Planning. Concepts, Models, Software, and Case Studies*. 5th ed. H. Stadtler, C. Kilger, H. Meyr (eds.). Springer, Heidelberg, NY, Dordrecht, London, 2015, pp. 99–106.
4. Fleischmann B., Koberstein A. Strategic network design. *Supply Chain Management and Advanced Planning. Concepts, Models, Software, and Case Studies*. 5th ed. H. Stadtler, C. Kilger, H. Meyr (eds.). Springer, Heidelberg, NY, Dordrecht, London, 2015, pp. 107–123.

5. Ivert L.K. *Use of Advanced Planning and Scheduling (APS) systems to support manufacturing planning and control processes*. PhD thesis. Chalmers Univ. of Technology, Göteborg, Sweden, 2012, 111 p. Available at: <http://publications.lib.chalmers.se/records/fulltext/162616/162616.pdf> (accessed January 27, 2016).
6. Jonsson P., Kjellsdotter L., Rudberg M. Applying advanced planning systems for supply chain planning: three case studies. *Int. Jour. of Physical Distribution & Logistics Management*. 2007, vol. 37, iss. 10, pp. 816–834.
7. Kallrath J., Maindl T.I. *Real Optimization with SAP® APO*. Springer, Berlin, Heidelberg, NY, 2006, 340 p.
8. Lasschuit W., Thijssen N. Supporting supply chain planning and scheduling decisions in the oil and chemical industry. *Comp. and Chem. Eng.* 2004, no. 28, pp. 863–870. Available at: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.470.6859&rep=rep1&type=pdf> (accessed January 27, 2016).
9. Al-Qahtani K.Y., Elkamel A. *Planning and Integration of Refinery and Petrochemical Operations*. Wiley-VCH Verlag GmbH & Co, Germany, 2010, 209 p.
10. Lebreton B., Meyr H., Rosić H., Seipl C., Wetterauer U. Architecture of selected APS. *Supply Chain Management and Advanced Planning. Concepts, Models, Software, and Case Studies*. 5th ed. H. Stadtler, C. Kilger, H. Meyr (eds.). Springer, Heidelberg, NY, Dordrecht, London, 2015, pp. 341–361.
11. *Global Tax Planning*. 2013, 2 p. Available at: <https://tax.thomsonreuters.com/wp-content/pdf/global-tax/GlobalTaxPlanning.pdf> (accessed January 27, 2016).
12. Nieckels L. *Transfer Pricing in Multinational Firms: A Heuristic Programming Approach and a Case Study*. NY, John Wiley Publ., 1976, 202 p.
13. Cohen M.A., Fisher M., Jaikumar R. International manufacturing and distribution networks: a normative model framework. *Managing International Manufacturing*. K. Ferdows (Ed.). North-Holland, Amsterdam, 1989, pp. 67–93.
14. Cohen M.A., Lee H.L. Resource deployment analysis of global manufacturing and distribution networks. *Jour. of Manufacturing Operations Management*. 1989, no. 2, pp. 81–104.
15. Vidal C.J., Goetschalckx M. A global supply chain model with transfer pricing and transportation cost allocation. *European Jour. of Operational Research*. 2001, no. 129, pp. 134–158.
16. Vidal Holguin C.J., Goetschalckx M. A solution to the transfer pricing problem by successively reducing the duality gap. *Heuristica*. 2008, vol. 15, pp. 107–131.
17. Miller T., de Matta R. A global supply chain profit maximization and transfer pricing model. *Jour. of Business Logistics*. 2008, vol. 1, no. 29, pp. 175–199.
18. de Matta R., Miller T. Formation of a strategic manufacturing and distribution network with transfer prices. *European Jour. of Operational Research*. 2015, vol. 241, iss. 2, pp. 435–448.
19. Perron S., Hansen P., Le Digabel S., Mladenovic N. *Transfer Pricing in a Global Supply Chain*. GERAD, February 17, 2008, 31 p.
20. Sukhobokov A.A. *Issledovanie i razrabotka modeley i arkhitektury sredstv kontrollinga dlya mezhregionalnykh predpriyatiy v sostave sistem klassa ERP II* [Research and Development of Models and Architecture for Controlling Methods used for Multiregional Companies as a part of ERP II Systems]. PhD thesis, 2009, 196 p. (in Russ.).
21. Tikhonov I. Using Big Data for the optimization of internal supply chains of multinational companies. *Cloud of Science*. 2015, vol. 2, no. 3, pp. 419–429. Available at: https://cloudofscience.ru/sites/default/files/pdf/CoS_2_419.pdf (accessed January 27, 2016).
22. Kochetov Yu.A., Mladenovich N., Hansen P. Local search with alternating neighborhoods. *Diskretny analiz i issledovanie operatsy* [Discrete Analysis and Operations Research]. 2003, vol. 10, no. 1, pp. 11–43 (in Russ.).
23. Krivonozhko V.E. *Razvitiye konechnykh metodov resheniya zadach optimizatsii. Dekompozitsionny podkhod* [Development of Finite Methods for Solving Optimization Problems Using Decomposition]. PhD thesis. ISA RAS, 1996 (in Russ.).
24. Khokhlyuk V.I. *Parallelnye algoritmy tselochislennoy optimizatsii* [Parallel Algorithms for Integer Optimization]. 2007, 140 p. Available at: <http://math.nsc.ru/LBRT/u1/hohl/book.pdf> (accessed January 27, 2016) (in Russ.).
25. Brown H., Schumm C. Parallelization of SAP SCM optimizer using grid computing. *Proc. Conf. Supply Network and Logistics Management*. 2007, SAP Univ. Publ., St. Leon-Rot, 2007.
26. Sukhobokov A.A., Lakhvich D.S. Big Data tools influence on the development of simulation scientific disciplines. *Nauka i obrazovanie* [Science and Education]. 2015, no. 3, pp. 207–240. Available at: <http://technomag.edu.ru/doc/761354.html> (accessed January 27, 2016) (in Russ.).
27. Lakhvich D. Supply chains and transfer prices optimization using Apache Hadoop infrastructure and IBM ILOG CPLEX solver. *Cloud of Science*. 2015, vol. 2, no. 3, pp. 430–436. Available at: https://cloudofscience.ru/sites/default/files/pdf/CoS_2_430.pdf (accessed January 27, 2016).
28. Stadtler H., Kilger C., Meyr H. Conclusions and Outlook. *Supply Chain Management and Advanced Planning. Concepts, Models, Software, and Case Studies*. 5th ed.: H. Stadtler, C. Kilger, H. Meyr (eds). Springer, Heidelberg, NY, Dordrecht, London, 2015, pp. 501–509.
29. Wurm F. Planning and Controlling Consolidated Earnings based on Consolidated Cost of Goods Manufactured. *Controlling*, iss. 5, June 2005, Version 1.0-15.11.05, IM&C GmbH, 19 p. URL: http://www.im-c.net/fileadmin/imc/daten/Financial_Management_GCP/artikel_original_EN.pdf (accessed January 27, 2016).
30. *Presentation GCP Engine for Corporate Controlling*. IM&C, 2011, 47 p. Available at: http://www.im-c.net/fileadmin/imc/daten/Financial_Management_GCP/Supply_Value_Chain_EN.pdf (accessed January 27, 2016).
31. Gruat-La-Forme F.A., Botta-Genoulaz V., Campagne J.-P., Millet P.-A. Advanced planning and scheduling system: an overview of gaps and potential sample solutions. *Proc. Int. Conf. on Industrial Engineering and Systems Management*. 2005, Marrakech, Morocco 2005, pp. 683–695. Available at: <https://halshs.archives-ouvertes.fr/hal-00395107/document> (accessed January 27, 2016).

УДК 519.687

DOI: 10.15827/0236-235X.116.015-022

Дата подачи статьи: 16.06.16

2016. Т. 29. № 4. С. 15–22

МОДЕЛИРОВАНИЕ РАЗМЕЩЕНИЯ СЕРВИС-ОРИЕНТИРОВАННЫХ ПРИЛОЖЕНИЙ В ПРОГРАММНО-УПРАВЛЯЕМОЙ ИНФРАСТРУКТУРЕ ВИРТУАЛЬНОГО ЦЕНТРА ОБРАБОТКИ ДАННЫХ

*И.П. Болодурина, д.т.н., профессор, prmat@mail.osu.ru;**Д.И. Парфенов, к.т.н., fdot_it@mail.osu.ru**(Оренбургский государственный университет, просп. Победы, 13, г. Оренбург, 460018, Россия);**В.Н. Решетников, д.ф.-м.н., профессор, rvn_@mail.ru**(Центр визуализации и спутниковых информационных технологий НИИСИ РАН, Нахимовский просп., 36, корп. 1, г. Москва, 117218, Россия)*

Развитие информационных технологий в последнее время обуславливает использование распределенных сервис-ориентированных бизнес-приложений. При этом актуальны исследование механизмов управления размещением приложений в облачной среде и эффективное выделение соответствующих ресурсов для их работы в рамках существующих ограничений.

Целью данного исследования являются моделирование структуры и механизмов взаимодействия типовых облачных сервис-ориентированных приложений, а также определение подходов к оптимизации их размещения в системе хранения. Авторами разработана обобщенная модель сервис-ориентированного облачного приложения, включающая в себя три базовых слоя: модель приложения, модель сервиса и модель ресурса. В качестве ресурса в модели выступает некий объект, описывающий поведение и характеристики отдельных элементов инфраструктуры в зависимости от текущего состояния и параметров программно-управляемой инфраструктуры виртуального центра обработки данных. Отличительная особенность модели в том, что ресурс является универсальным, а это позволяет рассматривать его как с точки зрения пользователя, для которого он представляет собой закрытую систему, так и с точки зрения самой программно-управляемой инфраструктуры виртуального центра обработки данных, для которого он является открытой системой. Новизна модели заключается в одновременном описании размещения данных, связанных с ними приложений, а также состояния виртуальной среды, учитывающей сетевую топологию. Для детализации модели ресурса программно-управляемой инфраструктуры виртуального центра обработки данных в качестве отдельного элемента разработана модель виртуального самоорганизующегося хранилища данных, позволяющая определить их параметры и связи приложений с виртуальными и физическими узлами хранилища данных.

Ключевые слова: *облачные приложения, облачные сервисы, вычислительные ресурсы, виртуальные центры обработки данных, программно-управляемые устройства хранения, облачные вычисления, машинное обучение, OpenStack.*

В последние годы технология облачных вычислений стала популярной для предоставления доступа к приложениям и сервисам с целью обеспечения работы бизнес-услуг [1]. Существуют три основных подхода к предоставлению услуг в сфере облачных вычислений: инфраструктура как услуга (IaaS), платформа как услуга (PaaS) и ПО как услуга (SaaS). Применение этих подходов при развертывании облачных вычислительных платформ имеет множество преимуществ, таких как надежность и качество обслуживания [2]. В то же время существует ряд ограничений со стороны как пользователей, так и поставщиков облачных услуг. Для потребителя облачные ресурсы представляются бесконечными с точки зрения масштабируемости, однако, если рассмотреть экономический аспект их потребления, видим, что возможности их существенно сужаются. Со стороны провайдеров облачных услуг, предоставляющих доступ к ресурсам, набор сервисов и вычислительных мощностей ограничен. Для максимизации экономической составляющей за счет увеличения числа пользователей поставщики вынуждены применять политики, позволяющие эластично использовать выделяемые ресурсы, минимизируя при этом операционные затраты.

Актуальной проблемой является управление ресурсами и облачными приложениями в современных виртуальных и физических *центрах обработки данных* (ЦОД), поскольку оно напрямую влияет на эксплуатационные расходы. Крупные корпорации отрасли информационных технологий, такие как Amazon, Google, Salesforce, IBM, Microsoft, Oracle и др., в течение последних нескольких лет разрабатывают обновленные подходы к управлению ресурсами и объектами в ЦОД, используемых для размещения решений для облачных приложений [3, 4]. Основной тенденцией в данной области развития современных информационных технологий является оптимизация использования ресурсов ЦОД.

В проведенных ранее исследованиях авторами статьи были разработаны подходы к оптимизации хранения данных облачных приложений, а также к повышению эффективности доступа к ресурсам облачной среды [5]. Однако это не решает проблему размещения экземпляров самих приложений в облачной среде. Кроме этого, проведенный обзор исследований в данной области показал, что проблема оптимизации выбора ресурсов для конкретных типов облачных приложений недостаточно изучена [6–9].

Для решения обозначенных проблем авторы предлагают подход, основанный на управлении размещением облачных приложений и выделением соответствующих ресурсов для их работы.

Для понимания принципов работы облачных приложений необходимо определить их место в инфраструктуре ЦОД. На сегодняшний день строительство и эксплуатация выделенных физических ЦОД экономически нецелесообразны. Поэтому все чаще имеется в виду виртуальный ЦОД. Такой подход позволяет организовывать собственную инфраструктуру на базе арендуемых вычислительных мощностей. Виртуальный ЦОД с точки зрения облачного приложения является изменяющимся объектом. При этом его состояние в любой произвольный момент времени t можно описать в виде ориентированного графа: $VirtDC=(Node(t), Connect(t))$, где вершины $Node(t)=\{Node_1, Node_2, \dots, Node_n\}$ представляют собой активные элементы облачной инфраструктуры виртуального ЦОД (вычислительные узлы, системы хранения данных и др.), используемые активными экземплярами приложений $CloudAppl(t) = \{CloudAppl_1, CloudAppl_2, \dots, CloudAppl_c\}$, запущенные на виртуальных ресурсах; ветви – $Connect(t) = \{Connect_1, Connect_2, \dots, Connect_v\}$ – активные подключения пользователей к облачным приложениям.

Особенность работы облачных приложений в том, что слой виртуализации скрывает от конечных пользователей количество и расположение запущенных экземпляров сервисов или приложений, с которыми они работают. Чаще всего им доступны только адрес агрегирующего узла и имя приложения. Облачная система самостоятельно выбирает оптимальную виртуальную машину для выполнения запроса. Таким образом, с точки зрения пользователя, облачное приложение является черным ящиком, который осуществляет обработку поступающих запросов.

Прежде чем говорить о способах размещения облачных приложений в виртуальном ЦОД, необходимо определить их структуру, основные параметры и ключевые характеристики работы, влияющие на эффективность использования этих приложений. Для этого была разработана обобщенная модель сервис-ориентированного облачного приложения.

Обобщенная модель сервис-ориентированного облачного приложения представляет собой многослойную структуру, описываемую в виде графов, характеризующих связи отдельных элементов. Модель может быть представлена в виде трех базовых срезов, детализирующих связи отдельных облачных объектов инфраструктуры виртуального ЦОД: приложений, связанных с ними сервисов, а также предоставляемых им для работы ресурсов.

Сервис-ориентированное облачное приложение – это взвешенный ориентированный ациклический

граф зависимостей по данным, в котором вершинами являются задачи, направленные на получение или обработку информации из источников в соответствии с запросом пользователя, а дугами – зависимости приложений от источников данных между соответствующими вершинами: $CloudAppl=(G, V)$, где G – вершины графа задачи, направленные на получение или обработку информации по запросу пользователя; V – дуги зависимости приложений от источников данных.

Каждая вершина (G) характеризуется предъявляемыми ресурсными требованиями, количеством запускаемых экземпляров, оценкой времени выполнения запросов пользователей, коммуникационными шаблонами передачи данных между источниками и вычислительными узлами: $G=\{Res, NAppl, Utime, SchemeTask\}$, где Res – требования приложения к ресурсам; $NAppl$ – требования к количеству запускаемых экземпляров приложений; $Utime$ – оценка времени выполнения запросов пользователей; $SchemeTask$ – коммуникационные шаблоны передачи данных приложений.

Каждая дуга (V), соединяющая приложение с требуемым источником данных, характеризуется типом передаваемых данных, методом получения доступа к источнику информации, физическим типом объекта доступа, объемом трафика, требованиями к качеству обслуживания при передаче данных (QoS): $V=\{Tdata, Mdata, Fdata, Vdata, Qdata\}$, где $Tdata$ – тип передаваемых данных; $Mdata$ – метод получения доступа к источнику информации (REST, JSON и др.); $Fdata$ – тип объекта доступа (доступ к файлу в СХД, к локальному файлу, к распределенной БД, к сервису данных и пр.); $Vdata$ – объем трафика, оцениваемый на основе информации о данных (Мб); $Qdata$ – требования к QoS.

Оригинальность модели заключается в том, что для каждого приложения производится консолидированная оценка эффективности работы с источниками данных, что позволяет прогнозировать работу всей системы в целом.

Как отмечалось ранее, одним из ключевых срезов обобщенной модели сервис-ориентированного облачного приложения является облачный сервис. Он служит автономным источником данных для приложения, выполняющего роль консолидированного обработчика информации. Как правило, облачный сервис является узкоспециализированным и предназначен для выполнения ограниченного набора функций. Преимущество подключения облачного приложения к сервису, в отличие от прямого доступа к данным, в изолированной обработке информации, что позволяет сократить время исполнения запроса пользователя. Облачный сервис можно описать как закрытую систему в виде ориентированного графа зависимостей функционала от поступающего набора данных. Для описания доступа требуются адрес агрегирующего вы-

числительного узла и имя сервиса, а также требуемый на выходе формат данных. Агрегатор сервиса сам выбирает оптимальную для запроса виртуальную машину, на которой он и исполняется. Также все его приложения распределены между виртуальными машинами или физическими серверами предустановленного набора, их новые экземпляры масштабируются динамически в зависимости от количества поступающих запросов на выполнение функции конкретного сервиса со стороны облачного приложения, конечного пользователя или других облачных сервисов.

$CloudServ = \{AgrIP, NameServ, Format\}$, где $AgrIP$ – адрес агрегирующего вычислительного узла сервиса; $NameServ$ – имя сервиса; $Format$ – требуемый на выходе формат данных.

Для описания размещения облачных сервисов и приложений в инфраструктуре ЦОД была также реализована модель облачного ресурса. Под облачным ресурсом будем понимать объект ЦОД, описывающий поведение и характеристики отдельных элементов инфраструктуры в зависимости от его текущего состояния и параметров. В качестве объектов могут выступать дисковые массивы, в том числе отдельные устройства хранения, виртуальные машины, программно-конфигурируемые хранилища данных, а также СУБД различных видов (SQL/NoSQL) и др. Кроме этого, каждый облачный сервис или приложение предъявляет требования к количеству ядер, размерам оперативной и дисковой памяти, наличию специальных библиотек на физических или виртуальных узлах, используемых для запуска их среды.

$CloudRes = \{TRes, Param, State, Core, Rmem, Hmem, Lib\}$, где $TRes$ – тип ресурса; $Param$ – параметры доступа; $State$ – состояние ресурса; $Core$ – количество вычислительных ядер; $Rmem$ – размер оперативной памяти; $Hmem$ – размер дисковой памяти; Lib – требования к наличию специальных библиотек.

Отличительная особенность модели в универсальности ресурса, что позволяет рассматривать его как с точки зрения пользователя, для которого он представляет собой закрытую систему, так и с точки зрения самой программно-управляемой инфраструктуры виртуального ЦОД, для которого он является открытой системой.

Новизна модели заключается в одновременном описании размещения данных, связанных с ними приложений, а также состояния виртуальной среды, учитывающей сетевую топологию.

Для детализации модели ресурса программно-управляемой инфраструктуры виртуального ЦОД в качестве отдельного элемента разработана модель виртуального самоорганизующегося хранилища данных (SDS), представляемая также в виде ориентированного мультиграфа, включающего в себя в качестве вершин элементы виртуального ЦОД, отвечающие за размещение данных приложений

(виртуальные дисковые массивы, СУБД и т.д.). При этом в качестве дуг выступают связи данных с облачными приложениями и с сервисами. В качестве весов использована консолидированная оценка, состоящая из характеристики востребованности данных, а также загруженности конкретных устройств как на физическом, так и на виртуальном уровне.

$Stg = (StgNode, Data)$, где $StgNode$ – элементы виртуального ЦОД, отвечающие за размещение данных приложений; $Data$ – связи данных с облачными приложениями и сервисами.

Каждый элемент виртуального самоорганизующегося хранилища данных обладает индивидуальными характеристиками:

$$Stg_{ki} = (MaxV_{ki}, P_{ki}^{stg}, Vol_{ki}(t), \bar{R}_{ki}(t), \bar{W}_{ki}(t), s_{ki}^{stg}(t)),$$

где $MaxV_{ki} \in N$ – максимальная емкость для хранения (Mb); $P_{ki}^{stg} = \{p_{kij}^{stg}\}_j$ – множество сетевых портов; $Vol_{ki}(t) \in N \cup \{0\}$ – доступный объем для хранения в Mb; $\bar{R}_{ki}(t)$ и $\bar{W}_{ki}(t)$ – скорость чтения и записи; $s_{ki}^{stg}(t) \in \{online, offline\}$ – состояние.

Система хранилищ данных для облачных приложений напоминает слоеный пирог и работает на принципах самоорганизации ресурсов. Адаптивная модель динамической реконфигурации ресурсов является основой самоорганизации управления хранилищами данных. Кроме того, принципы самоорганизации ресурсов применяются при оптимизации структуры управления самой облачной платформой на базе адаптивного выбора управляющих узлов. В разработанной авторами модели управления ресурсами облачной системы условно можно выделить два ключевых компонента: виртуальные узлы и вычислительные ресурсы. Управление ресурсами в виртуальных самоорганизующихся хранилищах осуществляется посредством специализированного ПО (агенты управления), осуществляющего синхронизацию и сбор технической информации о вычислительных узлах, выделенных для размещения данных. Для снижения объемов потока трафика, формируемого агентами управления, обмен данными происходит между узлами с одинаковым способом хранения информации. При формировании больших групп для организации виртуальных самоорганизующихся хранилищ данных возможно выделение подгрупп и назначение дополнительных управляющих узлов. При назначении управляющего агента предпочтение отдается наименее загруженным узлам. Децентрализованный подход, основанный на принципах самоорганизации управления, позволяет исключить фактор потери связи с управляющим узлом во время работы. Каждый управляющий агент на виртуальном вычислительном узле системы хранения в любой момент времени имеет данные о других узлах, входящих в его группу или подгруппу, а также

информацию об управляющих узлах других групп, что позволяет провести выборы и делегировать полномочия новому управляющему узлу. Это обеспечивает надежность разработанной системы управления виртуальными хранилищами данных. В описанной концепции управления виртуальными самоорганизующимися хранилищами данных можно выделить три слоя управления, подчиняющихся друг другу по иерархическому принципу: первый (локальный) – управление на уровне оперативного доступа; второй (групповой) – управление на уровне группы ресурсов; третий (глобальный) – управление на уровне обмена данными во всей облачной системе. На базе описанной модели управления виртуальными самоорганизующимися хранилищами данных реализован алгоритм размещения данных.

Новизна обобщенной модели сервис-ориентированного облачного приложения заключается в одновременном описании подзадач и источников данных разного типа с указанием направления и способа передачи данных, а также требуемых для этого ресурсов.

Обобщенная модель облачного сервис-ориентированного приложения положена в основу при разработке алгоритмов планирования облачных сервисов и приложений поверх виртуальных машин и физических серверов инфраструктуры виртуального ЦОД, а также в алгоритме планирования размещения и организации доступа к данным облачных сервисов и приложений, что является оригинальным подходом.

Реализационная часть

По сравнению с имеющимися аналогами ключевой особенностью спроектированного алгоритма управления размещением облачных приложений в самоорганизующемся хранилище ЦОД являются эвристический анализ потоков запросов, а также классификация трафика по типу передаваемой информации [10–12]. Гибкость предлагаемого решения обусловлена виртуализацией хранилища данных. Это позволяет организовывать динамическое размещение данных и приложений на физических устройствах для обеспечения постоянного доступа к услугам. Разработанное решение дает возможность масштабировать сервисы и приложения, размещаемые в облачной системе на несколько виртуальных устройств хранения. Это обеспечивает сокращение времени отклика приложения и повышает отказоустойчивость облачной платформы в целом. Формирование самоорганизующейся системы хранения с использованием средств виртуализации ресурсов дает возможность не только повысить надежность инфраструктуры, но и собирать информацию о востребованности различных облачных ресурсов, используемых приложениями и

сервисами в процессе работы. Алгоритм управления размещением приложений и сервисов в виртуальном самоорганизующемся хранилище данных использует модель облачного ресурса, характеризующую параметры и структуру связи объектов ЦОД.

Формализованная в рамках настоящего исследования модель использует мультиагентный подход для сбора информации о состоянии носителей информации и вычислительных узлах, задействованных для работы облачной системы. Для обработки полученной информации используются алгоритмы интеллектуального анализа данных (Data Mining), основанные на нейросетевом подходе. Результатом применения данного подхода являются две карты, одна из которых характеризует размещение виртуальных хранилищ данных внутри самой облачной платформы с привязкой к физическим устройствам, другая описывает непосредственно востребованность информации в системе хранения. На основе алгоритма эвристического анализа построенных карт и интеллектуального прогнозирования управляющая система перераспределяет данные внутри виртуального хранилища и сами узлы хранения между физическими устройствами. При этом перестроение карт происходит постоянно, а интервалы времени изменения зависят от интенсивности обращения приложений и сервисов к системе хранения.

В рамках исследования определен наиболее оптимальный интервал времени для анализа карт, при котором план перестроения самоорганизующихся виртуальных хранилищ данных облачной платформы будет наименее ресурсоемким, но в то же время максимально эффективным.

Предложенный алгоритм управления размещением приложений в самоорганизующемся виртуальном хранилище данных позволяет также повысить производительность работы всей облачной платформы в целом. При этом благодаря адаптивной маршрутизации и балансировке потоков данных, передаваемых между запущенными экземплярами облачных приложений, обеспечивается не только должное качество обслуживания, но и компактное расположение устройств [13]. Кроме того, алгоритм управления размещением облачных приложений решает задачу оптимизации количества задействованных физических устройств хранения, что, в свою очередь, позволяет повысить экономическую эффективность использования облачных платформ. Для получения результата используется информация от агентов и управляющих узлов облачной платформы [14].

Для эффективной обработки поступающего потока запросов пользователей облачные приложения и сервисы, как правило, задействуют сразу несколько типов ресурсов инфраструктуры виртуального ЦОД. При этом характеристики времени

полнения каждого запроса зависят от параметров доступа к конкретному облачному ресурсу. Для повышения эффективности обработки запросов в облачной системе необходимо проводить упреждающую подготовку ресурсов, одновременно оптимизируя характеристики доступа, относительно времени отклика приложений. Разработанное алгоритмическое решение осуществляет проактивное планирование размещения облачных ресурсов, тем самым подстраивая каждый экземпляр самоорганизующегося виртуального хранилища данных под текущую востребованность данных и облачных приложений. Для оптимизации характеристик доступа к облачным ресурсам алгоритм позволяет осуществлять адаптивную балансировку потоков запросов пользователей в зависимости от текущей нагрузки на выделенный виртуальный узел самоорганизующейся системы хранения данных. На рисунке 1 представлена блок-схема работы алгоритма размещения облачных приложений в самоорганизующемся виртуальном хранилище данных ЦОД.

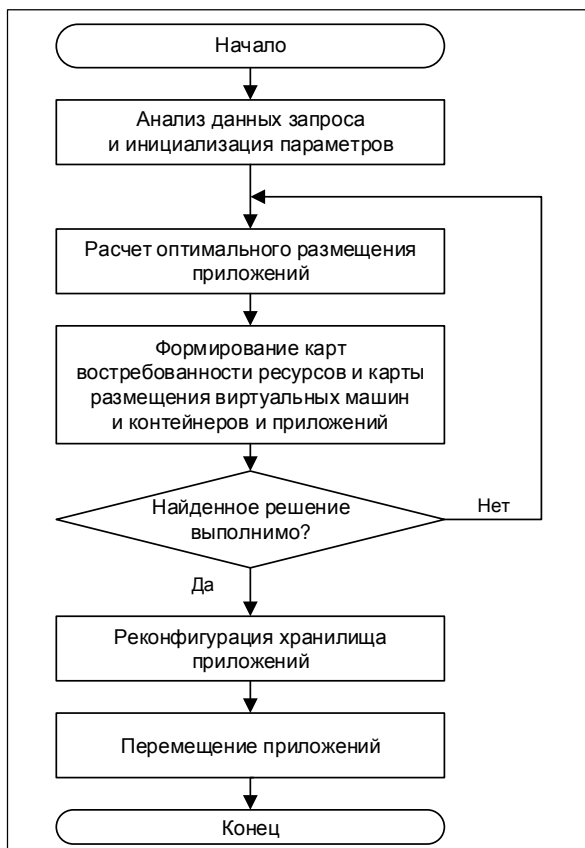


Рис. 1. Обобщенная блок-схема алгоритма управления размещением облачных приложений в программно-конфигурируемых хранилищах данных виртуального ЦОД

Fig. 1. A generalized flow chart of a control algorithm for setting cloud applications in software defined data storage of virtual DPC

Всего в работе алгоритма управления размещением облачных приложений в самоорганизующемся виртуальном хранилище ЦОД можно выделить три этапа планирования и оптимизации выполнения потока запросов, формируемых пользователями.

На первом этапе работы алгоритма проводится анализ параметров запроса, поступающего от пользователя к приложению, а именно, определяются тип приложения и вид передаваемых данных. Запрос данных является многокомпонентным, то есть для организации канала задействуются сразу несколько элементов системы. Поэтому на втором этапе система управления определяет наиболее подходящие ресурсы, способные гарантировать выполнение запроса. Если среди ресурсов существует вариативность, на основе информации о прогнозе востребованности каждого из доступных вариантов ресурса строится ранжированный список с привязкой к виртуальному хранилищу данных. При этом учитываются только наименее загруженные хранилища данных. На третьем этапе работы алгоритма анализируются текущие параметры ресурсов и прогнозируется время, необходимое на выполнение текущего запроса. Полученные результаты сохраняются для дальнейшего использования при самообучении алгоритма. В случаях высокой ресурсоемкости запроса для предварительной оценки в алгоритме применяется поэлементный анализ объектов доступа, входящих в запрос.

Экспериментальная часть

Для оценки эффективности разработанного алгоритма управления размещением облачных приложений в самоорганизующемся хранилище данных виртуального ЦОД авторами исследована работа облачной системы, построенной на базе Openstack с различными параметрами. В качестве эталона для проведения эксперимента применялись типовые алгоритмы, используемые облачными системами для распределения экземпляров облачных приложений и востребованных ими данных по вычислительным узлам системы хранения. Для анализа эффективности и производительности работы алгоритмов на различных системах хранения были определены типовые условия эксперимента, включающие традиционные устройства на магнитных дисках (HDD), твердотельные накопители (SSD) и виртуальные хранилища (SDS). Для экспериментального исследования создан прототип облачной среды, включающий в себя основные узлы, а также программные модули для разработанных алгоритмов, выполняющие перераспределение запросов пользователей к данным в самоорганизующемся виртуальном хранилище.

В облачной системе OpenStack реализован модуль, применяющий разработанные алгоритмы управления размещением облачных приложений в самоорганизующихся хранилищах данных виртуального ЦОД для рационального использования вычислительных ресурсов облачной системы и эффективного размещения виртуальных машин по физическим узлам, а также связанных с ними данных. В ходе эксперимента для анализа данных создан поток запросов, аналогичный реальным запросам пользователей к облачным приложениям и сервисам, основанный на архивных данных за последние три года, содержащих параметры и интенсивность обращения к различным видам ресурсов облачной системы, классифицированных по типам данных и структуре запроса. Для воспроизведения нагрузочного эксперимента использовались средние значения полученных данных. Поток запросов пользователей в ходе эксперимента распределялся между вычислительными узлами облачной системы в зависимости от следующих параметров доступа: тип клиента, тип сервиса. Общее количество запросов, сгенерированных в ходе проведения испытаний и поступивших на вход системы, составило 100 000.

Для проведения экспериментального исследования построены три технические площадки, включающие в себя физические системы хранения на базе магнитных жестких дисков (площадка 1) и твердотельных накопителей (площадка 2), а также виртуальное хранилище данных (площадка 3). На каждой из технических площадок последовательно проводился ряд вычислительных экспериментов, направленных на интенсивное выполнение операции чтения (эксперимент 1) (рис. 2), записи (эксперимент 2) (рис. 3) и одновременно операций чтения и записи данных (эксперимент 3) (рис. 4).

Для проведения экспериментальных исследований был определен интервал времени, соответствующий максимальному значению пиковой активности пользователей в облачной системе. Время каждого эксперимента составило один час. Результаты экспериментальных исследований доказывают, что размещение облачных приложений с использованием предложенного алгоритма более эффективно независимо от типа физических устройств, используемых в хранилище данных.

Результаты экспериментальных исследований показывают возможность снизить на 20–25 % количество отказов в обслуживании при размещении информации, востребованной приложениями и пользователями, в виртуальных самоорганизующихся хранилищах данных, что подтверждает целесообразность использования алгоритма для предоставления эффективного доступа к приложениям и сервисам, развернутым в виртуальном ЦОД на базе облачной платформы.

В дополнение к данным экспериментам авторами проведено исследование, направленное на

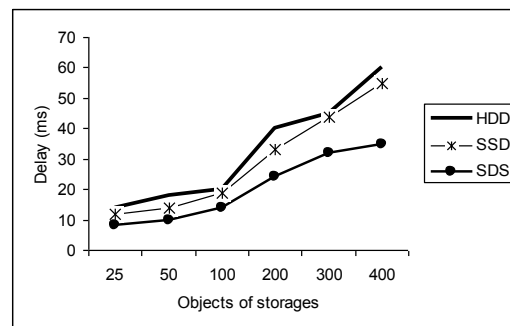


Рис. 2. Анализ времени отклика приложений при выполнении запросов на чтение данных (эксперимент 1)

Fig. 2. An analysis of application response time when running read request (experiment 1)

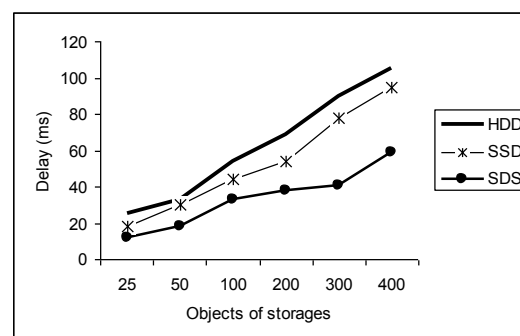


Рис. 3. Анализ времени отклика приложений при выполнении запросов на запись данных (эксперимент 2)

Fig. 3. An analysis of application response time when running write request (experiment 2)

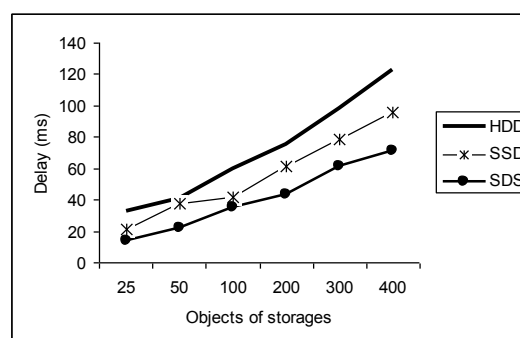


Рис. 4. Анализ времени отклика приложений при одновременном выполнении запросов на чтение и запись данных (эксперимент 3)

Fig. 4. An analysis of application response time when running read and write request simultaneously (experiment 3)

оценку количества задействованных в работе устройств хранения. На рисунке 5 представлен график, показывающий эффективность использования ресурсов системы хранения данных экспериментального стенда виртуального ЦОД.

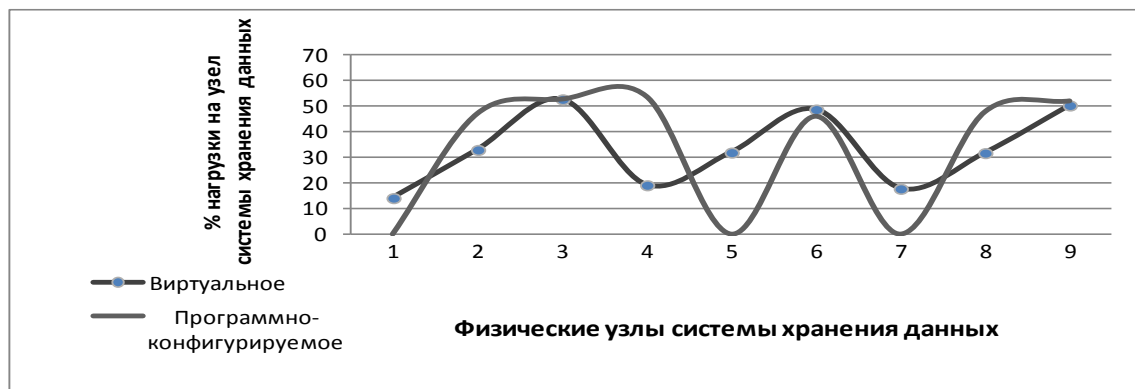


Рис. 5. График эффективности использования вычислительных ресурсов системы хранения данных в программно-управляемой инфраструктуре виртуального ЦОД

Fig. 5. A diagram of computing resource usage efficiency in a cloud system

За счет оптимального выделения ресурсов на каждом узле системы хранения гарантировано обеспечение совместной работы всех запущенных экземпляров облачных приложений, что актуально при значительной нагрузке на систему хранения. Разработанный алгоритм управления размещением облачных приложений и сервисов в виртуальной самоорганизующейся системе хранения данных позволяет высвободить от 20 до 30 % устройств хранения. Таким образом, предложенный алгоритм показал свою эффективность как для традиционных хранилищ, так и для систем с неоднородной конфигурацией физических узлов и виртуальных машин.

В результате исследования разработана модель сервис-ориентированного облачного приложения, описывающая структуру, связи, а также концепцию применения виртуальных самоорганизующихся хранилищ данных для размещения приложений и данных в облачной системе с учетом технических ограничений виртуального ЦОД. Разработанная модель позволила реализовать алгоритм управления размещением облачных приложений в виртуальной самоорганизующейся системе хранения. В ходе экспериментальных исследований установлено, что применение разработанного алгоритма позволяет сократить время отклика облачных приложений и сервисов и, как следствие, повысить производительность обработки запросов пользователей. При этом использование виртуальной самоорганизующейся системы хранения позволяет сократить количество физических устройств, что говорит об эффективности разработанного решения.

Работа выполнена при поддержке РФФИ, научные проекты №№ 16-37-60086 мол_а_дк, 16-07-01004.

Литература

1. Грибова В.В., Клещев А.С., Крылов Д.А., Москаленко Ф.М., Тимченко В.А., Шалфеева Е.А. Базовая технология разработки интеллектуальных сервисов на облачной платформе

IACPaaS. Ч. 2. Разработка агентов и шаблонов сообщений // Программная инженерия. 2016. № 1. С. 14–20.

2. Майер-Шенбергер В., Кукер К. Большие данные. Революция, которая изменит то, как мы живем, работаем, мыслим. М.: Манн, Иванов и Фербер, 2014. 240 с.

3. Bolodurina I., Parfenov D., Shukhman A. Approach to the effective controlling cloud computing resources in data centers for providing multimedia services Proc. 2015 Intern. Siberian Conf. on Control and Communications, SIBCON, 2015, pp. 714–717.

4. Болодурина И.П., Парфенов Д.И. Комплексное решение эффективного доступа к данным в гибридных облачных системах с учетом особенностей мультимедийных сервисов // Вестн. Оренбургского гос. ун-та. 2015. № 13 (188). С. 114–118.

5. Болодурина И.П., Парфенов Д.И. Управление потоками данных в высоконагруженных информационных системах, построенных на базе облачных вычислений // Системы управления и информационные технологии. 2015. № 1.1 (59). С. 111–118.

6. Тарасов В.Н., Полежаев П.Н., Шухман А.Е., Ушаков Ю.А., Коннов А.И. Математические модели облачного вычислительного центра обработки данных с использованием OpenFlow // Вестн. Оренбургского гос. ун-та. 2012. № 9. С. 150–155.

7. Устименко О.В. Перспективы развития распределенного хранения сверхбольших объемов данных // Матер. XVIII междунар. науч. конф. ОМУС ОИЯИ, Дубна, 2014. С. 198–207.

8. Wuhib F., Stadler R., Lindgren H. Dynamic resource allocation with management objectives – Implementation for an OpenStack cloud. Network and service management, 2012 Proc. 8th Intern. Conf. and 2012 Workshop on Systems Virtualization Management, 2012, pp. 309–315.

9. Bocchi E., Drago I., Mellia M. Personal Cloud Storage Benchmarks and Comparison. IEEE Transactions on Cloud Computing. 2015, vol. 99. DOI: 10.1109/TCC.2015.2427191.

10. Darabseh A., Al-Ayyoub M., Jararweh Y., Benkhelifa E., Vouk M., Rindos A. SDStorage: A Software Defined Storage Experimental Framework. Proc. of Cloud Engineering (IC2E), IEEE Press, 2015, pp. 341–346.

11. Plakunov A., Kostenko V. Data center resource mapping algorithm based on the ant colony optimization. Proc. of Science and Technology Conf. (MoNeTeC), IEEE Press, 2014, pp. 1–6.

12. Wu X., Kumar V., Ross J. Top 10 algorithms in data mining. Jour. Knowledge and Information Systems.

13. Singh A., Korupolu M., Mohapatra D. Server-storage virtualization: integration and load balancing in Data Centers. Proc. of the 2008 ACM/IEEE Conf. on Supercomputing. IEEE Press, 2008, pp. 1–12.

14. Болодурина И.П., Решетников В.Н., Парфенов Д.И. Распределение ресурсов в информационной системе дистанционной поддержки образовательного процесса // Программные продукты и системы. 2012. № 3. С. 151–155.

MODELING OF PLACING SERVICE-ORIENTED APPLICATIONS IN A SOFTWARE-DEFINED INFRASTRUCTURE OF THE VIRTUAL DATA CENTER

I.P. Bolodurina¹, Dr.Sc. (Engineering), Professor, prmat@mail.osu.ru

D.I. Parfenov¹, Ph.D. (Engineering), fdot_it@mail.osu.ru

V.N. Reshetnikov², Dr.Sc. (Physics and Mathematics), Professor, rvn_@mail.ru

¹ Orenburg State University, Pobedy Ave. 13, Orenburg, 460018, Russian Federation

² Center of Visualization and Satellite Information Technologies SRISA, Nakhimovsky Ave. 36/1, Moscow, 117218, Russian Federation

Abstract. The information technology development in recent years leads to using distributed service-oriented business applications. Thus, research on mechanisms of managing hosted applications in a cloud environment is currently important, as well as efficient allocation of appropriate resources for their work within the existing constraints.

The aim of the study is to simulate the structure and interaction mechanisms of typical cloud service-oriented applications, as well as definition of approaches to optimization of their placement in a storage system. The authors developed a generalized model of a service-oriented cloud application that includes three basic layers: an application model, a service model, and a resource model. As a resource in the model they consider some kind of object that describes the behavior and characteristics of individual elements of an infrastructure depending on the current status and settings of the virtual data center of a software-defined infrastructure. A distinctive feature of the model is that it is a universal resource that allows us to consider it from user's point of view (for whom it is a closed system) and from the point of view of the software-defined infrastructure of a virtual data center (for which it is an open system). The novelty of the model is simultaneous describing data placement, associated applications, as well as the state of the virtual environment, taking into account the network topology. To work out the resource model in the virtual data center of software-defined infrastructure in detail the authors developed a model of a software-defined storage as a separate element. It allows defining data parameters and applications' connections with virtual and physical nodes of a data storage.

Keywords: cloud applications, cloud services, computing resources, virtual data centers, software-defined storage devices, cloud computing, data mining, OpenStack.

Acknowledgements. The work has been done with support from RFBR, scientific projects no. 16-37-60086 мол_а_дк, 16-07-01004.

References

1. Gribova V.V., Kleshev A.S., Krylov D.A., Moskalenko F.M., Timchenko V.A., Shalfeyeva E.A. Basic technology of intellectual services development on the cloudy IACPaaS platform. Part 2. Agents and message templates development. *Programmaya inzheneriya* [Program Engineering]. 2016, no. 1, pp. 14–20 (in Russ.).
2. Mayer-Schönberger V., Cukie K. *Big Data: A Revolution That Will Transform How We Live, Work and Think*. John Murray Publ., 2013, 256 p. (Russ.ed.: Moscow, Mann, Ivanov i Ferber Publ., 2014, p. 240 p.).
3. Bolodurina I., Parfenov D., Shukhman A. Approach to the effective controlling cloud computing resources in data centers for providing multimedia services. *Proc. 2015 Int. Siberian Conf. on Control and Communications, SIBCON 2015*. 2015, pp. 714–717.
4. Bolodurina I.P., Parfenov D.I. Complete solution of efficient data access in a hybrid cloud systems according to multimedia services characteristics. *Vestnik Orenburgskogo gos. univ.* [Bulletin of the Orenburg State Univ.]. 2015, no. 13, pp. 114–118 (in Russ.).
5. Bolodurina I.P., Parfenov D.I. The flow of control in high loaded information systems based on cloud computing. *Sistemy upravleniya i informatsionnye tekhnologii* [Control systems and information technologies]. 2015, no. 1.1 (59), pp. 111–118 (in Russ.).
6. Tarasov V.N., Polezhaev P.N., Shukhman A.E., Ushakov Ju.A., Konnov A.L. Mathematical models of cloud computing datacenter based on openflow. *Vestnik Orenburgskogo gos. univ.* [Bulletin of the Orenburg State Univ.]. 2012, no. 9, pp. 150–155 (in Russ.).
7. Ustimenko O.V. Prospects for the development of ultra-large data volumes distributed storage. *Materialy XVIII mezhdunar. nauch. konf. obyedineniya molodykh uchenykh i spetsialistov OIJaI (OMUS)* [Proc. 18th Int. scientific conf. of the Association of Young Scientists and Specialists (AYSS)]. 2014, pp. 198–207 (in Russ.).
8. Wuhib F., Stadler R., Lindgren H. Dynamic resource allocation with management objectives – Implementation for an OpenStack cloud. Network and service management. 2012. *Proc. 8th Int. conf. and 2012 workshop on systems virtualization management*. 2012, pp. 309–315.
9. Bocchi E., Drago I., Mellia M. Personal Cloud Storage Benchmarks and Comparison. *IEEE Transactions on Cloud Computing*. 2015, vol. 99.
10. Darabseh A., Al-Ayyoub M., Jararweh Y., Benkhelifa E., Vouk M., Rindos A. SDStorage: A Software Defined Storage Experimental Framework. *Proc. of Cloud Engineering (IC2E)*. Tempe, IEEE Press, 2015, pp. 341–346.
11. Plakunov A., Kostenko V. Data center resource mapping algorithm based on the ant colony optimization. *Proc. of Science and Technology Conf. (Modern Networking Technologies) (MoNeTeC)*. Moscow, IEEE Press, 2014, pp. 1–6.
12. Wu X., Kumar V., Ross J. Top 10 algorithms in data mining. *Knowledge and Information Systems*. 2008, iss. 14. Available at: <http://www.cs.uvm.edu/~icdm/algorithms/10Algorithms-08.pdf> (accessed July 12, 2016).
13. Singh A., Korupolu M., Mohapatra D. Server-storage virtualization: integration and load balancing in Data Centers. *Proc. 2008 ACM/IEEE Conf. on Supercomputing*. Austin, IEEE Press, 2008, pp. 1–12.
14. Bolodurina I.P., Reshetnikov V.N., Parfenov D.I. Resource distribution in an information system related to educational process remote support. *Programmnye produkty i sistemy* [Software & Systems]. 2012, no. 3, pp. 151–155 (in Russ.).

УДК 004.5

DOI: 10.15827/0236-235X.116.023-026

Дата подачи статьи: 07.10.16

2016. Т. 29. № 4. С. 23–26

АРХИТЕКТУРА ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА ДЛЯ ВЗАИМОДЕЙСТВИЯ С МУЛЬТИАГЕНТНОЙ СРЕДОЙ

*И.А. Сидоров, к.т.н., научный сотрудник, ivan.sidorov@icc.ru
(Институт динамики систем и теории управления СО РАН,
ул. Лермонтова, 134, г. Иркутск, 664033, Россия)*

В работе рассматривается подход к созданию пользовательского интерфейса для взаимодействия с мультиагентной средой, ориентированной на решение ресурсоемких вычислительных задач и объединяющей множество разнородных программно-аппаратных ресурсов. Наличие большого числа пользователей в такой среде и разнообразие спектра решаемых ими задач обуславливают необходимость предоставления уровней обслуживания, соответствующих разным категориям пользователей и классам задач. На сегодняшний день наблюдается дефицит высокоуровневых инструментальных средств, позволяющих создавать и применять такого рода мультиагентные среды на основе агентов, представленных в виде сервисов, с ориентацией на некоторую предметную область. Кроме того, в существующих инструментариях мало внимания уделяется интерфейсу, с помощью которого различные категории пользователей (администраторы, разработчики, конечные пользователи) могут взаимодействовать с мультиагентной средой. В данной статье предлагается оригинальная архитектура пользовательского веб-интерфейса мультиагентной среды, основанная на перспективной концепции Agent-as-a-Service, для реализации взаимодействия между пользователями и агентами среды. Предлагаемая архитектура базируется на принципе построения толстого клиента, при котором все основные компоненты веб-приложения загружаются в веб-браузер пользователя и дальнейшая генерация веб-страниц осуществляется динамически на стороне клиента. Архитектурный каркас веб-приложения, функционирующего в веб-браузере пользователя, спроектирован с применением схемы MVC (модель-представление-контроллер), при которой модель данных веб-приложения, пользовательский интерфейс (представление) и логика работы (контроллер) разделены на три слабозависимых компонента, что позволяет выполнять модификацию какого-либо из компонентов с минимальным воздействием на другие. Представленный в статье подход к построению подсистемы пользовательского веб-интерфейса обладает рядом преимуществ: снижает нагрузку на узел агента, повышает время отклика веб-интерфейса, улучшает кросс-платформенность и встраиваемость компонентов пользовательского интерфейса в различные подсистемы мультиагентной среды, облегчает тестирование и поддержку веб-приложения, а также решает ряд проблем с обеспечением безопасности.

Ключевые слова: мультиагентные технологии, архитектура веб-приложений, визуальный пользовательский интерфейс.

В настоящее время все большую актуальность приобретают исследования, связанные с повышением интеллектуализации систем организации *распределенных вычислительных сред* (РВС) [1] на основе мультиагентных технологий. Актуальность таких исследований обусловлена необходимостью более эффективного интегрированного использования разнородных ресурсов РВС, а также высокоуровневой поддержки пользователей в процессе создания и запуска масштабируемых приложений в РВС. Пользователи являются, как правило, специалистами в разных предметных областях. Каждый пользователь предъявляет свои субъективные требования к выполнению приложений и использованию общих ресурсов среды. Наличие большого числа пользователей РВС и разнообразие спектра решаемых ими задач обуславливают необходимость предоставления уровней обслуживания, соответствующих разным категориям пользователей и классам задач. Одним из подходов для решения описанных задач является использование *мультиагентных систем* (МАС) с экономическими механизмами управления распределенными вычислениями, обеспечивающее повышение эффективности вычислительных процессов для пользователей в зависимости от степени их заинтересованности в таком повышении [2]. Однако на сего-

дняшний день наблюдается дефицит высокоуровневых инструментальных средств, позволяющих создавать и применять МАС на основе агентов, представленных в виде сервисов, с ориентацией на некоторую предметную область [3]. Кроме того, в существующих инструментариях мало внимания уделяется интерфейсу, с помощью которого различные категории пользователей (администраторы, разработчики, конечные пользователи) могут взаимодействовать с мультиагентной средой.

При проектировании и реализации пользовательских интерфейсов применяется широкий спектр различных архитектурных подходов (MVC, MVP, MVVM, MDA и др.), используются различные инструментарии и технологии для создания как серверной части веб-приложения (JavaEE, Node.js, Yii, Django и др.), так и клиентской части (jQuery, AngularJS, ReactJS и др.). При реализации современных веб-приложений все больше проявляются тенденции к применению сервис-ориентированного подхода [4] для организации базовой части веб-приложения в виде веб-сервиса. Однако на сегодняшний день недостаточно развиты средства для применения архитектурных подходов SOA и ROA [5] к интеграции разнородных компонентов веб-приложения. Расширение возможностей применения приведенных подходов, с точки зрения

автора, позволит выполнять безболезненную модификацию и расширение набора компонентов веб-приложения, взаимодействующих на основе протокола SOAP либо архитектурного стиля RESTful [6], осуществлять динамическую реконфигурацию и адаптацию пользовательского интерфейса для различных категорий пользователей, расширять функции пользовательского интерфейса администратором РВС с гарантированным уровнем безопасности и надежности системы.

В данной статье предлагается оригинальная архитектура пользовательского веб-интерфейса для взаимодействия с агентами MAC, основанная на сервис-ориентированном подходе. Эта архитектура включает компоненты взаимодействия с разработчиками, администраторами и конечными пользователями MAC. Ее отличительной особенностью является использование перспективной концепции Agent-as-a-Service для взаимодействия между пользователями и агентами MAC.

Подсистема пользовательского веб-интерфейса MAC должна удовлетворять следующим исходным требованиям:

- являться легковесной автономной подсистемой, способной к встраиванию в различные компоненты мультиагентной среды;
- использовать архитектурные подходы SOA и ROA к интеграции разнородных компонентов веб-приложения и подсистем агента;
- обеспечивать независимость от ресурсоемких СУБД, системных сервисов и служб;
- поддерживать проблемную ориентацию создаваемой MAC;
- поддерживать все современные операционные системы для серверной части веб-приложения и все современные веб-браузеры для клиентской части веб-приложения;
- удовлетворять высоким требованиям обеспечения безопасности MAC;
- использовать отечественные и/или свободно распространяемые зарубежные библиотеки и средства создания пользовательского веб-интерфейса;
- предоставлять высокий уровень расширяемости веб-интерфейса с возможностью адаптации для различных нужд MAC;
- предоставлять пользователю информационно-справочные материалы с базовыми функциями экспертной подсистемы для разрешения проблемных ситуаций.

С целью создания подсистемы пользовательского веб-интерфейса, удовлетворяющей приведенным требованиям, была спроектирована и реализована архитектура (см. рисунок), включающая следующие основные составляющие:

- уровень пользователей (разработчики, пользователи и администраторы MAC) и сторонних программных систем, взаимодействующих с агентами MAC по прикладному программному интерфейсу;

- веб-приложение, загружаемое в веб-браузер пользователя и содержащее в своем составе весь необходимый перечень визуальных и графических элементов, моделей данных, декларативных описаний отображения графических элементов и данных модели, а также функциональной составляющей, определяющей логику работы веб-приложения и взаимодействия с агентом MAC;

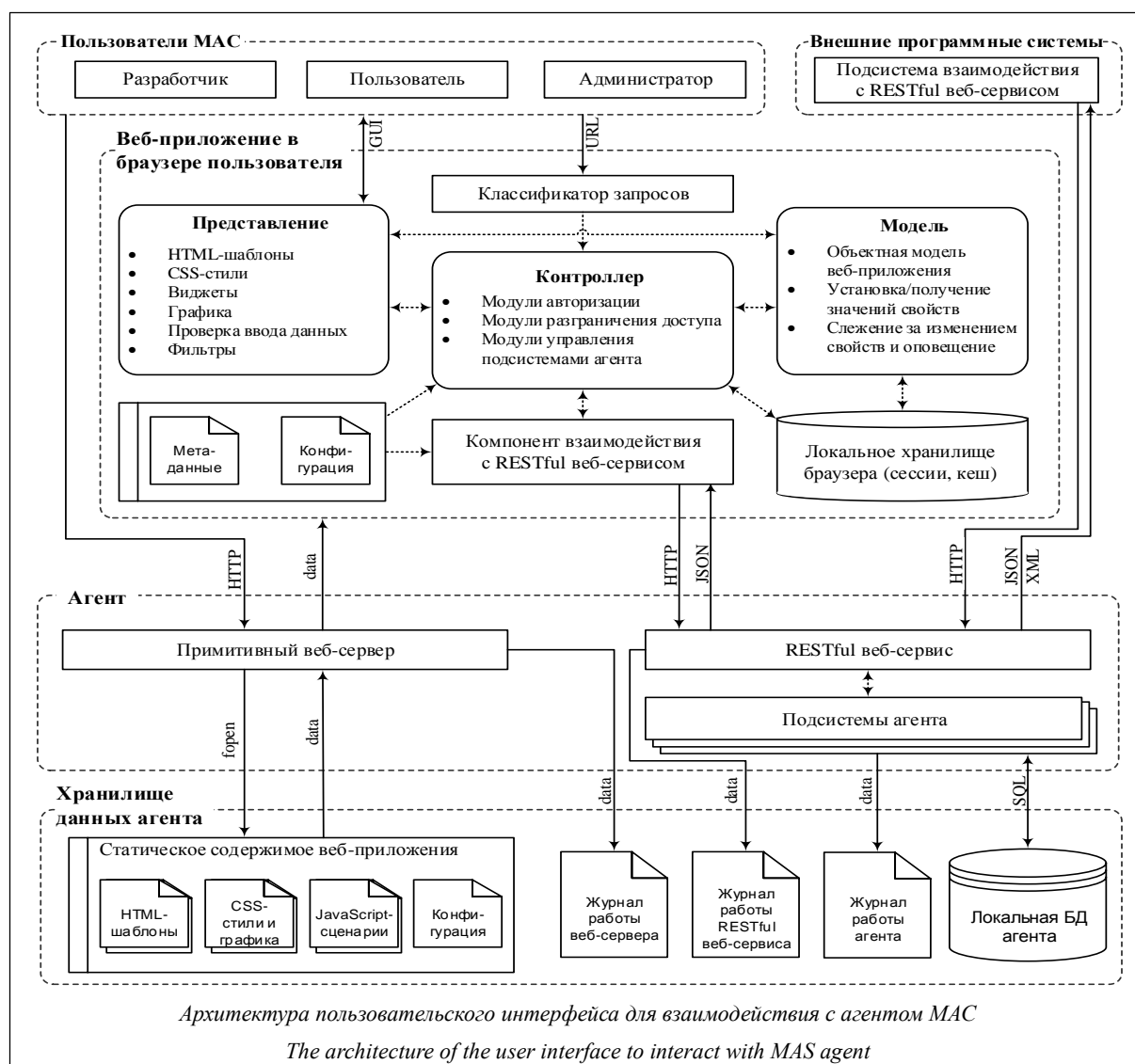
- примитивный веб-сервер, либо реализуемый в составе агента MAC, либо запускаемый независимо и выполняющий единственную задачу по передаче статических файлов веб-приложения в веб-браузер пользователя при первой загрузке;

- RESTful веб-сервис, реализуемый в составе агента и предоставляющий прикладной программный интерфейс для взаимодействия веб-приложения с подсистемами агента на основе протокола HTTP.

Приведенная архитектура базируется на принципе построения толстого клиента, при котором генерация веб-страниц осуществляется динамически на стороне веб-браузера. На сервере хранится статическая копия исходных файлов веб-приложения, которые передаются в веб-браузер при первом обращении пользователя и при дальнейшей работе пользователя с веб-приложением не обновляются. При этом единственной функцией веб-сервера является выполнение GET-запросов для передачи статических файлов по запросу веб-браузера. Реализация такого примитивного веб-сервера (с учетом соблюдения необходимых требований безопасности) составляет не более 150–200 строк программного кода на таких языках программирования, как Java, PHP, C++ (с использованием библиотек Boost или Qt), что позволяет осуществлять реализацию веб-сервера в составе агента. Кроме того, возможно использование существующих легковесных программных решений, таких как Mongoose [7], node.js [8], а также более комплексных программных продуктов Apache или nginx, однако применительно к поставленным задачам выбор таких решений не является рациональным.

Архитектурный каркас веб-приложения, функционирующего в веб-браузере пользователя, спроектирован с применением схемы MVC [9] (модель-представление-контроллер), при которой модель данных веб-приложения, пользовательский интерфейс (представление) и логика работы (контроллер) разделены на три слабозависимых компонента, что позволяет выполнять модификацию какого-либо из компонентов с минимальным воздействием на другие. Отдельные триплеты модели, представления и контроллера, выполняющие определенные задачи, объединяются в модули (к примеру, модули авторизации, управления настройками агента, управления ресурсами и др.).

При первом открытии в веб-браузер загружается главная страница (index.html), содержащая список ресурсов (HTML, CSS, JavaScript и др.), ко-



торые должны быть загружены дополнительно для инициализации и дальнейшей полноценной работы веб-приложения. Кроме того, главная страница содержит базовую структуру элементов пользовательского интерфейса и является оболочкой для всех отображаемых в дальнейшем веб-страниц, наполняемых данными требуемым контроллером на основе необходимого HTML-шаблона. Эти шаблоны содержат декларативное описание размещения графических и текстовых элементов с двусторонним связыванием с моделью данных.

При переходе по веб-страницам (по всему веб-приложению) не производится загрузка новых веб-страниц с веб-сервера, а загружаются только те данные, которые необходимо отобразить пользователю. Выявление переходов между веб-страницами выполняется классификатором запросов, который определяет действие пользователя и передает управление необходимому контроллеру. Если осуществляется переход на страницу, на которой должны быть отображены данные, получаемые от подсистем агента, контроллер делает запрос к ком-

поненту взаимодействия с RESTful веб-сервисом агента, который, в свою очередь, формирует и выполняет в асинхронном режиме Ajax-запрос по протоколу HTTP к RESTful веб-сервису агента. После получения ответа (в качестве формата передачи данных используется JSON) осуществляются проверка полученных данных, изменение значений параметров модели и дальнейшее отображение этих данных пользователю в составе необходимого HTML-шаблона.

В составе рассматриваемого пользовательского веб-интерфейса на текущий момент частично реализованы следующие модули:

- аутентификация и авторизация пользователей для выполнения дальнейших операций на основе выданного уникального ключа (token);
- взаимодействие с агентом: получение списка заданий и подробной информации об отдельном задании, добавление в очередь нового задания с указанием параметров запуска, изменение параметров исполняемого задания, удаление задания из очереди;

– выполнение операций с ресурсами агента (загрузка, получение, изменение и удаление файлов заданий);

– изменение настроек агента (изменение квот, политик управления заданиями и др.).

Для указанных операций используются четыре основных метода CRUD (Create, Retrieve, Update, Delete), реализуемых при построении RESTful веб-сервисов в виде методов HTTP-протокола: POST, GET, PUT и DELETE. При реализации описанного подхода использованы следующие средства и библиотеки: HTML5, CSS3, JavaScript, jQuery, Bootstrap, модифицированная версия Angular 2 [10].

Представленный в статье подход к построению подсистемы пользовательского веб-интерфейса обладает рядом преимуществ: снижает нагрузку на веб-сервер, функционирующий на узле агента, за счет вынесения логики приложения и функций генерации веб-страниц на сторону веб-браузера пользователя; повышает время отклика веб-интерфейса (передаются только данные); улучшает кросс-платформенность и встраиваемость компонентов пользовательского интерфейса в различные подсистемы мультиагентной среды; облегчает тестирование и поддержку веб-приложения, а также решает ряд проблем с обеспечением безопасности.

Исследование выполнено при частичной финансовой поддержке РФФИ, проекты №№ 15-29-07955-офи м и 16-07-00931, а также при частичной финансовой под-

держке Совета по грантам Президента Российской Федерации для государственной поддержки ведущих научных школ Российской Федерации НШ-8081.2016.9.

Литература

1. Гегель В.П., Линева А.В. Проблемы и перспективы достижения экзафлопного уровня производительности суперкомпьютерных систем // Вестн. Нижегород. ун-та им. Н.И. Лобачевского. 2012. № 3–1. С. 189–198.
2. Toporkov V.V., Emelyanov D.M. Economic model of scheduling and fair resource sharing in distributed computations. *Programming and Comp. Soft.*, 2014, vol. 40, no. 1, pp. 35–42.
3. Бычков И.В., Опарин Г.А., Феоктистов А.Г., Сидоров И.А., Богданова В.Г., Горский С.А. Мультиагентное управление вычислительной системой на основе метамониторинга и имитационного моделирования // Автоматизация. 2016. Т. 52. № 2. С. 3–9.
4. Buyya R., Vecchiola C., Selvi S.T. Mastering cloud computing. Burlington, Massachusetts, USA, Morgan Kaufmann Publ., 2013, 469 p.
5. Guo X., Shen J., Yin Z. On software development based on SOA and ROA. *Control and Decision Conf.*, Publishing Press, 2010, pp. 1032–1035.
6. Pautasso C., Zimmermann O., Leymann F. RESTful web services vs. big web services: making the right architectural decision. *Proc. 17th Intern. World Wide Web Conf.*, Beijing, China, 2008, pp. 805–814.
7. Mongoose Embedded Web Server Library. URL: <https://github.com/cesanta/mongoose> (дата обращения: 06.10.2016).
8. Gackenheim C. Node.js recipes: a problem-solution approach, Apress, 2013, 376 p.
9. Scott E.A. SPA design and architecture: understanding single page web applications, Manning Publ., 2015, 275 p.
10. Deeleman P. Learning Angular 2, Packt Publ., 2016, 354 p.

Software & Systems

DOI: 10.15827/0236-235X.116.023-026

Received 07.10.16

2016, vol. 29, no. 4, pp. 23–26

USER INTERFACE ARCHITECTURE FOR INTERACTION WITH MULTI-AGENT ENVIRONMENT

I.A. Sidorov¹, Ph.D. (Engineering), Research Associate, ivan.sidorov@icc.ru

¹ Institute of System Dynamics and Control Theory SB RAS, Lermontov St. 134, Irkutsk, 664033, Russian Federation

Abstract. The paper discusses an approach to the development of user interface to interact with a multi-agent environment, which is intended to solve resource-intensive tasks and includes a lot of loosely-coupled multivendor computing resources. Nowadays, there is a lack of toolkits which allow creating and using this kind of multi-agent environment, with a focus on a certain subject area. In addition, in the existing toolkits they do not pay enough attention to a graphical user interface, in which different categories of users (administrators, developers, and end users) can interact with the multi-agent environment. This paper proposes an original architecture of the user interface based on the promising concept of Agent-as-a-Service to implement the interaction between users and agents of the environment. The proposed architecture is based on the principle of building a thick client, in which components of a web-application are loaded into user's web-browser, and further web pages generation is performed dynamically on the client side. The architecture of a web application is designed using MVC scheme (model-view-controller), where a web-application model of data, user interface (view) and logic (controller) are divided into three weakly dependent components that allow modifying any component with minimal effect on the other. The presented approach to the development of the user web interface has several advantages: it reduces the load on an agent's node, increases the response time of a web interface, enhances portability and embeddability of user interface components in various subsystems of a multi-agent environment, facilitates testing and support of web applications, and solves a number of safety problems.

Keywords: multi-agent technologies, architecture of web applications, graphical user interface.

Acknowledgements. The research was supported by RFBR, projects no. 15-29-07955-ofi m and no. 16-07-00931-a, and partially supported by the Council for Grants of the President of the Russian Federation for state support of the leading scientific schools, project NSh-8081.2016.9.

References

1. Gergel V.P., Linev A.V. Problems and prospects for exaflop performance level in supercomputer systems. *Vestnik Nizhegorodskogo Universiteta* [Bulletin of Lobachevsky Univ. of Nizhni Novgorod]. 2012, vol. 3, no. 1, pp. 189–198 (in Russ.).
2. Toporkov V., Emelyanov D. Economic Model of Scheduling and Fair Resource Sharing in Distributed Computations. *Programming and Computer Software*. 2014, vol. 40, no. 1, pp. 35–42.
3. Bychkov I.V., Oparin G.A., Feoktistov A.G., Sidorov I.A., Bogdanova V.G., Gorsky S.A. Multi-agent Control of Computational Systems on the Basis of Meta-monitoring and Imitational Simulation. *Avtometriya* [Optoelectronics, Instrumentation and Data Processing]. Allerton Press, 2016, vol. 52, no. 2, pp. 107–112 (in Russ.).
4. Buyya R., Vecchiola C., Selvi S.T. Mastering Cloud Computing, Burlington, Massachusetts, USA, Morgan Kaufmann Publ., 2013, 469 p.
5. Guo X., Shen J., Yin Z. On software development based on SOA and ROA. *Control and Decision Conf. (CCDC)*. Publishing Press, 2010, pp. 1032–1035.
6. Pautasso C., Zimmermann O., Leymann F. RESTful Web Services vs. Big Web Services: Making the Right Architectural Decision. *Proc. 17th Int. World Wide Web Conf.* Beijing, China, 2008, pp. 805–814.
7. Mongoose Embedded Web Server Library. Available at: <https://github.com/cesanta/mongoose> (accessed October 6, 2016).
8. Gackenheim C. *Node.js Recipes: A Problem-Solution Approach*. Apress, 2013, 376 p.
9. Emmit S. *SPA Design and Architecture: Understanding Single Page Web Applications*. Manning Publ., 2015, 275 p.
10. Deeleman P. *Learning Angular 2*. Packt Publ., 2016, 354 p.

УДК 004.414.22

DOI: 10.15827/0236-235X.116.027-035

Дата подачи статьи: 01.08.16

2016. Т. 29. № 4. С. 27–35

МЕТОДОЛОГИЧЕСКИЕ АСПЕКТЫ УПРАВЛЕНИЯ ЖИЗНЕННЫМ ЦИКЛОМ ИНФОРМАЦИОННОЙ СИСТЕМЫ НА ОСНОВЕ ИНСТРУМЕНТОВ ФУНКЦИОНАЛЬНОЙ СТАНДАРТИЗАЦИИ

*О.В. Лукинова, д.т.н., ведущий научный сотрудник, lobars@mail.ru
(Институт проблем управления им. В.А. Трапезникова РАН,
ул. Профсоюзная, 65, г. Москва, 117342, Россия)*

В работе представлена методология управления полным жизненным циклом информационной системы, синтезирующая инструменты и модели функциональной стандартизации, теории открытых систем, управленческого консалтинга, базовых стандартов в области управления жизненным циклом информационной системы. Описаны основные положения и показана концептуальная модель, отражающая взаимосвязь задач, решаемых в рамках методологии, а именно: построение модели автоматизируемых бизнес-процессов предприятия средствами управленческого консалтинга, разработка модели информационной системы, а также ее стандартизация в соответствии с моделью OSE/RM (Open System Environment/Reference Model) при помощи функционального профиля, выбор варианта жизненного цикла, разработка его модели и профилирования. Исследованы различные варианты жизненных циклов, особое внимание уделено жизненному циклу покупной системы, а также системы, включающей, помимо локального компонента (развернутого на предприятии заказчика), совокупность интегрированных публичных сервисов. Показано (в терминах модели открытой среды OSE/RM) структурное представление не только целевой компоненты проектируемой информационной системы, но и системы ее защиты как единого создаваемого объекта. Приведена концептуальная модель механизмов защиты, совокупность которых составляет функциональную структуру системы защиты.

Ключевые слова: информационная система, функциональный и технологический профиль, модель открытой системы, свойства открытых систем, функциональная стандартизация.

Современные интегрированные информационно-телекоммуникационные системы представляют собой наиболее сложный класс *информационных систем* (ИС) с точки зрения методов и средств, используемых в ходе проработки и реализации их *жизненных циклов* (ЖЦ). К этим системам предъявляются весьма высокие требования, связанные с ответственностью их функционирования, необходимостью адаптации к непрерывно изменяющимся условиям бизнеса в рыночной экономике, информационного взаимодействия со смежными ИС. Кроме того, противоречие между возрастающей сложностью ИС и сокращением сроков и средств, отпускаемых на проектирование и сопровождение системы, требует внедрения в процесс управления ЖЦ определенных подходов.

Вопросам управления ЖЦ ИС посвящены многие публикации, например [1–3]. Данную работу отличают следующие особенности:

- создаваемая система проектируется как открытая на основе стандартизованного структурированного представления;
- проблема стандартизации и самой ИС, и ее ЖЦ, связанная с необходимостью применения открытых спецификаций, решается в рамках систематизированного подхода;
- предлагается новый способ описания видов деятельности ЖЦ, которые зачастую имеют неоднозначную интерпретацию в силу их существующей смысловой разницы;
- объекты *бизнес-процесс – приложение ИС – платформа* рассматриваются как пирамида услуг (сервисов), оказываемых по иерархии и в конечном счете ориентированных на пользователя.

Таким образом, в статье описан методологический подход к управлению ЖЦ сложных ИС, основанный на объединении современных инструментов *функциональной стандартизации* (ФС), моделей теории открытых систем, базовых стандартов и моделей управления ЖЦ, предпроектных консалтинговых исследований, что позволит повысить качество выходного продукта, процесса разработки, а также управляемость этим процессом.

Концептуальная модель методологии

Методология представляет собой инструмент для эффективного выполнения проектов по созданию, сопровождению и модификации ИС (причем под созданием здесь понимаются как разработка системы под заказ, так и внедрение покупного тиражируемого варианта). Она разработана для управления полным ЖЦ существования ИС на предприятии [1] и представлена следующими аспектами.

- Проведение консалтинговых исследований и построение модели бизнес-процессов, подлежащих автоматизации. Указанные модели интерпретируются как совокупность сервисов, требующихся предприятию для ведения своей деятельности.
- Представление ИС, приложения которой являются средой реализации бизнес-процессов, базируется на функциональных референсных моделях открытых систем, разработанных Комитетом IEEE POSIX (IEEE Std 1003.0-1005) и получивших развитие в [4, 5]. Модели не только являются основой для применения ФС, но и позволяют выстроить

требования системных интерфейсов между приложениями и платформенными сервисами.

- Моделирование различных вариантов ЖЦ ИС.
- Стандартизация ИС и ее ЖЦ с использованием инструментов методологии ФС, позволяющей описывать на нормативном уровне сложные объекты по частям в рамках систематизированного представления.

Концептуальная модель, отражающая взаимосвязь положений и задач, решаемых в рамках указанных аспектов, приведена на рисунке 1.

Как видно из рисунка, инициацией применения методологии является наличие у заказчика потребности в автоматизации своих бизнес-процессов. Поэтому первая задача, которую необходимо решить, – это проведение консалтинговых работ, в ходе которых строится бизнес-модель автоматизированных процессов предприятия. Бизнес-модель имеет самостоятельное значение для предприятия и, что крайне важно в контексте данной работы, является основой для формирования бизнес-требований к будущей ИС [6, 7].

Центральное место на схеме отведено объекту «Создание ИС», являющемуся, с одной стороны, средой реализации бизнес-процессов в виде своих приложений, а с другой – результатом (продуктом)

проекта по созданию ИС. Этот объект является водоразделом между сферой заказчика в части выставления бизнес-требований к ИС и сферой разработчика в части удовлетворения этих требований. Для получения качественного результата будущая ИС (или ее модификация) должна быть разработана как открытая система в соответствии с концепцией и моделью открытых систем *OSE/RM* (Open System Environment/Reference Model) [4], а также стандартизована как сложный объект (сложность в данном случае предполагает невозможность описания объекта в рамках одного стандарта) на основе инструментов ФС.

Верхняя часть схемы (рис. 1) представляет задачи, связанные с ЖЦ корпоративной ИС. Как видно из рисунка, необходимо построить модель варианта ЖЦ в соответствии с тем или иным базовым стандартом, наполнить ее соответствующими видами деятельности и стандартизовать те виды деятельности, которые будут выбраны и выстроены на ЖЦ (здесь также используются инструменты ФС).

Модель ИС как открытой системы

Важнейшим аспектом предлагаемой методологии является использование концептуальных поло-

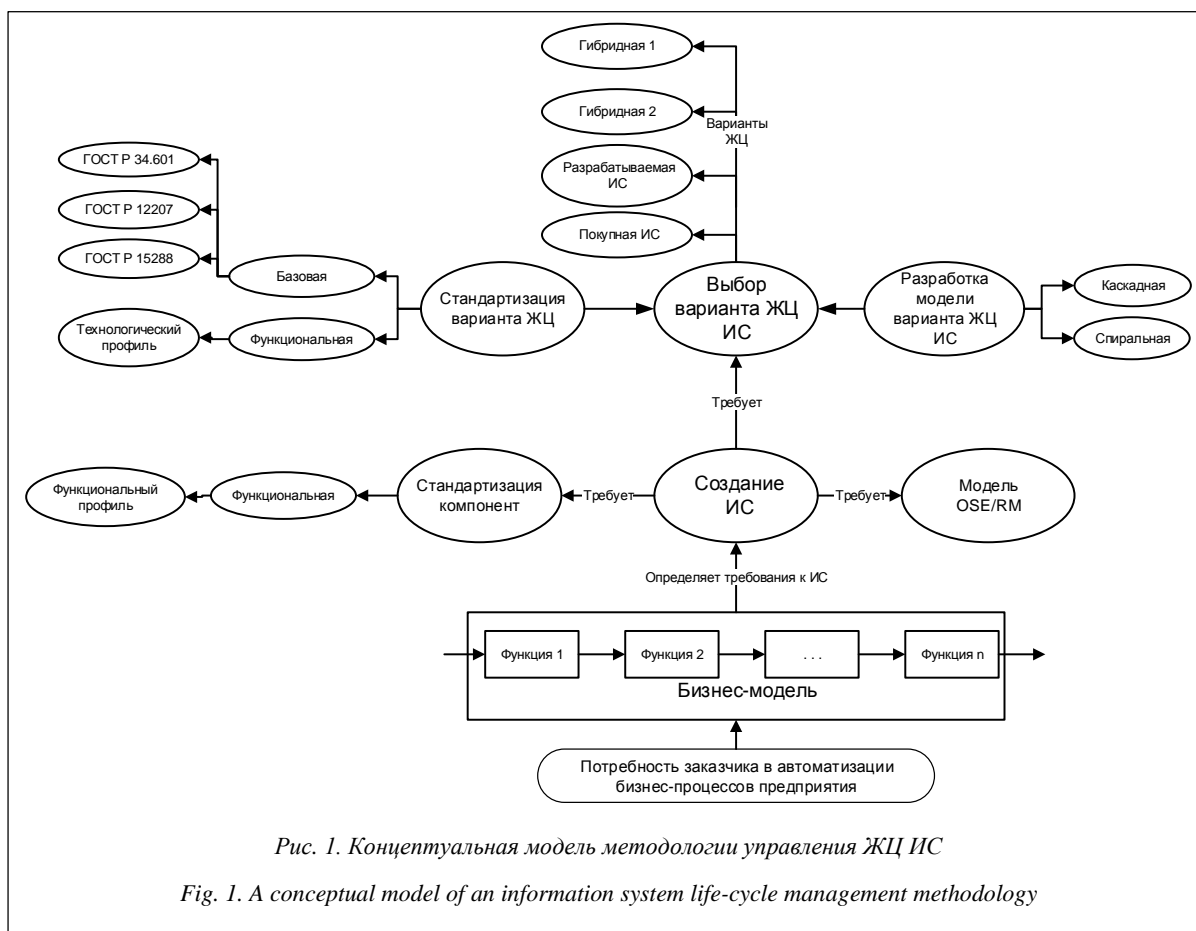


Fig. 1. A conceptual model of an information system life-cycle management methodology

жений теории открытой среды, в рамках которой ИС характеризуется определенными свойствами, а именно:

- расширяемость (*extensibility*) – возможность добавления новых прикладных функций ИС или изменения некоторых функций из числа уже реализованных без изменения при этом остальных функциональных частей (подсистем) ИС;

- мобильность приложений и данных (*portability*) – возможность перевода ИС на более совершенные аппаратно-программные платформы при их модернизации или замене с минимальными затратами, сохраняя вложенные инвестиции (обеспечивается соблюдением принятых стандартизованных программных интерфейсов (*API-Application Program Interface*) между приложениями и функциональной средой открытых систем);

- мобильность пользователей (*user portability*), обеспечиваемая дружественным пользовательским интерфейсом (поддерживается стандартизованными *API* среды по функциям пользовательского интерфейса);

- интероперабельность (*interoperability*), означающая возможность взаимодействия данной ИС с другими системами при необходимости обращения к информационным ресурсам;

- масштабируемость (*scalability*) – возможность изменения количественных характеристик (размерности решаемых задач, числа обслуживаемых пользователей и т.д.) путем настройки параметров, а не перепроектирования и программирования заново; рост количественных системных характеристик при добавлении определенных вычислительных ресурсов (например, процессоров, модулей оперативной и дисковой памяти в конфигурациях серверов);

- способность к интеграции – возможность объединения нескольких ИС различного назначения в единую интегрированную многофункциональную ИС.

Для представления ИС и реализации свойств открытости используется модель открытой среды *OSE/RM*, которая синтезирует как прикладную функциональность, так и другие аспекты системы – системы администрирования, защиты и пр. – на референсном (эталонном, справочном) уровне.

Модель (рис. 2) представляет ИС прежде всего как сочетание прикладной (*App*) и платформенной (*Platf*) компонент, которые структурируются на три уровня и четыре функциональные группы в каждом:

- компоненты служб и сервисов промежуточного, системно-прикладного слоя (*MW*);
- компоненты операционных систем или операционного слоя (*OW*);
- аппаратный слой (*HW*).

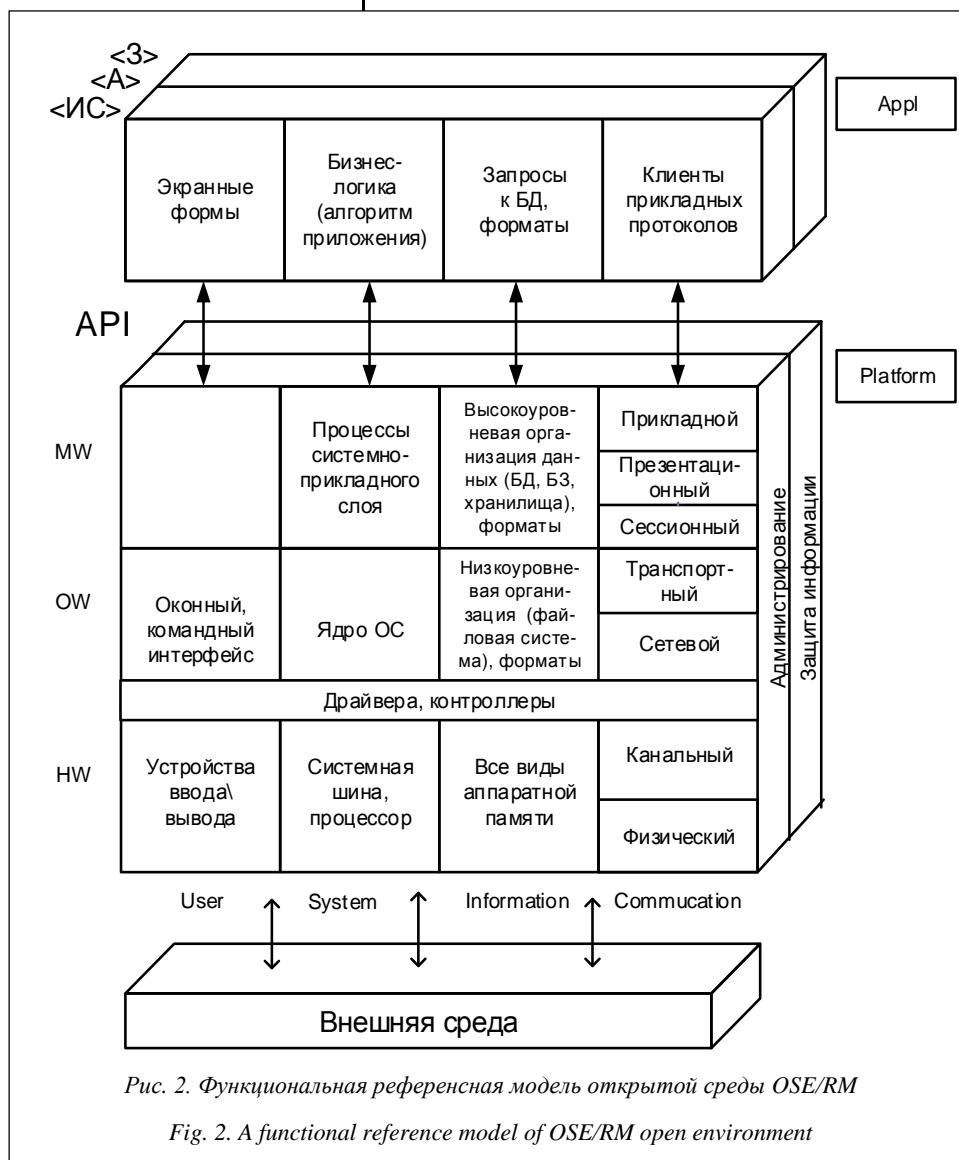


Рис. 2. Функциональная референсная модель открытой среды *OSE/RM*

Fig. 2. A functional reference model of *OSE/RM* open environment

Функциональные группы компонентов в данной модели составляют компоненты, обеспечивающие интерфейс с пользователем (*User*), все необходимые процессы в системе (*System*), организацию, представление, доступ и хранение данных (*Information*), а также компоненты телекоммуникационной среды, обеспечивающие взаимосвязь ИС (*Communication*) (данный уровень представляет собой модель взаимосвязи открытых систем (*OSI/RM* – *Open System Interconnection/Reference Model*)).

Кроме того, модель трехмерна. Она имеет дополнительные плоскости, в частности, администрирования и защиты, которые в зависимости от контекста аккумулируют сервисы, обслуживающие соответствующие клетки целевой плоскости.

Следует заметить, что при любых вариантах ЖЦ, отмеченных на рисунке 1, ИС может быть представлена моделью *OSE/RM*. При этом возможны два варианта: а) создание системы с нуля, которое заключается в разработке/покупке/аренде, интеграции реализаций всех компонент (клеток) модели; б) модификация существующей системы – внесение изменений в некоторые компоненты без изменения остальных.

Если ИС создаваемая, то ее частями (в соответствии с плоскостями модели *OSE/RM*) будут целевая компонента – передняя плоскость модели <ИС>, система администрирования – плоскость администрирования <А> и система безопасности – плоскость защиты <З>, обеспечивающая заданный уровень свойств конфиденциальности, целостности, доступности и др. относительно ресурсов бизнес-процесса.

Естественно, эти части желательно создавать параллельно в одном проекте, хотя на практике система безопасности зачастую дорабатывается в процессе эксплуатации целевого компонента.

Для модифицируемой ИС результатом проекта может быть изменение существующих конкретизаций некоторых клеток модели или разработка и добавление новых:

- приложение;
- программное средство платформы:
 - системная компонента операционного слоя;
 - изменение или разработка СУБД, СУБЗ, хранилища данных и т.п.;
 - телекоммуникационная компонента системы, обеспечивающая взаимосвязь систем (она представлена клетками столбца *Communication* модели *OSE/RM*, хотя возможна и разработка нового приложения);
 - интеграционная компонента системы, обеспечивающая взаимодействие систем в части данных, БД, приложений (строится на основе телекоммуникационной компоненты и соответствующих средств клетки «Процессы системно-прикладного слоя»);
 - аппаратное средство.

Аналогичный перечень может использоваться для всех трех плоскостей модели, то есть и для це-

левой системы, и для систем безопасности и администрирования.

Плоскость защиты <З> представляется в двух аспектах.

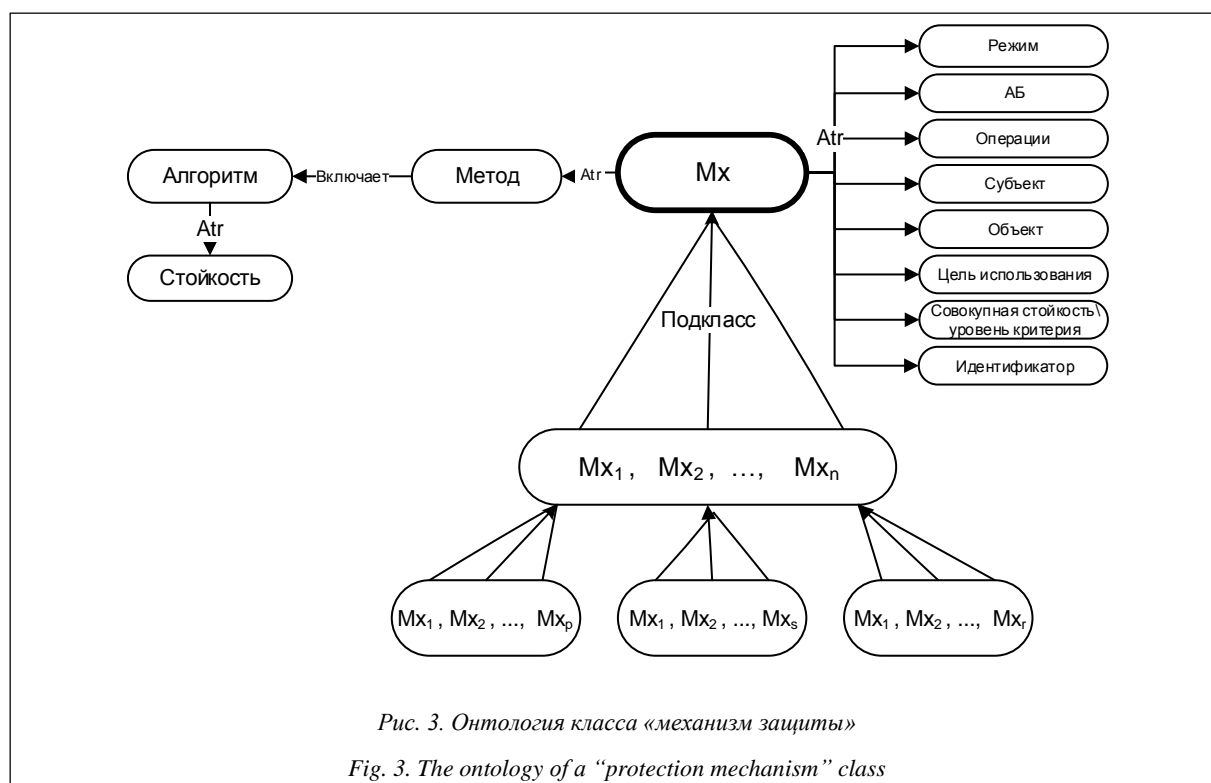
Первый аспект. Клетки плоскости интегрируют совокупности механизмов, обеспечивающих защиту реализаций соответствующих клеток базовой плоскости <ИС>, плоскости <А> и саму себя – межкатегорийный аспект. При этом защитные механизмы (*Mx*) – достаточно сложные по своему содержанию сущности. Для дальнейшей структуризации механизмов по плоскости необходимо представить внутреннюю структуру *Mx* в виде некоторой модели. Для этой цели были использованы семантические модели в виде онтологий. На рисунке 3 представлена онтология класса механизмов защиты {*Mx*}.

Каждый *Mx* класса референсный и представляет собой иерархию механизмов-подклассов, действие которых направлено на разные объекты ИС. Назовем механизмы-подклассы механизмами первой очереди *Mx*¹, второй очереди *Mx*² и т.д. в зависимости от номера иерархического слоя. Например, механизм идентификации и аутентификации имеет иерархию вложенности в две очереди. Действие конечных механизмов таксономии направлено на обеспечение безопасности различных клеток модели *OSE/RM*.

Каждый *Mx* в иерархии может быть осуществлен каким-либо {Методом}, а каждый метод представляет собой программную или аппаратную реализацию некоторого {Алгоритма}. Именно алгоритмы обеспечивают тот или иной уровень безопасности.

Участниками *Mx* являются *субъект, операции над объектом*, которые он осуществляет. Каждый из этих участников обладает некоторыми атрибутами. Поэтому концепт {*Mx*} имеет набор атрибутов: {Режим}, {АБ}, {Операции}, {Субъект}, {Объект}. Режим определяет то, как будет использоваться механизм, например, значения режима могут быть «периодический» или «избирательный». Атрибут {АБ} определяет атрибуты безопасности, присущие каждому участнику *Mx*. Для механизмов разных очередей перечни атрибутов или их значения могут быть различными. Атрибут {Операции} задает действия, которые осуществляются над объектом или субъектом, а значения атрибутов {Субъект}, {Объект} определяют конкретные субъекты и объекты, над которыми *Mx* осуществляет свои операции. {Совокупная стойкость/Уровень критерия} определяет уровень свойств безопасности, необходимый для защиты соответствующей клетки модели *OSE/RM*. Указанные сущности для разных механизмов различны. Конечные механизмы таксономий (листья онтологий) уже можно соотносить с клетками *OSE/RM*.

Второй аспект. Реализация *Mx* плоскости <З> в виде ИС требует ее структуризации по аналогии с



базовым представлением <ИС> в виде прикладной и платформенной компонент модели *OSE/RM* с поправкой на контекст. Прикладная компонента включает приложения, представляющие собой программную реализацию тех *Mx*, которые реализуют межкатегорийное представление каждой клетки *OSE/RM*.

Уровни платформенной компоненты имеют следующую специфику:

- к *HW*-слою, помимо стандартной аппаратуры, относятся также аппаратные защитные средства (*AMx*), которые используются в соответствующих клетках;
- операционный уровень может представляться как стандартной операционной системой, обслуживающей запросы приложений плоскостей <ИС>, <А> и приложений <З>, так и специфической, работающей только на приложения системы защиты;
- системно-прикладной слой в данном случае обслуживает и организует выполнение приложений защиты.

Описание вариантов ЖЦ ИС

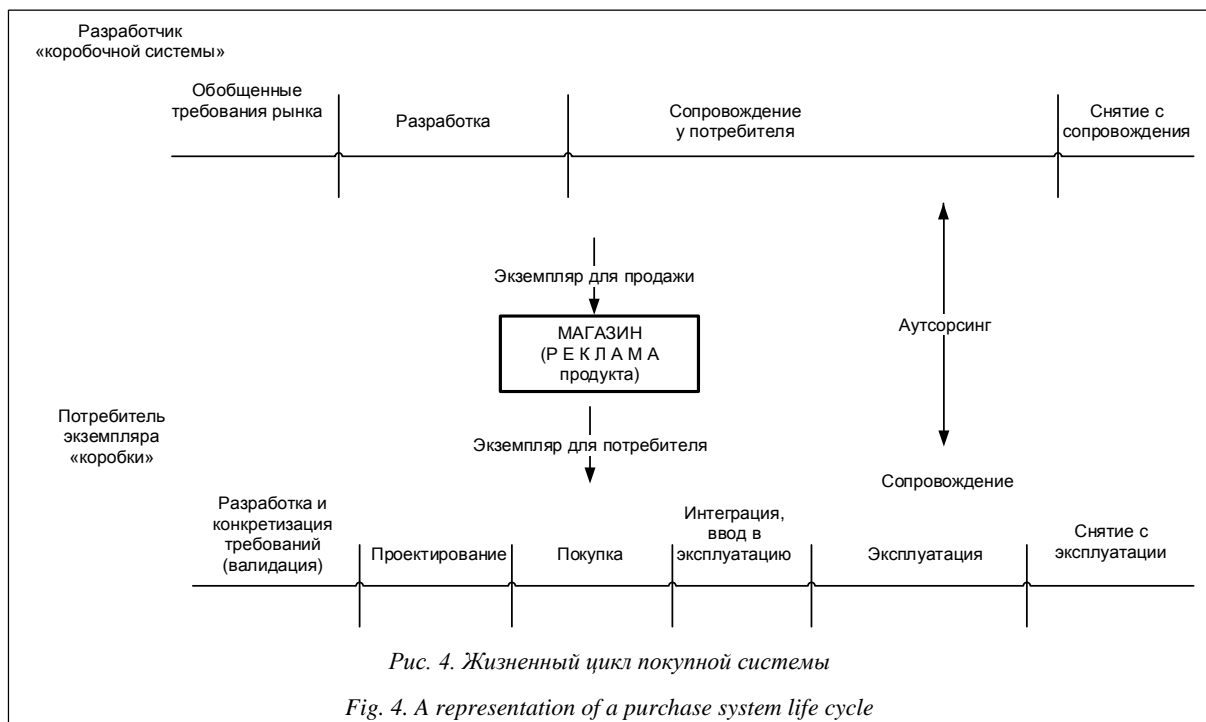
Способ приобретения предприятием КИС целиком и компонент ИС может быть различным. При этом ЖЦ во всех случаях строится по-разному и требует различных наборов работ. Поэтому методология предусматривает следующие варианты создания ИС.

1. В заказном варианте, когда предприятие заказывает разработку системы под свою спецификацию,

под реализацию конкретных бизнес-процессов. ЖЦ такой системы традиционен. На рисунке 1 этот вариант идентифицируется как «разрабатываемая ИС». Это может быть разработка с нуля, когда предприятие автоматизируется впервые, а может быть модификация уже существующей системы, но в любом случае это разработка под заказ.

2. Покупная система, когда компоненты системы приобретаются в готовом виде, зачастую у различных вендоров. ЖЦ покупной системы отличается от традиционного (рис. 4). Здесь речь идет о приобретении заказчиком *экземпляра системы* у производителя, поэтому на рисунке изображено представление ЖЦ как со стороны производителя, так и со стороны заказчика, поскольку, как правило, их отношения фактом покупки не ограничиваются, а продолжаются на стадии эксплуатации посредством договоров аутсорсинга на сопровождение системы.

Для заказчика важно не просто определиться с бизнес-требованиями, предъявляемыми к ИС, но и сопоставить их с тем, что предлагается на рынке, то есть произвести валидацию своих требований. Кроме того, в этом случае особую значимость приобретает процесс интеграции покупных компонент. Для этого необходимо наличие разработанных моделей *OSE/RM* и их нормативных спецификаций, которые фактически определяют перечень необходимых компонент системы и их характеристики. Кроме того, компоненты и их интерфейсы должны быть выполнены в соответствии с *USIC*-структуризацией открытых систем (рис. 2), что позволит без труда интегрировать их между собой.



3. ЖЦ гибридных ИС. В последнее время активно стали конфигурироваться так называемые гибридные КИС, когда для реализации бизнес-процесса заказчика к локальному компоненту добавляется ряд облачных сервисов. Это означает, что часть функций бизнес-процесса, то есть часть функциональных требований заказчика, реализуются за счет собственных вычислительных ресурсов, а другая часть – за счет публичных. При этом локальная часть может быть поставлена как в заказном, так и в тиражируемом варианте. ЖЦ этих вариантов имеет специфику и их следует рассматривать отдельно.

В качестве локального компонента рассматривается ИС, установленная и функционирующая непосредственно на предприятии заказчика. Под облачным сервисом понимается *стандартизованная* ИТ-услуга, которую *любой* провайдер может поставить на рынок и которой может воспользоваться *любой* пользователь. Это определение обуславливает следующие особенности, свойственные облачному публичному сервису (не путать с ИТ-услугами (сервисами) частных «облаков»).

- Функциональное содержание сервиса, реализованное любым поставщиком, должно отражать стандарты или иные законодательные акты, принятые в данной предметной области. Потребитель должен быть уверен: если он воспользовался, например, сервисом подготовки отчета в ПФР, этот отчет будет выполнен в соответствии с нормативными требованиями ведомства.

- Поскольку такой сервис должен быть доступным для любого пользователя, в Сети необходимо наличие общеизвестного ресурс-реестра, куда провайдер должен занести описание функци-

онального содержания предлагаемой услуги и интерфейса, по которому пользователь сможет к этой услуге обратиться. Сама же услуга (ее код) находится на ресурсах провайдера-поставщика.

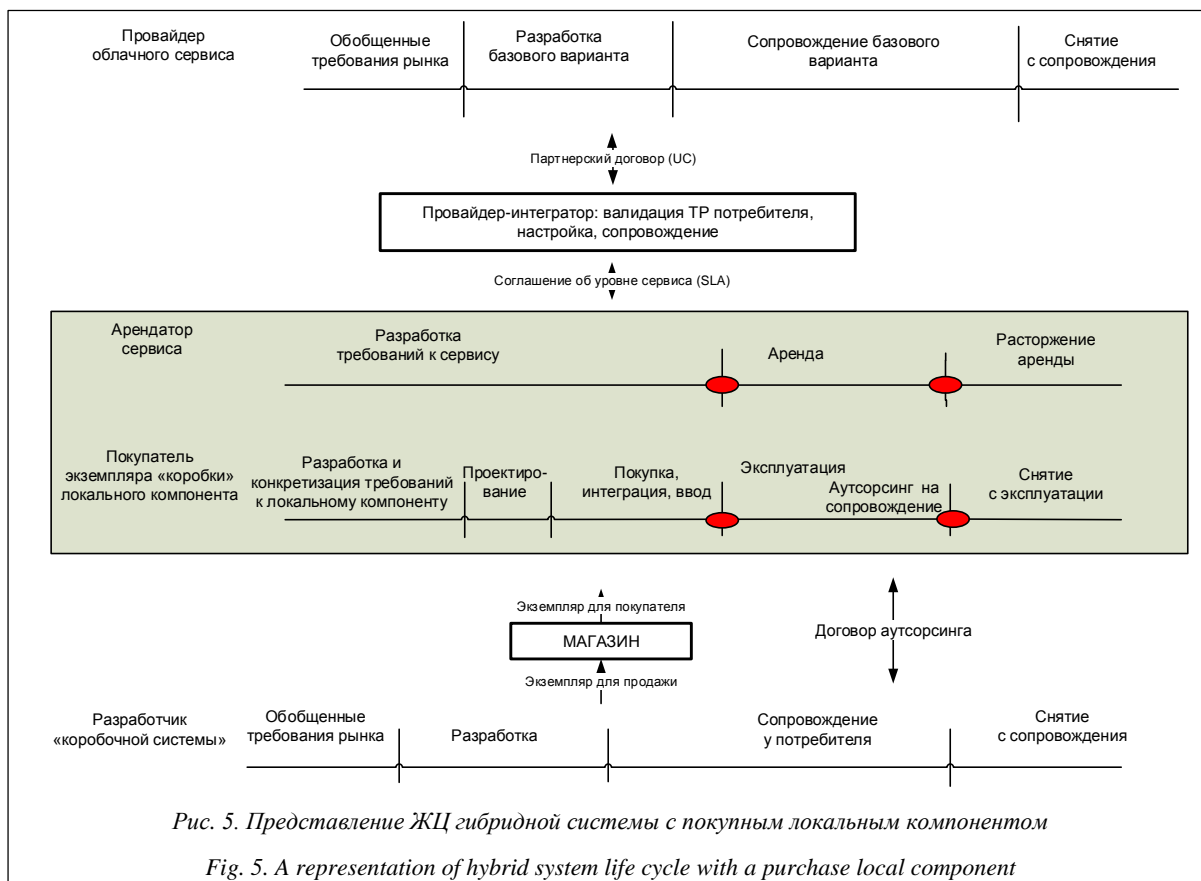
Этим свойством *multitenancy* (разделяемости провайдеров и разделяемости пользователей) отличаются ИТ-услуги, предлагаемые в частных облаках: услуги, предлагаемые на ресурсах корпоративных ЦОД, эксплуатируются на принципах функционального взаимодействия, когда разработчики и потребители всегда могут напрямую договориться о способах обращения к услуге и ее требуемом содержимом. ИТ-услуга публичного облака фактически базируется на идеологии web-сервисов.

- Третья особенность использования публичных сервисов заключается в том, что очень часто потребителю приходится «докручивать» сервис под свои нужды. (В частных облаках, кстати, такой проблемы нет.) Тогда возникает потребность введения в число участников процесса (предприятия и поставщика сервисов) провайдера-интегратора, задача которого состоит в том, чтобы осуществлять доводку облачных сервисов в соответствии с бизнес-процессом заказчика.

ЖЦ гибридной КИС, состоящей из покупного локального компонента и совокупности облачных сервисов, представлен на рисунке 5.

Схема отражает ЖЦ компонент гибридной системы с точки зрения четырех участников процесса:

- заказчика гибридной системы, являющегося покупателем тиражируемого экземпляра коробочной системы и арендатором облачных сервисов (средний фрагмент схемы);



– разработчика тиражируемой системы, осуществляющего продажу системы через магазин и, как правило, сопровождение на условиях аутсорсинга;

– провайдера облачных сервисов, осуществляющего разработку базового варианта сервиса на основе обобщенных требований рынка, сопровождение базового варианта сервиса и снятие его с сопровождения в случае отсутствия требований на аренду данного сервиса;

– провайдер-интегратора облачных сервисов, в рамках соглашений об уровне сервиса осуществляющего валидацию требований заказчика с предложениями рынка, настройку нужных сервисов, их интеграцию и сопровождение сервиса под требования заказчика.

ЖЦ гибридной системы с разрабатываемым локальным компонентом аналогичен соответствующим поправкам.

Профилирование ИС и ЖЦ

При создании и развитии сложных распределенных ИС требуются гибкое формирование и применение гармонизированных совокупностей базовых стандартов и нормативных документов для унификации и регламентирования заданных функций ИС. Для этого используется методология ФС. Структуризация функциональности ИС в виде модели *OSE/RM* создает основу для применения мето-

дологии ФС, основным инструментом которой является понятие профиля.

По ГОСТ Р ИСО/МЭК ТО 10000-1-99, *профиль* – это совокупность нескольких (или подмножество одного) базовых стандартов (и других нормативных документов) с четко определенными и гармонизированными подмножествами обязательных и факультативных возможностей, предназначенная для реализации заданной функции или группы функций.

Для стандартизации ИС используются функциональные профили, регламентирующие архитектуру и структуру системы и ее компонентов в соответствии с компонентами и плоскостями модели *OSE/RM*. Следовательно, для любой ИС должны быть определены профили

- приложений;
- среды ИС, включающие в себя спецификации программных интерфейсов между приложениями и средой;
- интерфейсов между ИС и внешней для нее средой;
- администрирования целевой системы;
- защиты информации (включает профили межкатегорийного представления и базовой функциональности системы безопасности) [8].

Кроме набора функций, эти профили должны описывать интерфейсы взаимодействия соответствующих компонентов системы как с другими компонентами того же горизонтального уровня мо-

дели, так и с компонентами выше- и нижележащего уровней.

Таким образом, проектирование ИС в значительной степени сводится к ее компоновке из стандартизованных узлов. Этот подход позволяет осуществлять развитие и модернизацию ИС путем добавлений или замены отдельных узлов без изменения других частей системы.

Кроме функциональных профилей, в составе профилей ИС должны рассматриваться *вспомогательные профили*, регламентирующие процессы создания, сопровождения и развития ИС, то есть профили процессов ЖЦ. Такие профили в [9] называются технологическими. Прежде чем строить технологический профиль, необходимо для выбранного варианта ЖЦ разработать модель, основанную на нормативно-правовой базе и соответствующим образом структурированную.

Из существующих в настоящее время моделей наиболее распространены две: каскадная и спиральная. Они принципиально различаются самим подходом к моделированию ЖЦ ИС и ее ПО. Каскадная модель более универсальна, то есть применима к производству разных изделий и позволяет достичь хороших результатов, когда в самом начале разработки можно достаточно точно и полно сформулировать все требования, которые в процессе разработки изменяются незначительно. Спиральная модель более ориентирована именно на ИС, особенно на программные продукты, поэтому при разработке ИС и их ПО она предпочтительнее каскадной. Что касается стандартов и методологических документов, относящихся к ЖЦ систем и их ПО, действующих в РФ, то базовыми являются ГОСТ 34.601-90, ISO/IEC 12207 и ISO/IEC 15288 и их российские аналоги ГОСТ Р ИСО/МЭК 12207-2010 и ГОСТ Р ИСО/МЭК 15288-2005. С точки зрения методологических подходов к управлению ЖЦ систем указанные документы имеют принципиальные различия. Так, ГОСТ 34.601-90 достаточно жестко устанавливает стадии и этапы создания ИС и отражает идеологию каскадной модели ЖЦ. Международные стандарты ISO/IEC 12207, ISO/IEC 15288 и соответствующие российские аналоги концептуально более развитые. В них заложена принципиально иная идея управления проектированием ИС.

В отличие от жесткой каскадной схемы ЖЦ, регламентируемой ГОСТ 34.601-90, указанные стандарты интерпретируются как библиотеки процессов, которые, вообще говоря, покрывают все пространство задач, возникающих при создании и сопровождении ИС. Но какие конкретно задачи надо решать при разработке конкретной ИС, то есть какие процессы выбирать из библиотеки, в какой последовательности их выстраивать, это дело исключительно экспертного суждения.

Профиль ЖЦ представляет собой набор стандартов, упорядоченный на основе выбранной мо-

дели ЖЦ, которой могут быть каскадная модель, каскадная модель с промежуточным контролем (эволюционная, итерационная) или спиральная модель. В [10] подробно описан алгоритм построения технологического профиля ЖЦ ИС в соответствии с нормативной базой.

Набор функциональных и технологических профилей составляет *полный профиль ИС*.

Профиль ЖЦ «закрывает» еще одну проблему указанных стандартов: все они имеют трехуровневую иерархию видов деятельности: стадия/процесс, этап/подпроцесс, работа/задача (здесь до «/» указаны термины по ГОСТ 34, после – по ГОСТ 12207), и если для первого уровня деятельности стандарты предусматривают достаточно содержательную интерпретацию (хотя бывают исключения), то для деятельности нижних уровней семантика зачастую размывается. И это понятно, суть выполняемых работ здесь слишком вариативна, зависит от многих факторов: технологии разработки, специфики разрабатываемой ИС, вариантов компоновки ЖЦ и т.п.

В заключение отметим, что результаты работы целесообразно использовать в следующих ситуациях. Во-первых, при реализации проектов по разработке сложных систем, включая систему безопасности, с целью представления результата проекта – открытой ИС, а также объекта управления проекта – ЖЦ ИС – в стандартизованном виде, а также понимания взаимосвязи между бизнес-процессами предприятия, ИС и защитой, выстраиваемой исходя из соображений безопасности бизнес-процесса. Во-вторых, даже если система состоит из покупных компонент, перечисленные цели не теряют своей актуальности при дальнейшем сопровождении ИС у заказчика.

Кроме того, изложенный подход позволяет заказчику сохранить уже сделанные инвестиции при изменении требований к ОИС и ее развитию, использовать информационные ресурсы, эксплуатируемые в других системах, сократить затраты на обучение персонала при переходе на новые версии ИС, не зависеть от одного поставщика технических и программных средств, повторно использовать готовые приложения, а также облегчает решение проблемы «унаследованных» систем и т.п.

Литература

1. Батоврин В.К., Бахтурин Д.А. Управление жизненным циклом технических систем: Сер. докл. СПб, 2012. Вып. 1. 59 с.
2. Вендров А.М. Проектирование программного обеспечения. М.: Финансы и статистика, 2005. 544 с.
3. Позин Б.А. Ввод в действие информационных систем и сопровождение их программного обеспечения // Новые технологии. 2010. № 4. С. 1–32.
4. Бойченко А.В., Кондратьев В.К., Филинов Е.Н. Основы открытых информационных систем. М.: Изд-во Евразийского открытого ин-та, 2004. 128 с.
5. Лукинова О.В. Методология проектирования систем защиты, построенных на основе референсной модели POSIX OSE/RM // Системы высокой доступности. 2012. № 3. С. 38–45.

6. Калянов Г.Н. Модели и методы теории бизнес-процессов (обзор) // Открытое образование. 2015. № 6. С. 4–9.
7. Васильев Р.Б., Калянов Г.Н., Левочкина Г.А. Направления стратегического ИТ-консалтинга // Автоматизация в промышленности. 2009. № 11. С. 3–8.
8. Лукинова О.В., Пугачев А.В. Особенности построения профилей систем безопасности ИС // Открытое образование.

2015. № 4. С. 80–87.

9. Липаев В.В., Филинов Е.Н. Формирование и применение профилей открытых информационных систем // Открытые системы. 1997. № 5. С. 3–11.

10. Бойченко А.В., Лукинова О.В. Управление жизненным циклом информационной системы на основе профилей // Системы высокой доступности. 2015. Т. 11. № 3. С. 64–68.

Software & Systems

DOI: 10.15827/0236-235X.116.027-035

Received 01.08.16

2016, vol. 29, no. 4, pp. 27–35

METHODOLOGICAL ASPECTS OF INFORMATION SYSTEM LIFE CYCLE MANAGEMENT BASED ON FUNCTIONAL STANDARDIZATION TOOLS

O.V. Lukinova¹, Dr.Sc. (Engineering), Leading Researcher, lobars@mail.ru

¹ V.A. Trapeznikov Institute of Control Sciences of RAS, Profsoyuznaya St. 65, 117997, Russian Federation

Abstract. The paper presents a methodology for managing the life cycle of information systems. It synthesizes tools and models of functional standardization, open system theories, management consulting, and basic standards in management of information system life cycle.

The paper also describes the basics and a conceptual model showing the relationship between problems to be solved using the methodology framework. The problems are the following: construction of a business process model using management consulting; information system model development in accordance with the OSE/RM model (Open System Environment / Reference Model) and its standardization using a functional profile; choosing a life cycle type, development of its respective model and profiling.

The paper investigates different types of a life cycle, focusing on a purchase system life cycle including a set of integrated public services in addition to a local component (deployed at a customer site). The structural representation of the information system target components together with its security system is shown in terms of the OSE/RM model. The article also describes a conceptual model of a protection mechanism (which is a base element of a protection system functional structure).

Keywords: information system, functional and technological profile, open system model, open system properties, functional standardization.

References

1. Batovrin V.K., Bakhturin D.A. *Upravlenie zhiznennym tsiklom tekhnicheskikh sistem: ser. dokl.* [Technical System Life Cycle Management: report series]. St. Petersburg, 2012, iss. 1, 59 p.
2. Vendrov A.M. *Proektirovanie programmnoy obespecheniya* [Software Design]. Moscow, Finansy i statistika Publ., 2005.
3. Pozin B.A. Implementation of information systems and their software support. *Novye tekhnologii* [New Technologies]. 2010, no. 4, pp. 1–32 (in Russ.).
4. Boychenko A.V., Kondratev V.K., Filinov E.N. *Osnovy otkrytykh informatsionnykh sistem* [The Basics of Opened Information Systems]. Kondratev V.K. (Ed.). Moscow, EAOI Publ., 2004, 128 p.
5. Lukinova O.V. Methodology of design of systems of the security, constructed on the basis of the reference POSIX OSE/RM model. *Sistemy vysokoy dostupnosti* [Highly Available Systems]. 2012, no. 3, pp. 38–45 (in Russ.).
6. Kalyanov G.N. Models and methods of a business process theory (review). *Otkrytoe obrazovanie* [Open Education]. 2015, no. 6, pp. 4–9 (in Russ.).
7. Vasilev R.B., Kalyanov G.N., Levochkina G.A. Directions of strategic IT consulting. *Avtomatizatsiya v promyshlennosti* [Automation in Industry]. 2009, no. 11, pp. 3–8 (in Russ.).
8. Lukinova O.V., Pugachev A.V. Construction features of IP security systems' sections. *Otkrytoe obrazovanie* [Open Education]. 2015, no. 4, pp. 80–87 (in Russ.).
9. Lipaev V.V., Filinov E.N. Formation and application of open information system profiles. *Otkrytye sistemy* [Open Systems]. 1997, no. 5, pp. 3–11 (in Russ.).
10. Boychenko A.V., Lukinova O.V. Management of life cycle of information system on the basis of profiles. *Sistemy vysokoy dostupnosti* [Highly Available Systems]. 2015, no. 3, iss. 11, pp. 58–63 (in Russ.).

УДК 004.94

DOI: 10.15827/0236-235X.116.036-044

Дата подачи статьи: 28.06.16

2016. Т. 29. № 4. С. 36–44

ИНФОРМАЦИОННАЯ МОДЕЛЬ СЕМАНТИЧЕСКОЙ БИБЛИОТЕКИ *LibMeta*

О.М. Атаева, младший научный сотрудник, oli@ultimeta.ru

(Вычислительный центр им. А.А. Дородницына Федерального исследовательского центра
«Информатика и управление» РАН, ул. Вавилова, 40, г. Москва, 119333, Россия)

В данной статье библиотеки рассматриваются как информационные системы, обеспечивающие основную функциональность для работы с библиотечными данными. Развитие технологий и возможностей сети переопределяет понятие как самих библиотек, так и ее ресурсов, которые сегодня не ограничиваются только библиографическими записями и их электронным представлением, а выводят на передний план семантику этих ресурсов.

Благодаря развитию технологий пользователь библиотеки получает дополнительные возможности для работы с ресурсами цифровых библиотек с помощью описания области своих интересов в терминах предметной области на основе стандартов с привлечением словарей, тезаурусов и онтологий. Это позволяет ему организовывать и описывать и собственные коллекции, и собственные ресурсы, при необходимости детализируя как описания ресурсов, так и область своих интересов путем уточнения ее терминов.

В работе рассматриваются основные требования к таким библиотекам и описывается информационная модель разрабатываемой системы, особенностью которой является возможность интеграции данных из источников, интегрированных в облако LOD.

Ключевые слова: семантические библиотеки, LOD, информационная модель, интеграция данных.

В данной статье, говоря о библиотеках, будем иметь в виду информационные системы, обеспечивающие основную функциональность для работы с библиотечными данными. Обычно при описании библиотек предполагают, что ее ресурсы представляют собой библиографические записи традиционных библиотек и электронные копии документов, описываемых этими записями. Но развитие технологий и возможностей сети переопределяет понятие как самих библиотек, так и ее ресурсов, которые не ограничиваются теперь только библиографическими записями и их электронным представлением, а также выводит на передний план семантику этих ресурсов. Для этого разработаны различные виды классификации ресурсов библиотеки – отраслевые рубрикаторы, позволяющие более детально определять тематическую направленность ресурсов. Зачастую этих средств для описания семантики недостаточно, со временем появляются новые требования к описанию ресурсов библиотек, что приводит к усложнению самих описаний и требует значительных трудозатрат на внедрение новых способов описаний, соответствующих текущим потребностям. Увеличивающийся поток поступающих объектов практически невозможно обработать вручную, нужны новые методы обработки и анализа поступающих данных.

В современных библиотеках сами ресурсы становятся более разнообразными и могут включать самые разнотипные объекты. Например, электронная библиотека «Научное Наследие России» [1], заявленная как проект по созданию библиотеки полнотекстовых научных трудов известных российских и зарубежных ученых и исследователей, включает в себя и описания музейных экспонатов, расширяя тем самым традиционные типы хранимых ресурсов классической библиотеки.

Semantic Web, Linked Open Data и библиотеки

В последнее десятилетие возросла популярность парадигмы Semantic Web [2], одним из практических воплощений которой стало сообщество, поддерживающее публикацию данных в сети согласно принципам LOD (Linked Open Data) [3]. Основным преимуществом этого подхода является возможность провязывания ресурсов из различных источников данных, при описании которых используются онтологии, содержащие метаописания самих метаданных ресурсов.

Основная идея LOD заключается в решении задач интеграции данных сети, для чего предлагается представлять информацию в формализованном виде, что делает ее доступной для машинной обработки. Единицей описываемых данных в Semantic Web является ресурс. Каждый ресурс обозначает какой-либо реальный объект, понятие или явление и имеет идентификатор URI (Unified Resource Identifier), который используется для описания знаний о сущности. Эти знания представляются в соответствии с моделью данных RDF (Resource Definition Framework) [4] в виде троек «субъект–предикат–объект».

Организация специального пространства связанных данных Linked Data основывается на практических решениях для публикации и связывания структурированных данных. Термин LOD описывает ту часть данных Linked Data, которая находится в открытом доступе и соответствует основным принципам LOD. Идея LOD очень привлекательна для различных организаций, многие из которых включили свои источники данных в это облако. Оказались провязанными самые различные типы ресурсов, которые представляют интерес для

пользователей библиотек с точки зрения обогащения данных как структурно, так и семантически.

Главные проблемы уже существующих наборов данных в LOD на текущий момент – это разнообразие терминов и разобщенность данных. В разных наборах данных могут использоваться различные онтологии для описания модели данных. Классический случай – библиографические онтологии, описывающие модель данных для ведения библиографических записей печатных изданий. Часто встречаются библиотеки, контент которых – это набор тематических ресурсов, для их поддержки составляется соответствующая онтология. Например, в качестве ресурсов могут выступать некоторые мультимедийные объекты, для описания которых классические библиографические онтологии, такие как BIBO или SPAR, непригодны. Эти проблемы являются следствием нестандартизованности процесса публикации набора данных в пространство LOD, а также иллюстрируют важность тщательного выполнения интеграции и реализации возможностей семантического поиска.

В связи с развитием технологий пользователь библиотеки может получать больше возможностей для работы с ресурсами цифровых библиотек с помощью описания области своих интересов в терминах предметной области на основе стандартов с привлечением словарей, тезаурусов и онтологий. Это позволит ему организовывать и описывать собственные коллекции и собственные ресурсы, при необходимости детализируя как описания ресурсов, так и область своих интересов посредством уточнения ее терминов.

Определения библиотек

Формально *электронная библиотека* представляет собой тройку объектов $\langle F, C, A \rangle$, где F – множество функций хранения и поиска, обеспечиваемых информационной системой для обработки объектов множества C . Объекты из C обладают фиксированным набором атрибутов (a_1, \dots, a_k) , $a_i \in A$. Этот набор будем называть *описанием множества C* или *метаданными множества C* .

Множество C будем называть контентом библиотеки, а любой объект $c \in C$ – информационным объектом. Тогда описание отдельного информационного объекта обозначим как $c(a_1, \dots, a_k)$. При этом значениями a_i могут быть только символьные наборы из некоторого алфавита L . Набор атрибутов и символьные значения этих атрибутов для объекта $c \in C$ – метаданные этого объекта. Значения атрибута a_i обозначим $c(a_i) \in L^*$, где L^* – множество всех строк (включая пустую строку), составленных из символов, входящих в L . Множество F состоит из функций вида $f: (a_1, \dots, a_j) \rightarrow In$, $In \subset C$, j принимает значения от 1 до k .

Фактически контент электронных библиотек представляет собой множество библиографиче-

ских записей объектов реальной классической библиотеки. В электронных библиотеках речь идет не о цифровом представлении копий реальных объектов, а лишь об их описаниях. В таких описаниях, например, встретив упоминание персоны в одном месте, невозможно точно установить соответствие с ее упоминанием в другом месте.

Цифровые библиотеки решают те же задачи поиска и хранения контента, что и электронные библиотеки, но существенно расширяют свою функциональность и определение своего контента. Во-первых, контент библиотеки становится мультимедийным. Это значит, что значениями атрибутов ее информационных объектов теперь могут выступать различные мультимедийные объекты, доступные для просмотра средствами самой цифровой библиотеки. В качестве мультимедийных объектов могут выступать совокупность аудио-, видео-, фото- и текстовых материалов. Во-вторых, расширяется функциональность за счет решения некоторых задач интеграции как метаданных, так и медийных объектов из внешних источников, доступных по сети.

При этом формальное определение представляет собой уже набор объектов $\langle F, C, A, M \rangle$, где F, C, A определяются так же, как и в электронных библиотеках; M – множество доступных мультимедийных объектов, $c(a_i) \in (M \cup L^*)$.

Множество F дополняется функциями вида $g: (X, a_1, \dots, a_j) \rightarrow Out$, где $Out \subset C(X)$ и $Out \subset C$, X – внешний источник, а множество объектов $C(X)$ может быть описано набором атрибутов (a_1, \dots, a_k) . Функции g предназначены для решения вопросов интеграции данных из внешних библиотек.

Семантические цифровые библиотеки являются следующим этапом в эволюции библиотек и обязаны своей популярностью семантическим технологиям, которые в значительной степени повлияли на переосмысление понятия контента библиотеки и послужили толчком для расширения и улучшения функциональности библиотек. В таких библиотеках данные лучше структурированы, выделены связи между ними, что улучшает поиск и дает возможность интегрировать данные различных типов. Обеспечивается интероперабельность с другими системами, необязательно являющимися библиотеками, так как основной задачей семантических технологий является предоставление метаданных в машиночитаемом формате.

Формально семантическая цифровая библиотека – это $\langle F, C, A, \Phi, T \rangle$, где F и A определяются так же, как и в цифровых библиотеках. Контент библиотеки $C = C_1 \cup C_2 \cup \dots \cup C_s$ представляет собой множество типов информационных ресурсов системы, для каждого из которых определен свой набор атрибутов (a_{i1}, \dots, a_{ik}) . Такое определение не означает исключение мультимедийных объектов, а подчеркивает обыденность мультимедийных объектов в семантических библиотеках, то есть $M \in C$,

и должно пониматься как добавление нового типа контента «мультимедийный объект» в библиотеку со своим набором атрибутов и отношений, каждый объект которого является абстрактным представлением реального объекта из множества M . Значения атрибутов $c(a_{ij}) \in (M \cup L^* \cup C)$; L^* , как и прежде, содержит область значений строковых атрибутов из A . Φ задает множество условий, накладываемых на представление контента, которое может, например, содержать ограничения, накладываемые на форматы значений $c(a_{ij})$. T – множество терминов предметной области, предназначенных для терминологической поддержки описания экземпляров информационных ресурсов множества C . Объектом T может быть подмножество элементов L^* или множество подмножеств из L^* . Элементы множества T могут быть связаны различными отношениями между собой, образуя простые таксономии (линейный словарь, классификатор с иерархическими связями) или сложные таксономии (таксономии с горизонтальными связями), а также могут быть связаны отношениями с объектами из множества C .

Важную роль в определении семантических библиотек при описании ее контента играют онтологии. Онтология модели контента фактически задается $\langle C, A, \Phi, T \rangle$, где множество C выступает как множество понятий онтологии, множество атрибутов A также содержит подмножество отношений между понятиями, а Φ задает множество функций интерпретации, заданных на понятиях и отношениях. Таким образом, множества C, A, Φ задают описание структуры контента библиотеки, тогда как объекты множества T терминологически ограничивают предметную область контента библиотеки.

Основные свойства семантических библиотек

Основным свойством семантических библиотек является возможность структурирования их разнородного контента и связывания данных из разных источников, что, в свою очередь, отражается на качестве данных контента.

Основные свойства семантической библиотеки, которые, на взгляд автора, являются определяющими для рассматриваемой системы:

- семантическая библиотека представляет собой интеграционный узел для разных источников данных, которые обогащают и пополняют ее набор данных;
- контент библиотеки описывается на семантическом уровне, что позволяет достичь лучшего взаимодействия между источниками данных;
- контент библиотеки может иметь разную степень гранулярности структуры в зависимости от рассматриваемых начальных условий при построении библиотеки;

- семантическое описание контента и его уровень гранулярности не зависят от технических характеристик реализации информационной системы библиотеки;

- понятийное описание контента библиотеки поддерживается его тезаурусом, который ограничивает предметную область ресурсов библиотеки терминологически.

Информационные системы в контексте семантических библиотек

Выделив модель контента семантической библиотеки и ее основные характеристики, отделим определяющее понятие контента семантической библиотеки от понятия реализующей библиотеку информационной системы [5]. Такой подход позволяет наращивать функциональность системы, добавлять новые подсистемы или изменять уже имеющиеся при неизменных остальных частях.

Информационная система IS задается набором подсистем F для решения задач обработки ее контента I , $IS = (F, I)$. Тогда IS представляется как организация совокупности своих подсистем $F = \cup F_i$ и своего контента I . Каждая из этих подсистем описывается своей предметной онтологией, и тогда можно представить *онтологию информационной системы* $OnIS$ объединением онтологий ее подсистем и онтологий ее контента $OnIS = OnF \cup OnI$, где $OnF = \cup OnF_i$ – объединение онтологий подсистем, $OnI = \langle C, A, \Phi, T \rangle$ – онтология контента. При описании онтологий информационных систем и ее модулей обычно опираются на абстрактные онтологии высокого уровня, определяя ее ключевые сущности. Более подробно этот подход освещается в [6].

Основные виды задач реализуются в информационной системе подсистемами:

- описания контента информационной системы;
- реализации задач интеграции данных из внешних источников;
- реализации задач интеграции данных из внутренних источников;
- поддержки коллекций;
- поиска и навигации по объектам системы;
- поддержки пользователей;
- управления тезаурусом;
- качества данных в системе.

Такое разбиение на подсистемы не является единственно возможным. Границы подсистем не могут быть строго определены. В системе существует область общих определяющих понятий, которые рассматриваются как принадлежащие нескольким подсистемам, в зависимости от того, какие процессы выполняются в конкретной подсистеме. Каждая из этих систем так или иначе взаимодействует с понятиями, определяющими контент этой библиотеки. Например, в перечисленных

подсистемах можно рассматривать как единую подсистему реализации задач интеграции данных из внешних и внутренних источников. С другой стороны, из подсистемы качества можно выделить отдельно систему выявления дубликатов. Это деление диктуется конкретной реализацией.

Краткий обзор некоторых семантических библиотек

Семантическая библиотека JeromeDL [7] является одной из попыток объединить возможности, предлагаемые концепцией и технологиями Semantic Web, с библиотечками, ориентируясь на тесное взаимодействие с пользователями. Фактически она представляет собой интегрированное приложение для ведения цифровой библиотеки, блогов и сервиса для закладок. В рамках цифровой библиотеки поддерживаются авторитетные файлы (для авторов, редакторов, издательств), таксономии, используемые для классификации по темам, тезаурус WordNet [8] для ключевых слов. Каждый ресурс описывается тремя типами метаданных: структурными, библиографическими и социальными. Каждый тип метаданных поддерживается соответствующими сервисами. Пользователю предоставляется комбинированное представление на основе этих метаданных. Основные модели для описания ресурсов, пользователей и их взаимодействия – библиографическая онтология MarcOnt [9], онтология FOAF [10], модель знаний SKOS [11] для описания таксономий.

Основными недостатками, на взгляд автора, являются ориентированность только на библиографические данные и слабая поддержка интеграции данных с другими источниками в рамках системы, в частности, с ресурсами из LOD. При необходимости добавления нового типа ресурсов приходится вносить изменения в систему на программном уровне. Одним из преимуществ этой системы является поддержка, помимо сервиса традиционного атрибутивного поиска, сервисов семантического поиска данных на естественном языке, доступ к данным на языке запросов SPARQL [12] для возможности машинной обработки. Следует отметить, что система распространяется бесплатно.

Остальные решения в этой области, такие как Greenstone [13] и Briks [14], так или иначе уступают указанному проекту или предназначены для использования в рамках специализированных предметных областей.

Одной из глобальных реализованных цифровых библиотек является проект Europeana [15], интегрирующий данные из институтов культурного наследия Европы. Многоуровневая организация провайдеров контента предназначена для автоматической оценки контента на соответствие модели данных EDM (Europeana Data Model) [16], которая была разработана в рамках проекта. В рамках этой

модели данных определены наборы классов и свойств для описания объектов культурного наследия. Одно из преимуществ EDM – возможность соблюдения принципов связанных данных при описании ресурсов. Масштаб этой библиотеки одновременно является и одним из препятствий для возможности индивидуальной тематической работы пользователя и позволяет причислить ее к глобальным семантическим библиотекам, среди которых также можно указать DBpedia [17], являющаяся ядром облака LOD.

Информационная модель цифровой библиотеки LibMeta

Наиболее полной эталонной моделью электронной библиотеки является разработка группы DELOS (Digital Library Reference Model, DLRM) [18]. Были определены базовые для электронной библиотеки понятия (конкретная ЭБ, система ЭБ, система управления ЭБ), выделены категории пользователей для этих понятий (разработчик, пользователь, администратор) и шесть основных высокоуровневых понятий/областей: контент, пользователь, функциональные возможности, качество, политики, архитектура.

Опираясь на концептуальную модель DELOS и ее определения, а также на идеи Semantic Web и Linked Open Data, была разработана персональная открытая семантическая библиотека LibMeta с системой поддержки работы пользователей с цифровыми ресурсами библиотек и их коллекциями для некоторой предметной области.

Основные требования к системе LibMeta. LibMeta характеризуется настраиваемым хранилищем метаданных для своих ресурсов и типами описываемых информационных ресурсов. Основные требования к описанию ресурсов – универсальность, структурированность и адаптируемость. Универсальность – независимость описания ее типов ресурсов и объектов от предметной области и области интересов пользователей. Структурированность описания обеспечивает поддержку связей между различными типами ресурсов как внутри системы, так и вне ее, исходя из определений LOD. Адаптируемость описания ресурсов обеспечивает возможность добавления новых свойств и связей в процессе развития системы и обеспечивает настройку пользовательских интерфейсов под эти изменения. Далее приведены основные понятия подсистем LibMeta, которые обеспечивают соответствие этим требованиям, выведенным на базе формальной модели семантической библиотеки [19].

Подсистема описания контента информационной системы. За универсальность определения контента системы отвечает набор понятий, составляющих информационную модель контента библиотеки LibMeta: информационный ресурс и

информационный объект, которые описывают экземпляры ресурсов. Информационный ресурс является основной единицей описания контента библиотеки, а информационный объект представляет экземпляры информационных ресурсов. Каждый из них имеет собственный уникальный идентификатор в соответствии с требованиями LOD. Фактически семантическое значение информационного ресурса является эквивалентным понятию класса онтологии с некоторыми ограничениями в его описании. Структура описания информационных объектов определяется понятиями *атрибут* и *набор атрибутов*, которые задаются при описании соответствующего ресурса. Атрибут является элементом описания свойства ресурса, а набор атрибутов – коллекцией атрибутов разных видов. Типы атрибутов следующие: атрибут, файловый, объектный, числовой, текстовый, строковый. При подключении подсистемы управления таксономиями появляется новый вид атрибута – таксономический. Помимо определения круга значений атрибута, важными характеристиками являются тип и количество его значений [19].

Эти понятия обеспечивают структурированное описание контента и обеспечивают поддержку его адаптируемости. Такой подход также обеспечивает описание конкретных ресурсов и их объектов в виде RDF-троек и предоставления SPARQL точки доступа для публикации данных в LOD.

Конкретная реализация модели контента библиотеки может быть основана на некоторой импортируемой онтологии, классы которой превращаются в ресурсы, свойства описываются в терминах атрибутов LibMeta, наборы атрибутов определяют фактически домены свойств онтологий. При построении модели ресурсов библиотеки на основе этой онтологии сохраняются все URI свойств, от-

ношений и классов выбранной онтологии. При необходимости при импортировании выбранной онтологии в систему можно изменить набор понятий, расширив или сократив его средствами системы.

Конечно, такой способ отображения онтологии на понятия системы LibMeta не сохраняет весь возможный перечень ограничений, накладываемых на свойства и классы онтологии изначально, но ее структурная часть сохраняется, что является достаточным для решения задач, определенных в рамках системы.

На рисунке 1 приведены основные понятия, используемые для конструирования описания предметной области. Некоторые из понятий будут пояснены далее.

Подсистема управления тезаурусом. Для описания тезауруса введены дополнительные понятия: *таксон* и *таксономия*. Таксон представляет собой элемент таксономии с определенным набором свойств, необходимым для его базового представления, а таксономия определяет набор доступных связей между составляющими таксономию таксонами и ресурсами системы. Для описания дополнительных связей между таксонами вводятся отношения между ними, которые позволяют определять и описывать новые связи в рамках информационной системы. По умолчанию в системе доступны только два типа связей между таксонами: иерархическая и нетипизированная горизонтальная. На рисунке 2 представлены поддерживаемые подпонятия таксономии согласно определению семантической библиотеки: словарь, классификатор и тезаурус. На рисунке 3 отображаются используемые по умолчанию связи в таксономиях между определяющими их таксонами. Для тезауруса можно доопределить атрибуты, помеченные на ри-

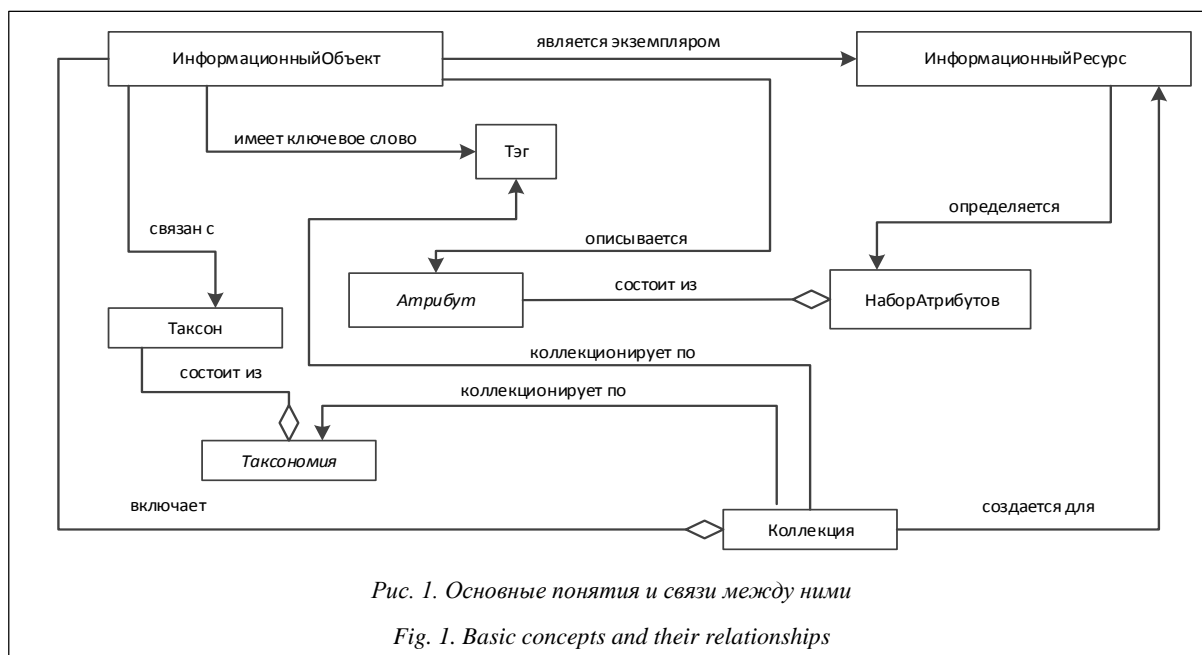


Рис. 1. Основные понятия и связи между ними

Fig. 1. Basic concepts and their relationships

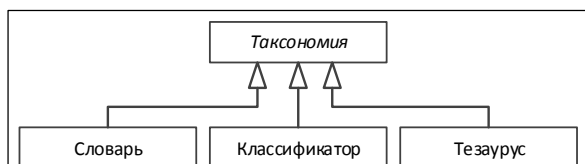


Рис. 2. Таксономии

Fig. 2. Taxonomies

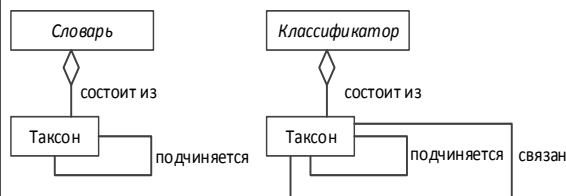


Рис. 3. Связь таксономий и таксонов

Fig. 3. Relations between taxonomies and taxons

сунке 4 как *T_ТекстовыйАтрибут*, *T_ОбъектныйАтрибут*, *T_ТаксономическийАтрибут*, которые позволяют расширить определение таксона, включать в его определение также информационные объекты и специфицировать горизонтальные связи между таксонами.

Для определения связи таксономий с информационными объектами вводится понятие *таксономического атрибута*. Это отношение обеспечивает возможность подключения любой из таксономий к любому типу ресурсов в процессе жизнедеятельности системы. Такой подход позволяет, с одной стороны, избежать избыточности на начальном этапе проектирования системы, с другой – обеспечить представление практически любых связей. Таксо-

номические атрибуты задаются при описании области значений атрибутов информационных ресурсов.

Подсистема поддержки коллекций. Для возможности ведения разнообразных коллекций объектов используется понятие *коллекция информационных объектов*, которая определяется на основе некоторой таксономии с указанием коллекционируемых типов ресурсов. Коллекция может объединять информационные объекты различных информационных ресурсов. На основе одной и той же таксономии можно определять несколько коллекций. Такой подход окажется чрезвычайно полезным для создания отдельных пользовательских коллекций.

Подсистема реализации задач интеграции данных из внешних источников. Для решения задач интеграции данных из источников LOD вводится понятие *источник данных*, которому ставятся в соответствие информационные ресурсы системы, и устанавливается соотношение набора атрибутов ресурса со свойствами ресурса из источника данных. Это позволяет генерировать SPARQL-запросы к источникам данных для извлечения конкретной информации. При этом пользователь оперирует привычными формами поиска, избегая необходимости написания самих запросов.

Для случая, когда конкретная реализация модели контента библиотеки основана на некоторой импортируемой онтологии и онтология используется в источнике данных, предусмотрен механизм взаимно однозначного отображения свойств и классов онтологий подключаемого набора данных в термины LibMeta полуавтоматическим способом. Таким образом, формируется интеграционный

узел, который позволяет устанавливать взаимосвязи с источниками данных, расположенными в LOD. На рисунке 5 приведена схема связей понятия *источника данных* с основными понятиями, определяющими контент библиотеки. Рисунк 6 иллюстрирует взаимодействие пользователя с подсистемой для получения результатов своего запроса.

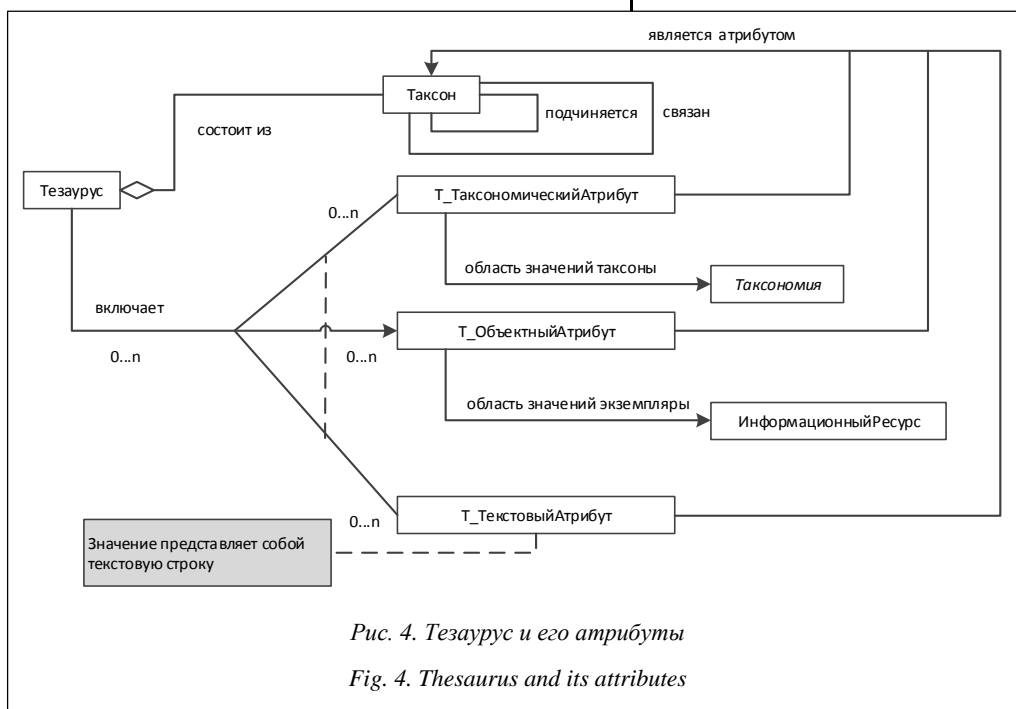


Рис. 4. Тезаурус и его атрибуты

Fig. 4. Thesaurus and its attributes

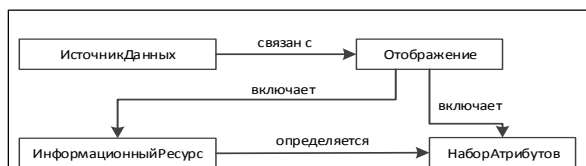


Рис. 5. Источник данных и основные понятия контента

Fig. 5. The data source and the basic concepts of content

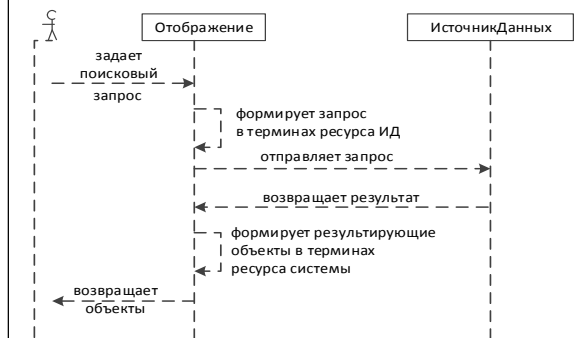


Рис. 6. Взаимодействие пользователя с источником данных

Fig. 6. User interaction with the data source

Подсистема реализации задач интеграции данных из внутренних источников. На самом деле эта подсистема оперирует теми же понятиями, что и предыдущая: *источник данных, ресурс, атрибуты ресурса*. Главное отличие состоит том, что интегрируемый источник не включен в LOD. Доступ к таким источникам обычно осуществляется по некоторому протоколу, который может быть широко используемым, таким как OAI-PMH, и довольно специфичным. Для таких случаев предусмотрена отдельная подсистема, которая поддерживает извлечение данных практически по любым протоколам и возвращает данные, готовые для включения в систему, в терминах информационных ресурсов системы.

Подсистема поиска и навигации по объектам системы. При описании информационных ресурсов и определении набора их атрибутов важную роль играют виды атрибутов, формирующие структурное описание ресурса. Атрибуты делятся на несколько пересекающихся видов: поисковые, описа-

тельные, административные, идентифицирующие. В подсистеме поиска важную роль играют именно поисковые атрибуты, используемые при выполнении атрибутного поиска по типам ресурсов. Результатом такого поиска являются объекты, краткое описание которых представлено пользователю посредством описательных атрибутов.

Подсистема поиска также поддерживает поиск по ключевым словам (или тегам) по всем ресурсам системы. Поиск выполняется по ключевым словам, которыми снабжены объекты системы.

Итак, основные понятия в этой подсистеме – *ресурс, объект, атрибут, вид атрибута, тег*. Тег объекта фактически представляет собой ключевое слово и обеспечивает функциональность определения набора семантических тегов информационных объектов из ключевых слов.

Подсистема качества данных в системе.

Подсистема качества данных непосредственно использует понятия информационного ресурса, информационного объекта и атрибута, дополняя собственными понятиями *ошибка, тип ошибки, правило, условие*. С помощью этих понятий определяются процесс (поток) работ для устранения проблем в данных, условие выявления ошибки определенного типа для значений атрибута объектов некоторого типа ресурсов и задается правило устранения ошибок [20].

Подсистема поддержки пользователей LibMeta. Важной составляющей любой информационной системы являются ее пользователи. Рассмотрим в общих чертах основные понятия подсистемы поддержки пользователей: *пользователь, роль, разрешение, информационный ресурс, информационный объект, область интересов*. Для каждого пользователя уровень доступа определяется его *ролью*, определяющей набор *прав доступа* для работы с информационными ресурсами и объектами. Для каждого пользователя системы определяется область его интересов, в описании которой может быть задействован тезаурус предметной области контента библиотеки, а также список пользователей со сходным кругом интересов (рис. 7). При этом каждый пользователь может создавать свои коллекции ресурсов в рамках своих интересов, пользуясь доступными средствами соответствующих подсистем.

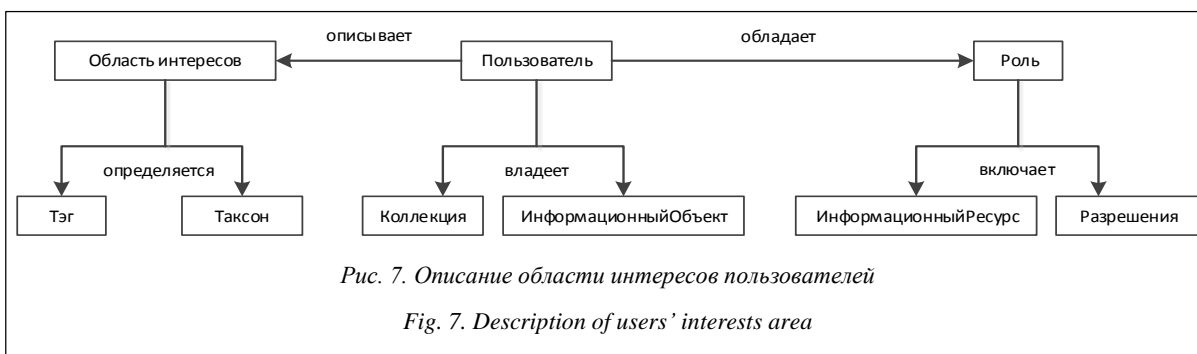


Рис. 7. Описание области интересов пользователей

Fig. 7. Description of users' interests area

Основная функциональность LibMeta. Функциональность LibMeta, доступная для всех публичных пользователей:

- просмотр ресурсов и их структуры;
- атрибутный и семантический поиск и навигация по доступным ресурсам системы;
- атрибутный и семантический поиск по источникам данных;
- просмотр общедоступных коллекций информационно-образовательных объектов.

Для авторизованного пользователя LibMeta обеспечивает дополнительно следующую функциональность:

- определение своей таксономии или расширение своей ветви определенного в системе основного терминологического тезауруса; фактически обеспечивается поддержка создания так называемых *аннотационных онтологий* [21] или *онтологий пользователей* (фолксономии) [22–24], которые представляют собой коллективный словарь пользователей, составленный в результате процесса проставления тегов ими для ресурсов;
- определение собственной коллекции ресурсов, основанной на использовании таксономии;
- организация совместных тематических коллекций для групп пользователей;
- атрибутный и семантический поиск по источникам данных с возможностью сохранения результатов поиска;
- доступ администратора системы ко всей вышеопределенной и дополнительной, доступной только ему функциональности:
 - ✓ расширение описания типов ресурсов или создание новых (по запросу пользователей);
 - ✓ включение объектов ресурсов в общедоступный список (по запросу пользователей);
 - ✓ обеспечение возможности редактирования определенных типов ресурсов или таксономий (для групп пользователей);
 - ✓ редактирование групп и ролей пользователей и набора доступных им операций;
 - ✓ редактирование и настройка основного терминологического тезауруса и его связей.

Таким образом, в статье рассмотрена информационная модель семантической библиотеки, определены основные понятия, лежащие в основе проектирования персональной открытой семантической библиотеки LibMeta. Рассмотрены общая архитектура семантической библиотеки и подход к интеграции источников данных.

В данный момент сконструирована библиотека на основе данных из «Научного Наследия России» и выполнено связывание с данными из источников LOD. Второй пример сконструированной библиотеки касается предметной области обыкновенных дифференциальных уравнений и публикаций по математике и также связан с данными из LOD.

Дальнейшее направление работ связано с созданием модуля автоматического тегирования доку-

ментов или выделения их ключевых слов как для отдельных описаний объектов, так и для их коллекций. В качестве описаний, к примеру для публикаций, может использоваться ее полный текст. В случае объемных текстов предполагается возможность предварительного создания автореферата текстов.

Это позволит выполнять тематическую кластеризацию документов, построение иерархии ключевых слов по темам для персональных коллекций, использовать семантические теги при поиске объектов на естественном языке как в самой библиотеке, так и в подключенных источниках данных.

Литература

1. Научное наследие России. URL: <http://e-heritage.ru> (дата обращения: 25.06.2016).
2. Semantic Web. URL: <http://www.w3.org/standards/semanticweb/> (дата обращения: 25.06.2016).
3. Bizer C., Heath T., and Berners-Lee T. Linked data – the story so far. *Int. J. Semantic Web Inf. Syst.*, 2009, vol. 3, no. 5, pp. 1–22.
4. Resource Description Framework (RDF). URL: <https://www.w3.org/RDF/> (дата обращения: 25.06.2016).
5. Серебряков В.А. Что такое семантическая цифровая библиотека // *Электронные библиотеки: перспективные методы и технологии, электронные коллекции (RCDL): тр. 16-й Всерос. науч. конф. Дубна: Изд-во ОИЯИ, 2014. С. 21–25.*
6. Palano R., Pandurino A., Guido A.L. Conceptual design of web application families: the BWW approach. *Proc. 6th Workshop on Domain Specific Modeling*, Portland, USA, 2006, pp. 23–32.
7. Kruk S.R. et al. *JeromeDL-a semantic digital library*. 2007, 268 p.
8. Miller G.A. WordNet: a lexical database for English. *Communications of the ACM*, 1995, vol. 38, no. 11, pp. 39–41.
9. Kruk S.R., Synak M., Zimmermann K. MarcOnt--Integration ontology for bibliographic description formats. *Proc. Intern. Conf. on Dublin Core and Metadata Applications*, 2005, pp. 231–234.
10. URL: <http://xmlns.com/foaf/spec/> (дата обращения: 25.06.2016).
11. Isaac A., Summers E. SKOS simple knowledge organization system primer. Working Group Note, W3C. 2009. URL: <https://www.w3.org/TR/skos-primer/> (дата обращения: 25.06.2016).
12. URL: <https://www.w3.org/TR/sparql11-overview/> (дата обращения: 25.06.2016).
13. Witten I. H., Bainbridge D., Boddie S. J. Greenstone: Open-source digital library software with end-user collection building. *Online information review*. 2001, vol. 25, no. 5, pp. 288–298.
14. Aloia N., Concordia C., Meghini C. Implementing BRICKS, a Digital Library Management System. *SEBD*, 2007, pp. 4–15.
15. Europeana Collections. URL: <http://www.europeana.eu> (дата обращения: 25.06.2016).
16. Doerr M. et al. The europeana data model (edm). *World Library and Information Congress: Proc. 76th IFLA General Conf. and Assembly*, 2010, pp. 10–15.
17. URL: <http://dbpedia.org/> (дата обращения: 25.06.2016).
18. Candela L., Castelli D., Dobrev M., Ferro N., Ioannidis Y., Katifori H., Koutrika G., Meghini C., Pagano P., Ross S., Agosti M., Scholdt H., Soergel D. The DELOS Digital Library Reference Model Foundations for Digital Libraries. *IST-2002 2.3.1.12. Technology-enhanced Learning and Access to Cultural Heritage*. Version 0.98, December 2007. URL: http://www.delos.info/files/pdf/ReferenceModel/DELOS_DLReferenceModel_0.98.pdf (дата обращения: 25.06.2016).
19. Атаева О.М., Серебряков В.А. Основные понятия формальной модели семантических библиотек и формализация процессов интеграции в ней // *Программные продукты и системы*. 2015. № 4 (112). С. 180–187.

20. Атаева О.М., Бездушный А.Н. Моделирование потоков работ в задаче приведения данных // Информационное обеспечение науки. Новые технологии: сб. науч. тр. М.: Научный Мир, 2009. С. 323–329.

21. Castro L., Giraldo O.X., Castro A.G. Using the Annotation Ontology in semantic digital libraries. Proc. ISWC 2010. Posters & Demonstrations Track. Collected abstracts. Shanghai, China, Nov. 9, 2010, vol. 658, pp. 153–156. URL: <http://ceur-ws.org/Vol-658/> (дата обращения: 25.06.2016).

22. Kruk S.R., McDaniel B. Semantic digital libraries. Springer, Berlin, 2009, 245 p.

23. Lee Y., Yang S.Q. Folksonomies as subject access—a survey of tagging in library online catalogs and discovery layers. IFLA Publ. Series. 2012. URL: https://www.nlib.ee/html/yritus/ifla_jarel/papers/4-1_Yan.docx (дата обращения: 25.06.2016).

24. Spiteri L.F. The structure and form of folksonomy tags: The road to the public library catalog. Information technology and libraries, 2013, vol. 26, no. 3, pp. 13–25.

Software & Systems

DOI: 10.15827/0236-235X.116.036-044

Received 28.06.16

2016, vol. 29, no. 4, pp. 36–44

AN INFORMATION MODEL OF LibMeta SEMANTIC LIBRARY

O.M. Ataeva¹, Junior Researcher, oli@ultimeta.ru

¹ Institution of Russian Academy of Sciences Dorodnitsyn Computing Centre of RAS, Vavilov St. 40, Moscow, 119333, Russian Federation

Abstract. The article considers libraries as information systems that provide the core functionality for working with data. The technology development determines the concept of both the library and its resources, which are not limited only by bibliographic records and their electronic submission now, but also bring the semantics of these resources to the front.

Based on new opportunities offered by advances in technologies, a library user receives additional opportunities to work with digital library resources using descriptions of their range of interests in subject area terms based on standards with dictionaries, thesauri and ontologies. This allows organizing and describing his own collections, as well as his own resources, detailing a resource description and their area of interest by clarifying its terms.

The paper considers basic requirements for such libraries and describes the developed system information model. A feature of the system is the ability to integrate data from sources integrated in the LOD cloud.

Keywords: semantic library, LOD, data integration, information model.

References

1. *Nauchnoe nasledie Rossii* [Scientific Heritage of Russia]. Available at: <http://e-heritage.ru> (accessed June 25, 2016).
2. *Semantic Web*. Available at: <http://www.w3.org/standards/semanticweb/> (accessed June 25, 2016).
3. Bizer C., Heath T., Berners-Lee T. Linked data – the story so far. *Int. J. Semantic Web Inf. Syst.* 2009, vol. 3, no. 5, pp. 1–22.
4. *Resource Description Framework (RDF)*. Available at: <https://www.w3.org/RDF/> (accessed June 25, 2016).
5. Serebryakov V.A. What is a semantic digital library. *Elektronnye biblioteki: perspektivnye metody i tekhnologii, elektronnye kolleksii – RCDL: tr. 16 Vseross. nauch. konf.* [Proc. 16th All-Russian Scientific Conf. on Digital Libraries: Prospective Methods and Technologies, Electronic Collections]. Dubna, Joint Institute for Nuclear Research Publ., 2014, pp. 21–25 (in Russ.).
6. Palano R., Pandurino A., Guido A.L. Conceptual design of web application families: the BWW approach. *Proc. 6th Workshop on Domain Specific Modeling*. Portland, USA. 2006, pp. 23–32.
7. Kruk S.R. *JeromeDL—a Semantic Digital Library*. 2007, 268 p.
8. Miller G. A. WordNet: a lexical database for English. *Communications of the ACM*. 1995, vol. 38, no. 11, pp. 39–41.
9. Kruk S.R., Synak M., Zimmermann K. MarcOnt—Integration ontology for bibliographic description formats. *Intern. Conf. on Dublin Core and Metadata Applications*. 2005, pp. 231–234.
10. *FOAF Vocabulary Specification 0.99*. Available at: <http://xmlns.com/foaf/spec/> (accessed June 25, 2016).
11. Isaac A., Summers E. *Skos Simple Knowledge Organization System Primer*. Working Group Note, W3C. 2009.
12. *SPARQL 1.1 Overview*. Available at: <https://www.w3.org/TR/sparql11-overview/> (accessed June 25, 2016).
13. Witten I.H., Bainbridge D., Boddie S.J. Greenstone: Open-source digital library software with end-user collection building. *Online Information Review*. 2001, vol. 25, no. 5, pp. 288–298.
14. Aloia N., Concordia C., Meghini C. *Implementing BRICKS, a Digital Library Management System*. SEBD. 2007, pp. 4–15.
15. *Europeana Collections*. Available at: <http://www.europeana.eu> (accessed June 25, 2016).
16. Doerr M. The europeana data model (edm). *World Library and Information Congr.: 76th IFLA General Conf. and Assembly*. 2010, pp. 10–15.
17. *DBpedia*. Available at: <http://dbpedia.org/> (accessed June 25, 2016).
18. Candela L., Castelli D., Dobrev M., Ferro N., Ioannidis Y., Katifori H., Koutrika G., Meghini C., Pagano P., Ross S., Agosti M., Schuldt H., Soergel D. *The DELOS Digital Library Reference Model Foundations for Digital Libraries. IST–2002 2.3.1.12. Technology-enhanced Learning and Access to Cultural Heritage*. Version 0.98, 2007. Available at: http://www.delos.info/files/pdf/ReferenceModel/DELOS_DLReferenceModel_0.98.pdf (accessed June 25, 2016).
19. Ataeva O.M., Serebryakov V.A. The basic concepts of a semantic libraries formal model and its integration process formalization. *Programmye produkty i sistemy* [Software & Systems]. 2015, no. 4 (112), pp. 180–187 (in Russ.).
20. Ataeva O.M., Bezduzhny A.N. Modeling workflows in a data reduction task. *Informatsionnoe obespechenie nauki. Noveye tekhnologii. Sb. nauch. tr.* [Proc. Science Information Support. New technologies]. Kalenov N.E. (Ed.). Moscow, Nauchny Mir Publ., 2009, pp. 323–329 (in Russ.).
21. Castro L.J.G., Giraldo O.X., Castro A.G. Using the Annotation Ontology in semantic digital libraries. *Proc. 9th Int. Semantic Web Conf. (ISWC)*. 2010, pp. 153–156.
22. Kruk S.R., McDaniel B. *Semantic digital libraries*. Heidelberg, Springer Publ., 2009, 245 p.
23. Lee Y., Yang S.Q. *Folksonomies as Subject Access—A Survey of Tagging in Library Online Catalogs and Discovery Layers*. IFLA Pub. Series. Accepted for publication. 2012.
24. Spiteri L.F. The structure and form of folksonomy tags: The road to the public library catalog. *Information Technology and Libraries*. 2013, vol. 26, no. 3, pp. 13–25.

УДК 004.04: 519.767

DOI: 10.15827/0236-235X.116.045-057

Дата подачи статьи: 19.07.16

2016. Т. 29. № 4. С. 45–57

СЕМАНТИЧЕСКИЙ АНАЛИЗ И СПОСОБЫ ПРЕДСТАВЛЕНИЯ СМЫСЛА ТЕКСТА В КОМПЬЮТЕРНОЙ ЛИНГВИСТИКЕ

Т.В. Батура, к.ф.-м.н., старший научный сотрудник, tatiana.v.batura@gmail.com
(Институт систем информатики им. А.П. Ершова СО РАН,
просп. Лаврентьева, 6, г. Новосибирск, 630090, Россия)

Статья посвящена проблемам семантического анализа текстов. Рассмотрены различные методы: диаграммы зависимостей и семантические сети, подходы, основанные на лексических функциях и тематических классах, фреймовые и онтологические модели, логические модели представления знаний. На данный момент существуют различные методы представления смысла высказываний.

Создание новых методов семантического анализа текстов актуально для решения многих задач компьютерной лингвистики, таких как машинный перевод, автореферирование, классификация текстов и других. Не менее важна разработка новых инструментов, позволяющих автоматизировать семантический анализ.

Несмотря на то, что некоторые научные и технические идеи в области обработки текстов развиваются довольно быстро, многие проблемы семантического анализа остаются нерешенными. Большинство исследователей пришло к выводу, что словарь для поддержки семантического анализа должен оперировать смыслами и, следовательно, описывать свойства и отношения понятий, а не слов. Но возникает вопрос, как правильно структурировать и представлять информацию в подобных словарях, чтобы поиск по ним был удобным и быстрым, а кроме того, можно было бы учитывать изменения в естественном языке (исчезновение старых и возникновение новых понятий). В данной статье предпринята попытка систематизировать известные достижения в области семантического анализа и в какой-то мере найти ответ на этот и другие вопросы.

Ключевые слова: семантический анализ, автоматическая обработка текста, извлечение информации, семантические сети, логика предикатов, представление знаний, смысл высказывания.

Помимо знаний о структуре языка, семантика тесно связана с философией, психологией и другими науками, так как неизбежно затрагивает вопросы о происхождении значений слов, их отношении к бытию и мышлению. При семантическом анализе необходимо учитывать социальные и культурные особенности носителя языка. Процесс человеческого мышления, как и язык, который является инструментом выражения мыслей, очень гибкий и трудно поддается формализации. Поэтому семантический анализ по праву считается самым сложным этапом автоматической обработки текстов.

На данный момент существует много методов представления смысла высказываний, однако ни один из них не является универсальным. Над соотношением смысла тексту работали многие исследователи. Так, И.А. Мельчук [1] ввел понятие лексической функции, развил понятия синтаксических и семантических валентностей и рассмотрел их в контексте толково-комбинаторного словаря, который представляет собой языковую модель. Он показал, что значения слов соотносятся не непосредственно с окружающей действительностью, а с представлениями носителя языка об этой действительности. В.Ш. Рубашкин и Д.Г. Лахути [2] ввели иерархию синтаксических связей для более эффективной работы семантического анализатора. Самыми важными являются обязательные ролевые связи, далее идут связи кореференции, затем факультативные ролевые связи и только потом предметно-ассоциативные. Известный лингвист Е.В. Падучева [3] предлагает рассматривать тематические классы слов, в частности глаголов, по-

скольку они несут основную смысловую нагрузку. Существенной в данном подходе является идея разделения понятий языка на некоторые семантические группы с учетом того, что эти понятия имеют некоторый нетривиальный общий смысловой компонент. Элементы таких групп склонны иметь один и тот же набор зависимых понятий.

Универсальный язык представления знаний должен быть удобным для осуществления вывода новых знаний из уже имеющихся, а значит, необходимо создать аппарат для проверки правильности высказываний. Здесь как раз полезны логические модели представления знаний. Например, семантический язык, предложенный В.А. Тузовым [4], содержит в себе формализмы логики предикатов, в нем присутствуют атомарные понятия, функции над этими понятиями и правила вывода, с помощью которых можно описывать новые понятия. Не исключено, что в направлении создания подобных семантических языков будет развиваться научная мысль в будущем.

Исследование семантики в рамках теории «Смысл ⇔ Текст»

При создании теории «Смысл ⇔ Текст» в [1] введено понятие *лексической функции*. Лексическая функция – определенное смысловое соотношение, например, «равенство по смыслу» (syn), «противоположность по смыслу» (anti), «обобщающее понятие» (gener) и др. Пусть имеется ряд лексических единиц – слов и словосочетаний. Тогда данная лексическая функция ставит в соответствие каждой из этих единиц набор лексических единиц,

находящихся с исходной единицей в соответствующем смысловом соотношении.

Значения одной лексической функции от разных аргументов могут полностью или частично совпадать; могут совпадать и значения разных функций от одного аргумента. На взгляд автора, говорить о лексических функциях как о многозначных не совсем корректно и удобно. Удобнее говорить о лексических предикатах [5]. Помимо простых лексических предикатов, для описания лексической сочетаемости могут использоваться и их комбинации – составные предикаты.

Особую роль при исследовании семантики в подходе И.А. Мельчука играют валентности слов, то есть способность слов вступать в связи с другими словами. Различают два вида валентностей слова: синтаксические и семантические. Хотя это разделение иногда довольно условно. Ситуации, описываемые словами на естественном языке, имеют, как правило, от одного до четырех смысловых компонентов, или семантических актантов. В то же время каждому слову сопоставляются глубинно-синтаксические актанты – зависимые слова, соответствующие подлежащему и дополнениям. Семантические валентности определяются лексическим анализом ситуации, задаваемой конкретным словом. Синтаксические валентности определяются количеством синтаксических актантов, представленных непосредственно в тексте и заданных контекстом.

С формальной точки зрения мы имеем конструкцию, описанную ниже. Чтобы не связывать с каждым глаголом (и другими словами) отдельный предикат, будем рассматривать предикат, размерность которого больше на 1: $P^{val}(y, x_1, x_2, \dots, x_n)$, при этом y будет само слово, а x_1, x_2, \dots, x_n – его валентности. Отличать синтаксические и семантические актанты можно с помощью мультииндексов, указывающих, какие именно актанты заданы в тексте. Запись $P^{val}_{i_1 i_2 \dots i_k}(y, x_{i_1}, x_{i_2}, \dots, x_{i_k})$ означает, что заданы актанты i_1, i_2, \dots, i_k . В частности, если заданы все актанты, получаем $P^{val}_{1 \dots n}(y, x_1, x_2, \dots, x_n)$. Некоторые варианты (наборов мультииндексов) могут быть недопустимыми в языке. Если набор i_1, i_2, \dots, i_k допустим, имеет место импликация

$$\forall y \forall x_1 \dots \forall x_n (P^{val}_{1 \dots n}(y, x_1, x_2, \dots, x_n) \rightarrow \\ \rightarrow P^{val}_{i_1 i_2 \dots i_k}(y, x_{i_1}, x_{i_2}, \dots, x_{i_k})).$$

Одно из главных теоретических изобретений И.А. Мельчука – *толково-комбинаторный словарь*, отражающий прежде всего нетривиальную сочетаемость лексем. Получается, что язык – это очень большая модель, в которой определены лексические предикаты, действующие описанным выше образом.

Словарная статья толково-комбинаторного словаря может быть представлена в виде кортежа

$A = \langle w, P_1, \dots, P_n, Val \rangle$, где w – основное слово; P_1, \dots, P_n – лексические предикаты, связанные со словом; Val – информация о валентности слова.

В этом случае набор статей в толково-комбинаторном словаре можно считать некоторой подмоделью исходной модели, являющейся языком. Лексические предикаты, определенные теперь на более узком множестве, будут действовать аналогично.

Теория «Смысл \Leftrightarrow Текст» с самого начала создавалась для применения в прикладной проблематике автоматического перевода. По замыслу И.А. Мельчука, с ее помощью, в отличие от традиционных нестрогих теорий, следовало обеспечить построение «действующей» модели языка. Теория «Смысл \Leftrightarrow Текст» действительно была использована в некоторых системах машинного перевода, разработанных в России, – прежде всего в системе англо-русского автоматического перевода ЭТАП, созданной группой под руководством Ю.Д. Апресяна. Все эти системы относятся к экспериментальным, то есть их промышленное использование не представляется возможным. Несмотря на то, что они включают много полезной лингвистической информации, в целом ни одна из них пока не обеспечила прорыва в качестве перевода.

На взгляд автора, основная ценная идея этой теории состоит в том, что значения слов соотносятся не непосредственно с окружающей действительностью, а с представлениями носителя языка об этой действительности (иногда называемыми концептами). Природа концептов зависит от конкретной культуры; система концептов каждого языка образует так называемую наивную картину мира, которая во многих деталях может отличаться от научной картины мира, являющейся универсальной. Задача семантического анализа лексики в теории «Смысл \Leftrightarrow Текст» именно в том, чтобы обнаружить наивную картину мира и описать ее основные категории. Другими словами, важная роль этой теории состоит в описании не только объективной, но и субъективной картины мира.

Хотя интерес к теории И.А. Мельчука угасает, разметка синтаксического корпуса «Национальный корпус русского языка» [6] выполняется лингвистическим процессором ЭТАП-3, основанным на принципах теории «Смысл \Leftrightarrow Текст».

Идеи Ю.Д. Апресяна в разработке процессора ЭТАП несколько отличаются от идей И.А. Мельчука. Центральное место в исследованиях Апресяна занимает синонимический словарь нового типа [7]. Для этого словаря была разработана подробная схема описания синонимических рядов, где каждый элемент ряда характеризовался с точки зрения семантики, синтаксиса, сочетаемости и других свойств. В словаре собрано и обобщено максимальное количество информации о языковом поведении русских синонимов.

Концептуальный и прецедентный анализ

На этапе морфологического и семантико-синтаксического анализа текстов основными единицами, обозначающими понятия, являются слова. При таком подходе считается, что смысл словосочетаний и фраз может быть выражен через смыслы составляющих их слов. Такой подход опирается на предположение, что словосочетания, встречающиеся в языке, можно разделить на свободные и несвободные. Другой подход основывается на том, что неделимыми единицами смысла являются категории и понятия, состоящие не из самостоятельных слов, а из словосочетаний [8]. Такие категории и понятия называются концептами. Идея концептуального анализа как неотъемлемой составляющей семантического анализа встречается в исследованиях [2, 9, 10]. В данной работе кратко изложены взгляды на то, какие задачи должны решаться средствами концептуального семантического анализа.

С точки зрения используемых методов и средств семантический анализ должен предусматривать два этапа: этап интерпретации грамматически выраженных (синтаксических и анафорических) связей и этап распознавания связей, не имеющих грамматического выражения. Неоднозначности должны разрешаться самим процессом анализа по критерию степени смысловой удовлетворительности получаемого в каждом варианте результата.

Ключевым пунктом системы семантического анализа является эффективная словарная поддержка. В этом смысле любая система семантического анализа является тезаурусно ориентированной. Процедуры семантического анализа во всех без исключения случаях опираются на функциональность понятийного словаря. Словарь для поддержки семантического анализа должен оперировать смыслами и, следовательно, описывать свойства и отношения понятий, а не слов, поэтому его можно назвать концептуальным словарем [2]. В некотором смысле роль концептуального словаря могут выполнять семантические сети.

В семантическом интерпретаторе прежде всего следует специфицировать различаемые типы семантических отношений в тексте: ролевые (связи по валентности предиката), предметно-ассоциативные (отношения между объектами, процессами, значимые в предметной области) и др.

Принимаются следующие основные правила интерпретации синтаксических связей.

1. Тип устанавливаемого семантического отношения определяется семантическими классами и в определенных случаях более детальными семантическими характеристиками синтаксического «хозяина» и «слуги».

2. Предлоги рассматриваются не как самостоятельный объект интерпретации, а как дополнитель-

ная (семантико-грамматическая) характеристика связи между синтаксическим «хозяином» предлога и управляемым им словом.

3. Для разрешения лексической и синтаксической омонимии, фиксируемой синтаксическим анализатором, семантический интерпретатор использует систему эмпирически устанавливаемых предпочтений. На уровне типов семантических отношений устанавливается следующий порядок предпочтений (соответствует уменьшению приоритета связи): функциональные связи и связи, устанавливающие факт смысловой избыточности; ролевые связи, определяемые как обязательные, при наличии семантически согласованного актанта; связи кореференции; ролевые связи, определяемые как факультативные; специфицируемые предметно-ассоциативные связи; неспецифицируемые предметно-ассоциативные связи.

Все большее значение приобретает анализ «по образцу» (прецедентный анализ) [11], основанный на использовании корпуса предварительно размеченных текстов. Система анализа должна обеспечивать не только извлечение знаний из конкретного текста, но и накопление результатов как на синтаксическом, так и на семантическом уровне для использования их далее в качестве прецедентов.

Одним из наиболее масштабных и значимых проектов, осуществляемых в настоящее время, является создание Национального корпуса русского языка [6]. В нем участвует большая группа лингвистов многих научных центров России.

Национальный корпус русского языка – коллекция электронных текстов, снабженных обширной лингвистической и метатекстовой информацией. Корпус представляет все разнообразие стилей, жанров и вариантов русского языка 19–20 вв. В настоящее время в нем используются пять типов разметки: метатекстовая, морфологическая (словоизменяемая), синтаксическая, акцентная и семантическая. Остановимся лишь на семантической разметке.

При семантической разметке большинству слов в тексте приписываются один или несколько семантических и словообразовательных признаков и пр. Разметка текстов осуществляется автоматически в соответствии с семантическим словарем корпуса. Поскольку ручная обработка семантически размеченных текстов очень трудоемка, семантическая омонимия в корпусе не снимается: многозначным словам приписываются несколько альтернативных наборов семантических признаков.

В основу семантической разметки положена система классификации русской лексики, принятая в БД «Лексикограф», которая разрабатывалась под руководством Е.В. Падучевой и Е.В. Рахилиной. Подход Е.В. Падучевой часто рассматривается как особое направление в изучении семантики русского языка. В ее работах рассмотрен большой

класс вопросов по этой теме. Наиболее интересными являются исследования тематических классов русских глаголов [3, 12, 13]. Тематический класс объединяет слова с общим семантическим компонентом, который занимает центральное место в их смысловой структуре. Различают, например, глаголы восприятия, знания, эмоций, принятия решения, речевых действий, движения, звука, бытийные глаголы и др.

Для Национального корпуса русского языка был существенно увеличен словарь, расширен состав и усовершенствована структура семантических классов, добавлены словообразовательные признаки. Словник семантического словаря базируется на морфологическом словаре системы «Диалинг» (общим объемом порядка 120 тыс. слов), представляющем собой расширение грамматического словаря русского языка А.А. Зализняка. Текущая версия семантического словаря включает слова знаменательных частей речи: существительные, прилагательные, числительные, местоимения, глаголы и наречия. Лексико-семантическая информация имеет различную структуру для разных частей речи.

Существенной в данном подходе является идея разделения понятий языка на некоторые семантические группы с учетом того, что эти понятия имеют некоторый нетривиальный общий смысловой компонент. Элементы таких групп склонны иметь один и тот же набор зависимых понятий. В таком случае словарь для поддержки семантического анализа должен оперировать смыслами и, следовательно, описывать свойства и отношения понятий, а не слов. Остается вопрос, как правильно структурировать и представлять информацию в подобных словарях, чтобы поиск по ним был удобным и быстрым, а кроме того, можно было бы учитывать изменения в естественном языке (исчезновение старых и возникновение новых понятий).

При обсуждении проблем семантики часто упоминают принцип композициональности. Согласно ему, смысл сложного выражения определяется смыслами его составных частей и правилами, применяемыми для их объединения. Поскольку предложение состоит из слов, получается, что его смысл можно представить набором значений слов, входящих в него. Но не все так просто. Смысл предложения также опирается на порядок слов, фразирование и отношения между словами в предложении, то есть учитывает синтаксис.

Как видим, концептуальный анализ позволяет утверждать, что в некоторых случаях принцип композициональности нарушается. Ошибочно утверждать, что смысл словосочетаний и фраз может быть выражен через смысл составляющих их слов. Это не всегда верно. Однако главная проблема такого подхода заключается в том, что выделение тематических классов и составление семантических словарей — чрезвычайно трудоемкий процесс,

сильно зависящий от индивидуального восприятия и интерпретации понятий конкретным человеком.

Сетевые модели представления знаний

Тезаурусы, семантические сети, фреймвые и онтологические модели. Тезаурус — разновидность словаря общей или специальной лексики, в котором указаны семантические отношения между лексическими единицами. В отличие от толкового словаря тезаурус позволяет выявить смысл не только с помощью определения, но и посредством соотнесения слова с другими понятиями и их группами, благодаря чему может использоваться для наполнения баз знаний систем искусственного интеллекта. В тезаурусах обычно используются следующие основные семантические отношения: синонимы (*смелый–храбрый*), антонимы (*добрый–злой*), гипонимы (*животное–собака*), гиперонимы (*собака–животное*), меронимы (*автомобиль–двигатель, колесо*), холонимы (*двигатель, колесо–автомобиль*) и паронимы (*индеец–индиец*).

Пример тезауруса — WordNet [14]. Базовой словарной единицей WordNet является синонимический ряд (синсет), объединяющий слова со схожим значением. Синсеты состоят из слов, принадлежащих той же части речи, что и исходное слово. Каждый синсет сопровождается небольшой формулировкой, разъясняющей его значение. Синсеты связаны между собой различными семантическими отношениями. WordNet содержит около 155 тысяч различных лексем и словосочетаний, организованных в 117 тысяч синсетов. Вся БД разбита на три части: существительные, глаголы и прилагательные/наречия. Слово или словосочетание может находиться более чем в одном синсете и принадлежать более чем одной категории части речи.

Семантическая сеть — модель предметной области, имеющая вид ориентированного графа, вершины которого соответствуют объектам предметной области, а дуги (ребра) задают отношения между ними [15]. Объектами могут быть понятия, события, свойства, процессы. Таким образом, семантическая сеть отражает семантику предметной области в виде понятий и отношений. Причем в качестве понятий могут выступать как экземпляры объектов, так и их множества.

Семантические сети возникли как попытка визуализации математических формул. За визуальным представлением семантической сети в виде графа стоит математическая модель, в которой каждая вершина соответствует элементу предметного множества, а дуга — предикату. Терминология, используемая в этой области, различна. Чтобы добиться некоторой однородности, узлы, соединенные дугами, принято называть графами, а структуру, где имеется целое гнездо из узлов или где существуют отношения различного порядка между графами, сетью.

Заметим, что среди семантических отношений, применяемых для описания сетей, могут быть не только семантические отношения, используемые в тезаурусах, но и другие связи: функциональные (определяемые обычно глаголами *производит, влияет, ...*), количественные (*больше, меньше, равно, ...*), пространственные (*далеко от, близко от, под, над, ...*), временные (*раньше, позже, в течение, ...*), атрибутивные (*иметь свойство, иметь значение*), логические (*И, ИЛИ, НЕ*) и пр.

Несмотря на некоторые различия, сети удобны для чтения и обработки компьютером, являются наглядным и достаточно универсальным средством представления семантики естественного языка. Однако их формализация в конкретных моделях представления, использования и модификации знаний оказывается достаточно трудоемкой, особенно при наличии множественных отношений между ее элементами. Сеть способна разрастись до большого размера, и, как следствие, поиск вывода в ней будет слишком сложным.

В сложных семантических сетях, включающих множество понятий, процесс обновления узлов и контроль связей между ними, как видим, усложняют процедуру обработки информации. Стремление устранить эти недостатки послужило причиной появления особых типов семантических сетей, таких как фреймовые модели.

Фрейм – структура для описания понятия или ситуации, состоящая из характеристик этой ситуации и их значений [16]. Фрейм можно рассматривать как фрагмент семантической сети, предназначенный для описания понятий со всей совокупностью присущих им свойств. Особенность фреймовых моделей представления знаний в том, что все понятия, описываемые в каждом из узлов модели, определяются набором атрибутов и их значениями, которые содержатся в слотах фрейма (*имя фрейма, слот 1, слот 2, ..., слот N*). Графически это выглядит аналогично семантической сети, но принципиальное отличие заключается в том, что каждый узел во фреймовой модели имеет обобщенную структуру, состоящую из множества слотов, каждый из которых содержит имя, указатель наследования, указатель типа данных и значение.

Слот – это атрибут, связанный с узлом в модели, основанной на фреймах, и являющийся составляющей фрейма. *Имя слота* должно быть уникальным в пределах фрейма. *Указатель наследования* показывает, какую информацию об атрибутах слотов во фрейме верхнего уровня наследуют слоты с теми же именами во фрейме более низкого уровня. *Указатель типа данных* содержит информацию о типе данных, включаемых в слот. Обычно используются следующие типы данных: указатель на имя фрейма верхнего уровня, текст, список, таблица, присоединенная процедура и др. *Значением слота* могут быть экземпляр атрибута, другой фрейм или фасет, оно должно соответствовать указанному типу дан-

ных и условию наследования. Помимо конкретного значения, в слоте могут храниться процедуры и правила, которые вызываются при необходимости вычисления этого значения. Таким образом, слот может содержать не только конкретное значение, но и имя процедуры, позволяющей вычислить его по заданному алгоритму, а также одну или несколько продуктов, с помощью которых это значение определяется. В слот могут входить несколько значений. Иногда слот включает компонент, называемый фасетом, который задает диапазон или перечень его возможных значений.

Различают фреймы-образцы (прототипы), хранящиеся в базе знаний, и фреймы-экземпляры, создаваемые для отображения реальных ситуаций на основе поступающих данных. Фреймовые модели являются достаточно универсальными, поскольку позволяют отразить все многообразие знаний о мире через фреймы-структуры (для обозначения объектов и понятий: *заем, залог, вексель*), фреймы-роли (*менеджер, кассир, клиент*), фреймы-сценарии (*банкротство, собрание акционеров, празднование именин*), фреймы-ситуации (*тревога, авария, рабочий режим устройства*) и др.

Важнейшим свойством теории фреймов является заимствованное из теории семантических сетей наследование свойств. И во фреймах, и в семантических сетях наследование происходит по *ISA*. Слот *ISA* указывает на фрейм более высокого уровня иерархии, откуда неявно наследуются, то есть переносятся, значения аналогичных слотов.

Основными преимуществами фреймов как модели представления знаний являются соответствие современным представлениям об организации долговременной памяти человека, а также ее гибкость и наглядность. Достоинства фреймовых моделей представления знаний проявляются, если родовые связи изменяются нечасто и предметная область насчитывает немного исключений.

Недостаток фреймовых моделей в их относительно высокой сложности, что проявляется в снижении скорости работы механизма вывода и увеличении трудоемкости внесения изменений в сформированную иерархию [17]. Поэтому при разработке фреймовых систем большое внимание уделяется наглядным способам отображения и эффективным средствам редактирования фреймовых структур. Фреймовые модели не позволяют организовать гибкий механизм логического вывода, поэтому фреймовые системы либо представляют собой объектно-ориентированные БД, либо требуют интеграции с другими средствами обработки знаний, например с логическими моделями.

В инженерии знаний под онтологической моделью понимается детальное описание некоторой предметной или проблемной области, которое используется для формулирования утверждений общего характера. Онтологии позволяют представить понятия в таком виде, что они становятся пригод-

ными для машинной обработки. Обычно выделяют следующие основные элементы онтологий: экземпляры, классы объектов (понятий), атрибуты (описывают свойства классов и экземпляров), функции (описывают зависимости между классами и экземплярами), аксиомы (дополнительные ограничения) [18].

Как видим, в центре большинства онтологий находятся классы, описывающие понятия предметной области. Атрибуты описывают свойства классов и экземпляров. Здесь прослеживаются аналогии с фреймовым подходом к формализации знаний. Многие понятия и принципы реализации, а также графическая форма представления на начальном этапе структуризации в онтологиях сходны с семантическими сетями. Основным отличием является ориентация онтологий на использование непосредственно компьютером, то есть структуры данных описаны не на естественном языке (как это принято в семантических сетях и тезаурусах), а на специальном формальном. С тезаурусами онтологии тоже имеют много общего. Но в отличие от них для онтологических моделей необходимыми требованиями являются внутренняя полнота, логическая взаимосвязь и непротиворечивость используемых понятий. В тезаурусах эти требования могут не выполняться.

Специализированные (предметно-ориентированные) онтологии – это представление какой-либо области знаний или части реального мира. В такой онтологии содержатся специальные для этой области значения терминов. К примеру, слово *поле* в сельском хозяйстве означает участок земли, в физике – один из видов материи, в математике – класс алгебраических систем.

Общие онтологии используются для представления понятий, общих для большого числа областей. Такие онтологии содержат базовый набор терминов, глоссарий или тезаурус, используемый для описания терминов предметных областей.

Современные онтологические модели являются модульными, то есть состоят из множества связанных между собой онтологий, каждая из которых описывает отдельную предметную область или задачу. Онтологические модели не являются статичными, они постоянно меняются.

Если использующая специализированные онтологии система развивается, может потребоваться объединение онтологий. Недостатком онтологических моделей является сложность их объединения. Онтологии даже близких областей могут быть несовместимы друг с другом. Объединение онтологий выполняют как вручную, так и в полуавтоматическом режиме. В целом это трудоемкий, медленный и дорогостоящий процесс.

Одна из существующих проблем в онтологическом подходе – представление знаний о времени и об изменениях знаний с течением времени. Однако большинство применяемых на практике языков

описания онтологий (например OWL и RDF) основываются на логике предикатов первого порядка и используют унарные или бинарные отношения. В этом случае для описания бинарных отношений с учетом времени требуется вводить в отношения дополнительный параметр, соответствующий времени. При этом бинарные отношения превращаются в тернарные и выходят за рамки описательных возможностей языка.

Еще одной важной задачей является описание знаний о времени с учетом возможной неполноты этих знаний. Эта задача обычно решается в рамках модальных темпоральных логик [19], например LTL, при помощи определенных модальных операторов. Но, поскольку язык описания знаний OWL основан на дескриптивной логике, воспользоваться таким решением для OWL-онтологий невозможно. Интересный способ представления знаний о времени с учетом неопределенности в онтологиях описан в работе А.Ф. Тузовского [20].

Семантические роли и семантические ограничения. Семантические сети позволяют представлять семантику отдельно взятого слова согласно его внутренней структуре. Если вместе с этой структурой учитывать грамматические особенности, то смысл высказывания может быть представлен в терминах семантических ролей и связанных с ними семантических ограничений.

Помимо термина «семантические роли», в литературе используются также понятия: тематические роли, тета-роли, глубинные падежи. Основоположниками данного направления исследований семантики принято считать Дж. Грубера и Ч. Филлмора. По своей сути эти понятия близки к семантическим и глубинно-синтаксическим актантам, исследованием которых занимался И.А. Мельчук. Приведем некоторые семантические роли, рассмотренные в работах [21, 22].

Агнс – одушевленный инициатор и контролер действия. Бенефактив (реципиент, посессор) – участник, чьи интересы косвенно затронуты в ситуации (получает пользу или вред). Инструмент – стимул эмоции или участник, с помощью которого выполняется действие. Источник – место, из которого осуществляется движение. Контрагент – сила или сопротивляющаяся среда, против которой выполняется действие. Объект – участник, который передвигается или изменяется в ходе события. Пациент – участник, претерпевающий существенные изменения. Результат – участник, который появляется в результате события. Стимул – внешняя причина или объект, вызывающие это состояние. Цель – место, в которое осуществляется движение.

В соответствии с числом аргументов и их семантическими свойствами множество глагольных лексем можно разбить на классы: глаголы физического воздействия (*рубить, пилить*), глаголы восприятия (*видеть, слышать*), глаголы способа речи (*кричать, шептать*) и др. Внутри каждого класса

существует более точное деление. Среди глаголов физического воздействия похожую семантическую предикатно-аргументную структуру имеют глаголы вида глагол (агенса, инструмент, объект): *break* – *разбить*, *crack* – *расколоть* и т.д. Другая предикатно-аргументная структура характерна для глаголов вида глагол (агенса, инструмент, цель): *hit* – *ударить*, *slap* – *шлепнуть*, *strike* – *ударить* и пр.

Было замечено, что существуют корреляции между морфологическими падежами, предложениями, синтаксическими ролями, с одной стороны, и семантическими ролями, с другой стороны, например, «cut with a knife», «work with John». Кроме того, следует учитывать, что у одного предикатного слова не может быть двух актантов с одной и той же семантической ролью. Различия в наборах ролей затрагивают в основном периферийные семантические роли (контрагент, стимул, источник) или сводятся к объединению/фрагментации ядерных ролей.

К сожалению, в результате многократных исследований не удалось установить взаимно-однозначное соответствие между семантическими ролями и падежами. Ситуация осложняется еще и тем, что сами роли нетривиально связаны между собой, а в естественных языках распространены такие генеративные приемы, как метафора и метонимия, которые порождают множество новых смыслов и не могут в принципе отражаться в статическом лексиконе.

Логические модели представления знаний

При построении логических моделей представления знаний вся информация, необходимая для решения прикладных задач, рассматривается как совокупность фактов и утверждений, которые представляются в виде формул в некоторой логике. Знания отображаются совокупностью таких формул, а получение новых знаний сводится к реализации процедур логического вывода. В основе логических моделей представления знаний лежит понятие формальной теории, задаваемое кортежем $S = \langle B, F, A, R \rangle$, где B – счетное множество базовых символов (алфавит); F – множество, называемое формулами; A – выделенное подмножество априори истинных формул (аксиом); R – конечное множество отношений между формулами, называемое правилами вывода.

Основной подход к представлению смысла в компьютерной лингвистике включает в себя создание представления смысла в формальном виде. Такое представление описывается языком представления смысла. Он необходим для того, чтобы ликвидировать разрыв между естественным языком и общесмысловыми знаниями о мире. Поскольку предполагается использовать этот язык для автоматической обработки текстов и при создании систем

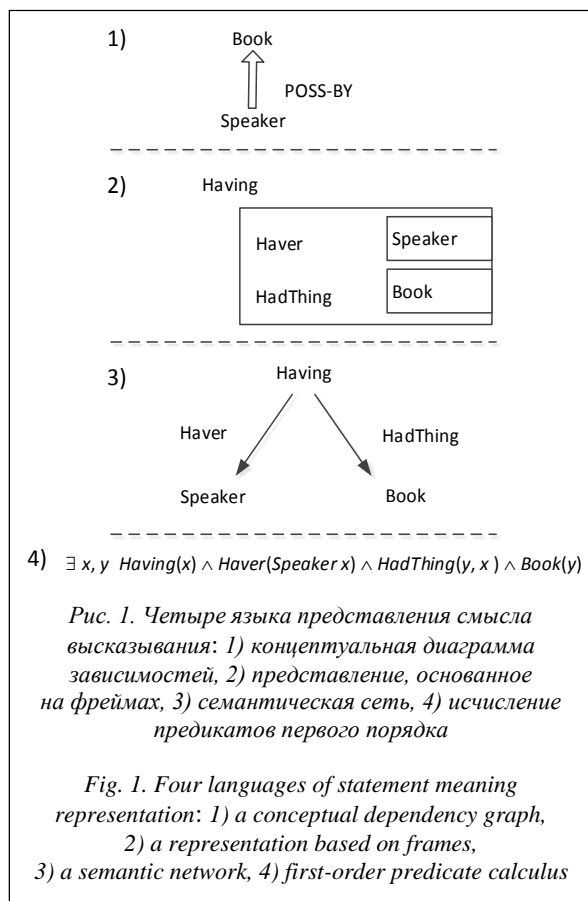
искусственного интеллекта, необходимо учитывать вычислительные требования семантической обработки, такие как необходимость определять истинность высказываний, поддерживать однозначность представления, представлять высказывания в канонической форме, обеспечивать логический вывод и быть выразительными.

В естественных языках существует большое разнообразие приемов для передачи смысла. Среди наиболее важных – способность передавать предикатно-аргументную структуру. Учитывая вышесказанное, получаем, что в качестве инструмента для представления смысла высказываний хорошо подходит логика предикатов первого порядка. С одной стороны, она относительно легко понимается человеком, с другой, хорошо поддается обработке (вычислительной). При помощи логики первого порядка могут быть описаны важные смысловые классы, включающие события, время и другие категории. Однако следует помнить, что высказывания, соответствующие таким понятиям, как убеждения и желания, требуют выражений, включающих модальные операторы. Язык представления смысла, как и любой язык, должен иметь свой синтаксис. Например, в работе [23] можно найти описание контекстно-свободной грамматики для исчисления предикатов первого порядка.

Семантические сети и фреймы могут быть рассмотрены в рамках логики предикатов первого порядка [24]. Например, смысл предложения *У меня есть книга* можно записать четырьмя различными способами с использованием четырех различных языков представления смысла (рис. 1).

Несмотря на то, что все эти четыре подхода различны, на абстрактном уровне они представляют собой общепринятое фундаментальное обозначение того, что представление смысла состоит из структур, составленных из множества символов. Эти символичные структуры соответствуют объектам и отношениям между объектами. Все четыре представления состоят из символов, соответствующих «говорящему», «книге» и набору отношений, обозначающих обладание одним другим. Важным здесь является то, что все эти четыре представления позволяют связать, с одной стороны, выразительные особенности естественного языка, а с другой – реальное состояние дел в мире.

Логические модели представления знаний обладают рядом преимуществ. Во-первых, в качестве «фундамента» здесь используется классический аппарат математической логики, методы которой достаточно хорошо изучены и формально обоснованы. Во-вторых, существуют достаточно эффективные процедуры вывода синтаксически правильных высказываний. В-третьих, такой подход позволяет хранить в базах знаний лишь множество аксиом, а все остальные знания (в том числе факты и сведения о людях, предметах, событиях и процессах) получать из этих аксиом по правилам вывода.



Вывод синтаксически правильных высказываний в логических моделях представления знаний опирается на правило резолюций, разработанное Дж. Робинсоном в 1965 году. Оно утверждает: если группа выражений, образующая посылку, является истинной, то применение правила вывода гарантированно обеспечит получение истинного выражения в качестве заключения. Результат применения правила резолюций называют резольвентой.

Метод резолюций (или правило устранения противоречий) позволяет проводить доказательство истинности или ложности выдвинутого предположения методом от противного. В методе резолюций множество предложений обычно рассматривается как составной предикат, который содержит несколько предикатов, соединенных логическими функциями и кванторами существования и всеобщности. Так как одинаковые по смыслу предикаты могут иметь разный вид, предложения сначала необходимо привести к унифицированному виду (к дизъюнктивной или конъюнктивной нормальной форме), то есть удалить кванторы существования, всеобщности, символы импликации, эквивалентности и др. Правило резолюций содержит в левой части конъюнкцию дизъюнктов. Поэтому приведение посылок, используемых для доказательства, к виду, представляющему собой конъюнкцию дизъюнктов, является необходимым этапом практически любого алгоритма, реали-

зующего логический вывод на базе метода резолюций [25].

Именно правило резолюций послужило основой для создания языка программирования Prolog. В языке Prolog факты описываются в форме логических предикатов с конкретными значениями. Правила вывода описываются логическими предикатами с определением правил вывода в виде списка предикатов над базами знаний и процедурами обработки информации. Интерпретатор языка Prolog самостоятельно реализует вывод, подобный вышеописанному. Чтобы инициировать вычисления, выполняется специальный запрос к базе знаний, на который система логического программирования генерирует ответы «истина» и «ложь».

Метод резолюций легко программируется, это одно из важнейших его достоинств, однако он применим только для ограниченного числа случаев, так как для его применения доказательство не должно иметь большую глубину, а число потенциальных резолюций не должно быть большим.

После того как язык Prolog приобрел большую популярность, в начале 80-х годов прошлого века появился термин «компьютеры пятого поколения». В то время ожидалось создание следующего поколения компьютеров, ориентированного на распределенные вычисления. Вместе с этим считалось, что пятое поколение станет основой для создания устройств, способных имитировать процесс человеческого мышления. Тогда же возникла идея создания аппаратной поддержки параллельных реляционных БД Grace и Delta [26, 27] и параллельного логического вывода (Parallel Inference Engine, PIE), опирающаяся на принципы языка Prolog. Каждый блок логического вывода сообщает о своей текущей рабочей нагрузке таким образом, чтобы работа могла быть передана в блок логического вывода с наименьшей нагрузкой [28]. Но, как известно, подобные попытки не позволили создать искусственный интеллект, а лишь послужили очередным подтверждением того, что человеческое мышление еще недостаточно изучено.

Системы с компонентами семантического анализа

Проект Open Cognition. В рамках проекта Open Cognition [29] разрабатывается анализатор Link Grammar Parser, который отвечает за обработку естественного языка. Link Grammar Parser начал разрабатываться в 1990-е гг. в университете Карнеги–Меллона [30]. Данный подход отличается от классической теории синтаксиса. Система приписывает предложению синтаксическую структуру, которая состоит из множества помеченных связей (коннекторов), соединяющих пары слов. Link Grammar Parser использует информацию о типах связей между словами. В настоящий момент поддерживаются словари для иврита, английского,

немецкого, русского, турецкого, персидского, арабского, латышского и вьетнамского языков.

Главной причиной, по которой анализатор называют семантической системой, можно считать уникальный по полноте набор связей (около 100 основных, причем некоторые из них имеют 3-4 варианта). В некоторых случаях тщательная работа над разными контекстами привела авторов системы к переходу к почти семантическим классификациям, построенным на синтаксических принципах.

Проект Open Cognition, в рамках которого развивается Link Grammar Parser, открытый и бесплатный, что является большим преимуществом для проведения исследований. Довольно подробное описание и исходный код можно найти на сайте [31]. Open Cognition продолжает развиваться, что также важно, поскольку есть возможность взаимодействовать с разработчиками. Наравне с Link Grammar ведется разработка анализатора RelEx [32], который позволяет извлекать отношения семантической зависимости в высказываниях на естественном языке и в результате представлять предложения в виде деревьев зависимостей. Он использует несколько наборов правил для перестроения графа с учетом синтаксических связей между словами. После каждого шага, согласно набору правил сопоставления, в полученном графе добавляются теги структурных характеристик и отношений между словами. Однако некоторые правила, наоборот, могут сокращать граф. Таким образом происходит преобразование графа. Этот процесс применения последовательности правил напоминает метод, используемый в ограничительных грамматиках. Главное отличие состоит в том, что RelEx работает с графовым представлением, а не с простыми наборами тегов (обозначающими отношения). Эта особенность позволяет применять более абстрактные преобразования при анализе текстов. Другими словами, основная идея состоит в том, чтобы использовать распознавание образов для преобразования графов. В отличие от других анализаторов, которые полностью опираются на синтаксическую структуру предложения, RelEx больше ориентирован на представление семантики, в частности, это касается сущностей, сравнений, вопросов, разрешения анафор и лексической многозначности слов.

Система «Диалинг». Эта автоматическая система русско-английского перевода разрабатывалась в 1999–2002 гг. в рамках проекта «Автоматическая обработка текста». В разное время в работе над ней принимали участие двадцать два специалиста, большинство из которых известные ученые-лингвисты. За основу системы «Диалинг» были взяты система французско-русского автоматического перевода, разработанная в ВЦП совместно с МГПИИЯ им. М. Тореза в 1976–1986 гг., и система анализа политических текстов на русском языке

«Политекст», разработанная в Центре информационных исследований в 1991–1997 гг.

Система «Политекст» была направлена на анализ официальных документов на русском языке и содержала полную цепочку анализаторов текста: графематический, морфологический, синтаксический и частично семантический. В системе «Диалинг» был частично заимствован графематический анализ, но адаптирован под новые стандарты программирования. Программа морфологического анализа была написана заново, поскольку скорость работы была низкой, но сам морфологический аппарат не изменился [33].

На графематическом уровне константами являются графематические дескрипторы: ЛЕ (лексема) – присваивается последовательностям, состоящим из кириллических символов; ЦК (цифровой комплекс) – присваивается последовательностям, состоящим из цифр, и т.д. На морфологическом уровне для обозначений используются грамемы: тв – творительный падеж, мн – множественное число, но – неодушевленность, св – совершенный вид, пе – переходность глагола и т.д. Возможные типы фрагментов на этапе фрагментационного анализа: главные предложения, придаточные предложения в составе сложного, причастные, деепричастные и другие обособленные обороты. Про каждый фрагмент известно, какие фрагменты в него непосредственно вложены и в какие он непосредственно вложен.

Основными составляющими применяемого в «Диалинге» семантического аппарата являются семантические отношения и семантические характеристики. Примеры семантических отношений: ИНСТР – «инструмент», ЛОК – «локация, местоположение», ПРИНАДЛ – «принадлежность» и пр. Они довольно универсальны и имеют сходство с предикатами и семантическими ролями. Семантические характеристики позволяют строить формулы с использованием логических связок «и» и «или». Каждому слову приписывается некоторая формула, составленная из семантических характеристик. В семантическом словаре «Диалинга» содержится около 40 семантических характеристик. Примеры семантических характеристик: ГЕОГР – географический объект; ДВИЖ – глаголы движения; ИНТЕЛ – действия, связанные с мыслительной деятельностью; НОСИНФ – носители информации; ЭМОЦ – прилагательные, которые выражают эмоции, и т.д. Некоторые характеристики являются составными, так как их можно выразить через другие. Семантические характеристики наравне с грамматическими характеристиками обеспечивают проверку согласования слов при интерпретации связей в тексте.

В данный момент все инструменты, разработанные в рамках проекта «Автоматическая обработка текста» (в том числе система «Диалинг»), являются свободным кроссплатформенным ПО.

Демоверсия и подробная документация доступны на сайте [34].

Другие системы семантического анализа. Существуют и другие системы, содержащие компоненты семантического анализа. Однако они имеют существенные недостатки для исследований: сложно найти описание, не являются бесплатными и свободно распространяемыми или не работают с текстами на русском языке. К таким системам относятся *OpenCalais* (<http://www.opencalais.com/opencalais-api/>), *RCO* (http://www.rco.ru/?page_id=3554), *Abbyy Compreno* (<https://www.abbyy.com/ru-ru/isearch/compreno/>), *SemSin* (<http://www.dialog-21.ru/media/1394/kanevsky.pdf>), *DictaScope* (<http://dictum.ru/>) и др.

Следует упомянуть систему извлечения данных из неструктурированных текстов *Pullenti* (<http://semantick.ru/>). Она заняла первое место на дорожках T1, T2, T2-m и второе место на T1-l на конференции «Диалог-2016» в соревновании Fact-RuEval. На сайте разработчиков системы *Pullenti* есть также демоверсия семантического анализатора, позволяющего по предложению строить семантическую сеть.

Инструментальная среда «ДЕКЛ» (<http://ipiranlogos.com/>) разработана в конце 90-х годов и использована для построения экспертных систем, оболочек для экспертных систем, логико-аналитических систем, лингвистических процессоров, обеспечивающих обработку и автоматическое извлечение знаний из потоков неформализованных документов на естественном языке.

Система машинного перевода «ЭТАП-3» предназначена для анализа и перевода текстов на русском и английском языках. Система использует преобразование текстов на естественном языке в их семантическое представление на языке Universal Networking Language. Как уже говорилось ранее, разметка синтаксического корпуса «Национальный корпус русского языка» [6] выполняется лингвистическим процессором ЭТАП-3, основанным на принципах теории «Смысл \Leftrightarrow Текст».

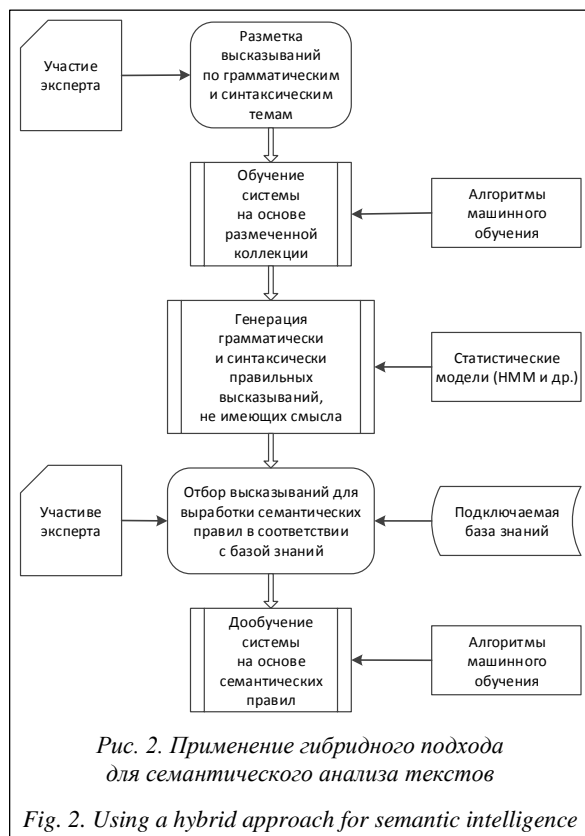
В последнее время появляется все больше систем представления баз знаний в виде графов. Поскольку объемы информации постоянно увеличиваются с невероятной скоростью, такие системы должны поддерживать построение и пополнение баз знаний в автоматическом режиме. Автоматическое построение баз знаний может осуществляться на основе структурированных источников данных. Примерами таких систем являются *Yago* (<http://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/yago/>), *DBpedia* (<http://wiki.dbpedia.org/>), *Freebase* (<https://developers.google.com/freebase/>), *Google's Knowledge Graph* (<https://developers.google.com/knowledge-graph/>), *OpenCyc* (<http://www.opencyc.org/>). Другой подход позволяет извлекать информацию из открытых ресурсов в Интернете без участия че-

ловека: *ReadTheWeb* (<http://rtw.ml.cmu.edu/rtw/>), *OpenIE* (<http://nlp.stanford.edu/software/openie.html>), *Google Knowledge Vault* (<https://www.cs.ubc.ca/~murphyk/Papers/kv-kdd14.pdf>). Подобные системы являются экспериментальными, каждая из них имеет свои особенности. Например, *Knowledge Vault* пытается учитывать неопределенности, каждому факту ставятся в соответствие коэффициент доверия и происхождение информации. Таким образом, все утверждения делятся на те, которые имеют высокую вероятность быть истинными, и те, которые могут быть менее вероятными. Предсказание фактов и их свойств осуществляется методами машинного обучения на основе очень большого количества текстов и уже имеющихся фактов. В данный момент *Knowledge Vault* содержит 1,6 млрд фактов. Система *NELL*, разрабатываемая в рамках проекта *ReadTheWeb* университетом Карнеги-Меллона, содержит более 50 млн утверждений с разными степенями доверия. Около 2 млн 800 тыс. фактов имеют высокую степень доверия. Процесс обучения *NELL* также еще не завершен.

Применение гибридного подхода в системах семантического анализа текстов. Учитывая вышесказанное, получаем, что при работе с семантикой текстов приходится иметь дело с гибкими, постоянно меняющимися структурами. Если на семантическом уровне представлять последовательность высказываний в виде графа, становится ясно, что этот граф нужно постоянно перестраивать по ходу повествования или диалога. Добавляется новая информация, исчезает старая и изменяется уже имеющаяся. Поэтому подходы, основанные на наборах заранее заданных правил, применяемых ранее для обработки текстов, дают недостаточно хорошие результаты. Основным их недостатком является невозможность эффективно осуществлять вывод новых правил.

Одним из возможных решений является применение гибридного подхода в системах семантического анализа текстов. Подразумевается, что гибридный подход сочетает в себе методы машинного обучения и методы, основанные на правилах. На рисунке 2 представлена основная идея применения гибридного подхода для систем семантического анализа текстов.

Семантический анализ текста сводится к анализу семантического пространства, то есть смысловой модели текста. Для изучения свойств семантического пространства может быть введено понятие размерности. Размерность семантического пространства – количество возможных вариантов сопоставления смысла тексту. Для синтаксического пространства размерность – количество возможных синтаксических ролей, корректно приписанных словам. Тогда можно утверждать, что размерность семантического пространства больше размерности синтаксического пространства ввиду многозначности не только семантических правил,



но и лексических единиц. Для текстов возникают также понятия семантических подпространств и проекций семантических пространств. В будущем планируется формализовать эти и другие понятия, исследовать свойства семантических пространств и подпространств, реализовать прототип системы семантического анализа.

В заключение отметим, что с развитием компьютерных технологий и постоянным ростом объемов текстовой информации исследования в области автоматической обработки текстов сфокусировались на прикладных аспектах. Однако в настоящее время возможности большинства программных инструментов ограничиваются морфологическим и синтаксическим анализом в сочетании с методами из теории вероятностей и математической статистики. Таким образом, лишь избранная часть относительно простых задач оказалась решенной, множество проблем предстоит решить в будущем.

Очевидно, причин для этого много. Например, существует мнение, что каждое правило в синтаксисе имеет свой аналог в семантике. Этот постулат называют гипотезой «правило к правилу» (rule-to-rule hypothesis) [35]. На самом деле это соответствие не является взаимно-однозначным, и в этом состоит главная сложность. Действительно, каждому синтаксическому правилу (дереву разбора) можно сопоставить семантическое правило (дерево разбора), но оно не будет единственным. Аналогично семантическому правилу сопоставляется

синтаксическое правило, но необязательно единственное. Именно эта неоднозначность приводит к неразрешимым на сегодняшний день проблемам в области автоматической обработки текстов. В связи с этим рассуждением возникает вопрос выбора нужного сопоставления из большого количества возможных вариантов.

Из всего вышесказанного можно сделать еще один очень важный вывод. Не следует рассматривать процессы генерации и интерпретации высказывания отдельно, так как они неразрывно связаны между собой. Выражая свою мысль, человек ориентируется на то, поймет ли его собеседник. В процессе генерации высказывания человек как бы перепроверяет себя, моделируя, как собеседник воспримет информацию. Похожий механизм действует при интерпретации высказывания. При осмыслении услышанного или прочитанного мы опять же сверяемся с нашими знаниями и представлениями о мире. Только благодаря этому нам удастся выбрать подходящий смысл.

Современные исследователи склоняются к мысли, что правильный выбор можно сделать, имея дополнительную базу знаний о мире. Такая база знаний должна содержать общесмысловую информацию о понятиях и отношениях между ними, чтобы при обращении к ней можно было определить подходящий контекст высказывания в автоматическом режиме. Она помогла бы учитывать накопленные знания о мире, которые в явном виде не присутствуют в конкретном высказывании, но непосредственно влияют на его смысл.

В данной статье предпринята попытка систематизировать известные на сегодняшний день достижения в области машинно-ориентированного семантического анализа. Расширенный вариант статьи доступен по ссылке <http://swsys-web.ru/methods-and-systems-of-semantic-text-analysis.html>.

Литература

1. Мельчук И.А. Опыт теории лингвистических моделей «Смысл–Текст». М.: Школа «Языки русской культуры», 1999. 346 с.
2. Лахути Д.Г., Рубашкин В.Ш. Семантический (концептуальный) словарь для информационных технологий // Научно-техническая информация. 2000. № 7. С. 1–9.
3. Падучева Е.В. Динамические модели в семантике лексики. М.: Языки славянской культуры, 2004. 608 с.
4. Тузов В.А. Компьютерная семантика русского языка. СПб: Изд-во СПбГУ, 2003. 391 с.
5. Батура Т.В., Мурзин Ф.А. Машинно-ориентированные логические методы отображения семантики текста на естественном языке: монография. Новосибирск: Изд-во НГТУ, 2008. 248 с.
6. Национальный корпус русского языка. URL: <http://www.ruscorpora.ru/> (дата обращения: 22.06.2016).
7. Апресян В.Ю., Апресян Ю.Д., Бабаева Е.Э., Богуславская О.Ю., Галактионова И.Г., Гловинская М.Я., Григорьева С.А., Иомдин Б.Л. и др. Новый объяснительный словарь синонимов русского языка. М.–Вена: Языки славянской культуры–Венский славистический альманах, 2004. 1488 с.
8. Хорошилов А.А. Методы автоматического установления смысловой близости документов на основе их концептуаль-

ного анализа // Электронные библиотеки: перспективные методы и технологии, электронные коллекции: тр. XV Всерос. науч. конф. RCDL' 2013. Ярославль: Изд-во ЯрГУ, 2013. С. 369–376.

9. Рубашкин В.Ш. Представление и анализ смысла в интеллектуальных информационных системах. М.: Наука, 1989. 189 с.

10. Лахути Д.Г., Рубашкин В.Ш. Средства и процедура концептуальной интерпретации входных сообщений на естественном языке // Изв. АН СССР: Сер. Технич. киберн. 1987. № 2. С. 49–59.

11. Рубашкин В.Ш. Семантический компонент в системах понимания текста // КИИ-2006: тр. 10 Национ. конф. по искусствен. интеллекту с междунар. участ. 2006. URL: <http://www.raai.org/resurs/papers/kii-2006/#dokladi> (дата обращения: 23.06.2016).

12. Падучева Е.В. Семантика вида и точка отсчета // Изв. АН СССР: Сер. лит. и яз. 1986. Т. 45. № 5. С. 18–25.

13. Падучева Е.В. Отпредикатные имена в лексикографическом аспекте // Научно-техническая информация. 1991. Сер. 2. № 5. С. 21–31.

14. WordNet. A lexical database for English. URL: <http://wordnet.princeton.edu/> (дата обращения: 23.06.2016).

15. Семантическая сеть. URL: https://ru.wikipedia.org/wiki/Семантическая_сеть (дата обращения: 23.06.2016).

16. Хабаров С.П. Представление знаний в информационных системах: конспекты лекций. URL: <http://www.habarov.spb.ru/bz/bz07.htm> (дата обращения: 23.06.2016).

17. Луценко Е.В. Представление знаний в информационных системах: электрон. учеб. пособие для студентов. Краснодар: Изд-во КубГАУ, 2010. 428 с.

18. Константинова И.С., Митрофанова О.А. Онтологии как системы хранения знаний // Информационно-телекоммуникационные системы: Всерос. конкурс. отбор статей, 2008. 54 с.

19. Темпоральная логика. URL: https://ru.wikipedia.org/wiki/Темпоральная_логика (дата обращения: 23.06.2016).

20. Разин В.В., Тузовский А.Ф. Представление знаний о времени с учетом неопределенности в онтологиях Semantic WEB // Докл. ТУСУР. 2013. № 2 (28). С. 157–162.

21. Fillmore Ch. The Case for Case. Proc. Texas Sympos. on Language Universals, 1967, 134 p.

22. Филлмор Ч. Дело о падеже // Новое в зарубежной лингвистике. М.: Прогресс, 1981. С. 369–495.

23. Рассел С., Норвиг П. Искусственный интеллект: современный подход. М.: Вильямс, 2007. 1408 с.

24. Jurafsky D., Martin J. Speech and language processing: an introduction to natural language processing, computational linguistics and speech recognition. 2008, 1024 p.

25. Вывод в логических моделях. Метод резолюций. URL: <http://www.aiportal.ru/articles/knowledge-models/method-resoluti-on.html> (дата обращения: 11.07.2016).

26. Boral H., Redfield S. Database Machine Morphology. Proc. 11th Intern. Conf. Very Large Data Bases, 1985, pp. 59–71.

27. Fushimi S., Kitsuregawa M., Tanaka H. An overview of the system of a parallel relational database machine GRACE. Proc. 12th Intern. Conf. Very Large Data Bases, 1986, pp. 209–219.

28. Tanaka H. Parallel Inference Engine. IOS Press Publ., 2000, 296 p.

29. Open Cognition. URL: <http://opencog.org/> (дата обращения: 23.06.2016).

30. Link Grammar Parser. AbiWord, 2014. URL: <http://www.abisource.com/projects/link-grammar/> (дата обращения: 20.06.2016).

31. The CMU Link Grammar natural language parser. URL: <https://github.com/opencog/link-grammar/> (дата обращения: 22.06.2016).

32. RelEx Dependency Relationship Extractor. OpenCog. URL: <http://wiki.opencog.org/wiki/home/index.php/RelEx> (дата обращения: 22.06.2016).

33. Сокирко А.В. Семантические словари в автоматической обработке текста (по материалам системы ДИАЛИНГ): дисс. ... канд. тех. наук. М.: МГПИИЯ, 2001. 120 с.

34. Автоматическая обработка текста. URL: <http://aot.ru/> (дата обращения: 23.06.2016).

35. Prószyński G. Machine Translation and the rule-to-rule hypothesis. New Trends in Translation Studies (In Honour of Kinga Klaudy). Budapest: Akadémiai Kiadó, 2005, pp. 207–218.

SEMANTIC ANALYSIS AND METHODS OF TEXT MEANING REPRESENTATION IN COMPUTER LINGUISTICS

T.V. Batura¹, Ph.D. (Physics and Mathematics), Senior Researcher, tatiana.v.batura@gmail.com

¹ A.P. Ershov Institute of Informatics Systems (IIS), Siberian Branch of the Russian Federation Academy of Sciences, Lavrentev Av. 6, Novosibirsk, 630090, Russian Federation

Abstract. The paper is devoted to the problems of semantic analysis of texts. The article discusses different methods, such as dependency diagrams, semantic network, approaches based on lexical functions and thematic classes, frame, ontological and logical models of knowledge representation. At the moment, there are many methods of representing sentence meaning.

Creating new methods of semantic analysis is significant in solving many problems of computational linguistics such as machine translation, automatic summarization, text classification and others. Development of new tools for semantic analysis is equally important.

Despite the fact that some of the scientific and technical ideas in natural language processing are evolved, many problems of semantic analysis remain unsolved. Most of researchers came to the conclusion that the dictionary for semantic analysis has to operate meanings and, therefore, describe the properties and relationships between concepts, rather than individual words. But there is a question: how to organize and represent information in these dictionaries to search it fast and conveniently, and in addition, take into account the changes in the natural language (the disappearance of old and the emergence of new concepts). This paper attempts to answer this and other questions. The article undertakes an attempt to systematize known achievements in the field of a semantic analysis, and in any measure to find the answer to this and other questions.

Keywords: semantic analysis, natural language processing, information retrieval, semantic networks, predicate logic, knowledge representation, meaning of the sentence.

References

1. Melchuk I.A. *Opyt teorii lingvisticheskikh modeley "Smysl–Tekst"*. Moscow, Yazyki russkoy kultury Publ., 1999, 346 p.
2. Lakhuti D.G., Rubashkin V.Sh. Semantic (conceptual) dictionary for information technologies. *Nauchno-tehnicheskaya informatsiya* [Scientific and Technical Information]. 2000, no. 7, pp. 1–9 (in Russ.).
3. Paducheva E.V. *Dinamicheskie modeli v semantike leksiki* [Dynamic Models in Lexis Semantics]. Moscow, Yazyki russkoy kultury Publ., 2004, 608 p.
4. Tuzov V.A. *Kompyuternaya semantika russkogo yazyka* [Russian Language Computer Semantics]. St. Petersburg, SPbGU Publ., 2003, 391 p.
5. Batura T.V., Murzin F.A. *Mashinno-orientirovannye logicheskie metody otobrazheniya semantiki teksta na estestvennom yazyke* [Computer Oriented Logical Methods of Text Semantics Representation on a Natural Language]. Monograph. Novosibirsk, NGTU, 2008, 248 p.
6. *Russian National Corpus*. Available at: <http://www.ruscorpora.ru/en/index.html/> (accessed June 22, 2016).
7. Apresyan V.Yu., Apresyan Yu.D., Babaeva E.E., Boguslavskaya O.Yu., Galaktionova I.G., Glovinskaya M.Ya., Grigoreva S.A., Iomdin B.L., Krylova T.V., Levontina I.B., Ptentsova A.V., Sannikov A.V., Uryson E.V. *Novy obyasnitelny slovar sinonimov russkogo yazyka* [The New Explanatory Dictionary of Russian Synonyms]. 2nd ed., Moscow, Vena, 2004, 1488 p.
8. Khoroshilov A.A. Methods of automatic setting documents' semantic adjacency based on their conceptual analysis. *Elektronnye biblioteki: perspektivnye metody i tekhnologii, elektronnye kollektsii: XV Vseross. nauch. konf. RCDL' 2013* [Proc. 15th All-Russian Scientific Conf. "Electronic Libraries: Prospect Methods and Technologies, Electronic Collections"]. Yaroslavl, YarGU Publ., 2013, pp. 369–376 (in Russ.).
9. Rubashkin V.Sh. *Predstavlenie i analiz smysla v intellektualnykh informatsionnykh sistemakh* [Meaning Representation and Analysis in Intelligent Information Systems]. Moscow, Nauka Publ., 1989, 189 p.
10. Lakhuti D.G., Rubashkin V.Sh. Tools and a Procedure of Conceptual Interpretation of Input Messages on a Natural Language. *Izvestiya AN SSSR. Tekhnicheskaya kibernetika* [News of USSR Academy of Sciences. Technical Cybernetics]. 1987, no. 2, pp. 49–59 (in Russ.).
11. Rubashkin V.Sh. Semantic component in text understanding systems. *Tr. 10 nats. konf. po iskusstvennomu intellektu s mezhdunar. uchastiem (KII-2006)* [Proc. 10th National Conf. on Artificial Intelligence with Int. Participation (KII-2006)]. 2006. Available at: <http://www.raai.org/resurs/papers/kii-2006/#dokladi> (accessed June 22, 2016).
12. Paducheva E.V. Semantic of a kind and a reference point. *Izvestiya AN SSSR. Seriya lit. i yaz.* [News of USSR Academy of Sciences. Literature and Language]. 1986, vol. 45, no. 5, pp. 18–25 (in Russ.).
13. Paducheva E.V. Edited names in a lexicographical aspect. *Nauchno-tehnicheskaya informatsiya* [Scientific and Technical Information]. 1991, iss. 2, no. 5, pp. 21–31 (in Russ.).
14. *WordNet. A lexical database for English*. Available at: <http://wordnet.princeton.edu/> (accessed June 23, 2016).
15. *Semantic network*. *Wikipedia – The Free Encyclopedia*. Available at: https://en.wikipedia.org/wiki/Semantic_network (accessed June 23, 2016).
16. Khabarov S.P. *Predstavlenie znany v informatsionnykh sistemakh* [Knowledge Representation in Information Systems]. Lecture notes. Available at: <http://www.habarov.spb.ru/bz/bz07.htm> (accessed June 23, 2016) (in Russ.).
17. Lutsenko E.V. *Predstavlenie znany v informatsionnykh sistemakh* [Knowledge Representation in Information Systems]. Electronic study guide. Krasnodar, KubGAU Publ., 2010, 428 p.
18. Konstantinova I.S., Mitrofanova O.A. Ontologies as knowledge storage systems. *Vseross. konkursny otbor obzorno-analiticheskikh statey po prioritetnomu napravleniyu "Informatsionno-telekommunikatsionnye sistemy"* [All-Russian Competitive Selection of Review and Analytical Articles on Priority Area "Information and Analytical Systems"]. 2008, 54 p.
19. *Temporal logic*. *Wikipedia – The Free Encyclopedia*. Available at: https://en.wikipedia.org/wiki/Temporal_logic (accessed June 23, 2016).
20. Razin V.V., Tuzovsky A.F. Time knowledge representation taking into account uncertainty in Semantic WEB ontologies. *Doklady TUSUR* [Proc. of TUSUR Univ.]. 2013, no. 2 (28), pp. 157–162 (in Russ.).
21. Fillmore Ch. The Case for Case. *Proc. of the Texas Symp. on Language Universals*. 1967, 134 p.
22. Fillmore Ch. *The Case for Case. Universals in Linguistic Theory*. In Bach and Harms (Ed.). NY, Holt, Rinehart, and Winston Publ., 1968 (Russ.ed.: Moscow, Progress Publ., 1981, pp. 369–495).
23. Russell S., Norvig P. *Artificial Intelligence: A Modern Approach*. 2nd ed., Prentice Hall Publ., 2002, 1132 p.
24. Jurafsky D., Martin J. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition*. 2008, 1024 p.
25. *Vyvod v logicheskikh modelyakh. Metod rezolyutsy* [Output in Logical Models. Resolution Method]. Available at: <http://www.aiportal.ru/articles/knowledge-models/method-resolution.html> (accessed July 11, 2016).
26. Boral H., Redfield S. Database Machine Morphology. *Proc. 11th Int. Conf. on Very Large Data Bases*. 1985, pp. 59–71.
27. Fushimi S., Kitsuregawa M., Tanaka H. An overview of the system of a parallel relational database machine GRACE. *Proc. 12th Int. Conf. on Very Large Data Bases*. 1986, pp. 209–219.
28. Tanaka H. *Parallel Inference Engine*. 2000, 296 p.
29. *Open Cognition*. Available at: <http://opencog.org/> (accessed June 23, 2016).
30. Link Grammar Parser. *AbiWord*. 2014. Available at: <http://www.abisource.com/projects/link-grammar/> (accessed June 20, 2016).
31. *The CMU Link Grammar natural language parser*. Available at: <https://github.com/opencog/link-grammar/> (accessed June 22, 2016).
32. RelEx Dependency Relationship Extractor. *OpenCog*. Available at: <http://wiki.opencog.org/wiki/home/index.php/RelEx> (accessed June 22, 2016).
33. Sokirko A.V. *Semanticheskie slovari v avtomaticheskoy obrabotke teksta (po materialam sistemy DIALING)* [Semantic Dictionaries in Text Automatic Processing (adapted from DIALING system)]. PhD thesis, 2001, 120 p. (in Russ.).
34. *Avtomaticheskaya obrabotka teksta* [Text Automatic Processing]. Available at: <http://aot.ru/> (accessed June 23, 2016).
35. Prószyński G. Machine Translation and the Rule-to-Rule Hypothesis. *New Trends in Translation Studies (In Honour of Kinga Klauzy)*. Budapest, Akadémiai Kiadó Publ., 2005, pp. 207–218.

УДК 004.912

DOI: 10.15827/0236-235X.116.058-062

Дата подачи статьи: 08.08.16

2016. Т. 29. № 4. С. 58–62

ИЗВЛЕЧЕНИЕ МЕТАДАННЫХ ИЗ ПОЛНОТЕКСТОВЫХ ЭЛЕКТРОННЫХ РУССКОЯЗЫЧНЫХ ИЗДАНИЙ ПРИ ПОМОЩИ ТОМИТА-ПАРСЕРА

*Р.С. Сулейманов, преподаватель, mail@ruslan.cc
(Московский педагогический государственный университет,
ул. Малая Пироговская, 1/1, г. Москва, 119991, Россия)*

При публикации материалов в электронных библиотеках возникает необходимость извлечения метаданных после перевода печатного текста в электронный, что при обработке текста вручную является трудозатратным процессом.

В данной работе рассматривается возможность извлечения метаданных с помощью Томита-парсера, предназначенного для извлечения фактов из текста на естественном языке. Для обеспечения наиболее точного извлечения были разработаны грамматики для анализа полнотекстовых изданий на русском языке, сформирован список метаданных, являющихся обязательными при публикации издания. Разработанные грамматики были апробированы на 100 изданиях, после чего на основании анализа сформулирован ряд закономерностей. С учетом выведенных закономерностей алгоритм был оптимизирован, что позволило повысить эффективность автоматического извлечения данных. Определена необходимость программной обработки полученных данных, например, удаления повторяющейся информации и приведения данных к общему виду перед их публикацией.

С помощью оптимизированного алгоритма проведен масштабный эксперимент по автоматизированному извлечению метаданных из 10 000 изданий, выполнено сравнение его результатов с множеством метаданных, полученных вручную. Предложенный метод автоматического извлечения данных позволил корректно извлечь 86,7 % метаданных, и еще 4 % могут быть использованы после корректировки. Наибольшие проблемы (21 % данных извлечен неверно) возникли с наименованиями материалов вследствие отсутствия четкой структуры. Для четко структурированной информации, такой как ISBN и коды рубрикаторов, процент извлечения приближается к 100 %. Однако было установлено, что, несмотря на увеличение скорости и простоту нахождения метаданных, полностью исключить человека из процесса невозможно.

Ключевые слова: метаданные, извлечение метаданных из электронных библиотек, извлечение метаданных из полнотекстовых изданий.

Постоянное увеличение объемов информации с одновременным ростом требований к их доступности является одной из глобальных задач в эпоху формирования цифровой инфраструктуры информационного общества, в котором информация становится одним из базовых активов, необходимых для развития страны, общества и личности. Развитие информационных технологий привело к созданию и совершенствованию новых форм генерации информации, однако сохранение и передача накопленных знаний по-прежнему являются важнейшей задачей, которая решается и путем формирования электронных библиотек. Эти библиотеки позволяют обеспечить доступ к полнотекстовым изданиям российских и зарубежных авторов с любого устройства с доступом в Интернет, в то время как доступ к материалам в обычных библиотеках ограничен вследствие их местонахождения и правил ознакомления с материалами. По данным Министерства культуры РФ, охват населения России библиотечным обслуживанием составляет 34,4 %, при этом количество посещений снижается, в то время как аудитория российского Интернета составляет 82 млн человек, то есть 66 % населения России.

Оцифровка печатных материалов – трудоемкий процесс, обусловленный большим объемом накопленных фондов, их состоянием и возникающими сопутствующими задачами. Если задачу наращивания

производительности процесса получения цифровых копий изданий можно решать за счет установки более мощных сканирующих устройств и/или увеличения их количества, то задачу последующей обработки издания для его размещения в электронной библиотеке так просто не решить [1]. К числу наиболее трудоемких задач, сопровождающих формирование цифровых копий, является извлечение метаданных, использующихся в качестве атрибутивной информации при публикации материалов в электронных библиотеках и позволяющих осуществлять навигацию по ней [2].

При наличии материала в существующей электронной библиотеке метаданные можно получить из нее благодаря интерфейсу программирования приложений (API – application programming interface), наличию XML или JSON-сокетам, а также простым синтаксическим анализом HTML-кода страниц электронной библиотеки.

В случае, если материал не содержится в других электронных библиотеках и оцифровывается впервые, извлечение метаданных возможно двумя способами: вручную или при помощи анализа полных текстов материала. Очевидно, что извлечение метаданных вручную трудозатратно и неэффективно.

В данной работе исследуется процедура анализа полных текстов печатных материалов при переводе их в цифровую форму с целью определения

возможности сокращения времени автоматизированного извлечения метаданных из полнотекстовых материалов.

Построение алгоритма извлечения метаданных

Перед извлечением метаданных из материала необходимо преобразовать его из печатной в электронную форму (например, путем сканирования с распознаванием символа или перепечатки), в результате чего будут получены исходные данные для анализа текста [3]. Для анализа предложений в данном исследовании был использован Томита-парсер, созданный российской компанией «Яндекс» в 2014 году на основе GLP-парсера (Generalized Left-to-right Rightmost derivation parser) – обобщенного восходящего магазинного анализатора, расширяющего алгоритм LR-парсера и предназначенного для разбора по недетерминированным и неоднозначным грамматикам. Томита-парсер анализирует текст на естественном языке с учетом синтаксиса языка и морфологии обрабатываемого текста [4].

Томита-парсер является программным обеспечением, открытым для свободного использования, однако при этом для работы с ним необходимо изначально сформировать исходные файлы.

Для решения задачи извлечения метаданных на естественном языке при помощи Томита-парсера требуется создать КС-грамматику, газзетиры и файлы, описывающие факты [5].

КС-грамматика – это набор правил, описывающих синтаксическую структуру извлекаемых цепочек. Газзетир представляет собой словарь с ключевыми словами, которые используются при анализе КС-грамматиками. В файлах, описывающих факты, строится связь между грамматикой и настраивается способ интерпретации грамматики в факт.

Для оценки эффективности анализа полных текстов печатных материалов для извлечения метаданных была сформирована выборка из 100 случайных книжных материалов БД электронной библиотеки «Научное наследие России» [6]. Проверялась возможность извлечения следующих метаданных, являющихся обязательной информацией об издании и требующихся при публикации в электронной библиотеке для обеспечения каталогизации и доступности материалов:

- название материала;
- сведения об авторах;
- код ISBN (уникальный номер книжного издания);
- год публикации;
- место публикации;
- сведения об издателе;
- коды рубрикаторов (УДК, ББК, ГРНТИ).

Для анализа текстов были сформированы и использованы грамматики, приведенные в таблице 1.

Анализ результатов эксперимента

В 87 % обработанных материалов были обнаружены повторяющиеся паттерны, анализ которых позволил выявить закономерности, в дальнейшем учитываемые для оптимизации алгоритма извлечения метаданных.

1. Все рассматриваемые метаданные содержатся на первых или на последних трех страницах оцифрованного материала. Таким образом, для извлечения необходимых метаданных из материала достаточен анализ указанных страниц, анализ других страниц для решения поставленных задач не подходит, что снижает количество анализируемой информации для обнаружения метаданных.

2. Название материала встречается в аннотации в двух возможных сочетаниях:

- «Название» / «Автор»;
- («Издание» или «Публикация») «Название».

3. Авторы могут указываться как перед, так и после названия. Возможно различное написание Ф.И.О. автора: как с использованием инициалов, так и с полным именем.

4. Код ISBN обычно обозначается путем представления слова ISBN перед цифровой последовательностью. В данном исследовании извлекались коды, соответствующие ISO 2108.

5. Год и место публикации указываются рядом. В качестве места публикации может выступать географический объект или наименование организации, например РАН, институт и пр.

6. Сведения об издательстве предваряются словом «ИЗДАТЕЛЬСТВО» или прилагательным «ИЗДАТЕЛЬСКИЙ» с существительными, например «ДОМ» или «ФИРМА».

7. Коды рубрикаторов предваряются наименованием рубрикатора, например «УДК», «КОДЫ».

Пример автоматического извлечения метаданных (фактов) приведен на рисунке.

Как видно из примера, полученные факты имеют ряд недостатков, в частности, данные повторяются в различных видах, как в случае с Ф.И.О. автора. Название книги также содержит информацию об авторе, в графе «место издательства» материала указаны и географические данные, и наименование организации. При обработке более широкого спектра материалов возможны другие сочетания дублирующейся и/или неточной информации.

Таким образом, возможно извлечение метаданных из полного текста материала, при этом на основании контрольной выборки из 100 книг были сформированы дополнительные уточняющие правила, позволяющие провести поиск метаданных с большей точностью и меньшими затратами вре-

Таблица 1

Грамматики для извлечения метаданных

Table 1

Grammars to extract metadata

Метаданные	Используемая грамматика
ISBN	S -> ('ISBN') (':') ('-') AnyWord<wfl="[0-9]{1,10}(-)?[0-9]{1,10}(-)?[0-9]{1,10}(-)?[0-9]{1,10}(-)?[0-9]{1}">; Isbn -> S interp (Material.Isbn);
Информация об издателе	PublisherDescr -> (Adj) 'издательство' 'издательский' Noun; ForFact -> Word<h-reg1, gnc-agr[1], rt> (Word<gnc-agr[1]>*); CityOnly -> Word<gram="reo">; ForCity -> CityOnlyinterp (Material.PlaceOfPublish); S -> (ForCity) PublisherDescrForFactinterp (Material.Publisher::not_norm); S -> (ForCity) PublisherDescrForFact<quoted>interp (Material.Publisher::not_norm);
Коды рубрикаторов	UDKStart -> 'удк' (':') ('-'); UDKDescr -> AnyWord<wff="/[0-9]{1,5}(\. -)?([0-9]{1,5})?((\.-)?([0-9]{1,5})?(\. -)?([0-9]{1,5})?)/>interp (Material.RubricsUDK) (','); UDK ->UDKStartUDKDescr+; BBKStart -> 'ббк' (':') ('-'); BBKDescr -> AnyWord<wff="/[0-9]{1,5}(\. -)?([0-9]{1,5})?((\.-)?([0-9]{1,5})?(\. -)?([0-9]{1,5})?)/>interp (Material.RubricsBBK); BBK ->BBKStartBBKDescr+; GrntiStart -> 'грнти' (':') ('-'); GrntiDescr -> AnyWord<wff="/[0-9]{1,5}(\. -)?([0-9]{1,5})?((\.-)?([0-9]{1,5})?(\. -)?([0-9]{1,5})?)/>interp (Material.RubricsGrnti); Grnti ->GrntiStartGrntiDescr+; S -> BBK UDK Grnti;
Дата и место публикации	CityOrOrg -> Word<gram="reo"> "пан" interp (Material.PlaceOfPublish); S ->CityOrOrg (',') AnyWord<wfl="18[0-9]{2} 19[0-9]{2} 20[0-1][0-9]">interp (Material.YearOfPublish);
Автор и наименование	Initial ->Word<wff="/[А-Я]\./>; Initials ->Initial<h-reg1>Initial<h-reg1>; FullName ->InitialsWord<gram="фам"> Word<gram="фам">Initials Word<gram="фам"> (',') Word<gram="имя">Word<gram="отч">; Person ->FullNameinterp (Material.Person::not_norm); Year -> (',') AnyWord<wfl="18[0-9]{2} 19[0-9]{2} 20[0-1][0-9]">interp (Material.YearOfPublish) (',') EOSent ; FromStart ->AnyWord<fw, h-reg1>AnyWord*; MaterialName ->FromStartinterp (Material.Name::not_norm) ('/') Person; NotFromStart ->AnyWord<h-reg1>AnyWord*; MaterialName -> 'научный' 'издание' NotFromStartinterp (Material.Name::not_norm);

мени и ресурсов. Однако полученные при этом данные не подходят в полной мере для незамедлительного использования и нуждаются в дополнительной корректировке.

После обработки полного текста с помощью Томита-парсера и извлечения данных возможно сохранение извлеченной информации в форматах Google Protobuf, обычного текста или XML, после чего может быть осуществлена корректировка. Дополнительная программная обработка требуется для удаления повторяющихся данных, приведения их к общему виду, распределения информации по

верным рубрикам и иной корректировки извлеченных данных из выбранного формата и формирования итогового набора метаданных. После обработки полученных фактов возможно их использование для присвоения атрибутивной информации соответствующим изданиям.

Оценка эффективности разработанной методики извлечения метаданных

Для подтверждения корректности извлечения метаданных из полных текстов электронных вер-

Material							
Name	Person	Isbn	YearOfPublish	PlaseOfPublish	Publisher	RubricsUDK	RubricsBBK
						7.0	
			2009	РАН			85
				Москва	Восточная литература		
Николаева Н.С. Образы Японии: очерки и заметки	Н.С. Николаева						
		978-5-02-036405-9					
		978-5-02-036405-9					
	Николаева Н.С.						
		978-5-02-036405-9					
			2009	РАН			

Пример извлечения фактов из полного текста книги

The example of fact extraction from a full book text

сий печатных материалов был проведен ряд экспериментов. В качестве тестовой площадки для их проведения использован набор данных, содержащий 10 000 русскоязычных книг из БД электронной библиотеки «Научное наследие России» [7]. Для всех материалов, подлежащих тестированию, были доступны метаданные, с которыми было проведено сравнение извлеченных данных методом сравнения полей.

В результате экспериментов получены результаты, представленные в таблице 2.

Таблица 2

**Корректность извлечения метаданных
из тестовой выборки материалов**

Table 2

Metadata extraction accuracy in a test data sample

Поле	Извлечено верно (%)	Извлечено неверно (%)	Требуется уточнение (%) *
Наименование материала	76	21	3
Сведения об авторах	91	7	2
Код ISBN	98	0	2
Год публикации	89	10	1
Место публикации	84	12	4
Сведения об издателе	79	14	7
Коды рубрикаторов	90	1	9
Результаты в среднем	86,7	9,3	4

* Выявлены ошибки при оптическом распознавании текста (OCR), данные извлечены не полностью либо извлечена лишняя информация.

Средний показатель корректно извлеченных метаданных составил 86,7 %, еще 4 % извлеченных фактов поддаются последующей корректировке и могут быть использованы после ее проведения. При этом наибольшие проблемы наблюдаются с извлечением наименований материалов, которые не имеют четко утвержденной структуры, могут содержать любое количество символов и знаков препинания.

Извлечение сведений об издательствах также проблематично, поскольку может не подчиняться вышеприведенным правилам и не быть обозначенным при печати, что затрудняет поиск данных по тексту, однако вместе с данными, поддающимися корректировке, процент доступного извлечения превышает 80.

Наиболее полной обработке поддаются код ISBN и коды рубрикаторов из-за четкой структуры, а также благодаря тому, что в большинстве случаев коды предваряются соответствующим названием, что значительно облегчает их поиск. При этом они поддаются корректировке, так что процент их извлечения приближается к 100.

Извлечение ISBN может помочь, если материал и данные о нем были ранее размещены кем-либо в сети Интернет. В таком случае метаданные можно уточнить путем поиска по кодам, например через Google Books ISBN API [8].

Таким образом, в данной работе предложены правила обработки книжных материалов для эффективности поиска метаданных. При проведении эксперимента было показано, что автоматизированное извлечение метаданных из полных текстов русскоязычных книг позволяет существенно сократить затраты на ввод метаданных в любую электронную библиотеку, однако полностью исключить участие человека в этом процессе невозможно. Для обеспечения максимально полной и достоверной информации о материале необходима редакторская проверка корректности полученных программным путем метаданных.

Процент успешного извлечения метаданных из полных текстов можно увеличить за счет улучшения качества оптического распознавания печатных материалов, а также улучшения КС-грамматик и газзетиров.

Литература

1. Кириллов С.А. Эволюция систем оцифровки печатных изданий на примере их использования в проекте ЭБ «Научное наследие России» // Информационное обеспечение науки. Новые технологии: сб. науч. тр. 2011. С. 227–237. URL: http://www.benran.ru/SEMINAR/SEM/Sb_11/sbornik/doc_227.pdf (дата обращения: 07.08.2016).

2. Антопольский А.Б. Системы метаданных в электронных библиотеках // Библиотеки и ассоциации в меняющемся мире: новые технологии и новые формы сотрудничества: сб. тр. VIII Междунар. конф. 2001. URL: <http://gpntb.ru/win/inter-events/crimea2001/tom/sec4/Doc5.html> (дата обращения: 07.08.2016).
3. Васильев А., Козлов Д., Самусев С., Шамина О. Извлечение метаданных и библиографических ссылок из текстов русскоязычных научных статей // RCDL2007: сб. тр. Всерос. науч. конф. Переславль: Изд-во Ун-та г. Переславля, 2007. Т. 1. С. 175–181.
4. Economopoulos G.R. Generalised LR parsing algorithms. PhD thesis, Univ. of London Royal Holloway, August 2006, 253 p.
5. Инструмент для извлечения структурированных данных из текста Томита-парсер. URL: <https://tech.yandex.ru/tomita/> (дата обращения: 06.08.2016).
6. Электронная библиотека «Научное наследие России». URL: <http://e-heritage.ru/index.html> (дата обращения: 06.08.2016).
7. Каленов Н.Е., Савин Г.И., Серебряков В.А., Сотников А.Н. Принципы построения и формирования электронной библиотеки «Научное наследие России» // Программные продукты и системы. № 4. 2012. С. 30–40.
8. Using the API. URL: <https://developers.google.com/books/docs/v1/using> (дата обращения: 06.08.2016).

Software & Systems

DOI: 10.15827/0236-235X.116.058-062

Received 08.08.16

2016, vol. 29, no. 4, pp. 58–62

EXTRACTION OF METADATA FROM THE FULL-TEXT ELECTRONIC MATERIALS WRITTEN IN RUSSIAN USING TOMITA-PARSER

R.S. Suleymanov¹, Lecturer, mail@ruslan.cc

¹ Moscow State University of Education, M. Pirogovskaya St., 1/1, Moscow, 119991, Russian Federation

Abstract. Publishing information in digital libraries requires metadata extraction after transforming initial material into e-text. This procedure is time-consuming in case of performing it manually. This paper considers metadata extraction using Tomita-parser method, which is software designed to extract facts from a natural language text. To ensure the most accurate extraction there were formulated spatial grammars for analyzing full-text books in Russian and a list of metadata for publication was made. Designed spatial grammars were tested on 100 editions, the analysis served as a base for observing a number of consistent patterns. The algorithm has been optimized with regard of derived patterns. This allowed improving the efficiency of automatic data extraction. The authors determined a need for manual data processing, such as removing repetitive information and data reduction to general view before publishing. The optimized algorithm helped to conduct a large-scale experiment of metadata automated extraction from 10,000 publications. Its results were compared to manually extracted data. The proposed method allows extracting correctly up to 86,7 % of meta-data with further 4% which can be used after adjustment. The biggest problem (21 % of data were extracted incorrectly) has been discovered in the names of the materials due to the lack of a clear structure. As for clearly structured information (such as ISBN and rubricator codes) the percentage of correct extraction approaches 100 %. However, despite the speed increase and easiness of metadata extracting, it was proved that it is impossible to completely eliminate a human from the process.

Keywords: metadata, metadata extraction from electronic libraries, metadata extraction from full-text materials.

References

1. Kirillov S.A. Print media digitizing system evolution on the example of their using in the EB project “Scientific heritage of Russia”. *Informatsionnoe obespechenie nauki. Noveye tekhnologii: sb. nauch. tr.* [Proc. “Russian Information Support. New technologies”]. 2011, pp. 227–237 (in Russ.).
2. Antopolsky A.B. Metadata systems in electronic libraries. *Sb. statey konf. “Biblioteki i assotsiatsii v menyayushchemsya mire: novye tekhnologii i novye formy sotrudnichestva”* [Proc. Conf. “Libraries and Associations in a Changing World: New Technologies and New Forms of Cooperation”]. Crimea, 2001. Available at: <http://gpntb.ru/win/inter-events/crimea2001/tom/sec4/Doc5.html> (accessed August 7, 2016).
3. Vasilev A., Kozlov D., Samusev S., Shamina O. Extracting metadata and references from Russian scientific papers. *Trudy konf. RCDL2007* [Proc. Conf. RCDL2007]. Pereslavl, Pereslavl Univ. Publ., 2007, vol. 1, pp. 175–181 (in Russ.).
4. Giorgios R. *Generalised LR Parsing Algorithms*. PhD thesis, Univ. of London Royal Holloway, 2006, 253 p.
5. *Instrument dlya izvlecheniya strukturirovannykh dannykh iz teksta Tomita-parser* [Tomita-parser tool to extract structured data from texts]. Available at: <https://tech.yandex.ru/tomita/> (accessed August 6, 2016).
6. *Nauchnoe nasledie Rossii* [Scientific Heritage of Russia]. Available at: <http://e-heritage.ru/index.html> (accessed August 6, 2016).
7. Kalenov N.E., Savin G.I., Serebryakov V.A., Sotnikov A.N. Scientific Heritage of Russia Digital Library: construction and sources aggregation philosophy. *Programmnye produkty i sistemy* [Software & Systems]. 2012, no. 4, pp. 30–40 (in Russ.).
8. *Using the API*. Available at: <https://developers.google.com/books/docs/v1/using> (accessed August 6, 2016).

MODELS OF HETEROGENEOUS AGENTS' BEHAVIOR IN A CONFLICT AND THEIR IMPLEMENTATION ALGORITHMS

N.Yu. Mutovkina¹, Ph.D. (Engineering), Associate Professor, letter-boxnm@ya.ru;

A.Yu. Klyushin¹, Ph.D. (Engineering), Associate Professor, klalex@inbox.ru;

N.A. Semenov¹, Dr.Sc. (Engineering), Professor, is@tstu.tver.ru

¹*Tver State Technical University, Nikitin Quay 22, Tver, 170026, Russian Federation*

Abstract. The article discusses pair models of agents' behavior in a conflict. Each agent has one of three possible behavioral types: deviating, compromise and coercing. A behavioral type determines an agent's nature and specifies agent's reaction that should be expected in response to specific actions. The authors select the need to share limited resources as the main cause of a conflict in simulation. The agents need resources to implement assigned tasks and to exist themselves. It is determined that in a conflict situation each agent tries to obtain more resources, than its opponent at the moment. The value of this commitment depends on an agent's behavioral type. The article identifies the technique of selecting the best behavioral model. It also proposes the measures to resolve conflict between agents. The authors applied the basic concepts of fuzzy logic in simulation.

Keywords: *conflict between agents, behavioral type, fuzzy logic, behavioral model, level of resource availability.*

The authors have already discussed the problem of interaction between agents in a *multi-agent system* (MAS) in [1–3]. They have also considered the models of agents' goal-seeking behavior and analyzed the impact of these models on system stability. The value of agents' winning depends on the behavior model which is chosen by an agent. Special attention is paid to the interaction model of agents in conflict. The conflict between agents is interpreted as a contest in which agents seek to achieve mutually exclusive statuses; each of them wants something that only one of them can receive [1]. The main causes of conflicts are: limited shared resources that agents have to share among themselves; the differences of agents' interests, goals and beliefs, behavioral patterns caused by behavioral types [4]; differences in the number and complexity of tasks assigned to agents. At the same time each agent owns a certain behavioral type r , $r \in R$, that reflects agent's preferences. A behavioral type is a value of an agent's state when performing a specific task; R is a set of possible preferences of an agent. In [5] the authors introduced the following semantic rules: the agent is deviating if $r \in (0, 0.5)$; the agent is called compromise (collaborating) if $r \in [0.5, 0.8]$ and the agent is coercive if $r \in (0.8, 1)$. According to the results of the study, compromise agents are ready to make concessions when sharing resources and tend to avoid conflict situations. Coercing agents usually seek to capture as many resources from other agents as possible. Deviating agents are characterized by the fact that the agent relies only on himself and prefers avoiding any interactions with other agents. Nevertheless, conflicts are inevitable when a problem is solved by two or more agents. Typically, the initiators of a conflict are coercing agents, but in some cases the agents of other behavioral types can also trigger a conflict. A conflict situation can arise even when there is only one agent when he starts having some doubts about the correctness of actions and behavior in different operation conditions. Then the

subject of research is an intrapersonal conflict [6]. Utopia is the idea of the possibility of achieving an absolutely peaceful state in MAS, when agents interact in harmony with each other and with themselves. As a rule, the efforts for total elimination of conflicts do not lead to a positive result. It should be noted, that in addition to a destructive function conflicts can have a positive impact: different points of view are identified, we may receive additional information, etc. [7]. So the focus should be on the control and management of conflicts in MAS.

Problem statement

When constructing a behavioral model of two agents in a conflict, the authors took into account the following considerations:

- each agent in MAS is seeking to increase his resource potential within his capabilities, even if he is not a rival for other agents. This desire is dictated by the logic: no resources no life;
- a conflict can occur between agents of the same behavioral type, as well as between agents of different types;
- each agent in the conflict will seek to increase his resources depending on the resources of other agent;
- the rates of increasing resources constrain the limitation of their total number, costs of available resources for new resources (the costs to support agent's activity) and "fatigue" of agents due to the struggle for resources;
- environment may affect the change of agent's resource potential regardless its behavioral type. For example, some agents may share their resources with this agent, while others, on the contrary, take aggressive actions.

A agents' behavioral model in conflict situations is a dynamic model with a discrete time t , $t = \overline{0, T}$, where

T is a completion moment of resource distribution.

Simulation uses the following notations:

- R_t^i, R_t^j are resource availability levels of agents i and j in moment of time t , $0 \leq R_t^i, R_t^j < 1$ respectively;
- η_t^i, η_t^j are reaction coefficients of the agents i and j on the acquisition of part of resources by an opponent $0 < \eta_t^i < R_t^i$, $0 < \eta_t^j < R_t^j$ respectively;
- λ_t^i, λ_t^j are the coefficients expressing “fatigue” of relevant agents from the struggle for resources in conflict, $0 < \lambda_t^i < R_{t-1}^i$, $0 < \lambda_t^j < R_{t-1}^j$;
- χ_t^i, χ_t^j are the coefficients expressing that agents realize the needed in resources, a level of “maturity instinct” in resource potential capacity, $0 < \chi_t^i, \chi_t^j < 1$;
- $\varepsilon_t^i, \varepsilon_t^j$ are random components that are associated with environmental influence on agents’ activities, $-1 < \varepsilon_t^i, \varepsilon_t^j < 1$. The closer their values to (-1) , the more hostile the environment. Positive values indicate the benevolence of the environment. A zero value indicates a neutral environment.

R_t^{total} is a total volume of resources in the system at the moment of time t determined by the formula:

$$R_t^{total} = R_{t-1}^{total} - (R_{t-1}^i + R_{t-1}^j),$$

$$t = 0, T, R_0^{total} = 1. \quad (1)$$

Thus, the model of the behavior of two agents in a conflict can be represented as:

$$\begin{cases} R_{t+1}^i = \eta_{t+1}^i \cdot R_t^j - \lambda_{t+1}^i \cdot R_t^i + \chi_{t+1}^i \cdot R_t^{total} + \varepsilon_{t+1}^i, \\ R_{t+1}^j = \eta_{t+1}^j \cdot R_t^i - \lambda_{t+1}^j \cdot R_t^j + \chi_{t+1}^j \cdot R_t^{total} + \varepsilon_{t+1}^j, \end{cases}$$

$$t = 0, T. \quad (2)$$

Values $\eta_t^i, \eta_t^j, \lambda_t^i, \lambda_t^j, \chi_t^i, \chi_t^j$ depend on the behavioral type agents and are calculated:

$$\begin{cases} \eta_{t+1}^i = \frac{[k_1(r_i)] \cdot (R_t^j / R_t^i)}{100}, \\ \eta_{t+1}^j = \frac{[k_1(r_j)] \cdot (R_t^i / R_t^j)}{100}, \end{cases} \quad (3)$$

where $k_1(r_i), k_1(r_j)$ are constants that are determined by a psycho-behavioral type of a corresponding agent. They show the agent’s desire to join the fight for the right to possess resources;

$$\begin{cases} \lambda_{t+1}^i = [k_2(r_i)] \cdot R_t^i, \\ \lambda_{t+1}^j = [k_2(r_j)] \cdot R_t^j, \end{cases} \quad (4)$$

where $k_2(r_i), k_2(r_j)$ are constants which depend on a type of an agent. It is determined empirically that agents prone to evasion and competition (aggression) spend more energy than compromise agents since the latter receive additional energy from positive emotions

that arise from resolving a conflict through compromise [8]. Such agent will always receive at least a part of a planned winning, but it’s better than nothing.

$$\begin{cases} \chi_{t+1}^i = 1 - \left| \left(R_t^{total} - (R_t^i + R_t^j) \right) \right|^{k_3(r_i)}, \\ \chi_{t+1}^j = 1 - \left| \left(R_t^{total} - (R_t^i + R_t^j) \right) \right|^{k_3(r_j)}, \end{cases} \quad (5)$$

where $k_3(r_i), k_3(r_j)$ are constants calculated as an average agent type (or average condition), i.e. $k_3(r_i) = \bar{r}_i$, $k_3(r_j) = \bar{r}_j$.

The resource allocation process ends with the following situation:

$$R_{t+1}^i + R_{t+1}^j \geq R_t^{total}. \quad (6)$$

The task is to analyze six pairwise interaction models of agents in a conflict. It is necessary for obtaining answers to the following questions:

- how much the resource distribution depends on a psycho-behavioral type of an agent and when the distribution is faster;
- how allocation of resources changes depending on a resource volume at the initial stage;
- what are resource allocation rates when reducing the overall level of resources R_t^{total} ;
- which interaction model is the most optimal for given initial conditions, i.e. allows allocating the resource R^{total} fully in a reasonable time?

The simulation algorithm

Construction of behavioral models of heterogeneous agents in a conflict includes several stages:

- analysis of psycho-behavioral types of interacting agents and determining the values of the coefficients $k_1(r_i), k_1(r_j), k_2(r_i), k_2(r_j), k_3(r_i), k_3(r_j)$. It is shown in table 1;
 - determining the values $R_0^i, R_0^j, R_0^{total}, \varepsilon_0^i$ and ε_0^j ;
 - calculating the values $\eta_{t+1}^i, \eta_{t+1}^j, \lambda_{t+1}^i, \lambda_{t+1}^j, \chi_{t+1}^i, \chi_{t+1}^j$ when $t = 0, T$ using the formulas (3)–(5);
 - calculating the values R_{t+1}^i, R_{t+1}^j using formulas (2) and defining the balance of resources R_t^{total} by the formula (1);
 - testing the condition (6); if it is fulfilled, the process stops;
 - calculating a cumulative summary of the distribution by the formulas:
- $$\begin{aligned} cumR_T^i &= \sum_{t=0}^T R_t^i, \quad cumR_T^j = \sum_{t=0}^T R_t^j, \\ cumR_T^{total} &= cumR_T^i + cumR_T^j; \end{aligned} \quad (7)$$
- forming the final tables – table 2 and table 3;
 - conclusions and suggestions based on the simulation results;

- numerical experiments with changing source data;
- analysis of the results of the experiments and general conclusions.

Modeling is possible in any software environment that supports drawing up charts and graphs and allows performing mathematical calculations. The authors have chosen MS Excel 2013 for its simplicity and clarity.

Demo

According to the abovementioned algorithm, we have determined the following values of the coefficients $k_1(r_i)$, $k_1(r_j)$, $k_2(r_i)$, $k_2(r_j)$, $k_3(r_i)$, $k_3(r_j)$ for the listed psycho-behavioral types of agents.

Table 1

The values of the constants that depend on a psycho-behavioral type of an agent

Agent type	Constants		
	k_1	k_2	k_3
Deviating	1,0	0,50	0,25
Compromise	1,0	0,35	0,65
Coercing	1,5	0,50	0,90

Let $T=5$, $R_0^{total} = 1$, $R_0^i = 0,01$, $R_0^j = 0,01$, $\varepsilon_0^i = 0$ and $\varepsilon_0^j = 0$, i.e.:

- at the zero phase there is 100 % resource that should be distributed in five iterations;
- each agent at the zero stage is assigned to 1 % of resource;
- the influence of external factors is absent, i.e. no one helps agents, but no one interferes with them.

In addition, we consider a situation when the resource is not replenished. It is believed that a given quantity of a resource should be enough for the task.

As a rule, the conflict intensifies in terms of nonrenewable resources. Therefore, one of the measures allowing minimizing a conflict between agents is the replenishment of resources. However, from the economic point of view, such event is not always possible.

Stages 3, 4 and 6 of the algorithm involve the formation of calculated tables and graphs with cumulative results (fig. 1 and fig. 2). The highest point on each chart indicates the full allocation of resources with creation of a specific deficit in virtue of the conflict. So at the end of the distribution we may take the previous step.

In figure 1a) illustrates the interaction of deviating agents, in figure 1b) illustrates the interaction of agents of a compromise type and in figure 1c) illustrates the behavior when dividing the resource between coercing type agents.

As it can be seen from figure 1, the behavior of compromise type agents is the most similar to the resource allocation condition. At the fourth iteration, they still have 37 % unallocated resource. When maintaining a conflict at the fifth iteration they have a resource shortage of 26 %. Therefore, even agents of a compromise type are not able to act effectively in the context of a conflict.

Figure 2d) shows the behavior of deviating agents (agent i resource level) and compromise agents (agent j resource level) types. Figure 2e) shows the interaction of deviating agents (agent i resource level) and coercive agents (agent j resource level). Figure 2f) shows the interaction of compromise agents (agent i resource level) and coercive agents (agent j resource level).

From the presented behavior models in figure 2 the best model is 2f).

Table 2 provides the information about which step ends the allocation of resources between agents resulting from the condition (6).

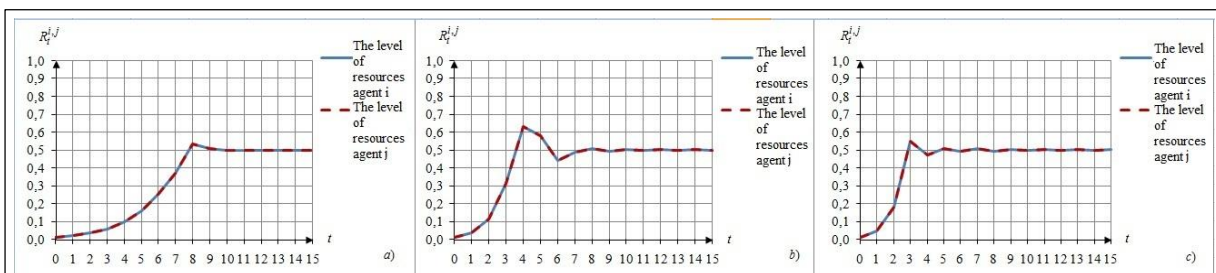


Fig. 1. Distribution curves of same type agents' resource

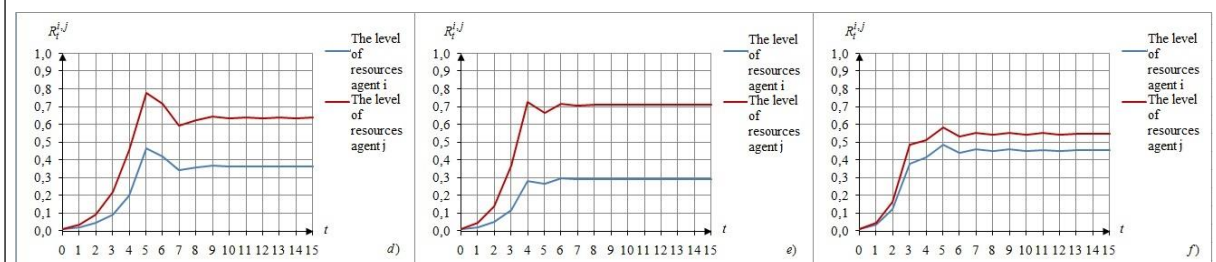


Fig. 2. Distribution curves of resource of different type agents

Table 2
The results of time distribution

Types of agents	Deviating	Compromise	Coercing
Deviating	7; 1a)	4; 2d)	3; 2e)
Compromise	4; 2d)	3; 1b)	4; 2f)
Coercing	3; 2e)	4; 2f)	2; 1c)

Table 2 shows that resource allocation time, that most closely matches the specified condition $T=5$, is equal to four iterations. It is typical for models of the interaction “Deviating \leftrightarrow Compromise” (2d)) and “Compromise \leftrightarrow Coercing” (2f)). Thus, we confirm a hypothesis about a constructive side of a conflict: negotiation between agents of different types, one of which is compromise, extend the field of agents’ activity; give them the opportunity to negotiate and to come closer to fulfilling these conditions. All other models of interaction are recognized as ineffective. In models 1b), 2e) and 1c) the distribution of resource occurs too quickly and the agents “don’t realize” the essence of the task because of confrontation. In the model 1a) the resource is distributed too slowly, so performing tasks by such agents is delayed.

Table 3 shows how much resource is left unallocated at the moment T .

Table 3
Unallocated resource in agents’ behavioral models of in a conflict

Types of agents	Deviating	Compromise	Coercing
Deviating	0,26; 1a)	0,35; 2d)	0,52; 2e)
Compromise	0,35; 2d)	0,37; 1b)	0,07; 2f)
Coercing	0,52; 2e)	0,07; 2f)	0,64; 1c)

The best distribution score is obtained in the model 2f), since it has only 7 % of unallocated resource on the fifth iteration of the distribution. The remaining values

show that agents need to change their strategy of behavior, otherwise they will experience a significant resource shortage that will not allow them to complete the task in the allotted time. Although the model 2f) is the most effective among all presented models, it is also not perfect.

To change the strategy of agents’ behavior and minimize a conflict we can use the following activities:

- if possible, to ensure reserve replenishment;
- assisting agents from the outside, for example, providing opportunities for other agents to share their resources with the considered agents;
- selecting agents that can effectively express themselves working in pairs despite the possibility of conflict between them.

For example, under the same conditions assisting the i -th agent only in the first iteration as $\varepsilon_1^i = 0,01$ (fig. 3a) allows a full distribution of the resource (100 %), but for 8 iterations.

Now 3d) model is the best, when the assistance is provided for a deviating agent. The assistance for a compromise-type agent in model 3f) causes negative emotions from a coercing-type agent, which only worsens a conflict.

Further numerical experiments showed that it is important to be attentive to the initial distribution of resources between agents and before assisting any of them to analyze the need for this. It is possible that the result will be like: “We wished our best, you know the rest!”

Additionally, we analyze the dependence of agents’ “fatigue” from their psycho-behavioral type, i.e. we check the fulfillment of the condition (8):

$$\lambda_i^i \lambda_i^j > \eta_i^i \eta_i^j, \quad (8)$$

where $\lambda_i^i \lambda_i^j$ is an assessment of agents’ “fatigue” from conflict; $\eta_i^i \eta_i^j$ is evaluation of agents’ reaction to the actions of each other.

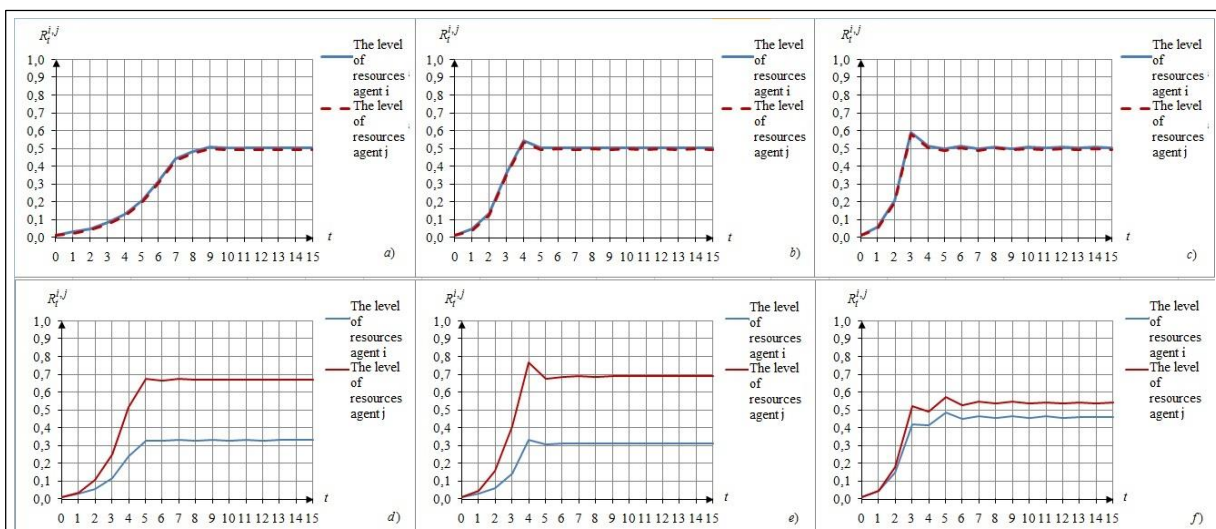


Fig. 3. Changes in agents’ behavior when providing a one-time assistance to the i -th agent

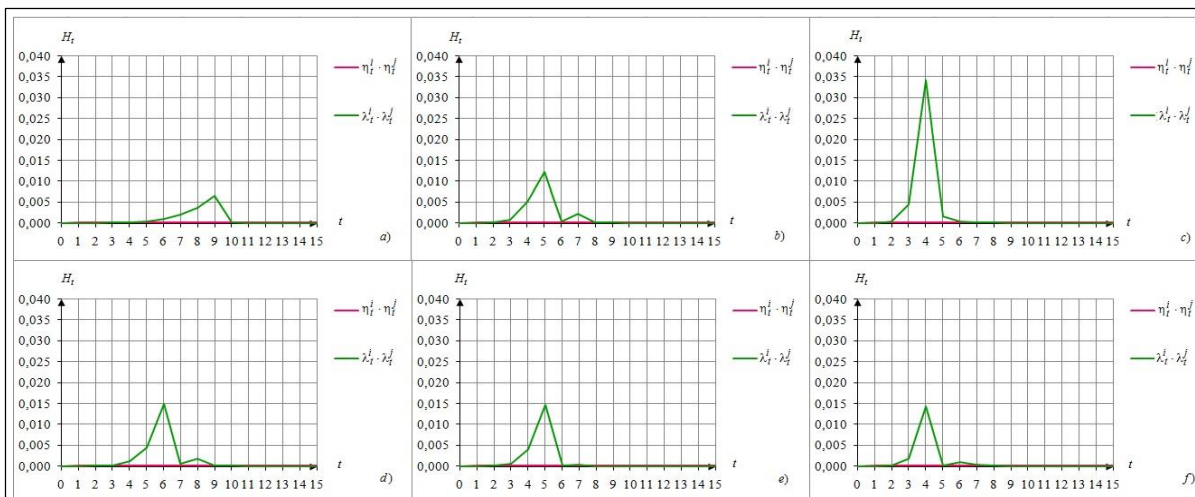
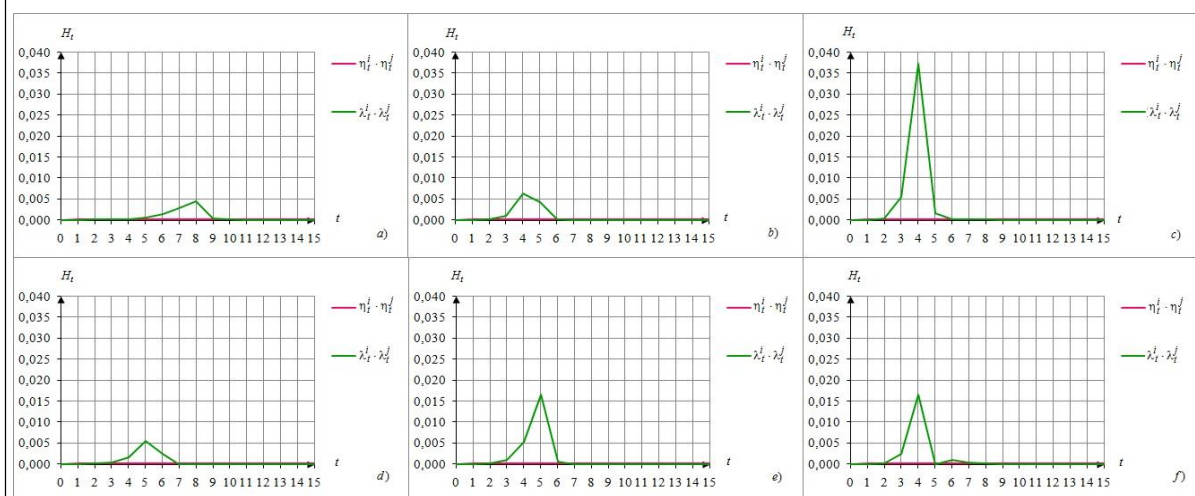


Fig. 4. The dependence of agents' "fatigue" on their type when interacting in conflict

Fig. 5. The dependence of agents' "fatigue" on their type when providing assistance to the i -th agent

The period of operation of the following inequality (8) shows the active phase of the conflict. In figure 4 shows the active phases of the conflict between agents in all considered models.

The graphs in figure 4 correspond to agents' behavior during sharing of resources presented in figure 1 and figure 2. It is obvious that coercing agents experience the greatest "fatigue" (fig. 4c), and deviating agents experience the lowest "fatigue" (fig. 4a). Other behavioral models are characterized by negotiation, so the peak of "fatigue" is about on the same level.

Figure 3 presents the models with little changes in the graphs of this dependence (fig. 5).

Now we can see the least "fatigue" in models a) and d). Based on the simulation results above, when assisting the deviating agent in the amount of 1 % on the first iteration, the model d) is recognized as the best.

Conclusion

Since this study considers an agent as an intelligent anthropomorphic entity characterized by mental prop-

erties, the description of pair models of agents' behavior in conflict is based on the hypothesis of the implementation of agents behavior proposed by American scientists K. Thomas and R. Kilmann [9, 10]. They identified five main styles of behavior in a conflict situation: adaptation (pliability), deviation, competition (rivalry), cooperation and compromise. Thus, classification features are: the degree of implementation of agent's own interests and achieving their goals; the degree in which they consider the interests of the opponent. Deviation and pliability strategies are peculiar to deviating agents. Compromise agents can follow the adaptation strategy, as well as and the compromise strategy. Coercing agents typically choose the strategy of confrontation. The simulation results show that both agents can win with only one strategy. It is cooperation.

Acknowledgements. The research was supported by the RFBR, project no. 15-01-05545-a.

References

1. Mutovkina N.Yu., Kuznetsov V.N., Klyushin A.Yu. The influence of goal-seeking behavior of agents on multi-agent system

stability. *Sistemy upravleniya i informatsionnye tekhnologii* [Systems of Control and Information Technologies]. 2014, no. 2 (56), pp. 43–48 (in Russ.).

2. Mutovkina N.Yu., Klyushin A.Yu., Kuznetsov V.N. Methods of harmonized management of conflict in a multi-agent system. *Sistemy upravleniya i informatsionnye tekhnologii* [Systems of Control and Information Technologies]. 2014, no. 3.2 (57), pp. 255–261 (in Russ.).

3. Klyushin A.Yu., Kuznetsov V.N., Mutovkina N.Yu. *The fuzzy models of the behavior of individuals and groups making decisions*. Tver, TvSTU Publ., 2015, part 2, 248 p. (in Russ.).

4. Fisher R., Ury W. *Getting to Yes: Negotiating agreement without giving in*. Penguin Group Publ., UK, 1981, 200 p.

5. Mutovkina N.Yu., Klyushin A.Yu., Kuznetsov V.N. Semantic definition of the agent type in a multi-agent system. The problem of interaction between agents. *Open Semantic Technologies for Intelligent Systems (OSTIS-2013). Proc. 3rd Int. Sc. Conf.*

(Golenkov V.V., ed.). Minsk, BGUIR Publ., 2013, pp. 309–316 (in Russ.).

6. Antsupov A. *Konfliktologiya* [Conflictology]. SPb, Piter, 2013, 304 p. (in Russ.).

7. Desyatov D.B. *The conflict theory*. Voronezh, Sc. Book Publ., 2008, 346 p. (in Russ.).

8. Ponomarev Yu.P. *Game models. Mathematical methods. Psychological analysis*. Moscow, Nauka, 1991, 160 p. (in Russ.).

9. Kilmann R.H., Thomas K.W. Interpersonal conflict-handling behavior as reflections of Jungian personality dimensions. *Psychological Reports*. 1975, vol. 37, no. 3, pp. 971–980. Available at: www.kilmanndiagnostics.com (accessed August 9, 2015).

10. Kilmann R.H., Thomas K.W. Developing a forced-choice measure of conflict-handling behavior: The “mode” instrument. *Educational and Psychological Measurement*. 1977, vol. 37, no. 2, pp. 309–325. Available at: www.kilmanndiagnostics.com (accessed August 9, 2015).

УДК 004.942

DOI: 10.15827/0236-235X.116.063-068

Дата подачи статьи: 11.08.16

2016. Т. 29. № 4. С. 63–68

МОДЕЛИ ПОВЕДЕНИЯ РАЗНОТИПНЫХ АГЕНТОВ В УСЛОВИЯХ КОНФЛИКТА И АЛГОРИТМЫ ИХ РЕАЛИЗАЦИИ

Н.Ю. Мutowкина, к.т.н., доцент, letter-boxnm@ya.ru;

А.Ю. Ключин, к.т.н., доцент, klalex@inbox.ru;

Н.А. Семенов, д.т.н., профессор, is@tstu.tver.ru

(Тверской государственный технический университет, наб. Аф. Никитина, 22, г. Тверь, 170026, Россия)

В статье рассматриваются парные модели поведения агентов в условиях конфликта, каждый из которых обладает одним из трех возможных поведенческих типов: уклоняющийся, компромиссный и принуждающий. Поведенческий тип определяет характер агента и указывает, какой реакции от агента следует ожидать в ответ на конкретные действия. В качестве основной причины конфликта при моделировании выбрана необходимость делить ограниченный объем ресурсов. Ресурсы необходимы для выполнения агентами поставленных задач и самого существования агентов. Установлено, что в ситуации конфликта каждый агент стремится получить большее количество ресурсов, чем есть в данный момент у соперника. Величина этого стремления зависит от поведенческого типа агента. В работе определена методика выбора лучшей модели поведения, а также предложены мероприятия по устранению конфликтной ситуации между агентами. При моделировании были применены основные постулаты нечеткой логики.

Ключевые слова: межагентный конфликт, поведенческий тип, нечеткая логика, модель поведения, уровень ресурсной обеспеченности.

Работа выполнена при поддержке РФФИ, грант № 15-01-05545-а.

Литература

1. Мutowкина Н.Ю., Кузнецов В.Н., Ключин А.Ю. Влияние целеустремленного поведения агентов на устойчивость многоагентной системы // Системы управления и информационные технологии. 2014. № 2 (56). С. 43–48.

2. Мutowкина Н.Ю., Кузнецов В.Н., Ключин А.Ю. Методы согласованного управления конфликтом в многоагентной системе // Системы управления и информационные технологии. 2014. № 3.2 (57). С. 255–261.

3. Ключин А.Ю., Кузнецов В.Н., Мutowкина Н.Ю. Нечеткие модели поведения лиц и групп, принимающих решения: монография. Ч. 2. Тверь: Изд-во ТГТУ, 2015. 248 с.

4. Фишер Р., Юри У. Путь к согласию, или Переговоры без поражения; [пер. с англ.]. М.: Наука, 1990. 160 с.

5. Мutowкина Н.Ю., Ключин А.Ю., Кузнецов В.Н. Семантическое определение типа агента в многоагентной системе. Проблема межагентного взаимодействия // Открытые семантические технологии проектирования интеллектуальных систем = Open Semantic Technologies for Intelligent Systems (OSTIS-2013): матер. III Междунар. науч.-технич. конф. Минск: Изд-во БГУИР, 2013. С. 309–316.

6. Анцупов А. Конфликтология. СПб: Питер, 2013. 304 с.

7. Десятов Д.Б. Теория конфликта: монография. Воронеж: Научная книга, 2008. 346 с.

8. Пономарев Ю.П. Игровые модели. Математические методы. Психологический анализ. М.: Наука, 1991. 160 с.

9. Kilmann R.H., Thomas K.W. Interpersonal Conflict-Handling Behavior as Reflections of Jungian Personality Dimensions. *Psychological Reports*. 1975, vol. 37, no. 3, pp. 971–980. URL: www.kilmanndiagnostics.com (дата обращения: 9.08.2015).

10. Kilmann R.H., Thomas K.W. Developing a Forced-Choice Measure of Conflict-Handling Behavior: The “Mode” Instrument. *Educational and Psychological Measurement*. 1977, vol. 37, no. 2, pp. 309–325. URL: www.kilmanndiagnostics.com (дата обращения: 9.08.2015).

УДК 004.925

DOI: 10.15827/0236-235X.116.069-072

Дата подачи статьи: 15.09.16

2016. Т. 29. № 4. С. 69–72

ОТОБРАЖЕНИЕ ТРЕХМЕРНЫХ ОБЪЕКТОВ С ИСПОЛЬЗОВАНИЕМ КЛАСТЕРНОЙ ВИЗУАЛИЗАЦИИ

А.М. Гуацинтов, научный сотрудник, algts@inbox.ru;

К.А. Мамросенко, к.т.н., руководитель, kirillam@ya.ru

(Центр визуализации и спутниковых информационных технологий НИИСИ РАН,
Нахимовский просп., 36, корп. 1, г. Москва, 117218, Россия)

В статье рассмотрены методы освещения трехмерных объектов, позволяющие существенно повысить реалистичность виртуальных трехмерных сцен, приведены их достоинства и недостатки, а также предложены подходы к организации данных об источниках света трехмерной сцены в памяти видеокарты.

Зачастую для корректного освещения трехмерной сцены требуется значительное количество источников света. При традиционном (forward rendering) подходе расчет освещения производится следующим образом: для каждой вершины в трехмерной сцене и каждого пикселя конечного изображения, выводимого на экран, последовательно производится расчет влияния каждого источника света в сцене, то есть для каждого пикселя выполняются расчет освещения и геометрическая обработка. Соответственно, при большом количестве источников света производительность подсистемы визуализации может быть существенно снижена.

Одним из решений данной проблемы является применение методов так называемой отложенной визуализации, в частности, отложенного освещения. Основная идея заключается в отделении этапа геометрической обработки от этапа освещения трехмерной сцены. Прорисовка изображения осуществляется в несколько этапов. Вся геометрия сцены прорисовывается один раз, при этом сохраняется информация о цвете, нормалях и глубине прорисовки для каждого пикселя в промежуточный G-Buffer, который используется на последующих этапах прорисовки. Однако прорисовка трехмерных сцен при помощи отложенной визуализации также имеет ряд ограничений.

В статье описаны несколько методов устранения некоторых недостатков отложенной визуализации с сохранением ее преимуществ. Это так называемые тайловая и кластерная визуализации. Приводятся данные измерений производительности системы визуализации, использующей различные методы освещения.

Ключевые слова: тренажерно-обучающая система, подсистема визуализации, тренажер, рендеринг, отложенная визуализация, освещение, кластерная визуализация, тайловая визуализация.

Тренажерно-обучающая система оператора сложной технической системы (далее ТОС) – техническое средство для исследований и подготовки операторов сложных технических систем, отвечающее требованиям методик подготовки, обеспечивающее получение знаний, навыков и умений, реализующее модель таких систем и осуществляющее контроль над действиями обучаемого. ТОС могут применяться для формирования индивидуальных профессиональных навыков и умений, а также для отработки групповых операций [1, 2].

Подсистема визуализации ТОС обеспечивает отображение результатов моделирования внешней среды и объекта управления с помощью устройств отображения информации. Воспроизведение визуальной картины должно быть с достаточной подробностью содержанием, позволяющим операторам ТОС успешно выполнять поставленные задачи [3]. Важную роль в достижении реалистичности отображаемой сцены играет освещение.

Для корректного освещения трехмерной сцены, как правило, требуется значительное количество источников света. При традиционном (forward rendering) подходе расчет освещения производится следующим образом: для каждой вершины в трехмерной сцене и каждого пикселя конечного изображения, выводимого на экран, последовательно производится расчет влияния каждого источника света в сцене, то есть этапы расчета освещения и геометрической обработки выполняются для каж-

дого пикселя. Кроме того, расчеты производятся даже для скрытых и перекрывающихся поверхностей, а также для тех пикселей, которые могут не попасть в итоговое изображение. Соответственно, при большом количестве источников света производительность подсистемы визуализации может быть существенно снижена.

Одним из решений данной проблемы является применение методов так называемой отложенной визуализации, например, отложенного освещения, основная идея которого в отделении этапа геометрической обработки от этапа освещения трехмерной сцены. Прорисовка изображения осуществляется в четыре этапа [4]: геометрическая обработка, освещение, постобработка и заключительный этап. Вся геометрия сцены прорисовывается один раз, при этом информация о цвете, нормалях и глубине прорисовки для каждого пикселя сохраняется в промежуточный G-Buffer, который используется на последующих этапах прорисовки [5]. Далее с использованием сохраненной в G-Buffer информации производится расчет освещения. Затем итоговое изображение копируется в кадровый буфер.

Однако прорисовка трехмерных сцен при помощи отложенной визуализации имеет множество ограничений. Во-первых, существуют значительные проблемы с отображением полупрозрачных объектов из-за того, что в G-Buffer сохраняется информация только о ближайшем к виртуальной камере трехмерном объекте. Во-вторых, из-за слож-

ности реализации ограничено применение аппаратного сглаживания изображения (Multisample Anti-aliasing – MSAA). В-третьих, отложенная визуализация характеризуется значительным расходом видеопамати, который увеличивается при повышении разрешения итогового изображения, а также значительным объемом данных, передаваемых по шине данных при прорисовке каждого кадра.

В настоящее время существует несколько методов устранения некоторых недостатков отложенной визуализации с сохранением ее преимуществ. Одним из них является тайловая (плиточная) визуализация, которая может применяться как с традиционным подходом к освещению трехмерной сцены, так и с отложенным [6–8]. Отложенная визуализация с помощью тайлов позволяет уменьшить объем передаваемых данных по шине, но увеличивает общий объем вычислений, связанных с освещением. С учетом мощностей современных графических адаптеров применение отложенной визуализации с помощью тайлов обеспечивает значительное увеличение производительности системы визуализации по сравнению с «чистой» отложенной визуализацией.

Принцип тайловой визуализации состоит в группировании пикселей итогового изображения, которое будет отображено на экране, в прямоугольные участки (тайлы), а также в использовании значения глубины каждого пикселя для определения дополнительных зон видимости – усеченных пирамид (view frustum). Соответственно, тайлы, в которых находятся пиксели со схожими значениями глубины, например, в данных пикселях отображается один и тот же трехмерный объект, определяют достаточно небольшой объем трехмерного пространства. В то же время для тайлов, пиксели которых существенно отличаются значениями глубины, дальняя граница ограничивающего объема тайла должна совпадать с максимальным значением глубины его пикселей. Это снижает эффективность отсечения обрабатываемых источников света и приводит к прямой зависимости производительности от выводимого изображения. Данный аспект особенно нежелателен в приложениях, работающих в режиме реального времени (не менее 25 кадров в секунду), так как в этом случае практически невозможно гарантировать стабильность частоты смены кадров системы визуализации.

Еще одним методом является так называемая кластерная визуализация, которая решает проблемы тайловой визуализации. Каждый кластер имеет фиксированную глубину, что снижает зависимость производительности от отображаемого изображения. Применение кластерного рендеринга позволяет использовать более миллиона источников света в виртуальной трехмерной сцене в реальном масштабе времени [9].

По аналогии с тайловой визуализацией кластерная визуализация может быть задействована как с

традиционным (forward) расчетом освещения, так и с отложенным (deferred). Основные этапы кластерной визуализации:

- прорисовка трехмерной сцены в G-Buffer (в случае использования кластерной отложенной визуализации);
- создание массива кластеров на основе информации о размерах трехмерной сцены или усеченной пирамиды видимости;
- поиск уникальных кластеров, задействованных в данном кадре прорисовки;
- сохранение информации об источниках света в найденных уникальных кластерах.

Одной из основных задач кластерной визуализации является создание массива кластеров, то есть разбиение трехмерного пространства на определенное количество сегментов. Необходимо, чтобы кластеры были небольшого размера для охвата как можно меньшего количества источников света, в то же время они должны быть достаточного размера для сохранения требуемого уровня производительности при обработке кластеров. Одним из методов создания массива кластеров является разбиение виртуального трехмерного пространства на равные промежутки – создание сетки. Этот метод позволяет быстро определять уникальный ключ каждого кластера, а также создавать кластеры одинакового размера. Однако данный подход требует ручной настройки для каждой отдельной трехмерной сцены, к тому же уникальный ключ каждого кластера может занимать значительное число битов в зависимости от размера сцены. Кроме того, из-за проецирования сетки кластеров на экран дальние кластеры становятся маленькими на итоговом изображении, вплоть до размера одного пикселя, что снижает общую производительность.

Еще один метод – создание массива кластеров на основе информации о видимой части трехмерной сцены. Аналогично тайловой визуализации итоговое изображение разбивается на определенное количество прямоугольников, а затем производится разбиение видимого пространства вглубь трехмерной сцены. Наиболее простым методом разбиения пространства является разделение глубины видимого пространства трехмерной сцены на равные сегменты в нормализованных координатах устройства (на отрезке значений [0, 1]). Однако распределение значений в нормализованных координатах устройства нелинейно, что приводит к неравномерным размерам кластеров: у ближней границы прорисовки виртуальной камеры кластеры становятся очень тонкими, в то время как у дальней границы очень длинными.

Чтобы кластеры наиболее соответствовали форме куба, применяется экспоненциальное разделение пространства по следующей формуле:

$$near_k = near \left(1 + \frac{2 \tan \theta}{S_y}\right)^k, \text{ где } near - \text{ближняя гра-}$$

ница зоны видимости виртуальной камеры; S_y – вы-

сота ячеек кластера по оси y ; 2θ – угол обзора виртуальной камеры; k – итерация разделения видимого виртуального пространства.

Количество делений видимого пространства должно быть ограничено во избежание существенных потерь производительности при обработке. Зачастую видимое пространство разделяют на 16 частей вглубь, а размер тайлов по высоте и ширине составляет от 32 до 64 пикселей.

После разделения пространства на кластеры необходимо определить список используемых кластеров. Одним из способов является передача информации обо всех кластерах в шейдерные программы. Преимущества такого подхода в простоте реализации и фиксированных размерах передаваемых данных, недостаток – в значительном расходе памяти на хранение информации об источниках света в каждом кластере. Например, для разрешения FullHD (1 920×1 080) и с учетом возможности сохранения информации о 256 источниках света в каждом кластере объем занимаемой информации составит приблизительно 4 Мб. С учетом двойной буферизации (в случае заполнения списка на центральном процессоре) объем составит 8 Мб.

Другим способом является определение наибольшего возможного количества одновременно используемых кластеров. Практически ни в одной трехмерной сцене источники света не охватывают все кластеры. Следует определить количество одновременно используемых кластеров в трехмерной сцене с максимальным количеством источников света и, например, использовать для записи информации о кластерах буфер данных, размер которого в два раза превышает тот, который ранее определен по числу кластеров.

На следующем этапе определяется количество источников света в каждом кластере. Основная задача – выявление пересечений ограничивающего контура источника света с кластерами. Для каждого типа источника света (наиболее часто применяются направленные источники света в виде конуса, и всенаправленные источники света в виде сферы) используется свой вид определения пересечений. Для всенаправленного источника света часто применяется ограничивающая сфера, что затрудняет определение пересечений с кластерами: большинство алгоритмов отсечения работают тогда, когда сфера значительно меньше проверяемого ограничивающего объема, в то время как в случае с кластерами ограничивающая сфера источника света зачастую в разы больше, чем кластер. Один из способов решения проблемы – использование ограничивающих кубов (Axis Aligned Bounding Box – AABB) для всенаправленных источников света и их сравнение с ограничивающим пространством каждого кластера.

Для направленных источников света определение пересечений при помощи ограничивающего куба дает значительное количество ложных сраба-

тываний, когда множество кластеров, которые на самом деле не пересекаются обрабатываемым источником света, все равно задействованы в дальнейших расчетах освещения. Одним из решений данной проблемы является создание усеченной пирамиды (frustum) для обрабатываемого источника света, посредством которой будут определяться пересечения с кластерами [10].

Информацию об используемых в обрабатываемом кадре системы визуализации кластерах и источниках света следует передать в память видеокарты для использования в шейдерных подпрограммах. Одним из способов является хранение информации о кластерах и источниках света в различных текстурах. В шейдерной подпрограмме определение конкретного кластера производится при помощи координат текущего обрабатываемого пикселя в итоговом изображении и значения глубины обрабатываемого пикселя. Информация об используемых кластерах может храниться в 3D-текстуре, в каждом текселе которой сохраняются информация о количестве направленных и всенаправленных источников в данном кластере, а также ссылка на данные об источниках света. Если же информация о различных источниках света занимает различное количество байтов, может быть создана промежуточная текстура с индексами элементов массива с источниками света. Тогда в трехмерной текстуре кластеров будет храниться информация об используемом индексе в промежуточной текстуре. В зависимости от количества одновременно используемых источников света объем передаваемых на видеокарту данных для каждого кадра подсистемы визуализации может составлять 50–100 Кб.

По производительности визуализация с использованием кластерного разбиения пространства и освещения в среднем в два раза превосходит визуализацию с использованием отложенного освещения: на тестовой сцене с использованием 1 024 источников света частота смены кадров системы визуализации с отложенной визуализацией составляла 96 кадров в секунду, для кластерной визуализации – 216 кадров в секунду. Кластерная визуализация с традиционным подходом к освещению (clustered forward) приблизительно на 20 % производительнее тайловой визуализации с традиционным подходом и примерно на 25 % медленнее тайловой визуализации с отложенным расчетом освещения. Измерения производились на оборудовании Intel Core i7-950, Nvidia Geforce 470, на тестовой сцене Crytek Sponza.

В настоящее время визуализация с использованием кластерного подхода к освещению начинает активно использоваться в мультимедийных проектах. Кластерная визуализация не ограничена пропускной способностью шины данных, что характерно для отложенной визуализации. Кроме того, в отличие от тайловой визуализации у кластерной

нет прямой зависимости между производительностью и отображаемым изображением. Возможность использования как традиционного расчета освещения, так и отложенного вместе с кластерным разбиением пространства позволяет использовать данный метод визуализации для различных проблемно-ориентированных отраслей, где требуется реалистичная визуализация трехмерных объектов в реальном масштабе времени.

Работа выполняется в рамках проекта № 2.9 программы фундаментальных исследований ОНИТ РАН.

Литература

1. Гиацинтов А.М., Мамросенко К.А., Решетников В.Н. Инструментальные средства предтренажерной и тренажерной подготовки операторов сложных технических систем // Программные продукты, системы и алгоритмы. 2014. № 1. URL: <http://swsys-ru/simulator-training-operators.html> (дата обращения: 14.09.2016).

2. Решетников В.Н. Космические телекоммуникации (начала). М.: Изд-во МАТИ, 2013. 184 с.

3. Мамросенко К.А., Решетников В.Н. Моделирование подстилающей поверхности в имитационных системах // Программные продукты и системы. 2015. № 4. С. 70–74.

4. Alamia M. Tutorial – simple OpenGL deferred rendering. Coding Labs. URL: http://www.codinglabs.net/tutorial_simple_def_rendering.aspx (дата обращения: 08.07.2016).

5. Борецков А. Steps3d – Tutorials – Deferred Shading. URL: <http://steps3d.narod.ru/tutorials/ds-tutorial.html> (дата обращения: 08.07.2016).

6. Olsson O., Assarsson U. Tiled shading. J. Graph. GPU Game Tools, 2011, vol. 15, no. 4, pp. 235–251.

7. Trebilco D. Light-indexed deferred rendering. Shader X7, 2009, pp. 243–256.

8. Harada T. Forward+: Bringing deferred lighting to the next level. URL: https://takahiroharada.files.wordpress.com/2015/04/forward_plus.pdf (дата обращения: 08.07.2016).

9. Olsson O., Billeter M., Assarsson U. Clustered deferred and forward shading. High Perform. Graph. 2012. С. 1–10.

10. Persson E. Practical clustered shading. URL: www.humus.name/Articles/PracticalClusteredShading.pdf (дата обращения: 08.07.2016).

Software & Systems

DOI: 10.15827/0236-235X.116.069-072

Received 15.09.16

2016, vol. 29, no. 4, pp. 69–72

3D OBJECTS RENDERING USING CLUSTERED SHADING

A.M. Giatsintov¹, Researcher, algts@inbox.ru

K.A. Mamrosenko¹, Ph.D. (Engineering), Head of the Research Center, kirillam@ya.ru

¹ Center of Visualization and Satellite Information Technologies SRISA, Nakhimovsky Ave. 36/1, Moscow, 117218, Russian Federation

Abstract. The article describes methods of lighting and shading of 3D objects that significantly enhance the realism of three-dimensional virtual scenes, their advantages and disadvantages. It also presents approaches for organization of light sources data in graphics card memory.

Usually a large number of light sources are required in order to shade the 3D scene correctly. With forward rendering lighting is computed in the following way: influence of each light source on the resulting image is computed for each vertex in 3D scene and each pixel of the framebuffer. Therefore, visualization subsystem performance may be significantly reduced when there are a lot of light sources in a scene.

In order to solve this problem we can use the methods of deferred rendering. The main idea of deferred rendering is to separate the geometry processing phase from lighting phase. An image is rendered in several phases. Scene geometry is rendered only once and information about color, normal and depth of each pixel is stored in temporary G-Buffer that is used in the following phases of rendering. But deferred rendering also has a number of significant downsides.

This article provides information about some methods that resolve most of drawbacks of deferred rendering while retaining its advantages. They include tiled visualization and clustered visualization. The paper provides performance measurements of visualization subsystem with various rendering methods.

Keywords: training simulation system, visualization subsystem, trainer, rendering, deferred shading, lighting, clustered rendering, tile rendering.

Acknowledgements. This work is done within the ONIT fundamental research program project no. 2.9.

References

1. Giatsintov A.M., Mamrosenko K.A., Reshetnikov V.N. A toolkit for pre-training and training operators of complex technical systems. *Programmnye produkty, sistemy i algoritmy* [Software, Systems and Algorithms]. 2014, no. 1. Available at: <http://swsys-ru/simulator-training-operators.html> (accessed September 14, 2016) (in Russ).

2. Reshetnikov V.N. *Kosmicheskie telekommunikatsii (nachala)* [Space telecommunications (beginning)]. Moscow, ITTs MATI Publ., 2013, 2nd ed., 184 p. (in Russ).

3. Mamrosenko K.A., Reshetnikov V.N. Terrain modelling in simulation training systems. *Programmnye produkty i sistemy* [Software & Systems]. 2015, no. 4, pp. 70–74 (in Russ).

4. Boreckov A. *Steps3d – Tutorials – Deferred Shading*. Available at: <http://steps3d.narod.ru/tutorials/ds-tutorial.html> (accessed July 8, 2016) (in Russ).

5. Alamia M. *Coding Labs. Simple OpenGL deferred rendering tutorial*. Available at: http://www.codinglabs.net/tutorial_simple_def_rendering.aspx (accessed July 8, 2016).

6. Olsson O., Assarsson U. Tiled Shading J. *Graph. GPU Game Tools*. 2011, vol. 15, no. 4, pp. 235–251.

7. Trebilco D. Light-indexed deferred rendering. *Shader X7*. 2009, pp. 243–256.

8. Harada T. *Forward+: Bringing deferred lighting to the next level*. Available at: https://takahiroharada.files.wordpress.com/2015/04/forward_plus.pdf (accessed July 8, 2016).

9. Olsson O., Billeter M., Assarsson U. Clustered Deferred and Forward Shading. *High Perform. Graph.* 2012, pp. 1–10.

10. Persson E. *Practical clustered shading*. Available at: www.humus.name/Articles/PracticalClusteredShading.pdf (accessed July 8, 2016).

УДК 004.42

DOI: 10.15827/0236-235X.116.073-077

Дата подачи статьи: 03.03.16

2016. Т. 29. № 4. С. 73–77

АНАЛИЗ УРОВНЕЙ ИНТЕГРАЦИИ КОМПОНЕНТОВ ГЕТЕРОГЕННЫХ ИНФОРМАЦИОННЫХ СИСТЕМ

О.В. Щекочихин, к.т.н., доцент, slim700@yandex.ru

*(Костромской государственный технологический университет,
ул. Дзержинского, 17, г. Кострома, 156005, Россия)*

П.В. Шведенко, магистрант, pitk1@mail.ru

*(Санкт-Петербургский национальный исследовательский университет
информационных технологий, механики и оптики (Университет ИТМО),
Кронверкский просп., 49, г. Санкт-Петербург, 197101, Россия)*

Интеграция разрозненных информационных систем предприятия на сегодняшний день является наиболее рациональным способом построения его единой информационной среды. Для ее реализации необходим класс интегрирующих инструментов. В статье проанализированы основные уровни интеграции гетерогенных информационных систем. Предложены авторская классификация уровней интеграции, а также варианты взаимодействия информационных систем в зависимости от целей интеграции. Проанализированы варианты интеграции на уровне брокеров, данных, сервисов и интерпретирования метаданных. Предложенная классификация является попыткой выделить ключевые, с точки зрения практической реализации, варианты интеграции информационных систем. Приведены примеры успешной реализации задач интеграции на каждом из предложенных уровней. Указаны сильные и слабые стороны каждого варианта интеграции. Уровень брокеров выгодно использовать в том случае, когда доступен исходный код интегрируемых приложений и можно дописать необходимый модуль интеграции. Интеграцию на уровне данных можно рекомендовать для информационных систем, схемы данных которых позволяют построить единую непротиворечивую структуру данных. Основными достоинствами уровня сервисов являются возможность многократного использования, слабая связанность сервисов друг с другом, отсутствие необходимости в единой БД или хранилище. При работе на уровне интерпретации метаданных появляется возможность быстро перенастраивать метаданные при интеграции данных из нового источника. Классификация дает возможность определить наиболее рациональный вид инструментов и набор методов создания интегрированной системы с заданными характеристиками. Авторы считают, что наиболее перспективным видом интеграции компонентов управляющих информационных систем, который требует дальнейшего изучения и совершенствования, является интеграция узкоспециализированных систем на уровне сервисов.

Ключевые слова: интеграция, информационная система, уровни интеграции.

В данной статье рассматриваются гетерогенные системы, то есть системы, состоящие из разнородных элементов, не обладающих свойством взаимозаменяемости.

Положение дел в области информационных технологий исторически сложилось так, что создан достаточно большой объем программных продуктов, библиотек, программных модулей, СУБД, систем программирования и т.п., существенно различающихся по функционалу. Задача современного проектирования *информационных систем* (ИС) сводится к тому, что на основе известных решений создается программная среда, которая выполняет множество функций. Это осуществляется через проектные решения, например, agile-технологии, позволяющие создать систему с актуальным функционалом в сжатые сроки.

Развитие теории проектирования ИС идет по двум направлениям. Первый путь связан с наращиванием функционала информационной среды, например, как это делается в IC, GreyGim. Второй путь разработки ИС нацелен на выполнение ограниченного класса функций.

Проще делать функционально ограниченные системы и использовать инструмент интеграции, чем полнофункциональные системы [1]. Для реализации такой парадигмы необходим класс инте-

грирующих инструментов, одним из которых являются стандартные программные интерфейсы. Успешный пример применения стандартных интерфейсов – CALS-технологии. В работе [2] сделан обзор стандартов и моделей данных, нацеленных на интеграцию ИС предприятия на основе MES- или ERP-систем. Однако этот пример является частным случаем и не может быть распространен на создание ИС другого класса. Заявкой на интеграцию явилось создание стандартов, объединенных вокруг модели взаимодействия открытых систем OSI. На представительском уровне модели OSI реализованы функции преобразования данных, преобразования между различными наборами символов, сжатие данных, шифрование.

Возможны следующие реализации идей интеграции однофункциональных и малофункциональных ИС: во-первых, встраивание универсального модуля интеграции в программный продукт; во-вторых, создание среды взаимодействия однофункциональных модулей через известные форматы передачи данных.

Однако не все ИС, использующиеся на предприятии, имеют возможности интеграции. В таком случае говорят об односторонней интеграции. В работе [3] приведен пример односторонней интеграции ИС электронного архива документов и пер-

сонифицированного учета в рамках отработки бизнес-процесса – бизнес процесс обработки документов персонифицированного учета Пенсионного фонда России.

Для решения практических задач интеграции ИС важно понимать ее цели, а также оценивать ресурсы, которые готово потратить предприятие на перестройку ИС. Именно поэтому исследование критериев классификации вариантов интеграции и методов интеграции данных так актуально и рассматривается во множестве работ. Ключевые подходы к решению проблем интеграции и их широкая вариативность представлены в работе [4]. Факторы, усложняющие интеграцию, и варианты минимизации негативного влияния этих факторов описаны в [5]. К числу таких факторов относят концептуальную разницу в архитектуре, технологическую разницу, несовместимость лицензий. В работе [6] предлагается рассматривать проблему интеграции с позиции представления данных и выделять синтаксический и семантический подходы.

В настоящей работе предлагается следующая система классификации уровней интеграции.

По степени автоматизации выделяются три уровня интеграции: ручная, автоматизированная и автоматическая. По методу создания связи между интегрируемыми элементами системы рекомендуется выделять уровни брокеров, данных, сервисов, интерпретирования метаданных.

Каждый уровень классификации имеет свои цели интеграции и инструменты и может быть рекомендован для решения соответствующего класса задач.

Классификация дает возможность определить наиболее рациональный вид инструментов и набор методов создания интегрированной системы с заданными характеристиками. Предложенная классификация является попыткой выделить ключевые, с точки зрения практической реализации, варианты интеграции ИС.

Зависимость степени автоматизации и методов создания связи при интеграции показана в таблице.

Интеграция на уровне брокеров. Данный уровень интеграции использует набор инструментов API- и СОМ-технологий. Цель интеграции – автоматическая передача данных и запуск исполняемого кода на выполнение. Для конечного пользователя интегрированные компоненты представляют единую систему.

Преимущества: универсальность – практически всегда можно создать дополнительный программный модуль, который будет обращаться в обе системы, к тому же разными способами (например, в одну через БД, а в другую через RPC).

Недостатки: сложность, трудоемкость, а следовательно, высокая стоимость разработки, внедрения и владения. Инструментами на данном уровне являются технологии CORBA, COM+, DCOM, RPC.

Зависимость степени автоматизации и метода создания связи

Dependence of a degree of automation on a connection making method

Метод создания связи	Степень автоматизации		
	Ручная	Автоматизированная	Автоматическая
Уровень брокеров	Отсутствует	Допускается	Предпочтительная
Уровень данных	Допускается	Допускается	Предпочтительная
Уровень сервисов	Допускается	Предпочтительная	Допускается
Уровень интерпретирования метаданных	Допускается	Допускается	Предпочтительная

Пример реализации конкретной архитектуры описан в работе [7]. Группой авторов предложена системная архитектура BizQuery, основанная на использовании многофункционального формата представления данных XML и языка запросов XQuery. Другим примером реализации брокера сообщений является использование web-служб для передачи данных в формате XML. В работе [8] приведено описание архитектуры взаимодействия систем документооборота через ненадежную среду передачи данных, где брокер сообщений берет на себя функции гарантированной и однократной доставки информации между системами.

Интеграция на уровне данных. На данном уровне интеграции приложения настраиваются на работу с единой или с несколькими связанными между собой БД. Цель интеграции – возможность конечных пользователей работать с едиными данными, обрабатывая их разными инструментами.

Преимущества: низкая стоимость интеграции, что при использовании одной СУБД очень заманчиво.

Недостатки: разрушение целостности данных. Если БД не экранирована хранимыми процедурами и не имеет необходимых ограничений целостности (в виде указания каскадных операций и триггеров), разные приложения могут приводить данные в противоречивые состояния. Если же БД экранирована и целостность обеспечивается, то и в этом случае в параллельно работающих с одной БД приложениях будут дублирующиеся части кода, выполняющие одинаковые или похожие операции. Кроме того, при изменениях структуры БД необходимо отдельно переписывать код всех приложений, с ней работающих. Модели интеграции на уровне данных описаны в работе [9]. Для решения проблемы авторы предлагают построить единую систему интеграции данных, основная задача которой – обеспечение доступа пользователям к данным из всех

корпоративных ИС без структурирования и накопления в единой БД.

При использовании единой объектно-ориентированной СУБД для задач интеграции возникают проблемы дублирования данных либо сложности их извлечения из иерархических структур. Для решения названных проблем предлагается использовать параллельные структуры, что позволяет извлекать экземпляры объектов, избегая избыточности данных и разных аномалий [10].

Интеграция на уровне сервисов. При интеграции на уровне сервисов чаще всего используются сервис-ориентированная архитектура (SOA) и шина данных [9]. Сервис представляет собой одну или несколько прикладных функций приложения, реализующих прикладную логику автоматизируемого процесса. Основные достоинства использования сервисов – возможность многократного применения и слабая связанность сервисов друг с другом.

Цель интеграции – быстрая отработка корпоративной бизнес-логики. Интеграция основана на фиксации интерфейсов и форматов данных с двух сторон.

Преимущества: низкая стоимость интеграции, быстрое объединение сильно различающихся систем без их модификации и дописывания дополнительных модулей.

Недостатки: присутствует фиксация; а если структуры или процессы изменяются, то образуются проблемы и узкоспециализированные, частные решения. При интеграции на уровне приложений важно применение стандартизованных компонентов.

На данном уровне возможна наименее затратная интеграция, где функции передачи данных будут выполнять человек, используя унифицированные интерфейсы отображения данных [5]. Это так называемая неавтоматизированная интеграция. Однако более перспективный вариант – автоматизированная интеграция. Примером успешной реализации технологической платформы массовой интеграции слабосвязанных информационных ресурсов в единую ИС является система ZooSPACE [11]. Комплекс ZooSPACE строится на произвольном количестве слабосвязанных самодостаточных узлов, которые функционируют в соответствии с единой политикой. Взаимодействие узлов между собой осуществляется посредством сетевых протоколов прикладного уровня на основе транспортного протокола TCP/IP.

Интеграция на уровне интерпретирования метаданных. Цель интеграции – ИС хранит данные в самоописывающем хранилище, а пользователь может модифицировать не только их, но и метаданные, перенастраивая ИС под постоянно меняющиеся требования решаемой задачи [12].

Преимущества: гибкость подхода и упрощение модификации систем, снижение необходимого

уровня квалификации пользователя при внесении изменений в структуру и функции.

Недостатки: сложность создания ПО с повышенным уровнем абстракции, в реализации метамодели и в сопоставлении информационных ресурсов в разных системах метаданных [13, 14].

Примером успешной реализации самоописывающего хранилища данных является система «COBRA++» [15], которая позволяет формировать информационную модель объектов предметной, быстро перенастраивать метainформацию при интеграции данных из нового источника.

В настоящее время условия развития современного предприятия таковы, что совершенствование корпоративной информационной среды идет эволюционно, путем постепенного внедрения узкоспециализированных программных продуктов в соответствии с текущими потребностями и возможностями, и на основании этих систем строится интегрированная система управления предприятием [16, 17]. Такой подход обусловлен множеством причин, главным образом, экономическими, развитие ИС всегда отвечает текущим потребностям предприятия.

Таким образом, в статье проанализированы основные уровни интеграции гетерогенных ИС, предложена классификация уровней интеграции. Рассмотрены варианты взаимодействия ИС в зависимости от целей интеграции. Наиболее перспективным видом интеграции компонентов управляющих ИС, который требует дальнейшего изучения и совершенствования, является интеграция узкоспециализированных систем на уровне сервисов. Такой вариант интеграции позволит построить децентрализованную систему, в которую могут быть интегрированы как самодостаточные слабосвязанные компоненты, так и жестко связанные приложения, что позволит избежать дополнительных расходов на перестройку работающих ИС и существующих БД на предприятии.

Литература

1. Взаимосвязь информационных подсистем предприятия. URL: <http://www.intuit.ru/studies/courses/13862/1259/lecture/24012> (дата обращения: 27.01.2016).
2. Решетников И.С., Тупышев А.М., Владимирова М.В., Гревцев В.А. Стандарты интеграции многоуровневых информационных систем // Автоматизация в промышленности. 2009. № 9. С. 23–27.
3. Порай Д.С., Тарханов И.А. Односторонняя интеграция информационных систем // Тр. Ин-та системн. анализа РАН. 2007. Т. 29. С. 59–70.
4. Когаловский М.Р. Методы интеграции данных в информационных системах. URL: <http://www.ipr-ras.ru/articles/kogalov10-05.pdf> (дата обращения: 20.11.2015).
5. Тимакин О.А., Радзивон В. Описание интеграционных решений информационной системы и особенности ее использования // Евразийский научн. журн. 2015. № 12. URL: <http://journalpro.ru/articles/opisanie-integratsionnykh-resheniy-informatsionnoy-sistemy-i-osobennosti-ee-ispolzovaniya/> (дата обращения: 27.01.2016).
6. Шибанов С.В., Яровая М.В., Шашков Б.Д. Обзор современных методов интеграции данных в информационных систе-

мах // Надежность и качество: тр. Междунар. симпоз. 2010. URL: <http://cyberleninka.ru/article/n/obzor-sovremennyh-metodov-integratsii-dannyh-v-informatsionnyh-sistemah> (дата обращения: 27.01.2016).

7. Антипин К.В., Фомичев А.В., Гринев М.Н., Кузнецов С.Д., Новак Л.Г., Плещачков П.О., Рекуц М.П., Ширяев Д.Р. Оперативная интеграция данных на основе XML: системная архитектура BizQuery // Тр. Ин-та систем. программирования РАН. URL: <http://citforum.ru/internet/xml/bizquery/> (дата обращения: 26.01.2016).

8. Данилин А.В. Технология интеграции информационных систем на основе стандартов xml и web-служб (на уровне брокеров). URL: http://www.benran.ru/Magazin/cgi-bin/Sb_03/pr03.exe?118 (дата обращения: 26.01.2016).

9. Алаудинов А.Г. Построение единой системы интеграции данных в крупных корпорациях // Надежность и качество: тр. Междунар. симпоз. 2011. URL: <http://cyberleninka.ru/article/n/postroenie-edinoj-sistemy-integratsii-dannyh-v-krupnyh-korporatsiyah> (дата обращения: 26.01.2016).

10. Волков А.А., Шведенко В.Н. Модель формирования параллельных структур в объектно-ориентированных СУБД // Программные продукты и системы. 2011. № 3. С. 14–17.

11. Жижимов О.Л., Федотов А.М., Шокин Ю.И. Технологическая платформа массовой интеграции гетерогенных данных // Вестн. НГУ. Сер.: Информационные технологии. 2013. Т. 11. № 1. С. 24–41.

12. Ратманова И.Д., Павлов М.Н. Подход к организации средств интеграции данных в корпоративных информационно-аналитических системах // Информационные технологии. 2006. № 6. С. 2–11.

13. Антопольский А.Б., Ауссем В.И. Типология информационных ресурсов в стандартных системах метаданных: анализ и проблемы интеграции // Информационные ресурсы России. 2006. № 5. С. 3–6.

14. Веселова Н.С., Шведенко В.Н. Моделирование информационных ресурсов предприятия при процессной организации системы управления // Программные продукты и системы. 2014. № 4. С. 260–264.

15. Набатов Р.А., Шведенко В.Н. Технология быстрой разработки баз данных и приложений пользователя в системе «COBRA++» // Программные продукты и системы. 2008. № 2 (82). С. 39–41.

16. Кусов А.А. Проблемы интеграции корпоративных информационных систем // Управление экономическими системами: электронный научный журнал. 2011. № 28. С. 103–109. URL: <http://uecs.ru/marketing/item/411-2011-04-25-10-08-37> (дата обращения: 25.12.2015).

17. Сысолетин Е.Г., Аксенов К.А., Круглов А.В. Интеграция гетерогенных информационных систем современного промышленного предприятия // Современные проблемы науки и образования. 2015. № 1. URL: <http://www.science-education.ru/ru/article/view?id=19030> (дата обращения: 25.12.2015).

Software & Systems

DOI: 10.15827/0236-235X.116.073-077

Received 03.03.16

2016, vol. 29, no. 4, pp. 73–77

ANALYSIS OF INTEGRATION LEVELS OF HETEROGENEOUS INFORMATION SYSTEM COMPONENTS

O.V. Schekochikhin¹, Ph.D. (Engineering), Associate Professor, slim700@yandex.ru

P.V. Shvedenko², Graduate Student, pitk1@mail.ru

¹ Kostroma State Technological University, Dzerzhinsky St. 17, Kostroma, 156005, Russian Federation

² The National Research University of Information Technologies, Mechanics and Optics, Kronverksky Ave. 49, St. Petersburg, 197101, Russian Federation

Abstract. Integration uncoordinated enterprise information systems is by far the most efficient way to build a unified information environment in the enterprise. To implement such a paradigm shift we need an integrating tool class. This paper analyzes the basic levels of heterogeneous information systems integration. There is an author's classification of integration levels, as well as the variants of information systems interaction depending on integration goals. The authors analyze the integration options at the broker level, the data level, the service level and at the level of meta information interpretation. The proposed classification is an attempt to highlight the key information systems integration options in terms of practical implementation. The paper shows the examples of successful implementation of integration tasks on each of the proposed levels. It also considers the strengths and weaknesses of each integration option. The broker level is useful in the case when a source code of integrated applications is available and there is an opportunity to add a required integration module. Integration at the data level can be recommended for information systems and data circuits that allow us to construct a single consistent data structure. The main advantages of the service level is the ability to reuse them, loose coupling of services, a lack of necessity in a single database or repository. When working at the level of meta-information interpretation it is possible to quickly reconfigure the meta-information in data integration from a new source. The classification makes it possible to determine the most rational set of tools and methods to create an integrated system with specified characteristics. The authors believe that the most promising way of integration of control information systems components, which requires further study and improvement, is the integration of specialized systems at the service level.

Keywords: integration, information system, integration levels.

References

1. *Vzaimosvyaz informatsionnykh podsistem predpriyatiya* [Information Subsystems Interrelation in an Enterprise]. Available at: <http://www.intuit.ru/studies/courses/13862/1259/lecture/24012> (accessed January 27, 2016).
2. Reshetnikov I.S., Tupyshev A.M., Vladimirova M.V., Grevtsev V.A. Integration standards of multilevel information systems. *Avtomatizatsiya v promyshlennosti* [Automation in Industry]. 2009, no. 9, pp. 23–27 (in Russ.).
3. Poray D.S., Tarkhanov I.A. One-sided integration of information systems. *Trudy ISA RAN* [Proc. of ISA RAS]. 2007, vol. 29, pp. 59–70 (in Russ.).
4. Kogalovsky M.R. *Metody integratsii dannykh v informatsionnykh sistemakh* [Data Integration Methods in Information Systems]. Available at: <http://www.ipr-ras.ru/articles/kogalov10-05.pdf> (accessed November 11, 2015).

5. Timakin O.A., Radzivon V. A description of information system integration solutions and its usage features. *Evrasyyskiy nauchnyy zhurnal* [Eurasian Scientific Jour.]. 2015, no. 12. Available at: <http://journalpro.ru/articles/opisanie-integratsionnykh-resheniy-informatsionnoy-sistemy-i-osobennosti-ee-ispolzovaniya/> (accessed January 27, 2016).
6. Shibanov S.V., Yarovaya M.V., Shashkov B.D. Overview of data integration modern methods in information systems. *Trudy Mezhdunar. simp. "Nadezhnost i kachestvo"* [Proc. Int. Symp. "Reliability and Quality"]. 2010, vol. 1. Available at: <http://cyberleninka.ru/article/n/obzor-sovremennykh-metodov-integratsii-dannykh-v-informatsionnykh-sistemah> (accessed January 27, 2016).
7. Antipin K.V., Fomichev A.V., Grinev M.N., Kuznetsov S.D., Novak L.G., Pleshchikov P.O., Rekuts M.P., Shiryayev D.R. Operational data integration based on XML: BizQuery system architecture. *Trudy ISA RAN* [Proc. of ISA RAS]. Available at: <http://citforum.ru/internet/xml/bizquery/> (accessed January 26, 2016).
8. Danilin A.V. *Tekhnologiya integratsii informatsionnykh sistem na osnove standartov XML i web-sluzhb (na urovne brokerov)* [Technology of information systems integration based on standards XML and Web-services (at brokers' level)]. Available at: http://www.benran.ru/Magazin/cgi-bin/Sb_03/pr03.exe?118 (accessed January 26, 2016).
9. Alaudinov A.G. Creating data integration unified system in big corporations. *Trudy Mezhdunar. simp. "Nadezhnost i kachestvo"* [Proc. Int. Symp. "Reliability and Quality"]. 2011, vol. 1. Available at: <http://cyberleninka.ru/article/n/postroenie-edinoy-sistemy-integratsii-dannykh-v-krupnykh-korporatsiyah> (accessed January 26, 2016).
10. Volkov A.A., Shvedenko V.N. Model of forming parallel structures in object-oriented DBMS. *Programmnye produkty i sistemy* [Software & Systems]. 2011, no. 3, pp. 14–17.
11. Zhizhimov O.L., Fedotov A.M., Shokin Yu.I. Technological platform of heterogeneous data mass integration. *Vestnik NGU. Ser.: Informatsionnye tekhnologii* [Novosibirsk State University Journal of Information Technologies]. 2013, vol. 11, no. 1, pp. 24–41 (in Russ.).
12. Ratmanova I.D., Pavlov M.N. Approach to organization of data integration tools in corporative information and analytical systems. *Informatsionnye tekhnologii* [Information Technologies]. 2006, no. 6, pp. 2–11 (in Russ.).
13. Antopolsky A.B., Aussem V.I. Information recourse typology in standard metadata systems: analysis and integration problems. *Informatsionnye resursy Rossii* [Information Recourses of Russia]. 2006, no. 5, pp. 3–6 (in Russ.).
14. Veselova N.S., Shvedenko V.N. Enterprise information resources modeling for process organization of enterprise management system. *Programmnye produkty i sistemy* [Software & Systems]. 2014, no. 4 (108), pp. 260–264 (in Russ.).
15. Nabatov R.A., Shvedenko V.N. A fast development technology for databases and user applications in "COBRA++". *Programmnye produkty i sistemy* [Software & Systems]. 2008, no. 2 (82), pp. 39–41 (in Russ.).
16. Kusov A.A. Integration problems in corporative information systems. *Upravlenie ekonomicheskimi sistemami: elektronnyy nauch. zhurnal* [Management of Economic Systems]. 2011, no. 28, pp. 103–109. Available at: <http://uecs.ru/market/ing/item/411-2011-04-25-10-08-37> (accessed December 25, 2016).
17. Sysoletin E.G., Aksenov K.A., Kruglov A.V. Heterogeneous information systems integration in modern industrial organization. *Sovremennyye problemy nauki i obrazovaniya* [Modern problems of science and education]. 2015, no. 1 (in Russ.).

Примеры библиографического описания статьи

1. Щекочихин О.В., Шведенко П.В. Анализ уровней интеграции компонентов гетерогенных информационных систем // Программные продукты и системы. 2016. Т. 29. № 4. С. 73–77; DOI: 10.15827/0236-235X.116.073-077.
2. Schekochikhin O.V., Shvedenko P.V. Analysis of integration levels of heterogeneous information system components. *Programmnye produkty i sistemy* [Software & Systems]. 2016, vol. 29, no. 4, pp. 73–77 (in Russ.); DOI: 10.15827/0236-235X.116.073-077.

УДК 004.25

DOI: 10.15827/0236-235X.116.078-084

Дата подачи статьи: 16.03.16

2016. Т. 29. № 4. С. 78–84

ИЗМЕРЕНИЕ ПРОИЗВОДИТЕЛЬНОСТИ КОМПОНЕНТОВ ПОДСИСТЕМЫ ПАМЯТИ ДЛЯ ГЕТЕРОГЕННЫХ СИСТЕМ НА КРИСТАЛЛЕ

А.О. Ключев, к.т.н., доцент, kluchev@cs.ifmo.ru;

А.А. Антонов, аспирант, 153287@niuitmo.ru

*(Санкт-Петербургский национальный исследовательский университет
информационных технологий, механики и оптики (Университет ИТМО),
Кронверкский просп., 49, г. Санкт-Петербург, 197101, Россия)*

Проектирование специализированных вычислительных систем на базе гетерогенных платформ в рамках современных методологий предусматривает наличие модели системной архитектуры с информацией о характеристиках входящих в ее состав компонент. Производительность подсистемы памяти как ключевого связующего элемента в современных архитектурах является одной из важнейших характеристик, определяющих общесистемную производительность. Тенденция к интегрированию множества гетерогенных компонент в составе систем и сетей на кристалле, в том числе на уровне иерархии кэш-памяти, вносит проблемы при определении параметров реальных вычислительных платформ в силу того, что внутрисистемные тракты обмена оказываются технологически недоступными для прямых измерений, а общедоступная документация, как правило, лишь фрагментарно описывает внутреннюю организацию системы. Существующие методы непрямого измерения производительности компонентов кэша не гарантируют соответствие получаемой модели кэш-памяти реальному поведению исследуемой системы.

В статье предложен метод непрямого селективного измерения производительности отдельных компонент кэш-подсистемы, в рамках которого предусмотрен ряд технических приемов для верификации селективности измерений с использованием информации о фактическом количестве обращений к отдельным компонентам кэш-подсистемы. Проанализирована применимость метода для нескольких популярных типов систем на кристалле и приведены результаты его апробации на процессоре Intel Core i7 и в заказной вычислительной платформе для проектирования гетерогенных измерительных систем реального времени.

Ключевые слова: *встроенные системы, системы на кристалле, гетерогенные системы, реальное время, кэш-память, системный уровень, согласованное проектирование аппаратного и программного обеспечения.*

Гетерогенные системы используются в различных областях для создания высокопроизводительных и энергоэффективных решений, таких как информационно-измерительные системы, аудио- и видеосистемы, системы управления, оборудование сетей и др. Под гетерогенными обычно понимают системы с аппаратным параллелизмом программно-видимых процессов или потоков и с различной степенью специализации вычислительных элементов: от процессоров общего и специального назначения до узкоспециализированных функциональных ядер: акселераторов вычислительных функций, контроллеров ввода/вывода и других, реализованных в виде сложнофункциональных блоков (СФ-блок или IP-ядро) на специализированной интегральной микросхеме (application specific integrated circuit, ASIC) или на ПЛИС [1]. Помимо набора вычислительных элементов, гетерогенность также проявляется в подсистемах коммуникации между ними: неоднородных сетевых структурах, а также распределенной памяти с многоуровневым кэшированием.

Моделирование интегрированных гетерогенных вычислительных платформ в проектировании

Сложность и многообразие вариантов аппаратно-программной организации таких систем, между которыми ведется поиск компромисса по их

характеристикам, вкупе с нарастающими требованиями к темпам и качеству проектирования ведут к необходимости развития и внедрения комплексных методологий системного уровня (hardware-software co-design, platform-based design, ESL-based co-design), в которых особое внимание уделяется этапам раннего архитектурного и микроархитектурного проектирования вычислительной системы без фиксации деталей аппаратно-программной реализации ее компонент и связей между ними [2]. Данные этапы предусматривают разработку функциональной модели системы («золотой модели», как правило, исполняемой) и параметрической модели вариантов системной архитектуры для оценки нефункциональных характеристик системы. Далее, в рамках так называемого исследования проектного пространства (design space exploration), производятся итеративное аппаратно-программное разделение и отображение «золотой модели» на различные варианты архитектур и оценка полученных результатов. Метрики, применяемые в модели вариантов системной архитектуры, характеризуют элементы используемой базовой аппаратно-программной платформы (СФ-блоки, коммуникации, комплексные механизмы) и включают в себя производительность процессоров, энергопотребление, емкость памяти, пропускную способность каналов связи и др. [1]. Пример инструмента, реализующего вышеупомянутый подход, – SystemCo-Designer [3].

Растущие требования как к разнообразию функций, реализуемых вычислительными системами на сегодняшний день, так и к их нефункциональным характеристикам обуславливают тенденцию к применению вычислительных платформ на базе единой СБИС (технологий «система на кристалле» (СнК) и «сеть на кристалле» [4]), в рамках которой интегрировано множество компонентов повторного использования (процессоров общего назначения, видеопроцессоров, DSP и прочих блоков), называемых IP-ядрами, на этапе проектирования. Количество IP-ядер, интегрируемых в рамках СнК, неуклонно растет, что вкупе с фундаментальными ограничениями по целостности и задержке распространения сигналов ведет к усложнению организации коммуникационной подсистемы и, соответственно, к усложнению ее проектирования и оценке ее реальных характеристик [5]. С учетом того, что на сегодняшний день архитектурно обмен через общую память доминирует в роли коммуникации между процессорными элементами, измерение производительности внутрисистемных трактов обмена с подсистемой памяти становится неотъемлемой частью моделирования вычислительных платформ в составе маршрутов проектирования на их основе.

Проблемы измерения производительности подсистемы памяти в интегрированных вычислительных платформах

Существующая в современных архитектурах тенденция к интегрированию множества различных специализированных компонент в рамках единой СнК усугубляет проблемы как технологического, так и методологического плана. Блоки в составе СнК зачастую не только физически реализуются на одном кристалле, но и интегрируются на уровне иерархии памяти, в результате чего появляется возможность ускоренного обмена данными между ними. Интегрированные подсистемы памяти для гетерогенных архитектур на сегодняшний день являются актуальной темой исследований (например, кэш-когерентные подсистемы в [6]) и интенсивно внедряются в промышленных изделиях. В качестве примера можно назвать технологию так называемой гетерогенной системной архитектуры (heterogeneous systems architecture, HSA) для СнК [7], развиваемую компаниями AMD, ARM, Imagination, Qualcomm и др., в которой декларируется возможность унифицированного доступа к памяти со стороны различных компонентов в составе гетерогенной архитектуры. Примерами реальных СнК, в составе которых на уровне иерархии памяти интегрированы процессорные элементы различной степени специализации, являются СнК Intel Core i7 4XXX (Haswell) и 5XXX (Broadwell), AMD Carrizo, Xilinx Zynq-7000 и UltraScale, TI OMAP и DaVinci. В качестве примера

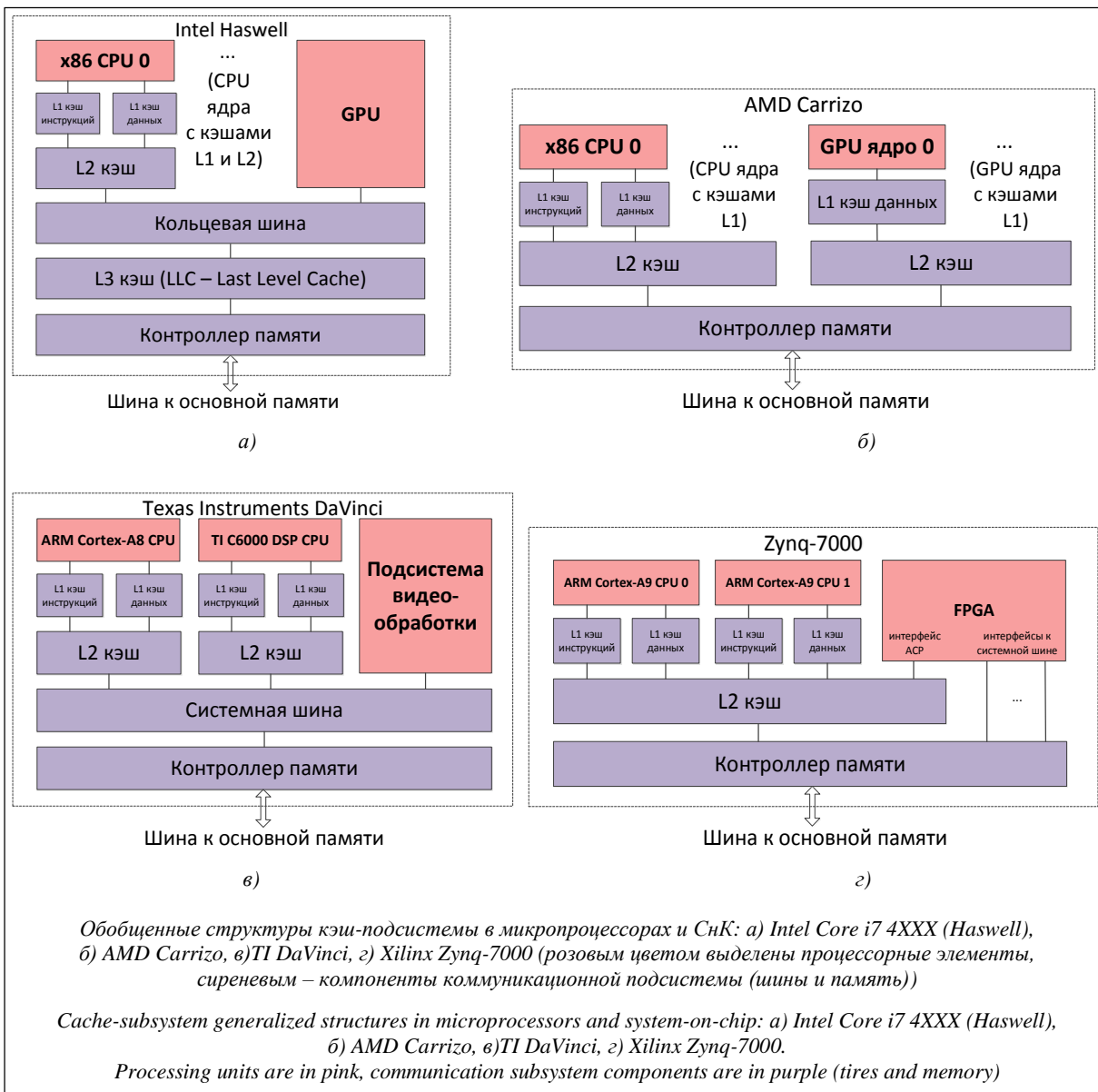
на рисунке приведены обобщенные структуры кэш-подсистем для некоторых из упомянутых платформ.

С точки зрения использования вышеупомянутых платформ в проектировании, внутрикристалльная реализация коммуникаций между блоками в составе гетерогенной архитектуры обуславливает фактическое отсутствие физического доступа к ним и способствует тому, что системный проектировщик крайне ограниченно осведомлен о деталях организации этих коммуникаций. Таким образом, необходимость работы с платформой на уровне отдельных вычислительных компонент и связей между ними вступает в противоречие с тем, что структура кэш-подсистемы, топология коммуникационной сети, протоколы обмена и так далее, как правило, относятся к микроархитектуре системы, а детали микроархитектурной реализации реальных коммерческих СБИС обычно являются коммерческой тайной соответствующих фирм-производителей и крайне поверхностно или фрагментарно описаны в документации, находящейся в открытом доступе. Приводимые оценки обычно являются оптимистичными и редко наблюдаются в реальности. Задача моделирования интегрированных вычислительных платформ для последующего отображения функциональной модели системы или оценки (верификации) показателей для разработанных систем на реальных образцах делает актуальным развитие методов и технологий селективного непрямого измерения производительности внутрисистемных трактов обмена.

Пример такого исследования – работа [8], где с использованием СнК Xilinx Zynq-7000, имеющей интерфейс для подключения к иерархии кэш-памяти со стороны блока программируемой логики, измерялась производительность данного интерфейса на примере задачи КИХ-фильтрации образа сигнала. Однако, к сожалению, большинство подобных исследований так или иначе привязаны к какой-либо платформе или приложению, и проблема характеристики внутрисистемных каналов связи в целом и особенно иерархии кэш-памяти остается на данный момент открытой.

Существующие методы и технологии не прямых измерений на уровне компонентов кэш-подсистем

В то время как оценке производительности вычислительных систем на примере тех или иных приложений посвящено значительное количество работ отечественных и зарубежных авторов, «погружение» в детали микроархитектуры реальных систем, организация измерений применительно к отдельным подсистемам и трактам обмена, что необходимо для планирования вычислений в рамках гетерогенных архитектур, является существенно более сложной и изощренной задачей.



К тому же эта область становится все более нетривиальной в силу развития и, соответственно, усложнения организации исследуемых систем. Однако можно назвать определенное количество разработок в данной области.

Большинство алгоритмов анализа характеристик компонентов иерархии кэш-памяти основано на наборе классических приемов, которые можно найти, например, в [9]. Измерение заключается в оценке времени выполнения специальной процедуры, осуществляющей обращения к памяти по определенным шаблонам. Рабочей процедурой является циклическое обращение к элементам массива, размещенного в памяти. Размер массива варьируется от минимального до максимального размера кэша (обычно удвоением). Обращение производится с прореживанием, которое позволяет обеспечить промахи по кэшу и также варьируется от нуля до максимального размера кэша. Повтор-

ные обращения к тем же адресам позволяют приблизительно оценить накладные расходы на измерения. В то же время в части интерпретации результатов и реальной микроархитектурной селективности измерений есть ряд проблем, в число которых входит необходимость вычисления размеров массива и адресов, возможность аппаратной параллелизации запросов, сложность последовательного размещения в силу работы механизма виртуальной памяти и так далее.

В рамках исследований в области адаптивного ПО в Корнелльском университете был разработан фреймворк X-Ray [10–12]. Он содержит набор измерительных средств (бенчмарков) микроархитектурного уровня (микробенчмарков) для оценки различных аспектов системной производительности. Посредством варьирования размера рабочего массива и степени прореживания производится измерение размера кэша, размера строки кэша, ассоци-

ативности и задержки. Последовательное размещение рабочего массива в памяти достигается через запрос больших страниц, специфический для операционной системы. Однако соответствие используемой модели реальной системе (то есть тех механизмов, параметры которых определяются, и уже реализованных в системе) остается проблемой.

Фреймворк LMBench [13] также предоставляет множество инструментов для измерения системной производительности с различных точек зрения – от производительности системных вызовов ОС до измерений микроархитектурного уровня. Например, имеется микробенчмарк, позволяющий произвести измерения размера кэша, размера строки кэша, задержки и ассоциативности для одного уровня кэша. Компания Intel рекомендует использовать LMBench для исследования иерархии памяти [14]. На графике зависимости пропускной способности и задержки от размера рабочего массива можно увидеть «ступеньки», которые соответствуют определенным уровням иерархии памяти, и таким образом оценить целевые характеристики.

Исследовательская группа Технического университета Дрездена проводила измерения производительности компонентов иерархии памяти для микропроцессоров Intel Nehalem, Sandy Bridge и AMD Bulldozer [15, 16]. Измерения выполнялись в пакете BenchIT. В системе инициализировались несколько потоков, которые отображались на различные процессорные ядра. Посредством определенной последовательности запросов кэш-память вводилась в начальное состояние. Далее с помощью встроенного в устройство таймера была измерена задержка кэш-памяти различных уровней и основной памяти. «Потолок» пропускной способности достигался посредством запуска нескольких потоков, которые обращались к одному и тому же компоненту памяти.

Более точный подход был предложен в [17]. Метод ориентирован на измерение кратковременных событий: например, увеличение задержки при наполнении регистров MSHR (miss information/status handling registers). Поскольку такие события нельзя измерить, усредняя результаты длительных измерений, авторы предлагают так называемую методологию SETE, которая предусматривает, например, использование встроенного высокоточного таймера и «разогрев» кэша инструкции путем выполнения нескольких запусков эксперимента с учетом только последнего результата.

В целом в существующих методах и технологиях измерений производительности компонентов кэш-подсистемы можно выделить ряд открытых проблем, связанных с возможным искажением результатов измерений. При подходе с постепенным увеличением размера рабочего массива и анализа «ступенек» на графике времени выполнения теста по факту обращения производятся к различным

уровням кэш-памяти, что затрудняет выделение случаев обращения именно к интересующему уровню кэш-памяти. Если структура теста ориентирована на определенный вариант организации кэш-памяти, без достоверной информации о том, что в конкретной системе реализован именно этот вариант, реальная селективность измерений также остается под вопросом, что допускает неверную интерпретацию результатов измерений.

Метод селективного измерения производительности компонентов кэш-памяти

Для решения проблемы обеспечения реальной селективности измерений авторами был разработан оригинальный метод селективного измерения производительности компонентов кэш-памяти. В целом данный метод основан на работе [9] с заимствованием следующих технических приемов.

Тесты представляют собой зацикленные обращения к памяти без вычислений между итерациями. Доступ к конкретному уровню кэша достигается посредством параметризации размера рабочего массива (он должен быть меньше, чем размер целевого кэша, но больше, чем кэш меньшего уровня). При первичной инициализации массива данные размещаются в кэше нужного уровня, в то время как последующие обращения формируют рабочую процедуру.

Требования к значениям элементов массива различаются в зависимости от измеряемой характеристики. Тесты пропускной способности не предъявляют каких-либо специфических требований. Для измерения задержки требуется, чтобы каждый элемент содержал адрес последующего элемента, что позволяет убрать влияние параллелизации и конвейеризации доступа к памяти. В литературе данный подход носит название «погона за указателем» (pointer chasing) [15]. Для недопущения попадания в кэши более низкого уровня применяется прореживание доступа.

Однако ввиду сложности механизмов управления кэш-памятью на сегодняшний день гарантировать обращения к требуемому уровню кэша без дополнительных проверок не представляется возможным. Другими словами, должно подтверждаться соответствие используемой модели иерархии памяти и ее реальной организации в системе. Поэтому с помощью встроенных средств мониторинга (так называемых счетчиков производительности) осуществляется проверка соответствия реального поведения подсистемы памяти ожидаемому (то есть специфицированному в рамках принятой модели подсистемы памяти).

Работа тестовой процедуры сопровождается подсчетом обращений к компонентам подсистемы памяти с помощью встроенных средств мониторинга. Запуск и останов счетчиков жестко синхронизируются с основной рабочей процедурой.

Встроенные счетчики обращений к компонентам кэш-памяти в различных вычислительных платформах**Built-in counters of references to memory cache components in different computing platforms**

Вычислительная платформа	Количество уровней кэш-памяти	Наличие встроенного блока измерения производительности	Количество счетчиков производительности	Доступные события кэш-подсистемы, актуальные для применения метода
Intel Haswell [18]	3	+	3 (назначение фиксировано) + 8 (программируемых)	Количество попаданий и промахов по кэшам L1, L2, L3
AMD Carrizo [19]	2	+	6 (на ядро) + 4 (сверный мост)	Количество попаданий в кэши L1 и L2
Texas Instruments DaVinci [20]	2	+	4	Количество обращений и промахов по кэшам L1 и L2
Xilinx Zynq-7000 [21]	2	+	6	Количество обращений и промахов по кэшу L1 инструкций и данных

Варьирование параметров производится до тех пор, пока не будут достигнуты целевые значения. В качестве начальных параметров следует брать размер рабочего массива в $\frac{1}{2}$ от исследуемого компонента кэш-памяти и шаг в пределах строки кэша для кэш-памяти 1-го уровня и, соответственно, кратный размеру строки для кэш-памяти последующих уровней. Варьирование следует производить согласно следующим правилам:

- если результаты опыта показали, что попаданий в кэш нужного уровня меньше ожидаемых за счет попаданий в кэш более низкого уровня, значит, сработала загрузка данных в кэш более низкого уровня, и поэтому следует увеличить степень прореживания обращений; если прореживание обращений достигает размера кэша более низкого уровня, это, скорее всего, означает, что аппаратное обеспечение детектировало шаблон обращений к памяти, следовательно, величину прореживания надо изменить;

- если результаты опыта показали, что попаданий в кэш нужного уровня меньше ожидаемых за счет попаданий в кэш более высокого уровня, значит, часть рабочего массива была выгружена из кэша нужного уровня с целью освобождения места для новых потенциально нужных данных, поэтому следует уменьшить размер рабочего массива с целью минимизации вероятности выгрузки данных в составе его в кэш более высокого уровня.

В результате, итеративно повторяя эксперименты в соответствии с приведенными правилами и используя информацию встроенных блоков измерения производительности, можно определить выборочную пропускную способность и задержку отдельных компонентов кэш-подсистемы в составе иерархии памяти целевой вычислительной платформы. Использование предложенных приемов позволяет верифицировать действительное количество обращений к компонентам иерархии памяти и за счет итеративных повторов измерений добиться реальной селективности измерений.

Применимость метода селективного измерения производительности компонентов кэш-памяти

Требования к наличию средств встроенного мониторинга производительности кэш-подсистемы является фактором, ограничивающим применимость представленного метода только теми системами, где данные средства реализованы. Исходя из содержания метода, для его применимости требуются как минимум счетчик циклов (таймер) и счетчик, программируемый на успешные обращения на каждый из интересующих компонентов кэш-подсистемы. Данные о доступности встроенных средств мониторинга производительности для рассмотренных ранее вычислительных платформ приведены в таблице.

Несмотря на то, что количество счетчиков и наличие тех или иных событий являются специфическими для платформы, можно видеть, что большинство современных вычислительных платформ соответствуют данным требованиям. Исключением в приведенных примерах является Xilinx Zynq-7000, для которого блок мониторинга производительности предусматривает подсчет только тех 58 событий, которые относятся непосредственно к ядру Cortex-A9 [21]. Поскольку кэш L2 для Cortex-A9 является внешним по отношению к ядру, отличить попадание в L2 от промаха, видимо, возможно только для подконтрольных трактов данных (например, при обращении к области памяти, отображенной на встроенную FPGA).

В рамках апробации метода был разработан пакет программных тестов для микропроцессора Intel Core i7 4770K [22]. Тесты были реализованы в виде модуля для загрузчика GNU GRUB 2.0 в среде без операционной системы, что нивелировало влияние ее эффектов на измерения. С помощью предложенного метода были проведены измерения с достижением соответствия общего количества запросов к

памяти и действительного количества обращений к каждому из исследуемых уровней кэш-подсистемы и коэффициентом вариации в пределах 10 %.

Также авторы использовали описанный метод в реальном проектировании на базе разработанной авторами заказной гетерогенной платформы для систем цифровой обработки сигналов [23]. В рамках данной платформы была реализована система обработки радиосигналов реального времени на процессорах ЦОС, ПЛИС и специализированных интегральных микросхемах. Предложенный метод позволил верифицировать селективную производительность внутрисистемных трактов обмена и осуществить прогнозирование предельной длительности циклов, необходимой для обработки отсчетов входного сигнала.

Таким образом, в соответствии с перспективными методологиями проектирования специализированных гетерогенных вычислительных систем необходимость моделирования системной платформы делает актуальной задачу селективного измерения пропускной способности и задержки коммуникационных компонентов между процессорными элементами в составе платформы. Тенденция к интеграции разнородных процессоров в рамках единой СБИС и, соответственно, единой иерархии памяти и внутрикристалльных коммуникаций вносит проблемы при осуществлении таких измерений. Для решения данной проблемы в статье предлагается оригинальный метод селективного измерения производительности компонентов кэш-памяти. Метод позволяет производить измерения пропускной способности и задержки отдельных уровней кэш-памяти с контролем селективности измерений с помощью данных о фактическом количестве обращений к отдельным блокам иерархии памяти. Предложенный метод был апробирован в рамках исследования кэш-подсистем микропроцессора Intel Core i7 и специализированной гетерогенной вычислительной платформы реального времени. Разработанный пакет тестов может использоваться для проведения измерений производительности компонентов кэш-подсистем для актуальных гетерогенных вычислительных платформ.

Литература

1. Pomante L., Serri P., Incerto E., Volpe J. HW/SW co-design of heterogeneous multiprocessor dedicated systems: a systemc-based environment. Proc. 2nd World Congr. Multimed. Comput. Sci. 2014, pp. 9–11.
2. Teich J. Hardware/software codesign: the past, the present, and predicting the future. Proc. IEEE. 2012, vol. 100, pp. 1411–1430.
3. Keinert J., Streubühr M., Schlichter T., Falk J., Gladi-gau J., Haubelt C., Teich J., and Meredith M. SystemCoDesigner – an automatic ESL synthesis approach by design space exploration and behavioral synthesis for streaming applications. ACM Trans. Des. Autom. Electron. Syst. 2009, vol. 14, no. 1, article 1, pp. 1–23.
4. Исаев М.В., Кожин А.С., Костенко В.О., Поляков Н.Ю., Сахин Ю.Х. Двухъядерная гетерогенная система на кристалле

«Эльбрус-2С+» // Вопросы радиоэлектроники. 2012. Т. 4. № 3. С. 42–52.

5. Ahmad B. Communication centric platforms for future high data intensive applications. PhD Thesis, Univ. of Edinburgh, 2009, 149 p.
6. Hechtman B.A., Sorin D.J. Evaluating cache coherent shared virtual memory for heterogeneous multicore chips. Proc. IEEE Int. Symp. Perform. Anal. Syst. Softw., 2013, pp. 118–119.
7. HSA Foundation. URL: <http://www.hsafoundation.com/> (дата обращения: 15.03.2016).
8. Sadri M., Weis C., Wehn N., and Benini L. Energy and Performance Exploration of Accelerator Coherency Port Using Xilinx ZYNQ. Proc. 10th FPGAWorld Conf. FPGAWorld'13. 2013, no. 5, 5 p. URL: http://www.googoolia.com/downloads/papers/sadri_fpgaworld_ver2.pdf (дата обращения: 15.03.2016).
9. Hennessy L.J., Patterson A.D. Computer architecture: a quantitative approach. Morgan Kaufmann Publ., 2003, 1141 p.
10. Yotov K. On the role of search in generating high performance BLAS libraries. PhD Thesis, Cornell Univ., 2006, 194 p.
11. Yotov K., Pingali K., Stodghill P. Automatic measurement of memory hierarchy parameters. Proc. 2005 ACM SIGMETRICS Int. Conf. Meas. Model. Comput. Syst. SIGMETRICS'05, NY, USA, ACM Press, 2005, 181 p.
12. Yotov K., Pingali K., Stodghill P. Automatic measurement of hardware parameters for embedded processors. 2005. URL: <http://ecommons.library.cornell.edu/bitstream/1813/5674/1/TR2005-1974.pdf> (дата обращения: 15.03.2016).
13. LMBench – Tools for Performance Analysis. URL: <http://www.bitmover.com/lmbench/lmbench.html> (дата обращения: 15.03.2016).
14. Measuring cache and memory latency and cpu to memory bandwidth for use with intel architecture. White Paper (Intel). 2008, pp. 1–14.
15. Molka D., Hackenberg D., Schone R., and Muller M.S. Memory performance and cache coherency effects on an intel nehalem multiprocessor system. Proc. 18th Int. Conf. Parallel Archit. Compil. Tech. IEEE, 2009, pp. 261–270.
16. Molka D., Sch R. Main memory and cache performance of intel sandy bridge and AMD Bulldozer Memory size. MSPC'2014. URL: https://tu-dresden.de/die_tu_dresden/zentrale_einrichtungen/zih/forschung/projekte/benchit/2014_MSPC_authors_version.pdf (дата обращения: 15.03.2016).
17. Fang Z., Mehta S., Yew P.-C., Zhai A., Greensky J., Gautham B., Binyu Z. Measuring microarchitectural details of multi- and many-core memory systems through microbenchmarking. ACM TACO, 2014, vol. 11, no. 4, article 55, pp. 1–26.
18. Intel® 64 and IA-32 Architectures Software Developer's Manual. Combined volumes. June 2015. URL: <http://www.intel.de/content/www/de/de/architecture-and-technology/64-ia-32-architectures-software-developer-manual-325462.html> (дата обращения: 15.03.2016).
19. BIOS and Kernel Developer's Guide (BKDG) for AMD Family 15h Models 60h-6Fh Processors. AMD. June 2015. URL: http://support.amd.com/TechDocs/50742_15h_Models_60h-6Fh_BKDG.pdf (дата обращения: 15.03.2016).
20. CortexTM-A8 Technical Reference Manual. Revision: r3p2. ARM. URL: <https://static.docs.arm.com/ddi0344/k/DDI0344.pdf> (дата обращения: 15.03.2016).
21. CortexTM-A9 Technical Reference Manual. Revision: r2p2. ARM. URL: http://infocenter.arm.com/help/topic/com.arm.doc/ddi0409f/DDI0409F_cortex_a9_neon_mpe_r2p2_trm.pdf (дата обращения: 15.03.2016).
22. Антонов А.А. Организация измерений производительности компонентов подсистемы памяти микропроцессоров и СМК. Магистер. дис. СПб: Изд-во Ун-та ИТМО, 2014. С. 1–95.
23. Kustarev P., Antonov A., Pinkevich V., Yanalov R. Using selective memory performance evaluation for time-critical embedded systems design. Proc. 15th Int. Multidiscip. Sci. Geoconf. SGEM-2015. URL: <http://www.sgem.org/SGEMLIB/spip.php?article5637> (дата обращения: 15.03.2016).

MEMORY SUBSYSTEM COMPONENTS PERFORMANCE EVALUATION FOR HETEROGENEOUS SYSTEMS-ON-CHIP

A.O. Klyuchev¹, Ph.D. (Engineering), Associate Professor, kluchev@cs.ifmo.ru; A.A. Antonov¹, Postgraduate Student, 153287@niuitmo.ru

¹ The National Research University of Information Technologies, Mechanics and Optics, Kronverksky Ave. 49, St. Petersburg, 197101, Russian Federation

Abstract. Specialized computer system design based on heterogeneous platforms within modern methodologies assumes a system platform model that contains information about the characteristics of its individual components. A memory subsystem is considered as a key communication element of current architectures, so its performance evaluation is one of the most important characteristics that determine the overall system performance. The tendency of multiple heterogeneous components integration within the systems and networks on-chip (including the cache memory hierarchy level) introduces new problems for computational platforms modeling, since in-system datapaths that connect the computational elements with cache memory hierarchy components become technologically inaccessible for direct measurements, while the publicly available documentation typically poorly describes the system's internal organization. Existing methods of indirect evaluation of cache subsystem components' performance do not guarantee the compliance of cache memory model being implied with the real behavior of the system under exploration.

The article proposes the method of indirect selective performance evaluation of individual cache subsystem components. The method provides a set of techniques for verification of evaluation selectiveness using information about the actual amount of requests to the individual cache subsystem components. The paper explores feasibility of the method using several actual instances of heterogeneous computer systems and provides the results of its approbation for Intel Core i7 processor and custom computational platform for real-time heterogeneous measurement systems design.

Keywords: embedded systems, systems-on-chip, heterogeneous systems, real-time, cache memory, Electronic System Level (ESL), HW/SW Co-Design.

References

1. Pomante L., Serri P., Incerto E. HW/SW Co-Design of Heterogeneous Multiprocessor Dedicated Systems: a SystemC-based Environment. *Proc. 2nd World Congr. Multimed. Comput. Sci.* 2014, pp. 9–11.
2. Teich J. Hardware/Software Codesign: The Past, the Present, and Predicting the Future. *Proc. IEEE*. 2012, vol. 100, Special Centennial iss., pp. 1411–1430.
3. Keinert J., Streubühr M., Schlichter T., Falk J., Gladigau J., Haubelt C., Teich J., Meredith M. SystemCoDesigner – an automatic ESL synthesis approach by design space exploration and behavioral synthesis for streaming applications. *ACM Trans. Des. Autom. Electron. Syst.* 2009, vol. 14, no. 1, pp. 1–23.
4. Isaev M.V., Kozhin A.S., Kostenko V.O., Polyakov N.Yu., Sakhin Yu.Kh. Dual-Core Heterogeneous System-on-Chip “Elbrus-2S+”. *Problems of Advanced Micro- and Nanoelectronic Systems Development (MES-2012)*. IDPM RAS Publ., 2012, vol. 4, no. 3, pp. 42–52 (in Russ.).
5. Ahmad B. *Communication Centric Platforms for Future High Data Intensive Applications*. PhD Thesis, Univ. of Edinburgh Publ., 2009, 149 p.
6. Hechtman B.A., Sorin D.J. Evaluating Cache Coherent Shared Virtual Memory for Heterogeneous Multicore Chips. *Proc. IEEE Int. Symp. Perform. Anal. Syst. Softw.*, 2013, pp. 181–119.
7. HSA Foundation. Available at: <http://www.hsafoundation.com/> (accessed March 15, 2016).
8. Sadri M., Weis C., Wehn N., Benini L. Energy and Performance Exploration of Accelerator Coherency Port Using Xilinx ZYNQ. *Proc. 10th FPGAWorld Conf. FPGAWorld '13*. 2013, no. 5, 5 p. Available at: http://www.googoolia.com/downloads/papers/sadri_fpga_world_ver2.pdf (accessed March 15, 2016).
9. Hennessy L.J., Patterson A.D. *Computer Architecture: A Quantitative Approach*. Morgan Kaufmann Publ., 2003, 1141 p.
10. Yotov K. *On the Role of Search in Generating High Performance BLAS Libraries*. PhD Thesis, Cornell Univ., 2006, 194 p.
11. Yotov K., Pingali K., Stodghill P. Automatic measurement of memory hierarchy parameters. *Proc. 2005 ACM SIGMETRICS Int. Conf. Meas. Model. Comput. Syst. SIGMETRICS '05*. NY, USA, ACM Press, 2005, 181 p.
12. Yotov K., Pingali K., Stodghill P. *Automatic Measurement of Hardware Parameters for Embedded Processors*. 2005. Available at: <http://ecommons.library.cornell.edu/bitstream/1813/5674/1/TR2005-1974.pdf> (accessed March 15, 2016).
13. *LMbench – Tools for Performance Analysis*. Available at: <http://www.bitmover.com/lmbench/lmbench.html> (accessed March 15, 2016).
14. Measuring Cache and Memory Latency and CPU to Memory Bandwidth for Use with Intel Architecture. *White Paper (Intel)*. 2008, pp. 1–14.
15. Molka D., Hackenberg D., Schone R., Muller M.S. Memory Performance and Cache Coherency Effects on an Intel Nehalem Multiprocessor System. *Proc. 18th Int. Conf. Parallel Archit. Compil. Tech. IEEE*. 2009, pp. 261–270.
16. Molka D., Sch R., Bridge-ep I.X.E.-S. Main Memory and Cache Performance of Intel Sandy Bridge and AMD Bulldozer Memory size. *MSPC 2014*. Available at: https://tu-dresden.de/die_tu_dresden/zentrale_einrichtungen/zih/forschung/projekte/benchit/2014_MSPC_authors_version.pdf (accessed March 15, 2016).
17. Fang Z., Mehta S., Yew P.-C., Zhai A., Greensky J., Gautham B., Binyu Z. Measuring Microarchitectural Details of Multi- and Many-Core Memory Systems through Microbenchmarking. *ACM TACO*. 2014, vol. 11, no. 4, pp. 1–26.
18. Intel® 64 and IA-32 Architectures Software Developer's Manual – Combined Volumes. 2015. Available at: <http://www.intel.de/content/www/de/de/architecture-and-technology/64-ia-32-architectures-software-developer-manual-325462.html> (accessed March 15, 2016).
19. BIOS and Kernel Developer's Guide (BKDG) for AMD Family 15h Models 60h-6Fh Processors. *AMD*. 2015. Available at: http://support.amd.com/TechDocs/50742_15h_Models_60h-6Fh_BKDG.pdf (accessed March 15, 2016).
20. CortexTM-A8 Technical Reference Manual. Revision: r3p2. *ARM*. Available at: <https://static.docs.arm.com/ddi0344/k/DDI0344.pdf> (accessed March 15, 2016).
21. CortexTM-A9 Technical Reference Manual. Revision: r2p2. *ARM*. Available at: http://infocenter.arm.com/help/topic/com.arm.doc.ddi0409f/DDI0409F_cortex_a9_neon_mpe_r2p2_trm.pdf (accessed March 15, 2016).
22. Antonov A. *Organizatsiya izmereniy proizvoditelnosti komponentov podsystemy pamyati mikroprotssorov i SnK*. [Organization of Memory Components Performance Evaluation for Microprocessors and SoCs]. Master's Thesis. ITMO Univ., 2014, pp. 1–95 (in Russ.).
23. Kustarev P., Antonov A., Pinkevich V., Yanalov R. Using selective memory performance evaluation for time-critical embedded systems design. *Proc. 15th Int. Multidiscip. Sci. Geoconf. SGEM 2015*. Available at: <http://www.sgem.org/SGEMLIB/spip.php?article5637> (accessed March 15, 2016).

УДК 004.891

DOI: 10.15827/0236-235X.116.085-088

Дата подачи статьи: 09.09.16

2016. Т. 29. № 4. С. 85–88

АРХИТЕКТУРА СИСТЕМЫ МОНИТОРИНГА ИНФОРМАЦИОННЫХ ТРЕНДОВ НА ОСНОВЕ СВОБОДНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

С.А. Беляев, к.т.н., доцент, beliaev@nicetu.spb.ru;

А.В. Васильев, студент, unlike-2010@mail.ru

*(Санкт-Петербургский государственный электротехнический университет «ЛЭТИ»
им. В.И. Ульянова (Ленина), ул. Профессора Попова, 5, г. Санкт-Петербург, 197376, Россия);*

С.А. Кудряков, д.т.н., зав. кафедрой, psi_center@mail.ru

*(Санкт-Петербургский государственный университет гражданской авиации,
ул. Пилотов, 38, г. Санкт-Петербург, 196210, Россия)*

Статья посвящена описанию программной системы, предназначенной для выявления источников информационных трендов в результате анализа публикаций на новостных сайтах, в социальных сетях и блогах. Основная функция системы – построение графов распространения информации в сети Интернет.

Авторы обосновывают актуальность данной задачи, несмотря на наличие готовых решений, выполняющих сканирование данных в Интернет. Отдельно отмечена проблема лавинообразного увеличения объема информации, требующей обработки.

В статье описана модель для формализации процесса анализа информационных трендов и отмечены отличия от опубликованных подходов к решению данной задачи. Предложены основные шаги по автоматизации решения на основе данной модели. Особое внимание уделено возможности и обоснованности использования программных продуктов с открытым исходным кодом для решения отдельных подзадач. Для построения системы предлагается многоуровневая архитектура, демонстрирующая возможность рационального использования свободного ПО, дана последовательность работы системы.

На основе описанной архитектуры и предложенной модели разработан программный комплекс, обеспечивающий решение задачи мониторинга информационных трендов. Приведены результаты тестирования комплекса на примере нескольких новостных сайтов. Предложены подходы по дальнейшему развитию решения.

Ключевые слова: информационные тренды, мониторинг Интернета, безопасность, система мониторинга, web-ресурсы, архитектура, модель.

В современном мире информация начинает цениться гораздо выше природных ресурсов, потому что именно благодаря ей развиваются и разрушаются экономики, начинаются и заканчиваются войны, формируются и распадаются государства. Многие институты изучают распространение и влияние информационных потоков современного общества, которое все меньше читает книги и смотрит телевизор и все больше времени проводит в сети Интернет. Распространение деструктивной информации в Интернете может иметь катастрофические последствия, поэтому возможность контроля основных информационных трендов и источников их появления является важнейшей задачей. В частности, появляется все больше случаев судебных разбирательств, касающихся сообщений в социальной сети ВКонтакте [1].

Системы мониторинга публикаций в Интернете существуют много лет, активно используются в различных областях и сферах жизни и решают, помимо прочих, задачи анализа средств массовой информации [2], информации по рынку продаж, мероприятий, услуг, контроля появления сообщений на заданную тематику [3] и т.п. Большинство предлагаемых решений предоставляет ссылки на публикации и статистику упоминания тех или иных тем, но без построения графов распространения информации [4].

В Рунете существует множество новостных блогов и информационных сайтов, которые за сутки посещают миллионы пользователей. На наполнение большинства из них работает множество людей, обеспечивающих сбор, систематизацию и подготовку публикаций, зачастую сенсационные сообщения появляются из одного-двух источников. Использование автоматизированных средств мониторинга информационных трендов в системе МВД [5] позволит:

- на ранних стадиях выявлять, предупреждать, пресекать и раскрывать деятельность преступных групп, отдельных лиц и общественных объединений;
- выявлять тенденции по распространению информации;
- увеличить область охвата по сравнению с традиционными методами поиска;
- сократить время реагирования при возникновении новых информационных трендов.

Система мониторинга информационных трендов отслеживает любые изменения и появление новых вбросов информации на информационно-развлекательных порталах, страницах новостных интернет-изданий, блогах и прочих интернет-ресурсах, контролирует время размещения сообщений, предполагаемых авторов и формирует графы распространения.

Особенность публикаций в Интернете заключается в том, что распространением информации занимаются не только профессиональные авторы, зачастую в текстах встречаются опечатки или грубые ошибки, могут использоваться сокращения или выполнена замена слов на их синонимы, что затрудняет сопоставление текстов. Для ослабления влияния данных особенностей система мониторинга информационных трендов использует алгоритмы нечеткого поиска и словари синонимов и сокращений.

Основная трудность связана с объемом обрабатываемой информации: только в социальных сетях, используемых в России, количество сообщений увеличивается более чем на 500 миллионов сообщений в месяц [6], не говоря о новостных сайтах. Сообщения в социальных сетях зачастую имеют явный механизм цитирования, при использовании блогов и подобных им сайтов далеко не всегда можно найти ссылки на источники информации, а увеличение объема отдельных статей в зависимости от используемого алгоритма поиска дубликатов (плагиата) может иметь сложность более $O(n^2)$. Соответственно, поиск может осуществляться только выборочно и по ограниченному подмножеству поисковых запросов, так как построение системы, которая «знает все», равнозначно созданию специализированного аналога поисковых гигантов, таких как Google или Яндекс.

Для формализации процесса анализа информационных трендов целесообразно сформулировать задачу в виде математической модели: $M = (A, G, H, C, L, S, R, T)$, где $A = \{a\}$ – список интернет-адресов новостных сайтов, социальных сетей и блогов, подлежащих контролю; $G: A \rightarrow H$ – функция получения HTML-кода с сайтов с учетом внутренних переходов между страницами по ссылкам, здесь $H = \{h\}$ – множество полученного HTML-кода; $C: H \rightarrow L$ – функция предварительного анализа и формирования множества $L = \{l = \langle a, d, v \rangle\}$, внутреннего представления публикаций, полученных из Интернета, в которых a – адрес публикации, d – дата ее появления, v – текст, подготовленный для анализа; $T = R(L, S)$ – множество обнаруженных информационных трендов, сформированных по элементам множества L с использованием словаря синонимов и сокращений $S = \{s\}$. Каждый элемент множества $T = \{t_i\}$ представляет собой кортеж $t_i = \langle d_i, gr_i, h_i \rangle$, где d_i – дата и время появления информационного тренда; gr_i – граф появления информации в сети Интернет; h_i – HTML-код первоисточника информационного тренда. Граф появления информации в узлах содержит ссылки на публикации, по дугам осуществляется переход к информации о сайтах, на которых публикация появилась позже. В отличие от аналогов [7] предложенная модель не учитывает некоторые особенности информационных трендов, таких как определение цели, интенсивность воздействия, организация

противодействия, что, с одной стороны, не позволяет в полной мере управлять информационными трендами, с другой – дает возможность упростить модель.

Ключевой особенностью предлагаемой модели является формализация информационных трендов в виде графов распространения информации в сети Интернет.

Модель функционирует следующим образом:

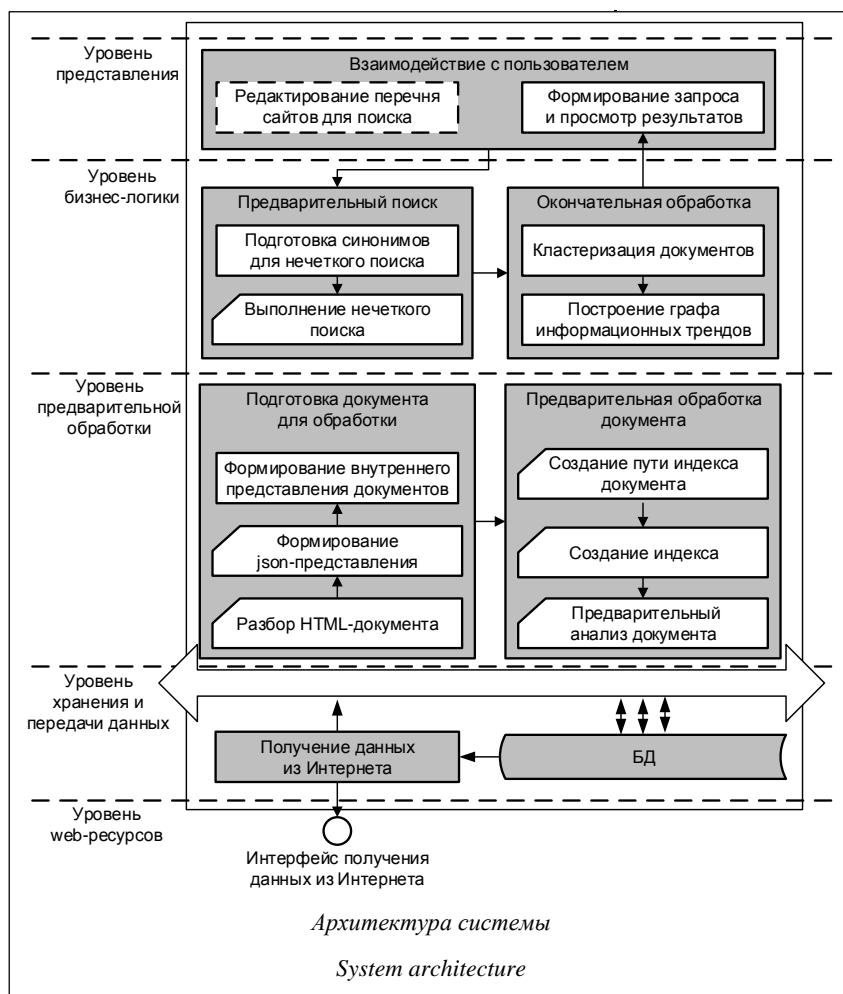
- сбор данных по заданному списку адресов и всем внутренним ссылкам;
- формирование внутреннего представления и предварительный анализ для дальнейшей обработки;
- построение графов появления информации в сети Интернет с использованием синонимов и алгоритмов нечеткого поиска.

Готовых решений, обеспечивающих построение графов появления информации в сети Интернет с учетом особенностей публикуемой информации, нет, но существуют программные продукты, в том числе свободное ПО, которые умеют частично решать данные задачи. Наиболее распространенные и универсальные – решения для локального хранения информации и выполнения нечеткого поиска, одно из наиболее популярных – Apache Lucene [8], в связи с этим дальнейшее построение системы выполнено с использованием данного продукта.

После получения HTML-документов из Интернета по заданному списку адресов осуществляется преобразование их во внутренний формат Lucene, а в качестве промежуточного представления используется JSON, создаваемый с помощью свободно распространяемой библиотеки GSON [9]. Библиотека GSON выбрана в связи с тем, что она быстрее формирует JSON для файлов объемом меньше 10 Мб по сравнению с JSON Simple, Jackson и JSONP. Решение на основе свободного ПО с открытым исходным кодом позволяет не только сократить время разработки, но и повысить его качество за счет использования отлаженных программных решений. Для полноценной реализации в системе мониторинга информационных трендов реализованы следующие алгоритмы:

- анализ HTML-документа с целью выбора статей и обеспечения перехода по внутренним ссылкам;
- нечеткий поиск [10] с использованием синонимов и сокращений;
- статистический анализ степени совпадения двух текстов;
- определение последовательности появления информации в сети Интернет.

С учетом описанной математической модели, возможностей по использованию свободного ПО и требуемых к разработке алгоритмов спроектирована архитектура системы (см. рисунок). В архитектуре прямоугольниками со «срезанным» верх-



ним левым углом представлены модули, реализованные с использованием свободного ПО.

Уровень web-ресурсов является источником данных, из которого с помощью интерфейса получения данных осуществляется сбор HTML-страниц. Полученные страницы поступают на уровень предварительной обработки, где сначала осуществляется подготовка документов, затем процесс обработки.

В рамках подготовки выполняются разбор HTML-страниц и выделение в них статей. В качестве инструмента выделения статей из страниц выбрана популярная библиотека с открытым исходным кодом Java HTML Parser [11]. Она обеспечивает API для извлечения данных и манипулирования ими, используя DOM, CSS и JQuery-подобные методы. Библиотека реализует WHATWG HTML5-спецификацию и разбирает HTML в ту же модель DOM, как это делают современные браузеры, такие как Google Chrome и Mozilla Firefox.

Затем формируется JSON-представление, а на его основании – внутреннее представление документа, которое может обрабатываться Apache Lucene. Дальнейшие этапы предварительной обработки по созданию пути индекса, созданию индекса и предварительному анализу докумен-

та выполняются непосредственно в Lucene. Результаты сохраняются в БД.

На уровне пользователя осуществляется настройка перечня сайтов для поиска, задается поисковый запрос и осуществляется визуальный анализ графов информационных трендов с просмотром соответствующих публикаций, их адресов и даты размещения.

Уровень бизнес-логики на основании запроса пользователя обеспечивает сбор необходимых синонимов для его уточнения, а затем нечеткий поиск с помощью механизмов Lucene. Результаты поиска кластеризуются с использованием стандартного алгоритма шинглов для определения меры совпадения двух документов, и для каждого кластера осуществляется построение графов распространения информации.

Тестирование системы проводилось на ограниченной выборке новостных сайтов (aif.ru, lenta.ru, ria.ru, rg.ru, bfm.ru, pikabu.ru). В результате было построено 25 гра-

фов распространения информации по этим сайтам. Среднее время выборки всех статей с одного сайта составило 24 минуты, среднее время выявления информационного тренда (обработка запроса, выделение кластера и построение графа распространения информации) – 27 секунд. В результате с одного сайта было обработано в среднем 230,65 Мб информации. Тестирование проводилось на персональном компьютере Lenovo y-500 с 6 Гб оперативной памяти и процессором Intel Core i5-3230 2.60GHz.

Оценка эффективности реализации показала, что значительно меньшую часть времени обработки запроса включает в себя нечеткий поиск с использованием синонимов и сокращений, что с учетом количества сайтов, использованных для тестирования, говорит о возможности дальнейшей оптимизации алгоритмов разбора HTML-страниц.

Использование данной системы позволяет осуществлять мониторинг появления информационных трендов и тем самым определять не только источник распространения конкретной новости, но и наиболее влиятельные источники информации, чтобы, с одной стороны, контролировать неожиданное появление новых источников, с другой – обеспечивать контроль минимального и достаточ-

ного количества информационных ресурсов. В дальнейшем целесообразно рассмотреть применимость различных вариантов алгоритмов, в первую очередь, разбора HTML-страниц и лишь затем кластеризации текстовой информации и возможности Lucene по настройке нечеткого поиска.

Литература

1. Борисов С.В., Васнецова А.С. Противодействие экстремистской деятельности – важный аспект обеспечения национальной безопасности // Правовая инициатива. 2014. № 3. URL: <http://49e.ru/ru/2014/3/8> (дата обращения: 03.09.2016).
2. Мониторинг СМИ. URL: <http://www.mlg.ru/solutions/pr/monitoring/> (дата обращения: 03.09.2016).
3. Программа СайтСпутник (FileForFiles & SiteSputnik) – поиск, сбор, мониторинг и анализ информации. URL: <http://sitesputnik.ru/> (дата обращения: 03.09.2016).
4. SCAN Система комплексного анализа информации. URL: <https://scan-interfax.ru/> (дата обращения: 03.09.2016).
5. Солодухин А.И., Романенко С.А., Беляев С.А., Медве-

дева Я.И. Подход к построению комплексной системы предупреждения преднамеренных информационных трендов на основе семантического анализа текстовых ресурсов в сети Интернет // Актуальные проблемы психологической безопасности: сб. тр. регион. совещ. СПб: Свое Изд-во, 2012. С. 79–85.

6. Социальные сети в России: цифры и тренды за февраль 2016 года. URL: <https://br-analytics.ru/blog/socialnye-seti-v-rossii-cifry-i-trendy-za-fevral-2016-g/> (дата обращения: 03.09.2016).

7. Потемкин А.В. Распознавание информационных операций средств массовой информации сети Интернет // Наукoведение. 2015. Т. 7. № 2. URL: <http://naukovedenie.ru/PDF/139TVN315.pdf> (дата обращения: 03.09.2016). DOI: 10.15862/139TVN315.

8. Welcome to Apache Lucene. URL: <http://lucene.apache.org/> (дата обращения: 03.09.2016).

9. Gson. URL: <https://sites.google.com/site/gson> (дата обращения: 03.09.2016).

10. Желудков А.В., Макаров Д.В., Фадеев П.В. Особенности алгоритмов нечеткого поиска // Инженерный вестн. 2014. № 12. URL: <http://engsi.ru/file/out/745418> (дата обращения: 03.09.2016).

11. Jsoup: Java HTML Parser. URL: <https://jsoup.org/> (дата обращения: 03.09.2016).

Software & Systems

DOI: 10.15827/0236-235X.116.085-088

Received 09.09.16

2016, vol. 29, no. 4, pp. 85–88

THE MONITORING OF INFORMATION TRENDS SYSTEM'S ARCHITECTURE BASED ON THE FREE SOFTWARE

S.A. Belyaev¹, Ph.D. (Engineering), Associate Professor, belyaev@nicetu.spb.ru

A.V. Vasilev¹, Student, unlike-2010@mail.ru

S.A. Kudryakov², Dr.Sc. (Engineering), Head of Chair, psi_center@mail.ru

¹ St. Petersburg Electrotechnical University "LETI", prof. Popova St. 5, St. Petersburg, 197376, Russia

² St. Petersburg State University of Civil Aviation, Pilotov St. 38, St. Petersburg, 196210, Russian Federation

Abstract. The article describes a software system designed to identify sources of information trends in the analysis of publications on news sites, social networks and blogs. The main feature of the system is the construction of a graph of information dissemination in the Internet.

The authors prove the relevance of this problem, despite the presence of ready-made solutions, which scan data in the Internet. The paper also focuses on a problem of exponential increase in the volume of information requiring processing.

The article describes a model to formalize the process of analyzing information trends and marked differences from published approaches to solving this problem. The authors propose basic steps for automation solutions based on this model. Special attention is paid to the possibility and validity of using software products with open source code for solving individual subtasks. To build the system the authors offer a layered architecture that demonstrates the possibility of rational use of free software and give a sequence of operation of the system.

On the basis of the architecture and the proposed model there is developed software that provides the solution to the problem of monitoring information trends. The results of testing based on several news sites. The paper proposes some approaches for further development of the solution.

Keywords: information trends, Internet monitoring, security, system monitoring, web-resources, architecture, model.

References

1. Borisov S.V., Vasnetsova A.S. Counteraction to the extremist activity – an important aspect of ensuring national security. *Pravovaya initsiativa* [The Legal Initiative]. Moscow, 2014, no. 3. Available at: <http://49e.ru/ru/2014/3/8> (accessed September 3, 2016).
2. *Monitoring SMI* [Mass Media Monitoring]. Available at: <http://www.mlg.ru/solutions/pr/monitoring/> (accessed September 3, 2016).
3. *Programma SaytSputnik (FileForFiles & SiteSputnik) – poisk, sbor, monitoring i analiz informatsii* [FileForFiles & SiteSputnik. Information Search, Capture, Monitoring and Analysis]. Available at: <http://sitesputnik.ru/> (accessed September 3, 2016).
4. *SCAN Sistema kompleksnogo analiza informatsii* [SCAN. System Information Complex Analysis]. Available at: <https://scan-interfax.ru/> (accessed September 3, 2016).
5. Solodukhin A.I., Romanenko S.A., Belyaev S.A., Medvedeva Ya.I. An approach to constructing a complex prevention system of malicious information trends based on text resource semantic analysis in the Internet. *Aktualnye problemy psikhologicheskoy bezopasnosti: sb. tr. region. soveshch.* [Proc. Regional Conf. on Important Problems of Psychological Security]. St. Petersburg, Svoe Izdatelstvo Publ., 2012, pp. 79–85 (in Russ.).
6. *Sotsialnye seti v Rossii: tsifry i trendy za fevral 2016 goda* [Social Networks in Russia: Numbers and Trends for February 2016]. Available at: <https://br-analytics.ru/blog/socialnye-seti-v-rossii-cifry-i-trendy-za-fevral-2016-g/> (accessed September 3, 2016).
7. Potemkin A.V. Recognition of mass media information operations in the Internet. *Naukovedenie* [Science Studies]. 2015, vol. 7, no. 2 (27). Available at: <http://naukovedenie.ru/PDF/139TVN315.pdf> (accessed September 3, 2016).
8. *Welcome to Apache Lucene*. Available at: <http://lucene.apache.org/> (accessed September 3, 2016).
9. *Gson*. Available at: <https://sites.google.com/site/gson> (accessed September 3, 2016).
10. Zheludkov A.V., Makarov D.V., Fadeev P.V. Characteristics of fuzzy search algorithms. *Inzhenerny vestnik* [Engineering Bulletin]. Moscow, N.E. Bauman MSTU Publ., 2014, no. 12. Available at: <http://engsi.ru/file/out/745418> (accessed September 3, 2016).
11. *Jsoup: Java HTML Parser*. Available at: <https://jsoup.org/> (accessed September 3, 2016).

УДК 004.4'242

DOI: 10.15827/0236-235X.116.089-093

Дата подачи статьи: 04.10.16

2016. Т. 29. № 4. С. 89–93

ГРАФИЧЕСКОЕ МОДЕЛИРОВАНИЕ ТЕХНОЛОГИЧЕСКОГО ПРОЦЕССА КАК ВСПОМОГАТЕЛЬНОЕ СРЕДСТВО СОСТАВЛЕНИЯ АЛГОРИТМА УПРАВЛЕНИЯ

*А.В. Парамонов, ведущий инженер-программист, paramonov-andrew@rambler.ru
(Машиностроительный концерн «ОРМЕТО-ЮУМЗ», просп. Мира, 12, г. Орск, 462403, Россия)*

Основным этапом в разработке программ для логических контроллеров в автоматизированных системах управления технологическими процессами является составление алгоритма управления. Одной из проблем, возникающих на этом этапе, является то, что алгоритм зачастую описывается в потенциально неполной или противоречивой словесной форме. Предлагаемые же формальные средства описания алгоритма требуют от постановщика задачи знания формальных теорий.

Целью данного исследования является получение такого средства описания алгоритмов, которое было бы доступно для всех участников разработки программ и обеспечивало бы полноту и непротиворечивость. Для достижения этой цели предлагается использовать графические модели технологических установок в качестве средства для описания алгоритмов. Описываются правила построения моделей. Предлагается методика представления алгоритма на модели с поддержанием полноты и непротиворечивости, что делает процесс его описания доступным для постановщика задачи. Излагаются правила описания алгоритма по модели в виде таблицы решений. Однако изложенные идеи могут быть использованы для получения других форм, например конечного автомата. В качестве основы для графических моделей используются иерархические структуры и реляционная модель.

Ключевые слова: программируемый логический контроллер, программирование контроллеров, система логического управления, технологическая установка, графическая модель.

Алгоритм логического управления может быть записан разными способами, и от того, насколько полна и непротиворечива запись, во многом зависит корректность написанной в дальнейшем программы для *программируемого логического контроллера* (ПЛК) [1]. При записи алгоритма естественными языками велик риск возникновения неполноты и/или противоречивости в силу их неоднозначности. Чтобы устранить этот риск, для описания алгоритмов используют формальные средства, например, булевы формулы [2], таблицы решений [3, 4], конечные автоматы [2], сети Петри [5], формулы темпоральной логики [6] и многие другие. Эти средства хороши, удобны и понятны для разработчика либо любого другого участника, владеющего формальной теорией, которая лежит в их основе. Однако при этом сохраняется разрыв между специалистом предметной области (*технологического процесса* (ТП)) и разработчиком. Специалист предметной области имеет знания о ТП, но у него нет знаний формальной теории и навыков корректной постановки задачи управления [7]. Разработчик же, наоборот, имеет поверхностные знания о ТП, но владеет формальной теорией, хотя тоже может не иметь навыков корректной постановки задачи управления. Из-за такого разрыва передача знаний о полном и непротиворечивом алгоритме управления от специалиста предметной области разработчику затрудняется.

Далее представлен метод описания алгоритма на основе графических моделей *автоматизированных технологических установок* (АТУ), визуально схожих с оригиналами. Метод поддерживается формальными средствами (таблицы решений, конечные автоматы и так далее), но вместо них

предоставляет специалисту предметной области графическое изображение технологической установки вместе с изображениями датчиков, органов управления, средств индикации и исполнительных механизмов. Специалист при составлении алгоритма имитирует процесс управления установкой, а соответствующий алгоритм записывается в нужной форме.

Графическая модель АТУ

Прежде чем описать графическую модель, дадим определение некоторым терминам.

Технологическая установка – комплекс оборудования и сооружений, предназначенный для проведения ТП. Примерами технологической установки являются ленточные конвейеры [8, 9], вагонопрокидыватели [10], трансбордеры, вагонотолкатели и так далее (рис. 1).

АТУ – совокупность технологических установок и АСУ процесса, проводимого на данных технологических установках.

Любую АТУ можно рассматривать как совокупность отдельных ее частей, организованных в иерархическую структуру. Например, любая АТУ, как правило, включает непосредственно металлоконструкцию ТУ, шкаф управления, пульт управления. На металлоконструкции, в свою очередь, располагаются различные датчики, а пульт включает различные органы управления.

Любую отдельно взятую часть АТУ, в том числе такую, которая, в свою очередь, также включает отдельные части, будем называть *элементом АТУ* (или иногда для краткости просто *элементом*).

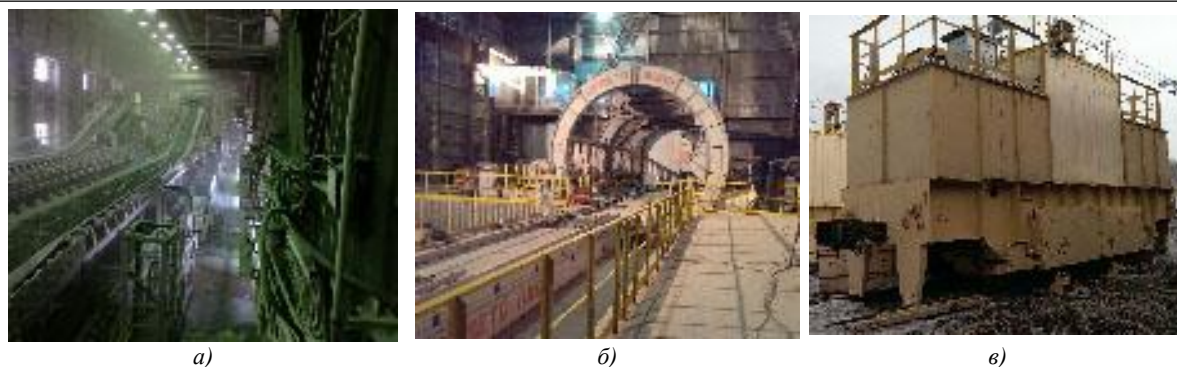


Рис. 1. Примеры технологических установок для ТП перегрузки насыпных материалов:
а) ленточный конвейер, б) вагоноопрокидыватель с трансбордером, в) вагонотолкатель

Fig. 1. Examples of processing stations for technological processes of bulk material handling:
a) belt conveyor, б) car dumper with transborder, в) car pusher

Далее моделирование иерархической структуры АТУ рассматривается с позиций реляционной теории и основывается на таких понятиях, как отношение, кортеж, кардинальность, атрибут, арность, тип и первичный ключ [10]. Однако понятие типа не используется в традиционном смысле. Вместо этого предопределим лишь два типа, общих для всей модели: *INT* – множество целых чисел и *STR* – множество строк символов. Модель АТУ представляется шестеркой отношений следующим образом: $M = (R_E, R_{CE}, R_I, R_{EI}, R_G, R_{CG})$, где R_E – множество элементов АТУ; R_{CE} – множество связей между элементами; R_I – множество изображений элементов или правил их построения; R_{EI} – множество соответствий элементов и их изображений; R_G – множество групп элементов; R_{CG} – множество связей между группами.

Рассмотрим каждое из шести отношений подробнее.

Отношение R_E определяет множество элементов и состоит (как и любое другое отношение) из заголовка и тела: $R_E = (H_E, B_E)$, $H_E = \{E\# : INT, NAME : STR, DESCR : STR, G\# : INT\}$, где $E\#$ – номер элемента; $NAME$ – название; $DESCR$ – описание; $G\#$ – номер группы элемента.

Атрибут $E^{\#}$ является первичным ключом, $G^{\#}$ – внешним ключом. Тело B_E представляет собой множество кортежей из значений атрибутов, указанных в заголовке. Каждый кортеж задается сле-

дующим образом: $b_E = \{E\# : v_1, NAME : v_2, DESCR : v_3, G\# : v_4\}$, где v_1, v_2, v_3, v_4 – значения соответствующих атрибутов.

Арность отношения R_E равна четырем. На рисунке 2а показано табличное представление данного отношения.

Условимся, что вторая строка таблицы представляет заголовок H_E , каждая последующая строка – в точности один кортеж из B_E , а каждый столбец – в точности один атрибут.

Для удобства последующего изложения, но без потери его строгости будем использовать табличные представления при определении отношений.

Как было сказано, структура АТУ иерархична. Для реализации данного обстоятельства в модели используется бинарное отношение R_{CE} . Если рассматривать R_E как множество вершин дерева элементов АТУ, то R_{CE} является множеством ребер и определяется в соответствии с рисунком 26.

Здесь *ANC#* – номер элемента предка; *DESC#* – номер элемента потомка.

Атрибуты *ANC#* и *DESC#* являются внешними ключами; первичным ключом является совокупность $\{ANC#, DESC#\}$.

Каждому элементу АТУ, за исключением отдельно обговоренных случаев, ставится в соответствие его графическое изображение. Совокупность таких изображений, определенным образом расположенных на плоскости, является изображением

$R_E = (H_E, B_E)$			
$E\# : INT$	$NAME : STR$	$DESCR : STR$	$G\# : INT$
v_1	v_2	v_3	v_4

$a)$

$R_{CE} = (H_{CE}, B_{CE})$	
$ANC\# : INT$	$DESC\# : INT$
v_1	v_2

$b)$

$R_I = (H_I, B_I)$	
$I\# : INT$	$IMAGE : STR$
v_1	v_2

$c)$

$R_{EI} = (H_{EI}, B_{EI})$			
$E\# : INT$	$I\# : INT$	$X : INT$	$Y : INT$
v_1	v_2	v_3	v_4

$d)$

$R_G = (H_G, B_G)$	
$G\# : INT$	$NAME : STR$
v_1	v_2

$e)$

$R_{CG} = (H_{CG}, B_{CG})$	
$ANC\# : INT$	$DESC\# : INT$
v_1	v_2

$f)$

Рис. 2. Табличное представление отношений: а) R_E , б) R_{CE} , в) R_I , г) R_{EI} , д) R_G , е) R_{CG}

Fig. 2. Tabular picture of the relations: a) R_E , б) R_{CE} , в) R_I , г) R_{EI} , д) R_G , е) R_{CG}

АТУ. Отношение R_I является множеством правил построения изображений элементов. На рисунке 2в показана таблица, представляющая R_I .

Здесь $I\#$ – номер правила; $IMAGE\#$ – имя файла с изображением или имя функции построения изображения.

Атрибут $I\#$ является первичным ключом.

Для соотнесения элементов с изображениями используется отношение R_{EI} , табличное представление которого показано на рисунке 2г.

Здесь $E\#$ – номер элемента; $I\#$ – номер правила построения изображения; X – координата x изображения на плоскости; Y – координата y изображения на плоскости.

В случае изображения АТУ в виде изометрической проекции в R_{EI} вводится атрибут Z , обозначающий третью координату.

Первичным ключом для R_{EI} является совокупность атрибутов $\{E\#, I\#\}$.

Каждый элемент АТУ будем относить к определенной, только одной, группе. Группы, как и элементы, организованы в иерархическую структуру. Отношения R_G и R_{CG} представляют соответственно множество групп и множество связей. На рисунке 2д показана таблица, представляющая бинарное отношение R_G .

Здесь $G\#$ – номер группы; $NAME$ – название группы.

Первичным ключом R_G является атрибут $G\#$.

Табличное представление бинарного отношения связей R_{CG} приведено на рисунке 2е.

Здесь $ANC\#$ – номер группы предка; $DESC\#$ – номер группы потомка.

Атрибуты $ANC\#$ и $DESC\#$ являются внешними ключами; первичным ключом является совокупность $\{ANC\#, DESC\#\}$.

Алгоритм построения графической модели

Представим алгоритм построения графической модели.

1. Определение базиса групп.

Построение модели имеет смысл лишь относительно определенного базиса групп. Базис, в свою очередь, задается отношениями R_G и R_{CG} . Минимальным, в смысле количества групп, является базис, показанный на рисунке 3.

2. Добавление корневого элемента.

Корневым элементом всегда является элемент, символизирующий всю АТУ в целом. В тело отношения R_E добавляется кортеж, представленный на рисунке 4.

Изображения данный элемент не имеет.

3. Добавление АТУ более низкого уровня.

Если вся АТУ как объект управления представляет комплекс более мелких АТУ, то соответ-

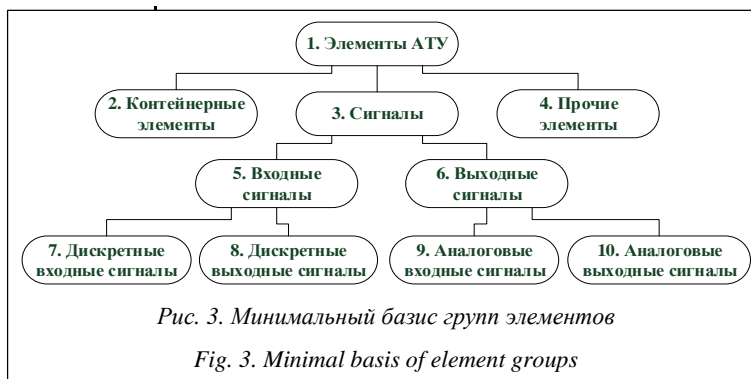


Рис. 3. Минимальный базис групп элементов

Fig. 3. Minimal basis of element groups

ствующие этим АТУ элементы добавляются в R_E с атрибутом $G\#$, равным 2. В отношение R_{CE} добавляются соответствующие связи.

4. Добавление металлоконструкций.

Для каждой АТУ добавляется металлоконструкция технологической установки в отношение R_E с атрибутом $G\#$, равным 4 (только в минимальном базисе). В отношение R_{CE} добавляется соответствующая связь, в отношение R_I – правило изображения металлоконструкции, а в R_{EI} – соотношение элемента с правилом изображения.

Добавление элементов по пунктам 5–10 происходит аналогично указаниям, приведенным в данном пункте.

5. Добавление шкафов, щитов и ящиков управления как частей АТУ.

6. Добавление датчиков в качестве дочерних элементов по отношению к тем металлоконструкциям, шкафам, щитам и ящикам, на которых эти датчики установлены.

7. Добавление исполнительных механизмов в качестве дочерних элементов по отношению к тем металлоконструкциям, шкафам, щитам и ящикам, на которых эти механизмы установлены.

8. Добавление органов управления в качестве дочерних элементов по отношению к тем шкафам, щитам и ящикам, на которых эти органы управления установлены.

9. Добавление средств индикации в качестве дочерних элементов по отношению к тем металлоконструкциям, шкафам, щитам и ящикам, на которых эти средства установлены.

10. Добавление сигналов в качестве дочерних элементов по отношению к датчикам, исполнительным механизмам, органам управления и средствам индикации. При этом в отношении R_E значение атрибута $G\#$ принимается равным 7, 8, 9 или 10 (рис. 3).

11. Приведение системы управления к логической.

R_E			
$E\# : INT$	$NAME : STR$	$DESCR : STR$	$G\# : INT$
1	<Название АТУ>	<Описание АТУ>	2

Рис. 4. Кортеж корневого элемента

Fig. 4. A root element tuple

Для каждого аналогового сигнала добавляются дискретные сигналы, отождествляемые с логическими высказываниями. Например, для аналогового сигнала могут быть добавлены три дискретных сигнала, отождествляемых с высказываниями: «значение аналогового сигнала меньше нуля», «значение больше нуля» и «значение равно нулю». Таким способом любая система управления с аналоговыми сигналами может быть приведена к логической.

При наличии изображений с прозрачным фоном (или функций для построения) для всех тех элементов АТУ, для которых эти изображения требуются, по модели восстанавливаются изображения АТУ: по одному для каждой металлоконструкции, каждого шкафа, щита и ящика.

Порядок наложения изображений друг на друга определяется отношением R_{CE} : те элементы, которые ближе к корню дерева, находятся ниже в изображении и наоборот.

Как показано выше, добавление элементов в модель почти всегда сопровождается установлением связей в R_{CE} . При этом не каждая связь может быть семантически правильной. Например, шкаф управления не может быть частью конечного выключателя. Вопросы обеспечения корректности модели в данной статье не рассматриваются. Отметим лишь, что для решения этой проблемы с успехом могут быть применены экспертные системы.

Описание алгоритма управления по графической модели

Алгоритм по модели может быть построен различными способами. Могут быть выбраны и различные формы представления алгоритма – таблицы решений, конечные автоматы, сети Петри и т.д. Только лишь для иллюстрации основных идей построения алгоритма по модели выберем наиболее простую для понимания форму – таблицы решений.

Перед началом построения модели имеется пустое отношение R_I (наряду с остальными отношениями, которые также являются пустыми). Любой логический дискретный сигнал может иметь три значения: безразличное, истина и ложь. Для каждого из этих значений добавим в отношение R_I отдельное изображение. Для определенности пусть это будет серый круг для безразличного значения, зеленый круг – для истинного и белый – для ложного (рис. 5).

R_I	
$I\# : INT$	$IMAGE : STR$
1	graycircle.jpg
2	greencircle.jpg
3	whitecircle.jpg

Рис. 5. Отношение R_I после добавления изображений значений логического сигнала

Fig. 5. R_I relation after adding pictures of logical signal values

В каждый момент времени с дискретным сигналом в модели может быть связано только одно из этих изображений, что определяется отношением R_{EI} . Добавление в отношение R_I столько изображений, сколько значений может принимать дискретный сигнал (включая безразличное), является обязательным условием вне зависимости от способа построения и формы представления алгоритма.

В терминах языка Tutorial D [10] множество идентификаторов условий таблицы решений – результат выражения $(R_E \text{ WHERE } G\# = 7) \{NAME\}$.

Множество идентификаторов действий – результат выражения $(R_E \text{ WHERE } G\# = 9) \{NAME\}$.

Один вектор условий можно получить как результат выражения $((R_E \text{ JOIN } R_{EI}) \text{ WHERE } G\# = 7) \{I\# \}$. Один вектор действий, в свою очередь, – результат выражения $((R_E \text{ JOIN } R_{EI}) \text{ WHERE } G\# = 9) \{I\# \}$.

В случае последовательной системы вектору условий соответствует результат выражения $((R_E \text{ JOIN } R_{EI}) \text{ WHERE } G\# = 7 \text{ OR } G\# = 9) \{I\# \}$.

Для удобства восприятия значения атрибута $I\#$ 1, 2, 3 могут быть переименованы в «-», «Y» и «N» соответственно.

Смена изображений сигналов, а точнее обновление кортежей в R_{EI} , определяет возможность добавления новых векторов условий и действий в таблицу решений. Таким образом, последовательной сменой изображений можно добиться определения всех мыслимых сочетаний значений сигналов и заполнения таблицы решений.

Выше было сказано, что добавление изображений для различных значений дискретного сигнала должно происходить в самом начале построения модели. Это может быть сделано и на любом другом этапе, но при этом должно быть определено соответствие номеров $I\#$ значениям сигналов.

По построенному в виде таблицы решений алгоритму может быть получена программа, например на языке ST [11], с помощью метода маски [3, 4]. При этом к таблице решений (как и к другим формам) также могут быть применены формальные методы обнаружения и устранения неполноты и противоречивости.

Таким образом, в статье описана методика моделирования АТУ на основе реляционной модели и реляционной алгебры. Модель представляет иерархическую структуру связанных элементов, структуру связанных групп, а также множество правил изображения элементов.

Основным назначением описанной методики является предоставление специалисту предметной области возможности построения полных и непротиворечивых алгоритмов без знаний теорий, лежащих в основе формальных средств. Так как знания данного специалиста сосредоточены в области ТП, предложено использование графических моделей, визуально схожих с реальными АТУ. Описаны пра-

вила построения моделей, правила представления и составления алгоритмов по модели.

Методика получает очень удобную для пользователя программную реализацию, если смена изображения дискретного сигнала осуществляется, например, кликом левой кнопки мыши по этому изображению. Часто используемые типовые элементы могут храниться отдельно для повторного использования.

При наличии метода трансляции предложенная модель может быть использована для получения программ человеко-машинных интерфейсов. Графическая модель, дополненная моделью поведения АТУ, может быть использована в качестве тренажера для обслуживающего персонала.

Литература

1. Парр Э. Программируемые контроллеры: рук-во для инженера; [пер. с англ. Б.И. Копылова]. М.: БИНОМ. Лаборатория знаний, 2007. 516 с.
2. Шалыто А.А. SWITCH-технология. Алгоритмизация и программирование задач логического управления. СПб: Наука, 1998. 628 с.

3. Фрайтаг Г., Года В., Якоби Х. и др. Введение в технику работы с таблицами решений; [пер. с нем. М.Г. Гаазе-Рапопорта]. М.: Энергия, 1979. 88 с.
4. Хамби Э. Программирование таблиц решений; [пер. с англ. С.В. Диева]. М.: Мир, 1976. 88 с.
5. Хадеев А.С., Браништов С.А. Метод синтеза LD-программ для ПЛК на основе концепции супервизорного управления дискретно-событийными системами, представленными в сетях Петри // Автоматизация, телемеханизация и связь в нефтяной промышленности. 2013. № 1. С. 42–48.
6. Кузьмин Е.В., Соколов В.А., Рябухин Д.А. Построение и верификация LD-программ ПЛК по LTL-спецификации // Моделирование и анализ информационных систем. 2013. Т. 20. № 6. С. 78–94.
7. Червенчук В.Д. Методы и средства синтеза алгоритмического и программного обеспечения систем управления с использованием таблиц решений: дис. ... канд. техн. наук. Омск, 1984. 123 с.
8. Волков Р.А., Гнутов А.Н., Дьячков В.К. и др. Конвейеры: справочник; [под общ. ред. Ю.А. Перетена]. Л.: Машиностроение, 1984. 367 с.
9. Журавлев Н.П., Маликов О.Б. Транспортно-грузовые системы: учебник. М.: УМНЦ, 2006. 368 с.
10. Дейт К.Дж. Введение в системы баз данных; [пер. с англ. М.Л. Степановой]. М.: Вильямс, 2005. 1328 с.
11. Петров И.В. Программируемые контроллеры. Стандартные языки и приемы прикладного проектирования. М.: СОЛОН-Пресс, 2004. 256 с.

Software & Systems

DOI: 10.15827/0236-235X.116.089-093

Received 04.10.16

2016, vol. 29, no. 4, pp. 89–93

GRAPHICAL MODELING OF TECHNOLOGICAL PROCESS AS A SUPPORT TOOL FOR DESIGNING OF CONTROL ALGORITHM

A.V. Paramonov¹, Leading Engineer-Programmer, paramonov-andrew@rambler.ru

¹ Heavy Engineering "ORMETO-YUMZ", Mira Ave. 12, Orsk, 462403, Russian Federation

Abstract. The main stage of program development for PLCs in automated control systems is designing a control algorithm. One of the problems we meet at this stage is that the algorithm is often described in a potentially incomplete or inconsistent verbal form. Existing formal tools of algorithm description require a person who designs the algorithm to know formal theories.

The aim of this work is to obtain a tool to design algorithms, which would be available for all participants in the development of programs and ensures completeness and consistency. To achieve this goal the author proposes using a graphical model of processing stations as a tool to design the algorithms. The paper describes the rules for designing models. The paper also proposes a method of algorithm presentation on a model with maintenance of completeness and consistency, which makes the designing process available to a person who designs the algorithm. The article provides the rules to describe the algorithm using the model in the form of decision tables. However, the ideas may be used for other forms, such as a finite-state automaton. The hierarchical structures and relational model are used as a basis for graphical models.

Keywords: programmable logic controller, PLC programming, logic control system, processing station, graphical model.

References

1. Parr E.A. *Programmable controllers. An engineer's guide*. 3rd ed., Newnes Publ., 2003, 448 p. (Russ. ed.: Moscow, Binom Publ., Laboratoriya znany, 2007, 516 p.).
2. Shalyto A.A. *SWITCH-tekhnologiya. Algoritimizatsiya i programmirovaniye zadach logicheskogo upravleniya* [SWITCH-Technology: Algorithmization and Programming of Logical Control Problems]. St. Petersburg, Nauka Publ., 1998, 628 p.
3. Freitag G., Gode W., Jacobi H., Lautz H., Simon J., Spittel U. *Einführung in die Entscheidungstabellentechnik* [Introduction to the Technique of Working with Decision Tables]. Berlin, Verlag Technik Publ., 1976, 88 p.
4. Humby E. *Programs from decision tables*. Macdonald and Co. Publ., 1973, 91 p. (Russ.ed.: Moscow, Mir Publ., 1976, 88 p.).
5. Khadeev A.S., Branishtov S.A. Method of PLC LD-program synthesis based on the concept of supervisory control of discrete-event systems represented by Petri nets. *Avtomatizatsiya, telemekhanizatsiya i svyaz v neftyanoy promyshlennosti* [Automation, Telemechanization and Communication in Petroleum Industry]. 2013, vol. 1, pp. 42–48 (in Russ.).
6. Kuzmin E.V., Sokolov V.A., Ryabukhin D.A. Design and verification PLC LD-programs by LTL-specifications. *Modelirovaniye i analiz informatsionnykh sistem* [Information Systems Modeling and Analysis]. 2013, vol. 6, pp. 78–94 (in Russ.).
7. Chervenчук V.D. *Metody i sredstva sinteza algoritmicheskogo i programmnoy obespecheniya sistem upravleniya s ispolzovaniem tablits resheniy* [Methods and Tools of control System Algorithms and Software Synthesis Using Decision Tables]. Ph.D. Thesis, Omsk, 1984, 123 p.
8. Volkov R.A., Gnutov A.N., Dyachkov V.K. *Konveyery: spravochnik* [Conveyors: a Reference Book]. Leningrad, Mashinostroeniye Publ., 1984, 367 p.
9. Zhuravlev N.P., Malikov O.B. *Transportno-gruzovye sistemy* [Transport-Cargo Systems]. Textbook, Moscow, UMNTS Publ., 2006, 368 p.
10. Date C.J. *An Introduction to Database Systems*. 8th ed., Addison Wesley Publ., 2004, 1336 p. (Russ.ed.: Stepanova M.L., Moscow, Vilyams Publ., 2005, 1328 p.).
11. Petrov I.V. *Programmiruemye kontrollery. Standartnye yazyki i priemy prikladnogo proektirovaniya* [Programmable Controllers. Standard Languages and Techniques for Applied Design]. Moscow, SOLON-Press, 2004, 256 p.

УДК 687.021:004.942

DOI: 10.15827/0236-235X.116.094-100

Дата подачи статьи: 29.09.16

2016. Т. 29. № 4. С. 94–100

МЕТОД ПОЛУЧЕНИЯ РАЗВЕРТКИ ДЕТАЛЕЙ ОДЕЖДЫ С УЧЕТОМ ДЕФОРМАЦИОННОЙ СПОСОБНОСТИ МАТЕРИАЛА

А.Е. Горелова, к.т.н., доцент, GorelovaAnn@mail.ru;

Н.А. Корнилова, д.т.н., доцент, начальник инженерингового центра, n Korn@mail.ru;

*З.Р. Григорьева, аспирант, zareta_grigoreva@inbox.ru
(Ивановский государственный политехнический университет,
ул. 8 Марта, 20, г. Иваново, 153037, Россия);*

*Д.А. Васильев, ведущий программист, vasmec@gmail.com
(ИИТ Консалтинг, ул. Суворова, 39-524, г. Иваново, 153012, Россия);*

*А.А. Корнилова, магистрант, лаборант, lmila.korn@gmail.com
(Санкт-Петербургский национальный исследовательский университет
информационных технологий, механики и оптики (Университет ИТМО),
Кронверкский просп., 49, г. Санкт-Петербург, 197101, Россия;
Санкт-Петербургский политехнический университет Петра Великого,
ул. Политехническая, 29, г. Санкт-Петербург, 195251, Россия)*

В работе предложен и исследован алгоритм получения развертки деталей швейного изделия (ЗДК) путем поэтапного преобразования триангуляционной сети деталей трехмерной конструкции изделия с использованием методов геометрического преобразования поверхности и физического моделирования поведения материала. На первом этапе осуществляется приведение объемной оболочки к плоскому виду путем проецирования ребер ячеек сети на плоскость развертывания. Затем применяется физическое моделирование, суть которого заключается в придании каждому узлу триангуляционной сети свойств материальной точки с массой, равной 1, а каждому ребру – упругой связи, стремящейся привести его длину к исходной длине. Процесс включает две временные стадии: рывок для выведения системы материальных точек из равновесия и переход детали к устойчивому равновесному состоянию (итерация), и продолжается до достижения минимальной суммарной силы по детали, что означает максимально возможное приближение геометрических характеристик ЗДК и развертки.

Оценка качества предложенного алгоритма осуществлена путем построения разверток шаровых сегментов полушеры. Показано, что развертка, получаемая в результате первого этапа преобразования, практически непригодна для раскроя детали из ткани, так как имеет недостаточную площадь. В результате физического моделирования происходит приведение площади детали к исходному значению, при этом периметр развертки всегда не меньше ЗДК. Также может наблюдаться несоответствие основных габаритных размеров деталей, что говорит о необходимости применения принудительного формования либо использования свойств материалов для получения формы изделия. Величина деформации определяет корректность выбора материалов для модели.

Метод реализован в САПР плотно облегающих изделий BustCAD и показал хорошие практические результаты.

Ключевые слова: САПР одежды, геометрическое моделирование, физическое моделирование, развертка трехмерной поверхности, деформация участка поверхности, свойства материалов.

Совершенствование методов автоматизированного проектирования – одно из наиболее перспективных направлений исследований в швейной промышленности. Актуальным является переход к трехмерному проектированию модели с последующим построением лекал и визуализацией посадки проектируемого изделия на виртуальном манекене. Полную реализацию этого направления проектирования сдерживает отсутствие полноценного обеспечения информации о закономерностях формообразования оболочек под влиянием конструктивных особенностей изделия и показателей свойств материалов.

Задача получения развертки одежды как неразвертываемой поверхности в 3D САПР сводится к определению возможной доли деформации и поиску замещающей геометрической фигуры, отличающейся от развертываемой на величину, не превышающую предел деформации [1].

Геометрические модели определяют трехмерную форму изделия набором развертывающихся линейных поверхностей, которые, с точки зрения дифференциальной геометрии, наследуют многие желательные характеристики и обеспечивают возможность использования параметрических моделей изделий. Так, в [2] предложен метод описания трехмерной поверхности конусами, в [3] разработан способ получения разверток на основе отображения на плоскость детали, инцидентной огибающей торсовой поверхности. В работе [4] использован метод, основанный на положениях теории чебышевских сетей: моделируется процесс укладки пространственной сети на плоскость с решением задачи сведения к минимуму искажений для каждого выделенного участка поверхности. В [5] исследована возможность использования значений гауссовой кривизны для получения плоской развертки произвольной нелинейной поверхности.

Исследования показали, что прямое применение данного алгоритма малоэффективно, поскольку отдельные куски поверхности для уменьшения своей гауссовой кривизны стремятся к минимальным размерам и в пределе стягиваются в точку.

Недостатком всех чисто геометрических методов является то, что в них не учитывается способность материалов к деформации и сжатию при одевании трехмерной формы, изготовленной из плоских деталей.

Физические модели направлены на использование в процессе разворачивания свойств текстильных материалов подвергаться растяжению, сдвигу и изгибу. В работах [6, 7] используется пространственная сеть, представляющая собой пружинно-массовую модель, длины сторон ячеек которой могут изменяться в процессе разворачивания, но при этом стремятся сохранить исходную длину. При разворачивании решается задача поиска замещающей плоской детали с минимальной внутренней энергией сети. Методы [8, 9] основаны на минимизации энергии растяжимых деформаций триангуляционной сети. Достоинством методов физического моделирования является возможность получать развертки деталей любой формы и размера, не ограничивая конструктора в модельном решении.

Авторами исследована возможность сочетания геометрических и физически ориентированных методов для получения развертки трехмерной детали одежды, наиболее соответствующей реальным условиям формообразования.

Теоретические исследования

Получение плоских шаблонов предложено осуществлять путем поэтапного преобразования триангуляционной сети деталей трехмерной конструкции изделия.

Поверхность исходной трехмерной конструкции изделия (3DK) представляет собой жесткую оболочку, объемная форма которой в той или иной степени повторяет поверхность манекена. Трехмерная конструкция имеет структуру нерегулярной триангуляционной сети, в которой количество и размер ячеек зависят от формы и размера аппроксимируемого участка поверхности [10]. Количество ячеек сети 3DK определяет качество восприятия модели, точность и длительность процедуры разворачивания.

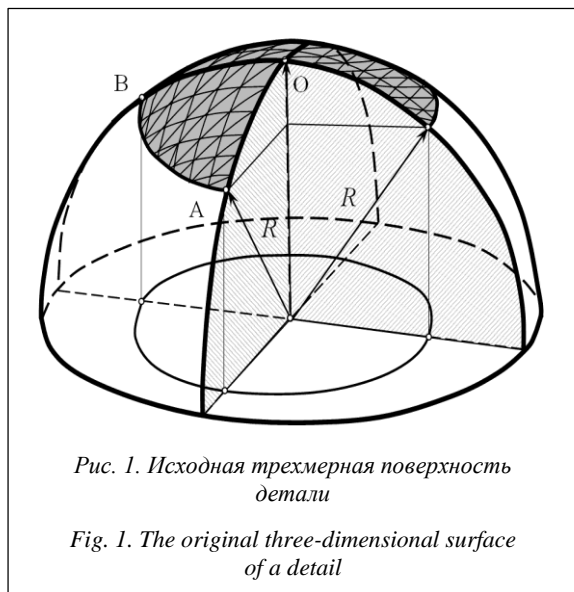
При создании модели изделия пользователь наносит на поверхности 3DK конструктивные линии и формирует детали. Исходная деталь представляет собой триангуляционную сеть, расположенную в трехмерном пространстве и ограниченную конструктивными линиями. Положение линий не всегда совпадает с вершинами ячеек, поэтому в их окрестностях происходит переразбиение исходной сети (увеличение количества элементов).

Геометрическими характеристиками трехмерной детали изделия являются количество ячеек N , их трехмерные параметры (длина и угол наклона ребер ячейки к каждой оси $M = \sum \vec{r}_{3D}$), периметр P_{3D} и площадь S_{3D} детали.

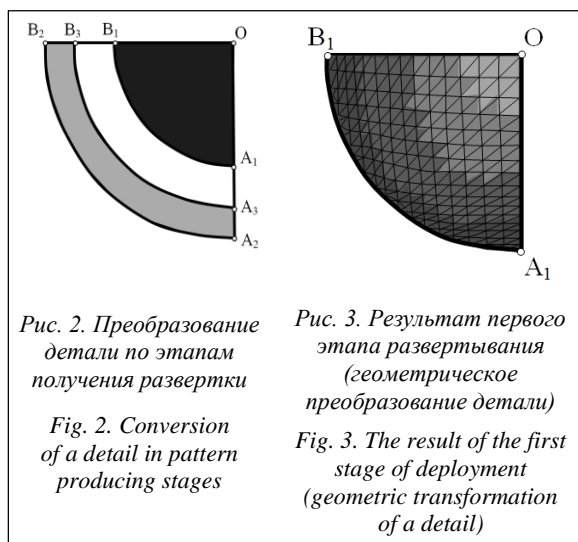
Задачей первого этапа является приведение объемной оболочки к плоскому виду. Для этого используется **геометрическое моделирование** ($3DK \rightarrow 2DK^1$), включающее проецирование каждого элемента структуры (ребра ячейки) \vec{r}_{3D} на плоскость разворачивания \vec{r}_{2D^1} . При ненулевой кривизне оболочки происходит изменение размеров каждого ребра ячейки (условная деформация, сопутствующая преобразованию, величина которой пропорциональна углу наклона к плоскости разворачивания), то есть $|\vec{r}_{3D}| \geq |\vec{r}_{2D^1}|$. Для минимизации возникающей деформации, то есть обеспечения близости длин исходных \vec{r}_{3D} и конечных \vec{r}_{2D^1} векторов, возможно применение методов геометрического преобразования, основанных на аппроксимации отдельных участков трехмерной поверхности разворачивающимися телами [10]. Более универсальным методом является поиск оптимального положения плоскости проецирования, который может быть реализован методом наименьших квадратов. При этом анализу подлежат суммы длин векторов, определяющих величины смещения наиболее удаленных точек триангуляционной сети в процессе проецирования трехмерной детали на плоскость. При разворачивании таких деталей, как рукав, спинка, переходящая на полочку, и т.д., целесообразно применение поэтапного геометрического моделирования, включающего проецирование на боковую поверхность цилиндра с дальнейшим преобразованием ее в плоскую деталь.

На рисунке 1 показан процесс получения развертки участка AOB сферы с радиусом R . Пример геометрического преобразования детали, расположенной на участке выпуклости, приведен на рисунках 2 и 3: A_1OB_1 – развертка участка AOB ($2DK^1$). Преобразование детали, расположенной на участке вогнутости, осуществляется аналогично. При условии сохранения значения R развертки участков будут идентичными.

Деталь $2DK^1$, полученная в результате геометрического преобразования (рис. 3), характеризуется неравномерно распределенной деформацией сжатия. Практически для каждой элементарной ячейки характерно соотношение $|\vec{r}_{3D}| \geq |\vec{r}_{2D^1}|$. Величина деформации увеличивается от центра к краю детали, минимальную деформацию имеют ребра поверхности, параллельные плоскости разворачивания, а максимальное сжатие наблюдается на участках с наибольшим углом наклона к данной плоскости.



На втором этапе с целью обеспечения соответствия площадей деталей $2DK$ и $3DK$ и перераспределения деформации по поверхности $2DK$ применяется физическое моделирование. Суть его заключается в придании каждому узлу триангуляционной сети свойств материальной точки с массой, равной 1, а каждому ребру – упругой связи, стремящейся привести его длину $\overline{r_{2D^1}}$ к исходной длине $\overline{r_{3D}}$. Процесс включает две временные стадии: рывок для выведения системы материальных точек из равновесия (результатом является деталь $2DK^p$) и переход детали к устойчивому равновесному состоянию (деталь $2DK$).



Рывок приводит в движение систему материальных точек в плоскости развертывания, увеличивая расстояние между ними в k_p раз:

$$\overline{r_{2D^p}} = k_p \overline{r_{2D^1}}. \quad (1)$$

При таком преобразовании площадь и периметр детали принимают значения, большие, чем у исходной трехмерной детали.

Для каждой материальной точки можно вычислить равнодействующую силу по положению соседних материальных точек:

$$\overline{F_{ij}} = \sum_{j=1}^n \overline{F_i}, \quad (2)$$

где n – количество соседних точек. После окончания рывка большинство сил взаимодействия направлены к центру детали ($F_{ij} > 0$). Силовое поле по детали распределяется неравномерно, деталь находится в максимально деформированном и неустойчивом состоянии. В ней преобладают деформации растяжения, а на участках, исходная деформация сжатия которых не компенсируется в результате рывка ($k_d < k_p$, $\overline{r_{2D^1}} \cdot k_p < \overline{r_{3D}}$), – сжатия. Пример преобразования детали развертки $3DK$ ($2DK^1$) приведен на рисунке 4: A_2OB_2 – результат воздействия рывка ($2DK^p$).

Переход к устойчивому равновесному состоянию (итерация) с физической точки зрения состоит в стремлении системы принять состояние с минимальным силовым полем по детали, уменьшив воздействие рывка. Процесс получения $2DK$ осуществляется в динамике, силовое поле по детали изменяется с течением времени:

$$\overline{F_{ij}^t} = k_{\text{ж}} (\overline{r_{2D^t}^{ij}} - \overline{r_{3D}^{ij}}), \quad (3)$$

где $k_{\text{ж}}$ – коэффициент жесткости связи между точками i и j ; $\overline{r_{2D^t}^{ij}}$ – расстояние между точками i и j в момент времени t .

В общем случае $k_{\text{ж}}$ распределен по триангуляционной сети равномерно. Для большего приближения к свойствам текстильных материалов коэффициент жесткости упругой связи в ребрах, направленных по разным осям, может быть различным и соответствовать жесткости материала по основе (петельным столбикам) $k_{\text{ж}}^o$ и утку (петельным рядам) $k_{\text{ж}}^y$, то есть $k_{\text{ж}}(k_{\text{ж}}^o, k_{\text{ж}}^y)$. Значение $k_{\text{ж}}$ определяется из требуемого соотношения площадей и длин (ширин) деталей $2DK$ и $3DK$. Для ткани стремятся выполнить условие $S_{2D} \approx S_{3D}$, для трикотажных полотен при выборе $k_{\text{ж}}^o$ и $k_{\text{ж}}^y$ учитывают растяжимость в разных направлениях и планируемую величину сокращения (увеличения) размеров вдоль петельных рядов (столбиков).

Сила взаимодействия материальных точек в момент окончания рывка ($t=0$) может быть определена следующим образом:

$$\overline{F_{ij}^{t=0}} = k_{\text{ж}} (\overline{r_{2D^1}^{ij}} - \overline{r_{3D}^{ij}}), \quad (4)$$

а в момент перехода к равновесному состоянию ($t=T, \sum F_{ij} \rightarrow 0$) –

$$\overline{F_{ij}^{t=T}} = k_{\text{ж}} (\overline{r_{2D}^{ij}} - \overline{r_{3D}^{ij}}). \quad (5)$$

Согласно формуле (5), конечное положение точки не зависит от значения рывка.

Положение материальных точек на каждом временном слое определяем с использованием метода численного интегрирования Верле [11]. Процесс продолжается до достижения минимальной суммарной силы по детали.

Псевдокод алгоритма:

пока (суммарная деформация > порога деформации) или изменение_деформации < порога изменения деформации)

для каждого ребра r_i
найти разницу δ текущей длины ребра и начальной (3D)

найти направление изменения $r_{i\text{norm}} = \text{normalize}(v2 - v1)$

для каждой вершины r_i ($v1, v2$)

$v1_{t+1} = v1_t + 0.5 * \delta * k_{ж}(r_{i\text{norm}})$

$v2_{t+1} = v2_t - 0.5 * \delta * k_{ж}(r_{i\text{norm}})$

Процесс окончания разворачивания характеризуется выражением $\sum F_{ij}^{t=T} = \min$, что означает максимально возможное приближение геометрических характеристик 3DK и 2DK. Для разворачивающихся участков $\sum F_{ij}^{t=T} = 0$, что означает $l_{2D} = l_{3D}$, $S_{2D} = S_{3D}$. Для неразворачиваемых поверхностей $\sum F_{ij}^{t=T} \neq 0$, то есть процесс остается до конца неуравновешенным. Состояние $\sum F_{ij}^{t=T} = \min$ для таких деталей возможно при наличии компенсирующих друг друга деформаций сжатия и растяжения, что приводит к соответствию площадей исходной и полученной деталей ($S_{2D} - S_{3D} \approx 0$), то есть $k_S \approx 1$.

Ошибки интегрирования могут приводить к движению и повороту всей сети в плоскости развертки. Поэтому вводится шаг компенсации положения детали как целого объекта. Для этого на каждой итерации вычисляются средний поворот (R_{cp}) и смещение (T_{cp}) всех узлов сети относительно начального момента времени t_0 , затем каждая вершина поворачивается на величину R_{cp} и смещается на T_{cp} .

Для разворачивания деталей с участками различной кривизны необходимо многократное повторение рывка и итерации до достижения желаемого результата. Для предотвращения складывания участков сети каждого треугольника сети вводится шаг зеркального отражения вершин относительно противоположащих ребер. Для этого для каждой пары треугольников, смежных с ребром r_i , определяется, лежат ли они по разные стороны от данного ребра; если это условие не выполняется, то одна из вершин симметрично отражается относительно оси, задаваемой ребром r_i .

Пример преобразования детали развертки 3DK (2DK^p) приведен на рисунке 5: фигура A_3OB_3 – результат физического моделирования (окончательная развертка трехмерной конструкции 2DK). В ней на участках выпуклости/вогнутости преобладают деформации сжатия, на границах – растяже-

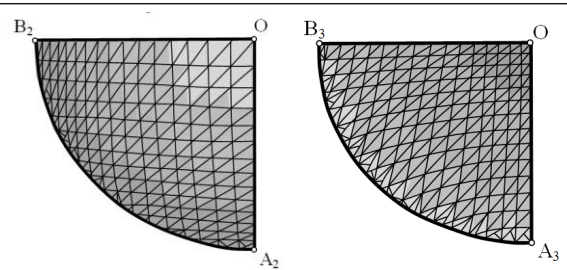


Рис. 4. Результат второго этапа разворачивания (преобразование детали во время рывка)

Fig. 4. The result of the second stage of deployment (conversion of a detail during jerk)

Рис. 5. Окончательный результат разворачивания (устойчивое равновесное состояние)

Fig. 5. The final result of the deployment (stable equilibrium state)

ния, а на переходных участках – устойчивое недеформированное состояние.

Результаты

Представленный метод разворачивания апробирован в специальной компьютерной программе. Для оценки адекватности предложенных геометрических и физических методов прослежено изменение параметров разверток шаровых сегментов полусферы радиусом 70 мм (рис. 6) по стадиям преобразования (рис. 7 – рывок, рис. 8 – окончательная развертка).

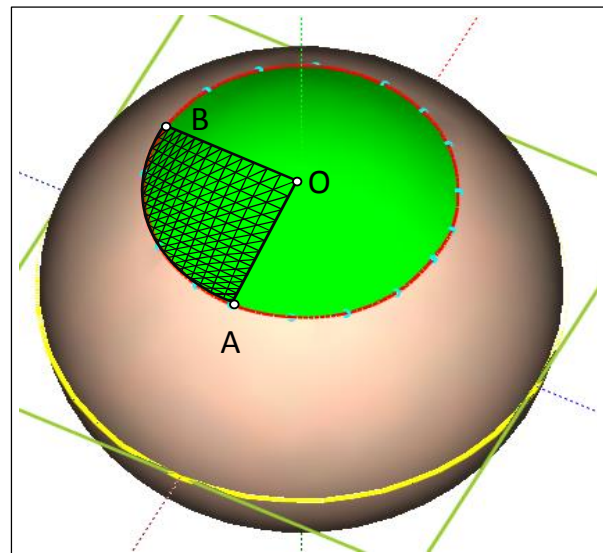


Рис. 6. Исходная трехмерная поверхность детали. Пример реализации в программе

Fig. 6. The original three-dimensional surface of a detail. An example of implementation in the program

В качестве критериев оценки соответствия размеров элементарных ячеек и деталей в исходной трехмерной конструкции (3DK) и в развертке

(2DK) использовали относительные характеристики типа

$$k_d^N = \frac{N_{2D}}{N_{3D}}, \quad (6)$$

где k_d – коэффициент деформации структуры; N – оцениваемый параметр (длина l , периметр P , площадь S); N_{3D} и N_{2D} – величина параметра, измеренного в исходной трехмерной конструкции (3DK) и в ее развертке (2DK) соответственно.

от 10 до 60 мм относительно экватора (большого диаметра), и их разверток представлены в таблице.

Обсуждение результатов

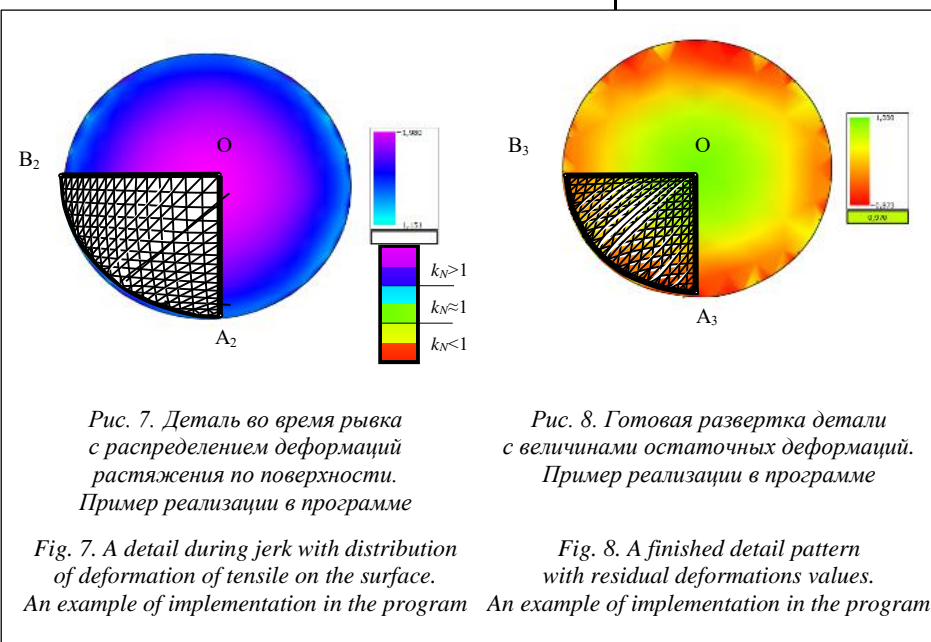
На первом этапе в результате геометрического преобразования периметр исходной детали изменяется незначительно: $k_d^P \approx 1$, $(P_{2D} - P_{3D}) \approx 0$, а площадь сокращается: $k_d^S < 1$, $(S_{2D} - S_{3D}) < 0$. Величина деформации увеличивается с увеличением размера детали.

То есть развертка, получаемая в результате первого этапа преобразования, практически непригодна для раскроя детали из ткани, так как имеет недостаточную площадь.

В результате рывка площадь и периметр увеличиваются в 1,5–2 раза.

На конечном этапе физического моделирования площадь детали приводится к исходному значению: $k_d^S \approx 1$, $(S_{2D} - S_{3D}) \approx 0$. При этом $k_P > 1$, $P_{2D} \geq P_{3D}$, то есть периметр развертки всегда не

меньше 3DK. Также может наблюдаться несоответствие основных габаритных размеров деталей $l_{2D} \leq l_{3D}$, что говорит о необходимости применения принудительного формования либо использования свойств материалов для получения формы изделия. Величина деформации определяет корректность выбора материалов для модели. Соблюдение деформации двумерной детали при изготовлении из-



При $k_d \approx 1$ имеет место соответствие геометрических параметров трехмерной конструкции и полученной двумерной детали ($(l_{3D} - l_{2D}) \approx 0$, $(P_{2D} - P_{3D}) \approx 0$, $(S_{2D} - S_{3D}) \approx 0$). При $k_d < 1$ наблюдается деформация сжатия развертки, а при $k_d > 1$ – растяжение.

Результаты измерения параметров исходных сегментов полусферы, расположенных на высоте

Изменение параметров разверток сегментов полусферы радиусом 70 мм по стадиям преобразования

Parameter variations of semi-sphere segments' development with in radius 70 mm by transforming stages

Высота сегмента относительно экватора, мм	Параметр	3DK	2DK ¹	k_d^1	2DK ^P	k_d^P	2DK	k_d
9,80	S, мм ²	26340,3	15036,6	0,57	60035,5	2,28	25772,3	0,98
	P, мм	434,58	434,58	1,00	869,07	2,00	577,53	1,33
19,60	S, мм ²	22026,1	14124,9	0,64	56395,2	2,56	21714,9	0,99
	P, мм	421,2	421,14	1,00	842,21	2,00	528,80	1,26
30,80	S, мм ²	17096,6	12328,8	0,72	22774,1	1,33	16980,7	0,99
	P, мм	393,68	393,51	1,00	535,17	1,36	467,10	1,19
40,60	S, мм ²	12803,1	10135,8	0,79	15875,8	1,24	12720,3	0,99
	P, мм	356,96	356,74	1,00	449,77	1,26	403,57	1,13
50,40	S, мм ²	8527,24	7338,13	0,86	9795,24	1,15	8561,35	1,00
	P, мм	303,64	303,71	1,00	354,32	1,17	329,63	1,09
60,20	S, мм ²	4231,26	3938,75	0,93	4375,12	1,03	4240,79	1,00
	P, мм	222,56	222,58	1,00	234,13	1,05	231,27	1,04

деля является обязательным условием создания трехмерной формы без образования конструктивных дефектов.

Для оценки возможности деформирования используемых материалов в процессе одевания изделия либо в процессе принудительного формования рекомендуется использовать следующие характеристики: относительное удлинение материала по нити основы и утка, способность к суживанию и оттягиванию по разным направлениям. Корректность выбора способа формообразования может быть оценена при сравнении коэффициентов деформации деталей развертки с предельно допустимыми значениями следующих показателей для выбранных материалов: растяжимость, способность к формованию, одевающая способность [12]. Для приближения условий деформирования плоских образцов к реальному процессу образования объемной детали данные показатели должны быть измерены с учетом предполагаемых формообразующих деформаций, например аналогично [13].

На основании изложенного сделаем следующие выводы. Предложенный авторами метод получения развертки позволяет учитывать форму объемной поверхности и свойства материалов на участках повторения изделием формы фигуры. При этом конечное положение и геометрические параметры детали развертки в большей степени зависят от кривизны оболочки и положения плоскости развертывания.

Метод реализован в САПР плотнооблегающих изделий BustCAD (г. Иваново, Россия) и на практике показал хорошие результаты.

Исследования выполнены при финансовой поддержке Минобрнауки России: задание № 11.1898.2014/К на выполнение научно-исследовательской работы в рамках проектной части государственного задания в сфере научной деятельности.

Литература

1. Фроловский В.Д. Математические модели и оптимизационные методы автоматизированного проектирования и подготовки производства корпусных изделий // Сб. науч. тр. НГТУ. 1997. Вып. 1 (6). С. 71–78.
2. Yuwei Meng, Charlie C.L. Wang, Xiaogang Jin. Flexible shape control for automatic resizing of apparel products. *Computer-Aided Design*, 2012, vol. 44, no. 1, pp. 68–76.
3. Зверева С.А. Согласованные конструктивные аналитические и компьютерные модели: автореф. дисс. канд. техн. наук. Донецк, 2000. 20 с.
4. Курышева В.Н. Разработка эмпирического метода одевания трехмерной поверхности тканью: дис. канд. техн. наук. М., 2006. 214 с.
5. Frolovsky V.D. Development and research of algorithm of construction unfolding surfaces. 2006. URL: http://graphicon.ru/html/2006/proceedings/papers/fr08_21_PavenkoFrolovsky.pdf (дата обращения: 28.09.2016).
6. Jituo L., Dongliang Z., Guondong L., Yanying P., Xing W., Yoshiyulci S. Flattening triangulated surface using a mass-spring model. *Advance Manufacturing Technology*, 2005, vol. 25, pp. 108–117.
7. Кирсанова Е.А. Методологические основы оценки и прогнозирования свойств текстильных материалов для создания одежды заданной формы: дис. докт. техн. наук. М., 2003. 380 с.
8. Charlie C.L. Wang, Shana S.-F. Smith, Matthew Ming-Fai Yuen. Surface flattening based on energy model. *Computer-Aided Design*, 2002, vol. 34, no. 11, pp. 823–833.
9. Charlie C.L. Wang, Kai Tang. Pattern computation for compression garment by a physical/geometric approach. *Computer-Aided Design*, 2010, vol. 42, no. 1, pp. 78–86.
10. Горелова А.Е., Корнилова Н.Л. Теоретическое обоснование математического описания опорной поверхности верхней плечевой одежды // Изв. вузов. Технология текстильной промышленности. 2006. Вып. 1 (288). С. 83–85.
11. Ландовский В.В. Методы интегрирования дифференциальных уравнений в задаче моделирования поведения ткани методом частиц. 2005. URL: <http://graphicon.ru/html/2005/proceedings/papers/Landovskiy.pdf> (дата обращения: 28.09.2016).
12. Корнилова Н.Л., Горелова А.Е. К вопросу учета формовочных свойств материалов при проектировании одежды // Изв. вузов. Технология текстильной промышленности. 2007. Вып. 6С. С. 85–88.
13. Rief S., Glatt E., Laourine E., Aibibu D., Cherif C., Wiegmann A. Modeling and CFD-simulation of woven textiles to determine permeability and retention properties. *AUTEX Research Jour.*, 2011, vol. 11, no. 3, pp. 78–83.

Software & Systems

DOI: 10.15827/0236-235X.116.094-100

Received 29.09.16

2016, vol. 29, no. 4, pp. 94–100

THE METHOD OF OBTAINING A SCAN OF CLOTHING DETAILS CONSIDERING DEFORMATIONAL APTITUDE OF A MATERIAL

A.E. Gorelova¹, Ph.D. (Engineering), Associate Professor, GorelovaAnn@mail.ru

N.L. Kornilova¹, Dr.Sc. (Engineering), Associate Professor, Head of the Engineering Center, nkorn@mail.ru

Z.R. Grigoreva¹, Postgraduate Student, zarema_grigoreva@inbox.ru

D.A. Vasilev², Leading Programmer, vasdmec@gmail.com

L.A. Kornilova^{3,4}, Graduate Student, Laboratory Assistant, lmila.korn@gmail.com

¹ Ivanovo State Polytechnic University, 8 Marta St. 20, Ivanovo, 153037, Russian Federation

² IIT Consulting, Ltd, Suvarov St. 39-524, Ivanovo, 153012, Russian Federation

³ The National Research University of Information Technologies, Mechanics and Optics, Kronverksky Ave. 49, St. Petersburg, 197101, Russian Federation

⁴ Peter the Great St. Petersburg Polytechnic University, Politekhnikeskaya St. 29, St. Petersburg, 195251, Russian Federation

Abstract. The research presents an algorithm for obtaining patterns of garment (3DK) using a phased conversion of triangulation part network of three-dimensional product construction using methods of geometric surface transformation and physical modeling of material behavior. On the first stage a 3D cover transforms into a flat view by projecting network cell edges on a deployment plane. Next, we use physical modeling, which consists in giving to each node of triangulation network the properties of a material point with a mass equal to 1, and giving to each edge an elastic connection, which seeks to result its length into an original length. The process involves two time steps: a spurt to unbalance the system of material points and transition of a detail to a sustainable equilibrium state (iteration). Then the process continues until reaching the minimum of total force on a detail. It means the maximum possible approximation of geometrical characteristics 3DK and a pattern.

Quality evaluation of the algorithm is implemented by constructing scans of orbicular hemisphere segments. The paper shows that the patterns obtained in the first stage of conversion are practically unsuitable for cutting details of fabric, because it has an insufficient area. As a result of physical modeling, the details area reduces to its original value, herewith the patterns perimeter is always at least 3DK. In addition, there can be seen discrepancy between the main overall dimensions of details, which indicates that there is a need to use forced molding or deformation of materials properties to obtain the garment shape. The magnitude of the deformation determines the correct choice of materials for a model.

The method is implemented in CAD of tight-fitting products BustCAD (Ivanovo, Russia) and showed good practical results.

Keywords: apparel CAD, geometric modeling, physical modeling, scanning three-dimensional surface, surface portion deformation, material properties.

Acknowledgements. The research is carried out with the financial support of the Education and Science Ministry of Russia: the task no. 11.1898.2014/K for execution a research work within the framework of the project of the state task in a scientific research area.

References

1. Frolovsky V.D. Mathematical models and optimization methods for computer-aided design and manufacture of case products. *Sbornik nauchnykh trudov NGTU* [Proc. NSTU]. Novosibirsk, 1997, vol. 1, no. 6, pp. 71–78 (in Russ.).
2. Meng Yu., Wang Ch.C.L., Jin X. Flexible shape control for automatic resizing of apparel products. *Computer-Aided Design*. 2012, vol. 44, no. 1, pp. 68–76.
3. Zvereva S.A. *Soglasovannye konstruktivnye analiticheskie i kompyuternye modeli* [Concerted constructive analytical and computer models]. MS Thesis. Donetsk, 2000, 20 p.
4. Kuryshva V.N. *Razrabotka empiricheskogo metoda odevaniya trekhmernoy poverkhnosti tkanyu* [Development of empirical method of dressing a three-dimensional surface into fabric]. MS Thesis. Moscow, 2006, 214 p.
5. Frolovsky V.D. *Development and research of algorithm of construction unfolding surfaces*. 2006. Available at: http://graphicon.ru/html/2006/proceedings/papers/fi08_21_PavenkoFrolovsky.pdf (accessed September 28, 2016).
6. Jituo L., Dongliang Z., Guondong L., Yanying P., Xing W., Yoshiyulci S. Flattening triangulated surface using a mass-spring model. *Advance Manufacturing Technology*. 2005, vol. 25, pp. 108–117.
7. Kirsanova E.A. *Metodologicheskie osnovy otsenki i prognozirovaniya svoystv tekstilnykh materialov dlya sozdaniya odezhdy zadannoy formy* [Methodological basics for assessing and prediction of textile material properties to create the desired shape of clothes]. Ph.D. Thesis. Moscow, 2003, 380 p.
8. Wang Ch.C.L., Smith Sh.Sh.-F., Yuen M.M.-F. Surface flattening based on energy model. *Computer-Aided Design*. 2002, vol. 34, no. 11, pp. 823–833.
9. Wang Ch.C.L., Tang K. Pattern computation for compression garment by a physical/geometric approach. *Computer-Aided Design*. 2010, vol. 42, no. 1, pp. 78–86.
10. Gorelova A.E., Kornilova N.L. Theoretical substantiation of a mathematical description of supporting surface of a brachial outerwear. *Izv. vuzov. Tekhnologiya legkoy promyshlennosti* [News of Higher Educational Institutions. Light Industry Technology]. 2006, vol. 1, no. 288, pp. 83–85 (in Russ.).
11. Landovsky V.V. *Metody integrirvaniya differentsialnykh uravneniy v zadache modelirovaniya povedeniya tkani metodom chastits* [Integration Methods of Differential Equations in the Problem of Modeling a Fabric Behaviour Using the Method of Particles]. 2005. Available at: <http://graphicon.ru/html/2005/proceedings/papers/Landovskiy.pdf> (accessed September 28, 2016).
12. Kornilova N.L., Gorelova A.E. On the question of accounting the molding properties of material in apparel design. *Izv. vuzov. Tekhnologiya legkoy promyshlennosti* [News of Higher Educational Institutions. Light Industry Technology]. 2008, vol. 6S, pp. 85–88.
13. Rief S., Glatt E., Laourine E., Aibibu D., Cherif C., Wiegemann A. *Modeling and CFD-simulation of woven textiles to determine permeability and retention properties*. AUTEX Research Jour. 2011, vol. 11, no. 3, pp. 78–83.

УДК 004.896

DOI: 10.15827/0236-235X.116.101-106

Дата подачи статьи: 14.06.16

2016. Т. 29. № 4. С. 101–106

ПРИМЕНЕНИЕ СИСТЕМЫ KLEE ДЛЯ АВТОМАТИЗАЦИИ ТЕСТИРОВАНИЯ ПРОГРАММ НА ЯЗЫКАХ C/C++

*А.Г. Зыков, к.т.н., zikov_a_g@mail.ru;**И.В. Кочетков, аспирант, melmacson@gmail.com;**В.И. Поляков, к.т.н., доцент, v_i_polyakov@mail.ru**(Санкт-Петербургский национальный исследовательский университет
информационных технологий, механики и оптики (Университет ИТМО),
Кронверкский просп., 49, г. Санкт-Петербург, 197101, Россия)*

В работе рассматривается верификация вычислительных процессов, в частности, автоматизация тестирования функциональных ошибок программных продуктов, которые крайне сложно исследовать и воспроизводить без непосредственного исполнения кодовых фрагментов. Основной идеей является применение программ или инструментов символического исполнения. В частности, подробно рассматривается система KLEE, представляющая собой символическую виртуальную машину, эмулирующую окружение. В ней параллельно выполняются символические процессы, каждый из которых – один из путей в исследуемой программе. Система построена на анализе LLVM байт-кода программы с применением STP-решателя для предикатов. Рассматриваются ее архитектура, состав компонент, принципы работы, базовые возможности, способ моделирования окружения, пример работы на основе тестирования утилиты `tr` системы MINIX и др.

Целью исследования являются изучение общих возможностей систем символического исполнения на примере набора KLEE и их применение для решения задачи автоматизации тестирования. Актуальность данной проблемы высока в связи с постоянно растущей сложностью ПО, которая ведет к увеличению сложности, длительности и, главное, стоимости тестирования и верификации программных продуктов. Новизна данного исследования заключается в том, что на основании детального изучения принципов функционирования системы рассмотрены альтернативные способы применения программ символического исполнения. К таким способам относятся тестирование с помощью эталона, поиск решения, восстановление схемы алгоритма программы. В качестве результата работы приведена статистика тестирования набора программ пакета COREUTILS 6.11.

Ключевые слова: автоматизация, тестирование, символическое исполнение, KLEE.

В тестировании можно выделить классы потенциальных проблем, которые невозможно или крайне трудно исследовать без исполнения части кода [1]. К таким проблемам можно отнести ошибки функциональной корректности алгоритма и др. В подобных ситуациях применение ручного или случайного (на основе случайно сгенерированных наборов входных данных) тестирования крайне неэффективно. Многие работы по данной теме рассматривают применение систем символического исполнения для анализа программы, поиска ошибок и автоматической генерации тестовых данных [2–4].

В данной работе рассмотрена система символического исполнения KLEE [5], которая позиционируется как усовершенствование системы EXE, разработанной в 2006 году. Она позволяет автоматизировать тестирование широкого спектра приложений, в том числе и взаимодействующих с окружением [6].

Принцип работы систем символического исполнения

Основная идея систем символического исполнения сводится к тому, что программа выполняется с использованием символических переменных, которые могут принимать любое значение, а не на разрабо-

танных вручную или случайно сгенерированных входных данных. Каждая инструкция в программе, манипулирующая конкретными данными (например $x = y * 2$), заменяется на ту, которая использует символические значения и переменные (например $x = \mu * 2$).

В ситуациях, когда программа достигает операторов ветвления, система отслеживает обе ветви, каждой из которых сопоставляет набор ограничений, запоминающийся на время исполнения этой ветви, называемый *условием пути*. Если ветвь исполнения завершается или генерирует исключение, на основе сохраненного набора условий пути можно сгенерировать тестовый набор, заменяя символические переменные на конкретные значения.

Таким образом, если учитывать постоянство исходной программы, ее выполнение на сгенерированном тестовом наборе должно привести к появлению той же ошибки, найденной при символическом исполнении.

Системы символического исполнения позволяют достигать высокого тестового покрытия на некоторых программах, однако для большинства программ [4] существуют проблемы, затрудняющие достижение подобного результата, а именно:

- экспоненциальный рост количества путей в коде программы;

- тестирование кода, взаимодействующего с окружением (ОС, сеть, пользователь);
- многопоточность и параллельные вычисления.

Принцип работы системы KLEE

Принцип работы системы KLEE проще описать на примере тестирования конкретной программы. Возьмем утилиту `tr` операционной системы MINIX. Несмотря на малый размер исходного кода (169 строк кода / 83 исполняемых), она отражает две типичные проблемы при тестировании.

- Сложность. Код программы реализует задачу трансляции и манипуляции символами, поступающими на вход. Все это скрывается под непрозрачным кодом разбора входных аргументов, сложными условиями ветвления и запутанным потоком управления.

- Зависимость от окружения. Исполнение в основном зависит от входных значений аргументов, поступающих с системного потока чтения. Аргументы управляют тем, какую функцию будет выполнять программа и в какую сторону (по ветке IF или ELSE) будет происходить ветвление в ней. Также программа сильно зависит от самой возможности чтения из системного потока, по которому могут поступать как валидные, так и невалидные (и даже небезопасные) данные, которые должны быть обработаны корректно и без ошибок. Поэтому возникает большое количество тестовых сценариев, связанных с граничными условиями и ключевыми значениями.

KLEE как представитель класса систем симульного исполнения работает по следующему алгоритму:

- заменяет все операнды на символьные;
- проходит все ветви программы и все выражения с помощью символьных переменных;
- вычисляет условие пути (значения переменных, необходимые для прохождения от точки входа до текущей позиции), которое привело к окончанию выполнения ветви (программы) или к ошибке; на основе этого условия формируется тестовый набор.

После этого можно использовать сгенерированные тестовые наборы для повторного автоматизированного тестирования программы, то есть, если та или иная ошибка в программе будет исправлена, выполнение полученного ранее теста не должно приводить к ее генерации.

Рассмотрим подробнее пример теста, который приводит к генерации ошибки:

```

1: void expand(char *arg, unsigned char *buffer) {      8
2:   int i, ac;                                          9
3:   while (*arg) {                                     10*
4:     if (*arg == '\\') {                               11*
5:       arg++;
6:       i = ac = 0;

```

```

7:   if (*arg >= '0' && *arg <= '7') {
8:     do {
9:       ac = (ac << 3) + *arg++; - '0';
10:      i++;
11:    } while (i < 4 && *arg <= '7');
12:    *buffer++ = ac;
13:  } else if (*arg != '0')
14:    *buffer++ = *arg++;
15:  } else if (*arg == '[') {                             12*
16:    arg++;                                              13
17:    i = *arg++;                                         14
18:    if (*arg++ != '-') {                               15!
19:      *buffer++ = '[';
20:      arg -= 2;
21:      continue;
22:    }
23:    ac = *arg++;
24:    while (i <= ac) *buffer++ = i++;
25:    arg++; /*Skip '[' */
26:  } else
27:    *buffer++ = *arg++;
28:  }
29: }
30: ...
31: int main(int argc, char* argv[ ]) {                  1
32:   int index = 1;                                      2
33:   if (argc > 1 && argv[index][0] == '-') {            3*
34:     ...                                              4
35:   }                                                  5
36:   ...                                              6
37:   expand(argv[index++], index);                      7
38:   ...
39: }

```

Строка 18 генерирует ошибку переполнения буфера на конкретной входной строке (`tr ["" ""]`).

KLEE проходит исходный код утилиты следующим образом.

1. Сначала KLEE собирает символьные аргументы командной строки, которые не содержат никаких ограничений, кроме нуль-терминации строк, и вызывает метод `main`.

2. Достигнув ветвления на строке 33 (`argc > 1`), KLEE вызывает STP-решатель, чтобы узнать, выполнение каких ветвей оператора возможно. Для данного ветвления существуют оба пути, поэтому KLEE разветвляет исполнение и следует по обоим путям, добавляя к соответствующим условиям путей новое ограничение (`argc > 1` для истинного, `argc ≤ 1` для ложного).

3. Так как количество активных путей больше одного, KLEE выбирает один для исполнения с помощью стратегии отбора. Будем считать, что мы следуем по пути, который приводит к генерации ошибок. Таким образом, по ходу исполнения KLEE добавляет дальнейшие ограничения на содержимое массива `arg` и разветвляется пять раз (на помеченных "*" строках): дважды на строке 33 и по разу на строках 3, 4, 15.

4. На каждой опасной операции (например разыменовании указателя) KLEE проверяет потенциальные причины для генерации исключения в условиях текущего пути. Ошибки не обнаруживаются до строки 18. На этой позиции происходит исключение из-за существования такого входного

значения, которое приводит к двойному инкременту указателя `arg` без проверки на окончание строки. Поэтому последний инкремент пропускает символ `'\0'` и начинает указывать на невалидную память.

5. KLEE генерирует конкретные значения для переменных `args` и `argv`, позволяющие при воспроизведении получить идентичное поведение. Чтобы продолжить исполнение текущего пути, KLEE добавляет ограничение, которое не приводит к генерации ошибки.

Архитектура

На высоком уровне KLEE работает как гибрид между операционной системой для символьных процессов и интерпретатором. Каждый символьный процесс имеет собственные регистровый файл, стек, кучу, программный счетчик и условия пути. Все тестируемые программы должны быть скомпилированы в LLVM-ассемблерный язык, RISC-подобный виртуальный набор инструкций. KLEE интерпретирует этот набор инструкций без каких-либо преобразований или упрощений [5].

KLEE оперирует множеством состояний процесса одновременно подобно диспетчеру процессов. В каждый момент времени выбирается одно из состояний, которое выполняет всего лишь одну инструкцию. Обработка продолжается до того момента, пока либо не останется возможных состояний, либо истечет таймаут, указанный пользователем.

В отличие от обычных процессов все хранилища данных для состояний (регистры, стек и т.п.) содержат не сырые данные, а выражения. Например, при вычислении операции

`%dst = sub i32%src0, %src1`

в регистр `%dst` записывается не результат `%src0 - %src1`, а выражение `sub(%src0, %src1)`. Для достижения большей эффективности при вычислении результата проверяется, являются ли операнды константами; если да, то в регистр записывается не выражение, а его результат.

При достижении оператора ветвления система проверяет, какие ветви возможны. Для этого используется STP-решатель. В случаях, когда возможно выполнение обеих ветвей, текущее состояние клонируется, к каждой из ветвей добавляется соответствующее ограничение, после чего анализ продолжается в обоих направлениях.

Каждая потенциально опасная инструкция проверяет, возможны ли ситуации, при которых входные значения приводят к ошибке. Например, при делении происходит проверка деления на нуль, при сложении и вычитании – проверка на переполнение и т.п. Если KLEE детектирует потенциальную ошибку, генерируется тестовый сценарий и исполнение текущего состояния завершается.

Операторы чтения и записи также являются потенциально опасными, например, при выполнении может произойти ошибка выхода за границы памяти. Память в KLEE представляется в виде набора различных массивов под каждый объект в проверяемом коде. Это сделано потому, что из-за бесконечного количества вариаций невозможно решить задачу поиска ограничений в случае, если представлять память единой для всего символьного процесса.

Многие операции (например проверка границ) требуют наличия специальной информации об объекте. Если указатель может ссылаться на N различных объектов, текущее состояние KLEE копируется N раз. В каждом состоянии разименованный указатель ссылается лишь на один конкретный объект.

При работе KLEE активно взаимодействует с модулями STP и SAT.

STP (SMT) – решатель задачи выполнимости формул в теориях. STP оперирует выражениями ($a > 1$, $x! = 2$ и т.п.) и взаимосвязью между ними. Результатом его работы является утверждение о выполнимости/невыполнимости входной теоремы (SAT/UNSAT). При работе все входные выражения в конечном итоге преобразуются в единую КНФ-форму, которая поступает на вход уже другому решателю, SAT [7].

SAT – решатель задачи о выполнимости булевой функции. Данный компонент оперирует набором переменных, скобок и операций И, ИЛИ и НЕ. Результатом его работы являются утверждение о выполнимости/невыполнимости входной булевой функции и, в случае выполнимости функции, битовый вектор, каждый из элементов которого соответствует значению одной из входных переменных, при котором функция принимает значение TRUE. Задача о выполнимости булевой функции является NP-полной [8].

Моделирование окружения

Известно, что при любой автоматизации тестирования (приемочного, модульного и т.п.) всегда встает проблема работы с окружением. Так как сама программа может работать на различных операционных системах, то и тестироваться она должна на различных операционных системах (например, запускать тесты C на ОС Linux или ОС Windows). При этом взаимодействие с окружением – всегда случайный фактор, который невозможно контролировать и который может приводить к ложным срабатываниям (например, ветвление в зависимости от значения системных переменных), а значит, невозможно гарантировать идентичный результат воспроизведения одного и того же теста. Следовательно, требуется моделировать окружение либо пропускать зависящие от него участки кода.

В системе KLEE все вызовы (open, read, write, stat, ioctl и др.), оперирующие окружением, взаимодействуют не с реальной операционной системой, а с ее моделью. Таким образом, символьный процесс работает в «модели» окружения, с которой он может взаимодействовать как с обычной, без необходимости каким-либо образом изменять исходный код. Все модели написаны на языке C и предоставляют возможность модификации и расширения по желанию разработчика.

Применение системы KLEE

Базовое предназначение системы KLEE – тестирование исходного кода программы на функциональные недостатки и генерация тестовых сценариев, покрывающих все пути активации программы. Однако возможности системы позволяют на ее основе решать и другие задачи, большинство из которых можно решать «из коробки», без внесения дополнительных модификаций. Некоторые из этих возможностей можно реализовать на большинстве инструментов символьного исполнения, так как в их основе лежат общие принципы работы. Перечислим возможные применения системы KLEE.

1. Функциональное тестирование исходного кода программы (описано выше).

2. Приемочное тестирование с помощью эталона.

- С помощью тестовых сценариев. Данный метод основан на том, что одни и те же тестовые сценарии можно воспроизводить на различных исходных файлах при соблюдении наименований входных переменных. При наличии одной эталонной программы (например образец решения лабораторной работы), которая решает известную задачу, сгенерированные по этой программе тесты можно считать набором приемочных тестов (test suite). Данный набор можно использовать для проверки программы-подобия (решения студентами лабораторной работы) на соответствие ожидаемым результатам. Таким образом, если программа-подобие генерирует исключения или дает неверный результат, можно считать ее некорректной.

- С помощью оператора Assert. Пример двух различных вариантов одной функции:

```
1: unsigned mod_opt(unsigned x, unsigned y) {
2:   if (y & ?) == y)
3:     return x & (y?1);
4:   else
5:     return x % y;
6: }
7: unsigned mod(unsigned x, unsigned y) {
8:   return x % y;
9: }
10: int main() {
11:   unsigned x, y;
12:   make_symbolpc(&x, sizeof(x));
```

```
13:   make_symbolpc(&y, sizeof(y));
14:   assert(mod(x, y) == mod_opt(x, y));
15:   return 0;
16: }
```

На данном примере можно убедиться в эквивалентности двух функций во всем диапазоне входных значений ($y! = 0$). Решая задачу на невыполнение условия в Assert, KLEE на основе перебора всех возможных путей приходит к выводу об эквивалентности двух функций на всем диапазоне значений.

3. Поиск решения. Данный метод основан на том, что задача системы KLEE – пройти все возможные пути активации программы. Поэтому, основываясь на этом принципе, для поиска решения математической или логической задачи можно написать простейшую тестовую программу, содержащую в себе данный предикат. При обработке системой KLEE будут сгенерированы тестовые сценарии, которые удовлетворят данный предикат, то есть будет найдено решение, удовлетворяющее поставленной задаче. Рассмотрим пример для простейшего уравнения:

```
1: if (x^2 + 4x + 4 == 0) {
2:   return true;
3: } else {
4:   return false;
5: }
```

Система KLEE сгенерирует набор, в котором $x = 2$, что является одним из решений уравнения. К задачам, решение которых можно найти с помощью символьного исполнения, относятся поиск пути в лабиринте и другие. Однако для них требуются более сложные тестовые программы.

4. Восстановление схемы алгоритма готовой программы. Данный метод основан на том, что система KLEE проходит и «исполняет» каждую инструкцию в программе, следуя от точки входа до останова, если таковой имеется. Поэтому, если требуется проанализировать алгоритм, заложенный в некотором исходном коде, и существует ряд препятствий (нет знания языка, исходный код крайне сложен или обфусцирован), можно использовать KLEE в качестве анализатора исходного кода, трассируя и преобразовывая его в некоторый универсальный алгоритмический язык. Этот результат можно визуализировать в классическую схему алгоритма, понятную без специальных знаний. Однако данный метод требует доработки системы, так как подобная функциональность не предоставляется «из коробки».

Результаты тестирования

Для получения реальных результатов тестирования авторы проанализировали весь набор программ пакета COREUTILS 6.11. Среднее покрытие

кода составило 94 %. Всего программа сгенерировала 3 321 набор входных данных, позволяющих покрыть весь указанный процент кода. Также было найдено 10 уникальных ошибок, признанных разработчиками пакета как реальные дефекты в программах, что является очень хорошим достижением, так как этот набор программ разрабатывается более 20 лет и большинство ошибок было устранено.

Ограничения

Применение системы KLEE имеет следующие ограничения:

- нет поддержки многопоточных приложений, символьных данных с плавающей точкой (ограничение STP), ассемблерных вставок;
- тестируемое приложение должно быть скомпилировано в LLVM-байт код (как и все его зависимые библиотеки);
- невозможно тестирование программ под системами семейства Windows.

Сделаем выводы. Система KLEE, рассмотренная в данной статье, является одной из систем символьного исполнения, которые все чаще фигурируют в статьях, так или иначе касающихся процесса автоматизации тестирования. KLEE «из коробки» можно применять не только для автоматизации тестирования ПО, но и для решения нестандартных задач.

Из преимуществ данной системы можно выделить следующие:

- достижение тестового покрытия, близкого к полному (90 %+);
- возможность тестирования программ, работающих с окружением;
- возможность воспроизведения полученных тестовых наборов в автоматическом режиме;
- возможность анализа программ, написанных на любом языке программирования, имеющем LLVM-интерпретатор;
- является свободным ПО.

Однако система имеет ряд недостатков:

- может работать только на системах семейства Linux, так как использует компоненты (STP, SAT, LLVM), которые крайне затруднительно установить на Windows;
- требует инструментировать исходный код тестируемых программ.

Дальнейшие исследования будут направлены на усовершенствование системы KLEE для решения проблемы don't care в сложных предикатах [9], добавление поддержки чисел с плавающей запятой в текущий STP проекта либо интеграцию KLEE с другим STP-решателем, так как это необходимо для тестирования математических программ, реализацию возможности тестирования многопоточных программ, а также на изучение и рассмотрение ближайших аналогов системы KLEE (KLOVER [10], S²E [11, 12], janala2 [13] и др.), анализ их преимуществ, ключевых особенностей, применяемых техник и оптимизаций.

Литература

1. Немолочнов О.Ф., Зыков А.Г., Поляков В.И. и др. Верификация в исследовательских, учебных и промышленных системах // Науч.-технич. вестн. СПбГУ ИТМО. Вып. 11. Актуальные проблемы анализа и синтеза сложных технических систем. 2003. С. 146–151.
2. Zamfir C., Candea G. Execution Synthesis: A technique for automated software debugging. ACM SI-GOPS/EuroSys European Conf. on Comp. Syst., 2010, pp. 321–334.
3. Boonstoppel P., Cadar C., Engler D. RWset: Attacking path explosion in constraint-based test generation. Tools and Algorithms for the Construction and Analysis of Systems (TACAS), 2008, pp. 351–366.
4. Artzi S. Finding bugs in web applications using dynamic test generation and explicit-state model checking. Soft. Eng., IEEE Transactions on 36.4, 2010, pp. 474–494.
5. Документация KLEE. URL: <http://klee.github.io/docs/> (дата обращения: 18.02.2016).
6. Cadar C., Dunbar D., and Engler D.R. KLEE: Unassisted and automatic generation of high-coverage tests for complex systems programs. Proc. 8th USENIX Symposium OSDI, 2008, pp. 209–224.
7. STP constraint solver. URL: <http://stp.github.io/> (дата обращения: 20.02.2016).
8. Домашняя страница MiniSAT. URL: <http://minisat.se/> (дата обращения: 15.02.2016).
9. Немолочнов О.Ф., Зыков А.Г., Поляков В.И. и др. Метод обнаружения недекларированных возможностей и значений DON'T CARE вычислительного процесса // Изв. вузов. Приборостроение. 2009. Т. 52. № 12. С. 32–39.
10. Guodong L., Indradeep G., Sreeranga R. KLOVER: A symbolic execution and automatic test generation tool for C++ programs. Proc. Intern. Conf. CAV 2011, Cliff Lodge, Snowbird, UT, USA, 2011, pp. 609–615.
11. Chipounov V., Kuznetsov V., Candea G. S2E: a platform for in vivo multi-path analysis of software systems. Proc. Intern. Conf. ASPLOS 2011, Newport Beach, CA., 2011, pp. 265–278.
12. Chipounov V., Kuznetsov V., Candea G. The S2E platform: design, implementation, and applications. Jour. ACM TOCS, 2012, vol. 30, no. 1, pp. 1–49.
13. A concolic testing engine for Java. URL: <https://github.com/ksen007/janala2> (дата обращения: 20.02.2016).

KLEE FOR AUTOMATIC TESTING PROGRAMS IN C/C++

A.G. Zykov¹, Ph.D. (Engineering), zykov_a_g@mail.ru

I.V. Kochetkov¹, Postgraduate Student, melmacson@gmail.com

V.I. Polyakov¹, Ph.D. (Engineering), Associate Professor, v_i_polyakov@mail.ru

¹ The National Research University of Information Technologies, Mechanics and Optics, Kronverksky Ave. 49, St. Petersburg, 197101, Russian Federation

Abstract. The article considers the problems of automation of software functional error testing. This kind of bugs is extremely difficult to find and reproduce without code execution. This problem is relevant due to ever-growing software complexity. It leads to increasing duration, complexity and cost of software testing and verification.

The purpose of the research is to analyze symbolic execution (evaluation) system basic features and their application in finding functional bugs and software testing in general. In particular, the article discusses the KLEE system in detail. This system is a symbolic virtual machine with environment emulation that executes parallel symbolic processes. It is based on analyzing a program LLVM byte-code. The article deals with KLEE architecture, components set, operation principles, features and some other aspects.

Novelty of this research is in the consideration of alternative methods of using symbolic execution tools. They are developed based on using their basic features. There is testing with reference program, problem solving (for example, finding a way in a labyrinth) and restoring an algorithm flowchart.

The article presents KLEE testing statistics (code coverage, bugs count, etc.) of COREUTILS 6.11 package, where there were found several serious bugs that had been missing for over ten years.

Keywords: automation, testing, symbolic execution, KLEE.

References

1. Nemolochnov O.F., Zykov A.G., Polyakov V.I. Verification in research, training and industrial systems. *Nauchno-tekhnich. vestnik SPbGU ITMO. Vyp. 11. Aktualnye problemy analiza i sinteza slozhnykh tekhnicheskikh sistem* [Scientific and Technical Journal of Information Technologies, Mechanics and Optics]. St. Petersburg, SPbGU ITMO Publ., 2003, pp. 146–151 (in Russ.).
2. Zamfir C., Candea G. Execution Synthesis: A Technique for Automated Software Debugging. *ACM SI-GOPS/EuroSys European Conf. on Computer Systems*. Paris, France, 2010, pp. 321–334.
3. Boonstoppel P., Cadar C., Engler D. RWset: Attacking path explosion in constraint-based test generation. *Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*. 2008, pp. 351–366.
4. Artzi S. Finding bugs in web applications using dynamic test generation and explicit-state model checking. *IEEE Transactions on Software Engineering*. 2010, vol. 36.4, pp. 474–494.
5. *KLEE Documentation*. Available at: <http://klee.github.io/docs/> (accessed February 18, 2016).
6. Cadar C., Dunbar D., Engler D. R. KLEE: Unassisted and automatic generation of high-coverage tests for complex systems programs. *Proc. 8th USENIX Symp. on Operating Systems Design and Implementation (OSDI)*. 2008, pp. 209–224.
7. *STP constraint solver*. Available at: <http://stp.github.io/> (accessed February 20, 2016).
8. *MiniSAT home page*. Available at: <http://minisat.se/> (accessed February 15, 2016).
9. Nemolochnov O.F., Zykov A.G., Polyakov V.I. Method of detecting undocumented features and DON'T CARE values of a computation process. *Izvestiya vysshikh uchebnykh zavedeny. Priborostroenie* [Journal of Instrument Engineering]. 2009, vol. 52, no. 12, pp. 32–39 (in Russ.).
10. Guodong L., Indradeep G., Sreeranga R. KLOVER: A Symbolic Execution and Automatic Test Generation Tool for C++ Programs. *Proc. Int. Conf. on Computer Aided Verification (CAV 2011)*. Cliff Lodge Publ., Snowbird, UT, USA, 2011, pp. 609–615.
11. Chipounov V., Kuznetsov V., Candea G. S2E: A Platform for In Vivo Multi-Path Analysis of Software Systems. *Proc. Int. Conf. on ASPLOS 2011*. Newport Beach, CA, 2011, pp. 265–278.
12. Chipounov V., Kuznetsov V., Candea G. The S2E Platform: Design, Implementation, and Applications. *Jour. ACM TOCS*. 2012, vol. 30, no. 1, pp. 1–49.
13. *A concolic testing engine for Java*. Available at: <https://github.com/ksen007/janala2> (accessed February 20, 2016).

Примеры библиографического описания статьи

1. Зыков А.Г., Кочетков И.В., Поляков В.И. Применение системы KLEE для автоматизации тестирования программ на языках C/C++ // Программные продукты и системы. 2016. Т. 29. № 4. С. 101–106; DOI: 10.15827/0236-235X.116.101-106.
2. Zykov A.G., Kochetkov I.V., Polyakov V.I. KLEE for automatic testing programs in C/C++. *Programmnye produkty i sistemy* [Software & Systems]. 2016, vol. 29, no. 4, pp. 101–106 (in Russ.); DOI: 10.15827/0236-235X.116.101-106.

МЕТОД РАСПОЗНАВАНИЯ СТРУКТУРЫ ТАБЛИЦЫ В ЭЛЕКТРОННЫХ ТАБЛИЧНЫХ ДОКУМЕНТАХ

Н.М. Ткешелашвили, аспирант, nmtkeshelashvili@corp.ifmo.ru;

С.В. Клименков, ассистент;

А.М. Дергачев, к.т.н., доцент

*(Санкт-Петербургский национальный исследовательский университет
информационных технологий, механики и оптики (Университет ИТМО),
Кронверкский просп., 49, г. Санкт-Петербург, 197101, Россия)*

Одним из популярных средств хранения деловой информации являются электронные таблицы. К сожалению, информация в них плохо структурирована. Определение структуры таблицы необходимо для корректного извлечения из нее данных в процессе автоматической обработки. В работе предложен метод распознавания структуры таблицы, основанный на визуальном подходе. Он опирается на тот факт, что в момент создания электронной таблицы заголовки, данные и агрегатные ячейки представляются таким образом, чтобы человек мог без проблем отличить их друг от друга. В разработанном методе типы данных и свойства форматирования ячеек представляются в виде набора битовых карт, рассматриваемых как графическое представление таблицы. Полагаясь на визуальные различия, позволяющие человеку отличать одни структурные элементы таблицы от других, а также на статистические зависимости внутри битовых карт, метод определяет ориентацию таблицы, расположение заголовков и данных и формирует структуру в виде набора объектов. Для определения направления таблицы используется метод Хафа. Для проверки корректности распознавания был выбран набор тестовых электронных таблиц, содержащих деловую информацию. Метод показал высокую точность: из более чем 100 тестовых файлов, содержащих более 20 000 строк, корректность распознавания составила 92 %.

Ключевые слова: автоматическая обработка документов, распознавание структуры, электронные таблицы.

Электронный документооборот применяется практически повсеместно, и одним из распространенных форматов документов являются электронные таблицы. Автоматическая обработка такого типа документов помогла бы решить многие трудоемкие задачи, связанные с извлечением информации, сопоставлением и сравнением данных из нескольких документов, построением и заполнением БД на основе электронных таблиц и так далее. Ключевой проблемой в данном случае является распознавание структуры таблицы, то есть определение таких ее частей, как заголовки, подзаголовки и непосредственно сами данные. Решению описанной проблемы посвящен ряд работ.

Решение, предложенное в [1], опирается на анализ содержащихся в таблице формул. Авторами вводится язык описания электронных таблиц и решается задача обнаружения структуры при помощи логического программирования на языке Prolog. В работе [2] рассматриваются поиск скрытых структурных свойств электронных таблиц и то, как эти свойства можно использовать для автоматических подсказок пользователю при редактировании электронных таблиц. Авторы ищут функциональные зависимости между наборами данных, по которым строится реляционная схема документа. В статье [3] описан метод плавного перехода между электронными таблицами и реляционной БД. Основой данного метода является гипотеза, что электронные таблицы и БД связаны отношениями уточнения. Для поиска таких отношений авторы ищут функциональные зависимости между данными. Использование вычислительных формул из

функциональных зависимостей позволяет сформулировать уточняющие шаги. Подход авторов использует теорию уточнения данных [4].

Описанные методы с разных сторон подходят к задаче распознавания структуры таблицы. Некоторые из них требуют дополнительных данных о структуре, составленных вручную, другие опираются на связь данных через формулы. Предлагаемый в настоящей работе метод основан на визуальном подходе к анализу структуры таблицы. Он опирается на графический образ таблицы, полученный из исходного файла электронных таблиц, а не на ее строгий макет. Исходными данными являются табличные документы форматов XLS и XLSX. Метод не предназначен для распознавания таблиц из сканированных или иных графических изображений.

Понятие структуры таблицы

Рассмотрим структуру таблицы с точки зрения нормативных источников (ГОСТ 2.105-95). Согласно государственному стандарту, в таблице выделяют головку, боковик и матричное ядро. Головка таблицы содержит заголовки столбцов, которые могут быть многоуровневыми, но должны составлять иерархию. Боковик содержит заголовки строк. Возможно использование врезанных заголовков. Над таблицей располагаются ее название и номер, ниже может размещаться иная дополнительная информация, например, дата создания, имя автора или единая единица измерения величин. Государственный стандарт регламентирует пра-

вила оформления таблиц, но никак не затрагивает данные, хранящиеся в таблице.

Другой подход заключается в формальном описании таблиц с позиции информационных объектов, представляющих собой совокупность аргументов и зависимых переменных [5]. Этот подход близок к реляционной модели, согласно которой строка таблицы представляется набором смысловых отношений между ячейками [6]. В такой модели таблица хранит информацию о множестве объектов с определенными именованными свойствами (атрибутами). Каждая запись, таким образом, может быть преобразована в пару «атрибут–значение». Названия свойств указываются в заголовках столбцов, а значения в ячейках. Значения свойств имеют одинаковый тип. Объект занимает одну строку таблицы. В документах свойства могут объединяться в иерархии, формируя различные варианты визуального представления одного и того же отношения.

Метод извлечения структуры таблицы

Таблицы, как правило, содержат объекты одного типа, расположенные по строкам или по столбцам в зависимости от ориентации таблицы. Построчное представление более распространено, постолбцовое типично для акцентирования внимания на сравнении свойств объектов при их небольшом количестве. В случае ориентации по строкам одна строка описывает один объект, а заголовки содержат названия свойств, общих для всех объектов. Таким образом, корректно определив местоположение заголовков и данных, то есть решив задачу обнаружения структуры таблицы, можно извлечь объекты в виде набора свойств и их значений. Распознавание структуры таблицы состоит из следующих шагов:

- шаг 1 – определение границ рабочей области таблицы;
- шаг 2 – определение направления данных;
- шаг 3 – поиск шаблона данных и заголовков;
- шаг 4 – обработка заголовков и врезанных заголовков.

Шаг 1. Определение границ рабочей области таблицы. В документах электронных таблиц для обозначения одной страницы документа применяется понятие «лист» (sheet). В общем случае на одном листе могут помещаться несколько таблиц, но рассматриваемый метод предполагает наличие на каждом листе только одной таблицы.

При создании нового документа электронных таблиц редактор отображает пустой лист определенного по умолчанию размера, полностью заполненный ячейками. Однако фактически ячейки создаются в момент их редактирования пользователем. Таким образом, документ не содержит ячейки, которые не были модифицированы, сокращая этим размер хранимых данных. Тем не менее в некото-

рых случаях создаются ячейки, не несущие полезной информации. Например, при редактировании ячейки, находящейся не в начале строки, будут созданы все ячейки от начала строки до этой ячейки. То есть, если таблица размещена на листе с некоторым смещением от его краев, это место будет заполнено пустыми ячейками. Также при выделении целиком столбца и назначении ему форматирования создаются все ячейки столбца в соответствии с высотой листа, принятой по умолчанию.

Под рабочей областью таблицы понимается диапазон ячеек, содержащих полезную информацию и подлежащих обработке. В связи с особенностями, указанными выше, на первом шаге сокращается рассматриваемое множество ячеек за счет отбрасывания пустых. Рабочая область определяется как минимальный прямоугольный диапазон размером R_w на R_h , включающий все непустые ячейки листа.

Шаг 2. Определение направления данных.

Таблицу, заголовки которой расположены над столбцами, называют столбцовой; если заголовки находятся в строках, таблица называется строчной. Под направлением данных понимается расположение объектов в таблице: по строкам или по столбцам.

Метод использует визуальные особенности структуры таблицы, такие как взаимное расположение элементов структуры, форматирование и типы данных ячеек. Каждая ячейка обладает набором атрибутов, относящихся как к самой ячейке, так и к данным внутри нее. Данные имеют формат записи и тип, к примеру, тип данных «числовой» и формат записи «проценты». Ячейка и данные могут иметь форматирование, к которому относятся цвет текста и фона, границы и отступы, выравнивание и пр. Каждому атрибуту соответствует набор возможных значений, например, атрибут «шрифт» может быть Arial, Verdana, Georgia и т.д.

На данном шаге для каждого значения каждого атрибута строится битовая карта. Размеры битовой карты совпадают с размерами рабочей области, единица означает наличие у данной ячейки атрибута с данным значением. На рисунке 1 приведен пример таблицы с информацией о книгах, имеющих в книжном магазине.

Справа от таблицы приведен пример битовых карт для атрибута «тип данных». На первой карте он принимает значение «текст», на второй – «числовой», на третьей – «дата». Множество битовых карт, сформированных для всех значений всех атрибутов, является своеобразным графическим образом таблицы. Нетрудно заметить, что расположение объектов в таблице влияет на получаемое на битовых картах изображение. Если объекты (в данном случае книги) расположены в строках, то их свойства, имеющие разные типы и форматы и организованные по столбцам, будут отображаться на битовых картах в виде вертикальных линий.

Книжный магазин «Наследие»		Типографский пер., 16		
		еж. 10:00 – 22:00		
Арти- кул	Название	Автор	Год	Цена
Зарубежная проза				
134289	Вино из одуванчиков	Рэй Брэдбери	1957	177 руб.
342565	Холодный дом	Чарльз Диккенс	1853	420 руб.
437903	Принц и нищий	Марк Твен	1881	315 руб.
936756	Айвенго	Вальтер Скотт	1819	261 руб.
673060	Дети капитана Гранта	Жюль Верн	1868	131 руб.
325367	Три мушкетера	Александр Дюма	1844	297 руб.
Русская проза				
056149	Лолита	Набоков В.В.	1955	144 руб.
135934	Спектр	Лукияненко С.В.	2002	435 руб.
325766	Отцы и дети	Тургенев И.С.	1862	392 руб.
263658	Гроза	Островский А.Н.	1860	130 руб.
786410	Смотритель	Пелевин В.О.	2015	510 руб.
154988	Дубровский	Пушкин А.С.	1841	213 руб.
Поэзия				
123471	Ромео и Джульетта	Уильям Шекспир	1594	388 руб.
388729	Фауст	Иоганн Гете	1960	149 руб.
254869	Евгений Онегин	Пушкин А.С.	1831	355 руб.
209853	Горе от ума	Грибоедов А.С.	1825	267 руб.





Рис. 1. Пример составления битовых карт для таблицы

Fig. 1. Bitmaps creation example

В то же время и основные, и врезанные заголовки будут представлять собой горизонтальные линии.

Отметим две важные особенности. Во-первых, врезанные заголовки размещаются горизонтально вне зависимости от ориентации таблицы. Во-вторых, заголовки обычно выделяют из остальных данных с помощью форматирования, а тип данных у них почти всегда строковый, в то время как свойства объектов включают разнообразные типы данных и форматы записи, но редко сильно выделяются форматированием. То есть линии, обнаруженные на картах типов данных и форматов записи, практически однозначно указывают на ориентацию таблицы, а на картах форматирования с большой вероятностью будут обнаружены ни о чем не говорящие горизонтальные линии.

Исходя из описанных особенностей, для определения ориентации таблицы используются только карты типов данных и форматов записи. Преобразование на множестве карт вертикальных линий будет указывать на столбцовую таблицу, горизонтальных – на строчную.

Для обнаружения линий на битовых картах используется метод Хафа [7, 8], в общем случае решающий задачу нахождения на изображении произвольных кривых, заданных параметрическим уравнением. При поиске прямых линий в качестве функции будет выступать уравнение прямой в полярных координатах. Выбор полярных координат обусловлен необходимостью представления вертикальных прямых. В соответствии с методом Хафа составляется матрица для двух параметров уравнения прямой – расстояния до начала координат (R) и угла наклона (θ). Угол наклона искомой прямой в данном случае может принимать значение 0 или 90 градусов, а расстояние R изменяется от 0 до максимального из измерений рабочей области $R_{\max} = \max(R_w, R_h)$ с шагом 1 (рис. 2).

На изображении выбираются точки интереса, для каждой из них ищутся все прямые, проходящие через эту точку, и заполняется матрица Хафа. В качестве точек интереса для конкретной битовой карты рассматриваются клетки, имеющие значение 1 и хотя бы одну клетку со значением 0 по соседству или являющиеся граничными для битовой карты. На рисунке 3 точки интереса закрашены темно-серым цветом.

Заполненная матрица Хафа показывает, на каких прямых лежит больше всего точек, и, соответственно, они наиболее вероятно присутствуют на изображении. Каждая строка итоговой матрицы точек нормируется на максимально возможное значение для данного измерения (то есть длину или ширину таблицы). Затем значения строк объединяются и сортируются. Вычисляется среднее арифме-

R	0	1	2	3	...	R_{\max}
0					...	
90					...	

Рис. 2. Применение метода Хафа для поиска прямых линий

Fig. 2. Using Hough method for detecting straight lines

1	1	1	1	1	0	1
0	1	1	1	1	0	1
0	1	1	1	1	0	1
0	1	1	1	1	0	1
0	1	1	1	1	0	0
0	1	1	1	1	0	1
0	1	1	1	1	0	1
0	1	1	1	1	0	1

Рис. 3. Пример выбора точек интереса

Fig. 3. Points of interest selection example

тическое всех значений, и отсекается часть сортированного списка, значения в которой больше среднего арифметического. Если среди оставшихся значений преобладают значения из первой строки матрицы ($\theta = 0$), таблица строчная, в противном случае она столбцовая.

Шаг 3. Поиск шаблона данных и заголовков.

Как правило, одному объекту в таблице соответствует одна строка. Однако не в каждой строке таблицы содержится объект. Строка может содержать основной заголовок, врезанный заголовок, агрегатные данные или быть пустой в целях визуального деления таблицы на блоки. Кроме того, в рабочую область попадает не только таблица, но и окружающая ее информация как над, так и под таблицей, которая также не является объектом.

В связи с этим встает задача отделения строк, содержащих объекты, от всех прочих строк. Опираясь на однотипность объектов и их большое количество, данную задачу можно решить статистически, используя битовые карты, созданные на предыдущем шаге алгоритма. В одинаковых столбцах у всех объектов будут содержаться схожие значения, принадлежащие к одному типу данных и, скорее всего, близкие по форматированию. Если рассматривать битовые карты в совокупности, можно обнаружить повторяющиеся шаблоны строк, соответствующие данным и врезанным заголовкам (рис. 4).

При поиске шаблонов целесообразно рассматривать не все имеющиеся битовые карты, во мно-

гих случаях можно ограничиться картами типов и форматов данных. Анализируя объединение битовых изображений, выберем все уникальные строки и подсчитаем частоту их повторения. Затем сгруппируем уникальные строки, имеющие не более D отличающихся ячеек, в шаблоны. Шаблоны отличаются от строк тем, что могут содержать ячейки, значения в которых не определены, то есть оно может быть как 0, так и 1. Параметр D в зависимости от количества ячеек в строке (n) принимает следующие значения:

$$D(n) = \begin{cases} 0, & n < 3, \\ 1, & 3 < n < 10, \\ \lceil n/5 \rceil, & n > 10. \end{cases} \quad (1)$$

Шаблон, которому соответствует наибольшее число строк исходной таблицы, принимается за шаблон данных.

Второй по частоте появления – шаблон врезанных заголовков. Таких шаблонов может быть несколько в случае наличия нескольких уровней заголовков. Все строки, удовлетворяющие шаблону врезанных заголовков, должны располагаться между строками данных. В противном случае они не могут считаться врезанными заголовками. К самой малочисленной группе относятся основные заголовки шапки таблицы, они обычно встречаются в единственном экземпляре, а также строки, не попадающие в вышеприведенные категории. Все выявленные шаблоны используются при построении структурированных объектов по данным, извлекаемым из табличных документов.

Шаг 4. Обработка заголовков и врезанных заголовков. Распознавание заголовков (или шапки) таблицы основано на нескольких предположениях. Во-первых, заголовок таблицы должен находиться непосредственно над первой строкой данных, то есть над первым объектом. Между заголовком и первым объектом может находиться только врезанный заголовок. Во-вторых, заголовок имеет иерархическую (древовидную) структуру. Если заголовок состоит более чем из одной строки и верхние уровни иерархии образованы с помощью объединения ячеек, то у каждой ячейки нижележащего уровня должен быть ровно один родитель на вышележащем уровне. В-третьих, над каждым полем объекта должен находиться ровно один заголовок. Это означает, что заголовок не может содержать пустые ячейки, несколько заголовков не могут соответствовать одному полю и один заголовок не может соответствовать нескольким полям.

Учитывая эти предположения, сформулируем последовательность действий при поиске заголовка.

Ищем в исходной таблице первую строку, совпадающую с одним из шаблонов данных.

Выбираем строку, находящуюся над первой строкой данных, и проверяем ее на непротиворечивость описанным выше правилам.

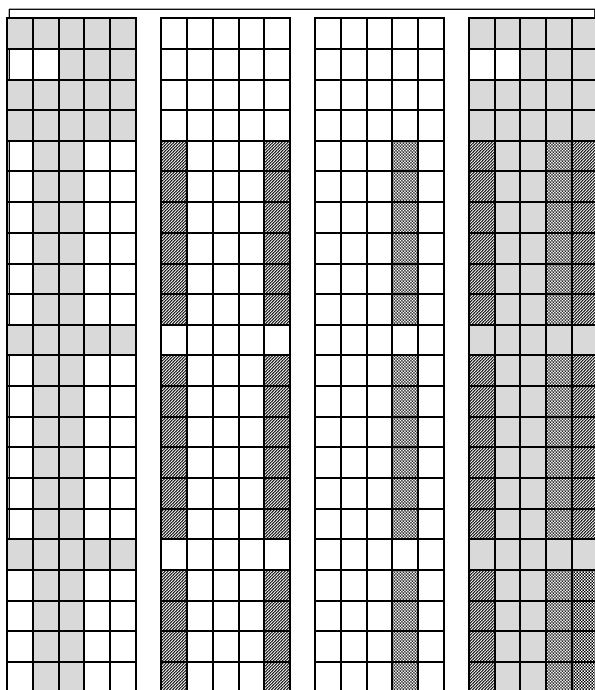


Рис. 4. Поиск шаблона данных по совокупности битовых карт (закрашенные ячейки имеют значение 1)

Fig. 4. Data template search by a set of bitmaps (filled cells have the value of 1)

Если строка противоречит одному из правил, переходим к следующей над ней строке, пока не найдем подходящую.

Когда подходящая строка найдена, она принимается за заголовки нижнего уровня.

Проверяем строку над ней для определения, является ли заголовок многоуровневым. Если вышестоящая строка образует с заголовками нижнего уровня непротиворечивую иерархию, удовлетворяющую описанным выше правилам, она принимается за следующий уровень многоуровневого заголовка. Этот шаг повторяется до тех пор, пока не попадется строка, не вписывающаяся в иерархию. Тогда поиск заголовка прекращается.

В результате выполнения описанных выше шагов получаем размеченную таблицу, каждая строка которой соотнесена с элементом структуры таблицы. Теперь возможно непосредственное извлечение из таблицы полезной информации, то есть содержащихся в ней объектов. Как говорилось ранее, объект представлен набором свойств и их значений, следовательно, из нескольких уровней заголовка нужно сформировать одну строку, которая будет служить именем свойству объекта. Это можно делать, например, так: «Заг.1:Заг.2 ... Заг.N». То есть записать полный путь до заголовка нижнего уровня, используя в качестве разделителя двоеточие. Значения свойств объектов выбираются из строк, соответствующих шаблону данных. Врезанные заголовки можно также рассматривать как свойства со значением true или false, обозначающие принадлежность объекта какой-то категории. Для хранения извлеченных объектов можно использовать формат, подобный JSON (JavaScript Object Notation – описание объекта языка JavaScript).

Тестирование разработанного алгоритма

Тестирование разработанного алгоритма проводилось на наборе таблиц из различных предметных областей. С целью учета всех возможных нюансов распознавания новые документы для тестирования не создавались, а использовались таблицы с реальными данными, составленные разными людьми и

находящиеся в свободном доступе. Все использованные таблицы можно разделить на следующие группы:

- простые (одноуровневый заголовок и данные);
- со сложным заголовком (многоуровневые заголовки);
- с врезанными заголовками;
- с многоуровневыми врезанными заголовками.

Оценка эффективности алгоритма распознавания структуры таблицы проводилась отдельно для каждой группы таблиц и типа элементов. Каждая строка может быть распознана как определенный элемент таблицы: данные, заголовок i -го уровня, подзаголовок i -го уровня и пр. Критерием эффективности выступало отношение правильно распознанных элементов таблицы к общему числу элементов данного типа в конкретной таблице:

$$k = \frac{E_{rec}}{E_{all}} \cdot 100 \%, \quad (2)$$

где E_{rec} – количество правильно распознанных элементов; E_{all} – общее количество элементов данного типа; k – коэффициент распознавания.

Общий коэффициент распознавания для всей таблицы определялся по формуле

$$K = \frac{\sum k_i}{N}, \quad (3)$$

где N – количество различных элементов, $i \in [0, N]$.

Результаты тестирования метода приведены в таблице. Для каждой группы таблиц в первой строке указано отношение количества правильно распознанных элементов заданного типа к общему количеству элементов такого типа в тестовых таблицах. Во второй строке указано количество элементов другого типа, которые были ошибочно распознаны как элементы заданного типа.

Основной заголовок таблицы распознавался методом правильно при условии, что он был корректно составлен и не содержал пустых ячеек. Ошибки в распознавании многоуровневых основных заголовков связаны с тем, что дополнительные данные о таблице, приводимые над ней, в ряде слу-

Результаты тестирования разработанного алгоритма

The results of algorithm testing

Группа таблиц	Уровень основного заголовка			Уровень врезанного заголовка			Данные (объект)	Другое	Распознано таблиц, %
	1-й	2-й	3-й	1-й	2-й	3-й			
Простые таблицы (32 шт.)	30/32	0/0	0/0	0/0	0/0	0/0	12362/12878	41/43	95
	0	2	0	493	0	0	1	24	
Многоуровневый заголовок (10 шт.)	10/10	9/10	4/4	0/0	0/0	0/0	3597/3910	8/9	94
	0	0	1	281	0	0	0	33	
Врезанные заголовки (23 шт.)	22/23	0/0	0/0	183/197	0/0	0/0	4202/4511	15/17	92
	0	0	0	17	306	2	0	1	
Многоуровневые врезанные заголовки (15 шт.)	15/15	0/0	0/0	51/51	85/104	5/12	1637/1742	7/7	86
	0	0	0	19	7	97	0	8	
Распознано элементов по типу, %	96	90	100	94	81	42	95	94	

чаев были оформлены таким образом, что их можно было интерпретировать как следующий уровень иерархии заголовка.

Неточности в обработке врезанных заголовков возникали при отсутствии форматирования, позволяющего отличить один уровень иерархии от другого, например, когда иерархия создавалась за счет многоуровневой нумерации в тексте ячеек. Также негативное влияние на распознавание ячеек данных и врезанных заголовков оказывали частично заполненные таблицы, не все ячейки данных которых содержали значения.

Тем не менее, разработанный метод показал высокую точность распознавания структурных элементов таблицы. Процент корректно распознанных строк для различных видов таблиц составил в среднем 92 %.

Таким образом, в статье рассмотрены современные методы определения структуры таблицы. Предложен метод распознавания структуры таблицы на основании визуальных особенностей отображения, который определяет направление таблицы, шаблоны данных и заголовков. Разработанный метод реализован на языке Java с использованием библиотеки POI для работы с документами форматов XLS и XLSX. Метод показал высокую точность в распознавании структуры таблиц, содержащих большое количество объектов. К таким документам относятся, например, электронные таблицы ценовых предложений. Дальнейшее совершенствование метода возможно за счет исполь-

зования битовых карт, созданных на основе смысловой близости данных, а также семантического анализа содержимого ячеек.

Работа выполнена в рамках темы НИР № 615869 «Методы проектирования ключевых систем информационной инфраструктуры».

Литература

1. Paine J. Spreadsheet structure discovery with logic programming. Proc. EuSpRIG 2004 Conf. Risk Reduction in End User Computing: Best Practice For Spreadsheet Users In The New Europe, 2004, 11 p.
2. Cunha J., Saraiva J., Visser J. From spreadsheets to relational databases and back. Proc. 2009 ACM SIGPLAN workshop on Partial evaluation and program manipulation, Savannah, GA, USA, 2009.
3. Cunha J., Saraiva J., Visser J. Discovery-based Edit Assistance for Spreadsheets. Proc. 2009 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC 2009), 2009, pp. 233–237.
4. Oliveira J.N. Transforming data by calculation. Intern. Summer School, Generative and Transformational Techniques in Software Engineering II, Springer, Heidelberg, 2008, vol. 5235, pp. 134–195.
5. Иванов Ю.Н., Емельянов Н.Е., Сотникова Р.А. Документы: типы, описания. Препр. М.: Изд-во ВНИИСИ, 1987. 62 с.
6. Codd E.F. The relational model for database management: vers. 2. Addison-Wesley Longman Publ., 1990.
7. Дегтярева А., Вежнев В. Преобразование Хафа (Hough transform) // Компьютерная графика и мультимедиа. 2003. № 1 (2). URL: <http://cgm.computergraphics.ru/content/view/36> (дата обращения: 09.06.2016).
8. Duda R.O., and Hart P.E. Use of the Hough transformation to detect lines and curves in pictures. Communications of the ACM 15.1, 1972, pp. 11–15.

Software & Systems

DOI: 10.15827/0236-235X.116.107-112

Received 10.06.16

2016, vol. 29, no. 4, pp. 107–112

TABLE STRUCTURE RECOGNITION METHOD FOR SPREADSHEETS DATA

N.M. Tkeshelashvili¹, Postgraduate Student, nmtkeshelashvili@corp.ifmo.ru

S.V. Klimenkov¹, Assistant; A.M. Dergachev¹, Ph.D. (Engineering), Associate Professor

¹ The National Research University of Information Technologies, Mechanics and Optics, Kronverksky Ave. 49, St. Petersburg, 197101, Russian Federation

Abstract. Spreadsheets are one of the most popular means to collect and represent big amount of business data. Unfortunately, the structure of spreadsheets in most cases is not defined. As a result, processing tools can not retrieve data automatically, without human interaction. When spreadsheets are being created, people make visual formatting to properly present headers, data and aggregation cells. The proposed method recognizes a spreadsheet structure based on visual elements distinction in a cell formatting as well as people do. The developed software based on this method takes the spreadsheet as an input and produces an output based on a simple object notation form. The method contains several steps. Firstly, a working area is defined as a square area of non-empty cells. After that, the whole working area is converted in the set of bitmaps, where thy value of 1 represents whenever cell has particular formatting. The second step contains discovering data sequence in business data – from the top to the down or from the left to the right. We apply Hough method on a subset of defined bitmaps at this step. Next, we find data and header patterns using simple statistics methods. We define most frequently used patterns as data cells. Special processing is made for subtitles and inserted titles. Such subtitles create additional property to resulting objects. The method has been tested on a large set of spreadsheets containing various business data, for example pricelists.

Keywords: tables processing, structure discover, spreadsheets, visual recognition.

Acknowledgements. The work has been performed as part of research topic no. 615869 “Design methods for information infrastructure key systems”.

References

1. Paine J. Spreadsheet structure discovery with logic programming. Proc. of EuSpRIG 2004 Conf. Risk Reduction in End User Computing: Best practice for spreadsheet users in the new Europe. 2004, 11 p.
2. Cunha J., Saraiva J., Visser J. From spreadsheets to relational databases and back. Proc. of the 2009 ACM SIGPLAN workshop on Partial evaluation and program manipulation. 2009, Savannah, GA, USA.
3. Cunha J., Saraiva J., Visser J. Discovery-based Edit Assistance for Spreadsheets. Proc. of 2009 IEEE Symp. on Visual Languages and Human-Centric Computing (VL/HCC 2009). 2009, pp. 233–237.
4. Oliveira J.N. Transforming data by calculation. Generative and Transformational Techniques in Software Engineering II. R. Lammel, J. Saraiva, J. Visser (Eds). International Summer School, vol. 5235, pp. 134–195, Springer Publ., Heidelberg, 2008.
5. Ivanov Yu.N., Emelyanov N.E., Sotnikova R.A. Dokumenty: tipy, opisaniya [Documents: Types, Descriptions]. Moscow, VNIISI Publ., 1987, 62 p.
6. Codd E.F. The relational model for database management: ver. 2. Addison-Wesley Longman Publ. Co., 1990.
7. Degtyareva A., Vezhnevets V. Preobrazovanie Khafa (Hough Transform). Kompyuternaya grafika i multimedia [Computer Graphics and Multimedia]. 2003, iss. 1 (2). Available at: <http://cgm.computergraphics.ru/content/view/36> (accessed June 9, 2016).
8. Duda R.O., Hart P.E. Use of the Hough transformation to detect lines and curves in pictures. Communications of the ACM. 1972, vol. 15.1, pp. 11–15.

УДК 519.68: 62-52

DOI: 10.15827/0236-235X.116.113-117

Дата подачи статьи: 28.01.16

2016. Т. 29. № 4. С. 113–117

МЕТОД ОБМЕНА ИНФОРМАЦИЕЙ МЕЖДУ ПРОГРАММНЫМИ СИСТЕМАМИ АВТОМАТИЗАЦИИ ТЕХНОЛОГИЧЕСКИХ И ПРОИЗВОДСТВЕННЫХ ПРОЦЕССОВ

С.Ю. Рябов, ведущий бизнес-аналитик, sergey.u.ryabov@gmail.com

(Synchronoss, Inc. 12012 Сансет-Хиллс Rd, 910, Рестон, VA 20190);

А.Г. Лютов, д.т.н., профессор, lutov1@mail.ru;

Ю.В. Рябов, к.т.н., доцент, ryabov_yuri_atp@mail.ru;

А.В. Вавилова, аспирант, alexdovolna@yandex.ru

*(Уфимский государственный авиационный технический университет,
ул. К. Маркса, 12, г. Уфа, 450000, Россия)*

Для повышения эффективности технологических и производственных процессов, их производительности и качества предложен метод обмена информацией между программными системами, основанный на применении универсального интерфейса управления информационными потоками на базе языка XML. В настоящее время многие приложения используют собственные форматы данных, что делает невозможным взаимодействие с другими приложениями. В данной работе проанализирован процесс обмена информацией между программными системами CAPP-PDM-MES. При этом создается универсальный XML-документ на основе набора унифицированных XSD-схем. XSD-схемы содержат информацию о характере данных в XML-документе, таким образом, XSD является базой знаний, содержащей метаданные. Предлагается создание единого информационного пространства, включающего XSD-базу знаний и универсальный интерфейс на базе языка XML для всего ПО, используемого в производственном процессе. Полученные при работе конкретной программной системы XML-документы могут быть обработаны Java- или Python-приложениями для создания универсального XML-документа на основе набора унифицированных XSD-схем. Передача информации с использованием универсальных XML-документов и XSD-схем рассмотрена на примере использования программы СПРУТ-ТП при формировании технологического документа «маршрутная карта». Показан фрагмент XML-схемы, отображающий структуру маршрутной карты. На основе XSD с использованием JAXB-приложений создаются XML-документы, содержащие конкретную информацию. Таким образом, набор XSD-файлов формирует базу знаний предметной области. Взаимодействие между программами предлагается осуществлять с помощью протокола SOAP, использующего для этого XML-документы, называемые сообщениями SOAP. Рассмотрен пример создания SOAP-сообщения типа «запрос» и «ответ». Спецификация SOAP содержит соглашения по преобразованию одностороннего обмена сообщениями, а также определяет, как осуществлять передачу всего XML-документа. В итоге получаем XML-файлы, содержащие данные в форме XML-документа и знания в форме XSD, понятные человеку и компьютеру, а потому удобные в использовании.

Ключевые слова: технологические и производственные процессы, XML-документ, XSD-схема, метаданные, протокол SOAP.

Функциональная совместимость и обмен информацией между различными программными системами – в числе главных целей предприятий, которые активно применяют в своей работе CAD/CAM/CAPP/PDM-системы [1]. Многие приложения используют собственные форматы данных, что затрудняет или даже делает невозможным взаимодействие с другими приложениями. Проблема обмена информацией между различными программными системами вызывает необходимость многократного ввода одной и той же информации, что закономерно ведет к снижению производительности технологических и производственных процессов.

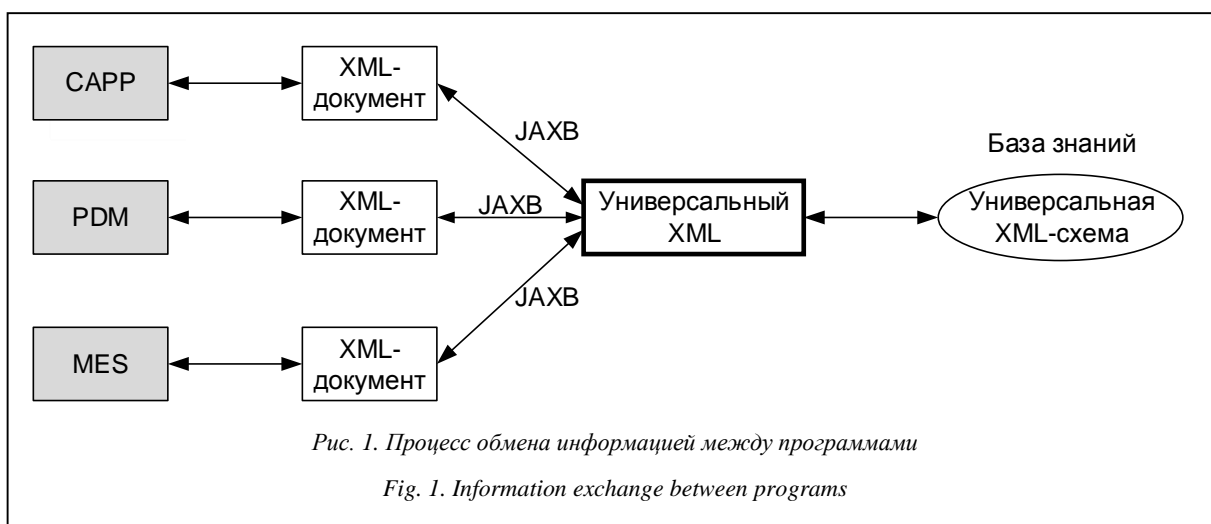
Постановка задачи

Для решения этой проблемы предлагается метод обмена информацией между программными системами, основанный на применении универсального интерфейса управления информационными потоками на базе языка XML [2, 3]. Так,

объединяя ПО в единую систему и создавая единое информационное пространство, можно облегчить и ускорить работу с программными системами технологических и производственных процессов [4]. Создание единого информационного пространства подразумевает, во-первых, единую базу знаний, во-вторых, универсальный интерфейс для всего ПО, используемого в производственном процессе.

Реализация универсального интерфейса подразумевает управление информационными потоками между отдельными программными средствами. По типу используемых информационных потоков в технологических и производственных процессах различают текстовые, графические и специфичные потоки [5].

В настоящее время обмен данными между отдельными программными средствами в некоторых программах автоматизированного проектирования осуществляется посредством языка STEP [6, 7]. Так, с его помощью реализован обмен данными между отдельными CAD- и CAM-программами.



Однако следует отметить, что язык STEP применяется прежде всего для обмена данными об изделиях, что существенно ограничивает область его применения. Во-вторых, STEP-файлы плохорасширяемые, то есть при добавлении нового параметра в уже используемый файл необходимо полностью переписывать код во всех экземплярах файла. В-третьих, для освоения языка STEP работникам потребуются дополнительные знания в области программирования, но на производстве далеко не все технологи и конструкторы их имеют.

Описание метода обмена информацией между программами

Для выполнения обмена информационными потоками предлагается использовать язык XML [1], у которого нет перечисленных выше недостатков. На рисунке 1 представлен обмен информацией между тремя программами (CAPP-PDM-MES) с помощью XML. Полученные XML-документы при работе конкретной программной системы могут быть обработаны Java- или Python-приложениями для создания универсального XML-документа на основе набора унифицированных XSD-схем [8, 9].

XML-схема (XSD-документ) содержит информацию о характере данных в XML-документе, таким образом, XSD является базой знаний, содержащей метаданные [9]. С помощью универсального XML-документа можно осуществлять обмен информацией между любыми программными системами.

Передачу информации от CAPP-системы с использованием универсальных XML-документа и XSD рассмотрим на примере использования программы СПРУТ-ТП. Результатом работы с данной программой является комплект технологических документов: маршрутная карта, операционные карты, карты эскизов, контрольные карты и т.д.

Маршрутную карту (см. по ссылке <http://www.swsys.ru/uploaded/image/2016-4/2016-4-dop/1.jpg>)

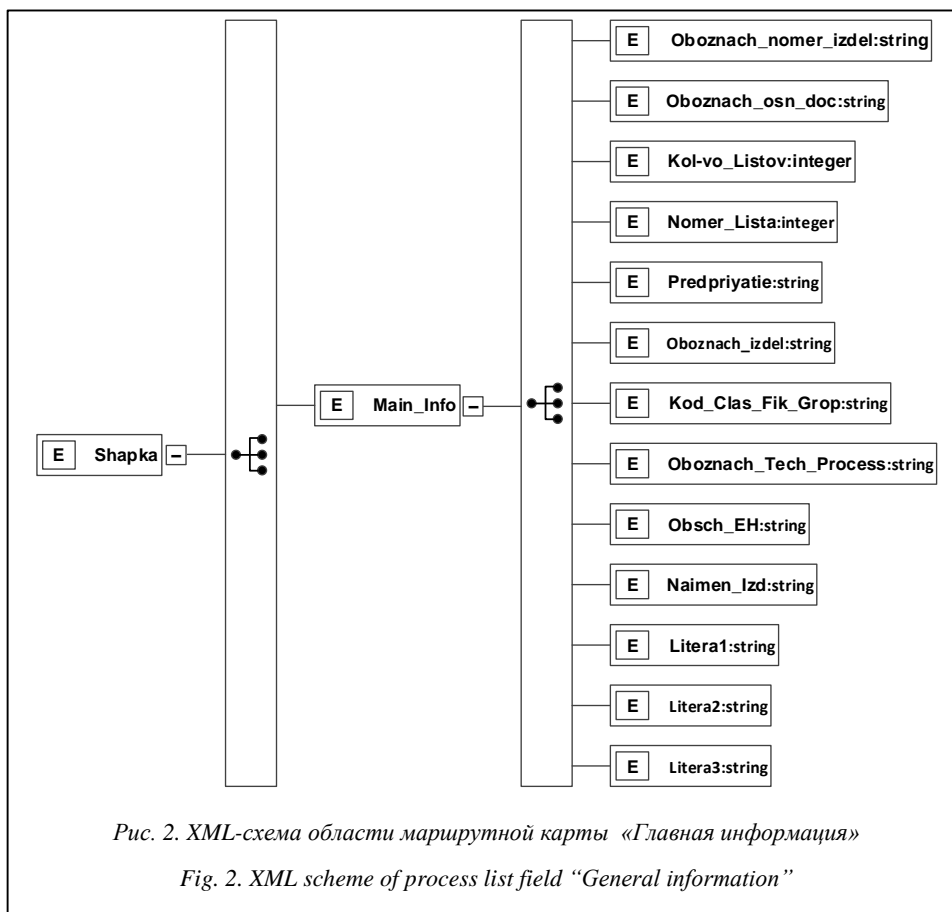
можно разграничить на две части: шапка, содержащая основную информацию по изделию, и строки, в которых описывается последовательность технологического процесса. Каждая часть, в свою очередь, состоит из областей. Так, в шапку маршрутной карты входят области «Создатели», «Главная информация», «Данные по материалу M01», «Данные по материалу M02» и «Изменения». Каждой области соответствует свой элемент XSD. На рисунке 2 показана XML-схема области «Главная информация». Она содержит поля «Обозначение номера изделия», «Обозначение основного документа» и другие.

В полях XSD отображается тип данных, например, поле «Обозначение номера изделия» имеет строковый формат (string), поле «Количество листов» – целочисленный (integer) и т.д. Помимо этого, XML-схема может содержать ограничения на вводимую информацию. Так, в поле «Литерал» может быть введена информация, состоящая только из символов А и Б.

Аналогичным образом создаются XSD операционной карты, карты эскизов и других технологических документов. На основе XSD с использованием JAXB-приложений создаются XML-документы, содержащие конкретную информацию. Таким образом, набор XSD-файлов формирует базу знаний предметной области.

Использование протокола SOAP

С целью обеспечения взаимодействия программ между собой необходимо применять протокол SOAP, использующий для этого XML-документы, называемые сообщениями SOAP [10]. Применение протокола SOAP инвариантно к используемой модели, поскольку он включает в себя только функции, необходимые для взаимодействия программ. Таким образом, SOAP не зависит от используемой платформы, и его реализация может быть осуществлена на любом языке программирования.



вания. Надо отметить, что в рамках протокола SOAP поддерживается транспортный протокол любого типа. Также протокол SOAP поддерживает любые методы кодирования данных, что позволяет посылать в сообщениях SOAP информацию любого типа. На рисунке 3 показана схема SOAP-взаимодействия.

Технологии web-сервисов включают протокол SOAP в качестве главной составляющей, обеспечивающей доставку данных. Он дает возможность поддерживать общий протокол передачи данных и отправителю, и получателю XML-документов, что обеспечивает согласованную передачу сообщений и эффективность сетевой связи. Спецификация протокола SOAP содержит соглашения по преобразованию однонаправленного обмена сообщениями с использованием принципа «запрос/ответ», а также определяет процедуру передачи всего XML-документа.

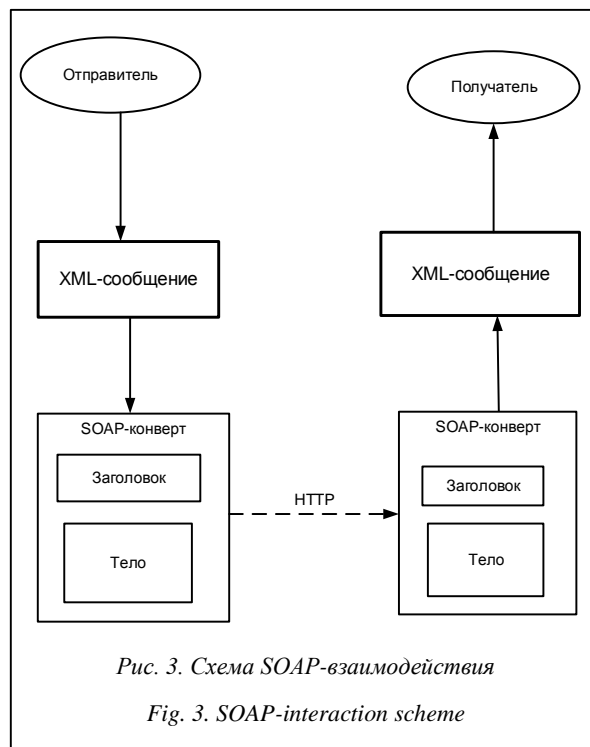
SOAP-сообщение является XML-документом и состоит из трех частей: конверт (SOAP Envelope), заголовок (SOAP Header) и тело (SOAP Body) [10].

Представим пример с приведенной выше маршрутной картой, создав SOAP-сообщения типа «запрос» и «ответ». В теле запроса содержится всего одна строка, в которой передается идентификационная информация о требуемом элементе (рис. 4). В нашем примере – это обозначение основного документа.

В ответ на запрос получен SOAP-ответ, который содержит XML-документ со всей информацией, хранящейся в маршрутной карте (рис. 5). Для примера показана основная информация маршрутной карты, остальная скрыта.

Таким образом, для организации информационных потоков между различными программными средствами рекомендуется использовать файлы в формате XML, которые отображают данные в виде XML-документа и знания в виде XML-схемы. Эти файлы легко расширяемы, понятны человеку и компьютеру и потому удобны в использовании. Обмен

информацией через XML-документы между программными средствами организуется с помощью SOAP-сообщений.



```

1. <soap:Envelope xmlns: SOAP-ENV="http://www.w3.org/2003/05/soap-envelope">
2.   <soap:Body>
3.     <getMarshKart xmlns="http://warehouse.example.com/ws">
4.       <obosnach_osn_doc>BA3-18.3.Ш.01.103 ТП</obosnach_osn_doc>
5.     </getMarshKart>
6.   </soap:Body>
7. </soap:Envelope>
8. |

```

Рис. 4. Пример SOAP-запроса

Fig. 4. SOAP-query example

```

1. <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
2.   <soap:Body>
3.     <Marshrutnaya_Karta xmlns:http://www.w3.org/2001/12/soap-encoding>
4.       <Shapka>
5.         <Sozdateli>
15.           <Main_Info>
16.             <Obosnach_nomer_izdel>BA3-18.3.Ш.01.103</Obosnach_nomer_izdel>
17.             <Obosnach_osn_doc>BA3-18.3.Ш.01.103 ТП</Obosnach_osn_doc>
18.             <Kol-vo_Listov>2</Kol-vo_Listov>
19.             <Nomer_Lista>1</Nomer_Lista>
20.           <Predpriyatie>СППВТ технология< Predpriyatie >
21.             <Obosnach_Izdel> BA3-18.3.Ш.01.103</Obosnach_Izdel>
22.             <Kod_Clas_Fik_Grop></Kod_Clas_Fik_Grop>
23.             <Obosnach_Tech_Process>BA3-18.3.Ш.01.103 ТП</Obosnach_Tech_Process>
24.             <Obsch_EH>мин</Obsch_EH>
25.             <Naimen_Izd>Весы автомобильные</Naimen_Izd>
26.             <Literal></Literal>
27.             <Litera2></litera2>
28.             <Litera3></Litera3>
29.           </Main_Info>
30.           <MO1>
36.           <MO2>
49.           <Izmenenia>
59.         </Shapka>
60.       <Stroka>
151.     </Marshrutnaya_Karta>
152.   </soap:Body>
153. </soap:Envelope>
154.

```

Рис. 5. Пример SOAP-ответа

Fig. 5. SOAP-response example

Предложенный метод организации информационных потоков между программными системами является универсальным с точки зрения обмена информацией и формирования знаний.

Применение метода обмена информацией между программами с помощью XML позволяет получить ряд преимуществ.

Средства разработки и стандартные библиотеки для XML существуют практически на всех платформах и для большинства популярных языков программирования. Методы работы с XML достаточно распространены, поэтому в разных системах используются одинаковые приемы. XML-информация может обрабатываться и восприниматься не только ПЭВМ, но и человеком. Интеграция по XSD формирует базу знаний предметной области. Представленная информация в формате XML не зависит от аппаратных и программных платформ, поэтому XML-формат позволяет связывать разнородные системы. К тому же высокая сте-

пень выразительности XML-документа дает возможность использовать XML-технологии для описания данных практически любой сложности.

Литература

1. Лютов А.Г., Рябов Ю.В. Применение интеллектуального управления для обеспечения качества производственных процессов // СТИН. 2015. № 7. С. 2–4.
2. Старых В.А., Дунаев С.Б., Коровкин С.Д. Спецификация и форматы обмена данными в разнородных информационных системах на базе XML-технологий. Библиотека портала CIT Forum. URL: <http://www.citforum.ru/internet/xml/xmltech/> (дата обращения: 10.11.2015).
3. Погорелов В.И., Щеглов Д.К., Рындин А.А. Методы обмена данными между системами поддержки жизненного цикла изделий на основе языка XML // Третьи Уткинские чтения: тез. докл. Рос. науч.-технич. конф. СПб: Изд-во БГТУ «Военмех», 2007. 171 с.
4. Скворцов А.В., Схиртладзе А.Г., Чмырь Д.А. Автоматизация управления жизненным циклом продукции: учебник для студ. ВПО. М.: Академия, 2013. 320 с.
5. Васенев К.А. Анализ информационных потоков про-

мышленного предприятия в контроллинге // Управление компаний. 2003. № 2. С. 40–45.

6. Куликов Д.Д., Соболев С.Ф. Интеллектуальные программные комплексы для технической и технологической подготовки производства. СПб: Изд-во СПбГУ ИТМО, 2012. 80 с.

7. Хабибуллин И. Самоучитель XML. СПб: БХВ-Петербург, 2003. 333 с.

8. W3C Extensible Markup Language (XML). URL: <http://www.w3.org/XML/> (дата обращения: 10.11.2015).

9. W3C XML Schema Definition Language (XSD) 1.1 Part 1: Structures W3C Recommendation 5 April 2012. URL: <http://www.w3.org/TR/xmlschema11-1/> (дата обращения: 10.11.2015).

10. Simple Object Access Protocol (SOAP) 1.1. URL: <http://www.w3.org/TR/2000/NOTE-SOAP-20000508/> (дата обращения: 10.11.2015).

Software & Systems

DOI: 10.15827/0236-235X.116.113-117

Received 28.01.16

2016, vol. 29, no. 4, pp. 113–117

METHOD OF INFORMATION EXCHANGE BETWEEN SOFTWARE SYSTEMS OF AUTOMATION OF TECHNOLOGICAL AND PRODUCTION PROCESSES

S.Yu. Ryabov¹, Lead Business Analyst, sergey.u.ryabov@gmail.com

A.G. Lyutov², Dr.Sc. (Engineering), Professor, lutov1@mail.ru

Yu.V. Ryabov², Ph.D. (Engineering), Associate Professor, ryabov_yuri_atp@mail.ru

A.V. Vavilova², Postgraduate Student, alexandrovna@yandex.ru

¹ Synchronoss, Inc. 12012 Sunset Hills Rd, 910, Reston, VA 20190

² Ufa State Aviation Technical University, K. Marks St. 12, Ufa, 450000, Russian Federation

Abstract. The article suggests a method of information exchange between software systems to improve technological and production processes efficiency, productivity and quality. The method is based on general information flows control interface based on XML language. Nowadays lots of applications use their own data formats. It makes communication with others applications impossible. The paper analyzes information exchange between CAPP-PDM-MES program systems process.

In addition, there is a general XML document based on unifying XSD schemes combination. XSD schemes contain information about XML document data type, so that XSD schemes are a knowledge base containing metadata. The authors suggest to create a consolidate information area including XSD data knowledge and general interface based on XML language for all software used in production process. Obtained XML documents can be processed by Java or Python applications to create general XML-document based on unifying XSD schemes combination. Information transfer using general XML-documents and XSD-schemes is considered by the example of creating technological document “process list” in SPRUT-TP. The presented XML-scheme segment represents a process list structural arrangement. Based on XSD using JAXB application there are XML-documents produced including explicit information. Thus, a set of XSD files combine a object domain knowledge base. The authors suggest to implement communication between programs by Simple Object Access Protocol (SOAP) using XML-documents called SOAP message. The paper contains an example of creating a “request” and “response” types of SOAP-message. SOAP specification contains agreements about transformation one-way message exchange and determines complete XML document transfer. The results include XML-files containing an XML-document form data and XSD-form knowledge that are understandable by a human and a computer and, therefore, accessible.

Keywords: technological and manufacturing processes, XML document, XSD scheme, meta data, Simple Object Access Protocol.

References

1. Lyutov A.G., Ryabov Yu.V. Intelligent control for manufacturing processes quality assurance. *STIN* [ENTO]. 2015, no. 7, pp. 2–4 (in Russ.).
2. Starykh V.A., Dunaev S.B., Korovkin S.D. Data exchange specification and forms in different information systems based on XML technologies. *CIT Forum*. Available at: <http://www.citforum.ru/internet/xml/xmltech/> (accessed November 10, 2015).
3. Pogorelov V.I., Shcheglov D.K., Ryndin A.A. Methods of data exchange between product life cycle support systems based on XML. *3 Utkinskie chteniya: tez. dokl. Ros. nauch.-tekhnich. konf.* [Proc. Russian Science and Fiction Conf. “3rd Utkin’s readings”]. St. Petersburg, BG TU Voenmekh Publ., 2007, 171 p. (in Russ.).
4. Skvortsov A.V., Skhirtladze A.G., Chmyr D.A. *Avtomatizatsiya upravleniya zhiznennym tsiklom produktsii* [Product life cycle control automation]. Textbook. Moscow, Academy Publ., 2013, 320 p.
5. Vasenev K.A. Enterprise information flows analyze in controlling. *Upravlenie kompaniy* [Company Control]. 2003, no. 2, pp. 40–45 (in Russ.).
6. Kulikov D.D., Sobolev S.F. *Intellektualnye programnye komplekсы dlya tekhnicheskoy i tekhnologicheskoy podgotovki proizvodstva* [Intelligent Program Complexes for Technical and Technological Preproduction]. St. Petersburg, SPbSU ITMO Publ., 2012, 80 p.
7. Habibullin I. *Samouchitel XML* [XML Teach-yourself Guide]. St. Petersburg, BHV-Peterburg Publ., 2003, 333 p.
8. *W3C Extensible Markup Language (XML)*. Available at: <http://www.w3.org/XML/> (accessed November 10, 2015).
9. *W3C XML Schema Definition Language (XSD) 1.1 Part 1: Structures W3C Recommendation*. 2012. Available at: <http://www.w3.org/TR/xmlschema11-1/> (accessed November 10, 2015).
10. *Simple Object Access Protocol (SOAP) 1.1*. Available at: <http://www.w3.org/TR/2000/NOTE-SOAP-20000508/> (accessed November 10, 2015).

УДК 004.056

DOI: 10.15827/0236-235X.116.118-124

Дата подачи статьи: 11.10.16

2016. Т. 29. № 4. С. 118–124

ПРОГРАММНАЯ СИСТЕМА СТРУКТУРНО-ПАРАМЕТРИЧЕСКОГО СИНТЕЗА СИСТЕМЫ ЗАЩИТЫ ИНФОРМАЦИИ

*В.Г. Жуков, к.т.н., доцент, vadimzhukov@mail.ru;**М.Н. Жукова, к.т.н., доцент, mariem@inbox.ru;**А.С. Тимохович, к.п.н., TASstepanich@yandex.ru;**Д.С. Волков, студент, 23_dimon@mail.ru**(Сибирский государственный аэрокосмический университет им. академика М.Ф. Решетнева,
просп. им. газ. «Красноярский рабочий», 31, г. Красноярск, 660014, Россия)*

Статья посвящена решению актуальной задачи разработки алгоритма структурно-параметрического синтеза системы защиты информации на примере создания защищенной автоматизированной системы. Данный алгоритм позволяет генерировать несколько возможных вариантов системы защиты информации с учетом требований государственных стандартов в области информационной безопасности, нормативно-методических документов ФСТЭК России и ФСБ России. Кроме этого, каждая сформированная архитектура системы защиты информации предполагает использование достаточно большого количества средств защиты информации, что усложняет задачу из-за появления различных вариантов реализации сформированной архитектуры системы. В работе рассматриваются только сертифицированные средства защиты информации для реализации в рамках построенной архитектуры системы защиты информации. Получившаяся задача оптимизации имеет множество особенностей и ограничений. Классические методы оптимизации в данном случае применить нельзя. Проведен анализ различных схем применения генетического алгоритма оптимизации, хорошо зарекомендовавшего себя при решении подобных задач оптимизации. Однако применение классического генетического алгоритма оказалось невозможным из-за особенностей требований нормативно-методических документов, регуляторов в сфере информационной безопасности, самой структуры системы защиты информации и компонент в виде средств защиты информации. Проведено множество модификаций алгоритма, позволяющих применять разработанный алгоритм при решении реальных практических задач. Разработанный алгоритм с модификациями реализован в виде программного средства. Работа программного средства протестирована при различных вариантах требований к создаваемой системе защиты информации. Эффективность структурно-параметрического синтеза системы защиты информации при использовании разработанного программного средства составляет в среднем 84 %.

Ключевые слова: *структурно-параметрический синтез, система защиты информации, информационная безопасность, проектирование, защищенность, генетический алгоритм.*

По данным аналитического центра InfoWatch, в первом полугодии 2015 года в мире обнародовано (в СМИ и иных источниках) и зарегистрировано 723 случая утечки конфиденциальной информации, что на 10 % превышает количество утечек, зарегистрированных за аналогичный период 2014 года [1]. Внешние атаки стали причиной 32 % утечек данных. В настоящее время в большинстве организаций ведется работа с информацией, требующей защиты в соответствии с нормативно-правовыми актами. А это значит, что необходимо проектировать *систему защиты информации (СЗИ)* так, чтобы избежать утечек [2].

Эта задача ложится на специалиста по информационной безопасности. Ему необходимо проанализировать рынок средств защиты информации и на основе требований создать эффективную систему защиты. При этом надо исходить из того, что основной целью организации является получение прибыли и, следовательно, затраты на систему защиты должны быть экономически оправданы, а сама система не должна создавать трудностей при работе персонала. Все работы по проектированию архитектуры СЗИ специалисту приходится делать практически вручную, затрачивая достаточно много времени. К тому же большую роль играют компетентность и опыт. Чтобы минимизировать

вероятность ошибки проектирования, временные и экономические затраты и помочь специалисту при выборе, необходимо использовать методы автоматизации. В работе рассматривается подход к построению и настройке СЗИ путем использования структурно-параметрического синтеза на базе генетического алгоритма оптимизации.

Проектирование СЗИ

Для обеспечения защиты необходимо использовать средства, которые существенно затрудняют для злоумышленника возможность атаки или полностью ее исключают и тем самым снижают риск потери данных в системе. Все средства защиты информации можно разделить на две большие группы: организационные и технические. В любой организации должны быть использованы как те, так и другие средства для создания системы защиты, соответствующей требованиям.

Организационные средства включают в себя организационно-технические и организационно-правовые и являются основой СЗИ, поскольку именно при их помощи определяются порядок и правила функционирования объектов защиты и деятельность должностных лиц по обеспечению защиты информации. Но во многих случаях создание эф-

фективной СЗИ невозможно без использования технических средств защиты.

Технические средства защиты можно разделить на аппаратные, программные и программно-аппаратные. Основная задача технических средств защиты – блокирование каналов утечки информации для предотвращения утечек, хищения, утрат, искажения, подделки информации.

Согласно Положению ФСТЭК «О сертификации средств защиты информации по требованиям безопасности информации», средства защиты информации, предназначенные для защиты сведений, составляющих государственную тайну, а также другой информации с ограниченным доступом, подлежащей защите в соответствии с действующим законодательством, обязательно должны быть сертифицированы. Поэтому для корректной работы разрабатываемого программного средства исходные данные должны быть взяты из реестра сертифицированных на территории России средств защиты информации.

Для качественной защиты информации недостаточно просто установить средства защиты на каждый элемент автоматизированной системы. Необходимо, чтобы все средства, методы и мероприятия, используемые для защиты информации, объединялись в единый целостный механизм – систему защиты.

СЗИ является совокупность органов и (или) исполнителей, используемой ими техники защиты информации, а также объектов защиты информации, организованная и функционирующая по правилам и нормам, установленным соответствующими документами в области защиты информации (ГОСТ Р 50922-2006).

При разработке системы защиты чаще всего используется типовый набор технических и организационных средств защиты информации. В статье [3] предлагаются следующие элементы, входящие в типовую схему организации защиты передачи и хранения информации:

- средства защиты от *несанкционированного доступа* (НСД);
- *межсетевой экран* (МЭ);
- средства криптографической защиты информации;
- средства аутентификации и хранения ключевой информации;
- антивирусное ПО;
- *системы обнаружения вторжений* (СОВ);
- средства анализа защищенности;
- операционная система.

Архитектура системы защиты должна соответствовать архитектуре защищаемой системы и быть максимально экономичной, поэтому некоторые из рассмотренных элементов могут отсутствовать в создаваемой системе.

Поскольку количество используемых средств защиты информации очень велико, возникает до-

полнительная сложная задача – правильно описать структуру и параметры системы для эффективного подбора средств и их последующей настройки.

Структурно-параметрический синтез СЗИ

Структурно-параметрический синтез – это процесс, в результате которого определяется структура объекта и находятся значения параметров составляющих ее элементов таким образом, чтобы были удовлетворены условия задания на синтез (технического задания). Так как все объекты и системы на определенном уровне рассмотрения имеют структуру, а элементы, составляющие структуру, имеют параметры, практически любая задача проектирования может быть сведена к задаче структурно-параметрического синтеза [4].

Использование методов структурно-параметрического синтеза для построения архитектуры и настройки параметров СЗИ сопряжено с несколькими проблемами. В работе [5] автор выделяет следующие:

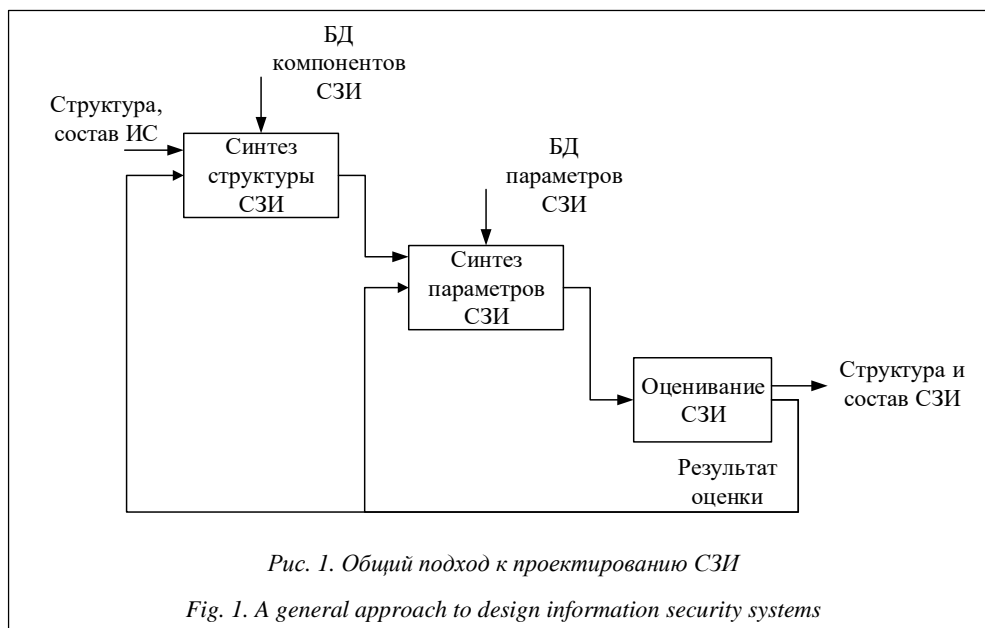
- недостаточные вычислительные мощности;
- отсутствие адекватных языков программирования (в частности, языков, поддерживающих парадигму объектно-ориентированного программирования);
- затруднения в многократном использовании программного кода;
- трудность реализации межплатформенных систем;
- отсутствие общей теории структурно-параметрического синтеза, пригодной для всех классов проектируемых устройств;
- отсутствие совершенного лингвистического обеспечения, поддерживающего процедуру структурно-параметрического синтеза.

Математические и компьютерные модели, применяемые при автоматизации структурно-параметрического синтеза объектов, существенно отличаются от моделей, используемых при автоматизации параметрического синтеза [6]. Так, если при параметрическом синтезе структура объекта в процессе синтеза остается постоянной, то в процессе структурно-параметрического синтеза изменяются как параметры объекта, так и его структура. Сравнение характеристик моделей, применяемых при автоматизации параметрического и структурно-параметрического синтеза, дано в работе [7].

Проектирование системы защиты с использованием алгоритмов синтеза в общем случае можно представить в виде обобщенной схемы, представленной на рисунке 1.

На вход схемы подаются необходимые параметры системы и БД, из которой выбираются средства для проведения синтеза структуры и синтеза параметров. Задача синтеза в данном случае является нетривиальной из-за необходимости учета

требований регуляторов и использования для формирования архитектуры системы защиты довольно большого перечня средств защиты информации. Из-за этих особенностей использование классических методов оптимизации становится сложным или невозможным и необходимо применять другие методы для решения задачи.



Генетический алгоритм оптимизации для решения задачи структурно-параметрического синтеза СЗИ

Требуется решение следующих задач:

- достижение максимума совпадений между заданной пользователем структурой и подобранными средствами защиты;
- достижение максимума совпадений между заданными пользователем параметрами средств защиты и подобранными средствами с этими параметрами;
- минимизация стоимости решения.

Таким образом, формируется следующая задача оптимизации:

$$F(X) = \begin{cases} 0, x_i \neq x_i^* \\ 1, x_i = x_i^* \end{cases} \rightarrow \max, \quad (1)$$

где $X = (x_1, \dots, x_n)$ при $n = 53$, $x \in [a_i; b_i]$, $C \leq C^*$ (стоимость), $C \rightarrow \min$.

Исходя из выбранной типовой структуры и параметров средств защиты, пользователь может задать 256 различных сочетаний элементов структур и 211 680 сочетаний различных параметров данных средств защиты.

Реестр сертифицированных средств в ФСТЭК содержит 1 226 записей, из которых элементами выбранной структуры являются следующие:

- средства защиты от НСД – 32 записи;
- МЭ – 387 записей;
- средства аутентификации и хранения ключевой информации – 15 записей;
- антивирусное ПО – 16 записей;
- СОВ – 14 записей;
- средства анализа защищенности – 4 записи;
- операционные системы – 64 записи.

Каждое из данных средств имеет набор параметров и стоимость. Для решения задачи предлагается использовать генетические алгоритмы.

Генетический алгоритм (англ. genetic algorithm) – это эвристический алгоритм поиска, используемый для решения задач оптимизации и моделирования путем случайного подбора, комбинирования и вариации искомых параметров с использованием механизмов, аналогичных естественному отбору в природе [8]. Первые работы по применению генетических алгоритмов были проведены в 1954 году, а особенно популярными они стали благодаря работе Джона Холланда и его книге «Адаптация в естественных и искусственных системах» в начале 70-х годов.

В статье [9] Холланд описывает типовую схему работы генетического алгоритма:

- а) задание целевой функции (приспособленности) для особой популяции;
- б) создание начальной популяции;
- в) отбор особей для скрещивания;
- г) подбор пар для скрещивания;
- д) скрещивание;
- е) мутирование;
- ж) вычисление значений целевой функции для каждой особи;
- и) формирование нового поколения (селекция);
- к) если выполняются условия остановки, цикл заканчивается, иначе – возвращаемся к шагу в).

В работе проведен анализ различных схем работы операторов. С учетом особенностей поставленной задачи сделан следующий выбор:

- схема отбора особей – пропорциональный отбор; данная схема наиболее просто реализуется и не требует дополнительных затрат вычислительной мощности на сортировку массива, необходимую в других способах;

- схема подбора особей в родительскую пару – случайный способ выбора; поскольку эта схема является универсальной для решения различных классов задач и численность популяции в рассматриваемой задаче достаточно мала;
- схема скрещивания – одноточечный кроссинговер;
- схема формирования нового поколения – в следующее поколение переходят и родители, и потомки без использования принципа элитизма;
- схема мутации – точечная мутация.

Программная система структурно-параметрического синтеза СЗИ

Для проведения структурно-параметрического синтеза СЗИ разработано программное средство, обобщенная схема которого показана на рисунке 2. На рисунке 3 приведена блок-схема работы программного средства.

Перед началом работы пользователь задает необходимые параметры системы при помощи элементов формы главного окна программы (см. <http://www.swsys.ru/uploaded/image/2016-4/2016-4-dop/3.jpg>). Пользователь системы может лишь выбрать необходимый класс защищенности автоматизированной системы и (или) класс персональных данных и получить автоматически сформированный список требований по структуре и параметрам системы, который будет подан на вход алгоритма для дальнейшего подбора подходящих средств защиты, либо самостоятельно осуществить выбор средств защиты информации в соответствии с существующими потребностями организации или требованиями регулирующих органов.

После нажатия на кнопку «Старт» происходят обработка введенной информации и генерация целевой функции, максимума которой необходимо достичь. Значением данной функции является количество бит, стоящих на гене на тех же местах, что и в значении, которое ввел пользователь. После этого происходит считывание информации о средствах защиты информации из текстового файла формата .txt, получаемого путем экспорта его из таблицы Excel. Каждый элемент в базе имеет следующий набор параметров:

- название;
- класс средств защиты из перечня типовых средств (8 бит);
- класс защищенности средств вычислительной техники (СВТ) от НСД (7 бит);

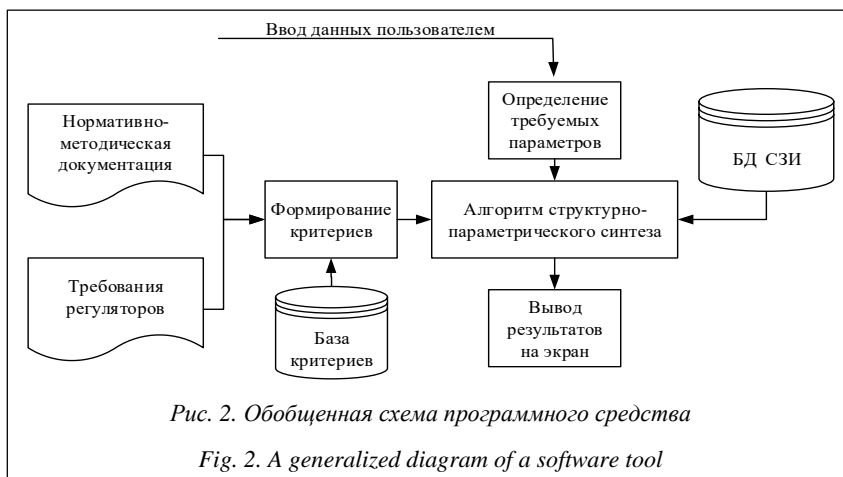


Рис. 2. Обобщенная схема программного средства

Fig. 2. A generalized diagram of a software tool

- класс защиты по недеklarированным возможностям (НДВ) (4 бита);
- класс МЭ (5 бит);
- оценочный уровень доверия (ОУД) (7 бит);
- класс СОВ (6 бит);
- класс средств антивирусной защиты (САВЗ) (6 бит);
- класс ФСБ (6 бит);
- стоимость.

На рисунке 4 показан порядок расположения бит в БД.

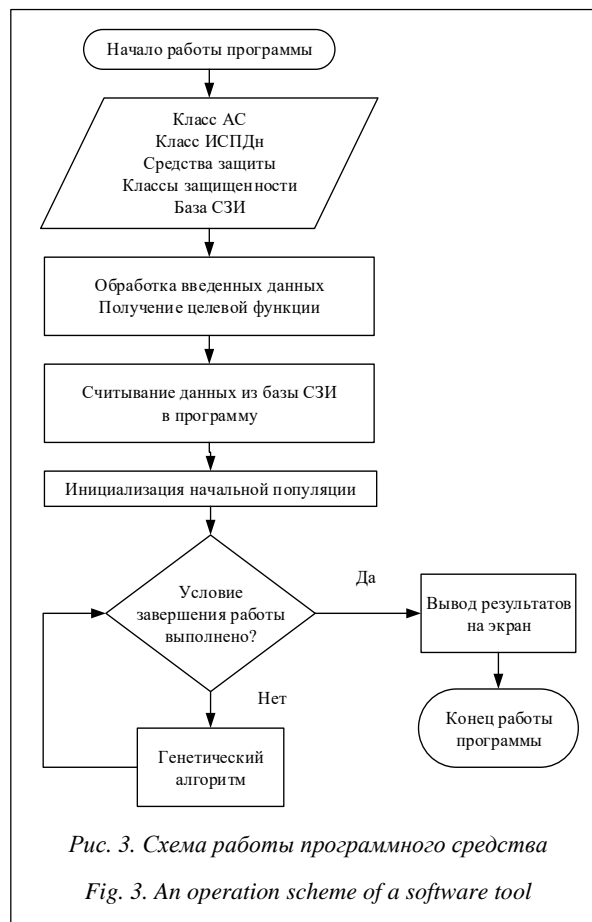


Рис. 3. Схема работы программного средства

Fig. 3. An operation scheme of a software tool

Название	СЗИ								СВТ от НСД							НДВ						МЭ				
	1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	1	2	3	4	1	2	3	4	5		
	ОУД							СОВ						САВЗ						ФСБ						
	1	2	3	4	5	6	7	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6	Цена

Рис. 4. Расположение бит в базе СЗИ

Fig. 4. The arrangement of bytes in an information security system base

Континет 3.7	СЗИ								СВТ от НСД								НДВ				МЭ					
	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	1	1	1	0	1	1	1	1			
	ОУД								СОВ				САВЗ				ФСБ									
	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	1	1	1	0	0	0	0

Рис. 5. Пример записи БД

Fig. 5. An example of a database record

Пример записи, имеющейся в БД, показан на рисунке 5.

Данное средство защиты имеет сертификаты ФСТЭК по 2-му уровню контроля НДВ, 2-му классу МЭ, 3-му классу СОВ и сертификат ФСБ по классу комплекса средств защиты (КСЗ).

После считывания БД и занесения ее в структуру происходит инициализация начальной популяции. Для этого каждый элемент из БД сравнивается со значением, которого необходимо достичь, и вся популяция сортируется. Затем из нее выбираются лучшие особи.

На следующем шаге происходит проверка условия завершения работы. Работа завершается, если найдено решение, удовлетворяющее заданным параметрам, или превышено максимальное число поколений. Второе условие необходимо, чтобы исключить возможность заикливания алгоритма в ситуациях, когда невозможно найти решение.

Для вычисления приспособленности каждой особи необходимо взять отношение среднего значения целевой функции в популяции к значению функции особи, для которой производится расчет. Приспособленность необходима для дальнейшего использования схемы пропорционального отбора.

Далее формируем промежуточное поколение, которое и будет результатом отбора особей. Все отобранные особи случайным образом разбиваются на пары, и производится скрещивание. При этом необходимо учитывать ряд особенностей, которые возникают в зависимости от поставленной задачи и способа реализации. Можно выделить следующие особенности реализации генетического алгоритма в программной системе:

- в программе для хранения характеристик используется тип `unsigned __int64`, вследствие этого для выполнения скрещивания необходимо использовать операции побитового сдвига влево и вправо;
- необходимость в каждом гене хранить информацию о том, результатом скрещивания каких особей он является, чтобы в конце работы алгоритма по подобранному решению можно было выявить, из каких средств защиты оно состоит.

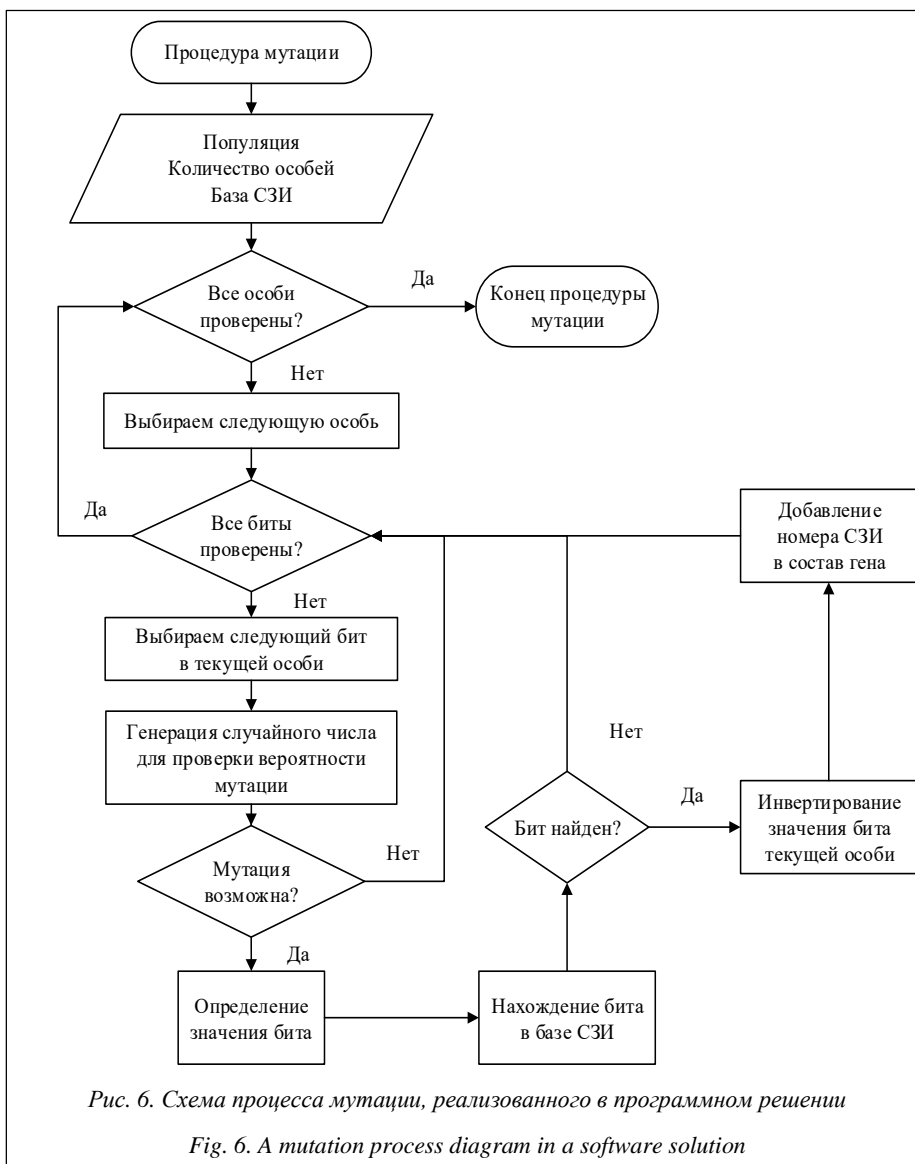
После проведения скрещивания выполняется процедура мутации для всех особей в промежуточной популяции. Для поставленной задачи при каждой мутации необходимо проверять ее возможность. Мутация возможна только в случае, если соответствующий бит имеется в БД. Процедура мутации осуществляется следующим образом:

- по порядку выбирается каждый бит в каждой особи;
- генерируется случайное дробное число в диапазоне от 0 до 1;
- сравниваются полученное число и вероятность мутации, заданная параметром работы алгоритма;
- если мутация оказалась возможной, определяется значение бита, стоящего на данной позиции, и проверяется его наличие в БД, иначе мутация не происходит и выбирается следующий бит;
- если в БД одно и более средств имеют данный бит, случайным образом выбирается одно из них, иначе мутация не происходит и выбирается следующий бит;
- значение бита меняется на противоположное в данной позиции;
- номер средства защиты, в котором был найден данный бит, вносится в состав гена.

Блок-схема данного процесса представлена на рисунке 6.

После мутации производится пересчет значений путем сравнения особей со значением, которое ввел пользователь, и отбор лучших из них. Далее происходят проверка завершения работы алгоритма и вывод решения на экран в виде списка необходимых средств защиты (см. <http://www.swsys.ru/uploaded/image/2016-4/2016-4-dop/4.jpg>).

Задача синтеза СЗИ является сложной из-за одновременного учета требований со стороны нормативно-методических требований и требований регуляторов в сфере информационной безопасности, а также огромного количества решений, возможных при применении тех или иных стратегий разработки СЗИ. Кроме этого, каждая сформирован-



ная архитектура СЗИ предполагает использование достаточно большого количества средств защиты информации, что усложняет задачу из-за появления различных вариантов реализации сформированной архитектуры СЗИ. В работе (из-за универсальности разрабатываемого решения – для применения в государственных или коммерческих организациях) принято решение рассматривать только сертифицированные средства защиты информации для реализации в рамках построенной архитектуры СЗИ. Получившаяся задача оптимизации имеет множество особенностей и ограничений. Классические методы оптимизации в данном случае оказались неприменимыми. Проведен анализ различных схем применения генетического алгоритма оптимизации, хорошо зарекомендовавшего себя при решении подобных задач оптимизации. Однако применение классического генетического алгоритма оказалось невозможным из-за особенностей требований нормативно-методических доку-

ментов, регуляторов в сфере информационной безопасности, самой структуры СЗИ и компонент в виде СЗИ. Проведено множество модификаций алгоритма, позволяющих применять разработанный алгоритм при решении реальных практических задач.

Разработанный алгоритм с модификациями реализован в виде программного средства, позволяющего универсально применять его для проектирования и настройки СЗИ для организаций различной формы собственности (государственные учреждения, коммерческие компании), а также различных масштабов. Работа программного средства протестирована с учетом различных вариантов требований к создаваемой СЗИ. Эффективность структурно-параметрического синтеза СЗИ при использовании разработанного программного средства составляет в среднем 84 %.

Литература

1. InfoWatch: Исследование утечек информации за первое полугодие 2015 года. URL: <http://www.infowatch.ru/analytics/reports/16340> (дата обращения: 30.03.2016).
2. Zhukova M., Stefarov A. Development of the protected telecommunication systems. Proc. 2015 Intern. Siberian Conf. on Control and Communications (SIBCON 2015). URL: <http://www.scopus.com/inward/record.url?eid=2-s2.0-84941051058&partnerID=40&md5=ad5536b24acf44c96921ec62626aa70f> (дата обращения: 30.03.2016).
3. InformationSecurity: Типовая схема организации защиты передачи и хранения информации. URL: http://www.itsec.ru/articles2/Inf_security/tipovaya-shema-organizacii-zaschiti-pere-dachi-i-hraneniya-informacii (дата обращения: 09.04.2016).
4. Жукова М.Н. О структурно-параметрическом синтезе защищенных информационных систем // Решетневские чтения: матер. XIX Междунар. науч. конф. Красноярск: Изд-во СибГАУ, 2015. Ч. 2. С. 282–284.
5. Акимов С.В. Анализ проблемы автоматизации структурно-параметрического синтеза // Доклады ТУСУР. 2011. № 2 (24). С. 204–211.
6. Бухтояров В.В., Жуков В.Г., Жукова М.Н., Золота-

рев В.В., Ковалев И.В., Попов А.М. Поддержка принятия решений при проектировании систем защиты информации: монография. М.: ИНФРА-М, 2014. 131 с.

7. Акимов С.В. Компьютерные модели для автоматизированного структурно-параметрического синтеза // Компьютерное моделирование 2004: тр. 5-й Междунар. конф. СПб: Нестор,

2004. Ч. 1. С. 191–197.

8. CodeNet: Что такое генетический алгоритм? URL: <http://www.codenet.ru/progr/alg/Smart/Genetic-Algorithms.php> (дата обращения: 31.03.2016).

9. Джон Х. Холланд. Генетические алгоритмы // В мире науки. 1992. № 9–10. С. 32–40.

Software & Systems

DOI: 10.15827/0236-235X.116.118-124

Received 11.10.16

2016, vol. 29, no. 4, pp. 118–124

PROGRAM SYSTEM OF STRUCTURAL AND PARAMETRICAL SYNTHESIS OF AN INFORMATION SECURITY SYSTEM

V.G. Zhukov¹, Ph.D. (Engineering), Associate Professor, vadimzhukov@mail.ru

M.N. Zhukova¹, Ph.D. (Engineering), Associate Professor, mariem@inbox.ru

A.S. Timokhov¹, Ph.D. (Education), TAShtepanich@yandex.ru

D.S. Volkov¹, Student, 23_dimon@mail.ru

¹Academician M.F. Reshetnev Siberian State Aerospace University, Krasnoyarsky Rabochy Ave. 31, Krasnoyarsk, 660014, Russian Federation

Abstract. The article is devoted to the solution of an important problem of structural and parametrical synthesis algorithm development for an information security system on the example of creating a protected automated system. This algorithm allows generating several possible options of an information security system considering requirements of state standards in the field of information security, standard and methodical documents of the Federal Service for Technical and Export Control of Russia and the Federal Security Service of Russia. Besides, each architecture of an information security system assumes using rather large number of information security means. It complicates the task because of various options of information security system created architecture implementation. The paper considers only certified means of information security for implementation within the constructed architecture of information security system. Such problem of optimization has a set of features and restrictions. Classical optimization methods in this case aren't applicable. The analysis of various schemes of application of an optimization genetic algorithm which has proved itself when solving similar problems of optimization. However, application of a classical genetic algorithm is impossible because of features of requirements of standard and methodical documents, regulators in the information security area, structures of information security system and a component in the form of information protection means. A set of algorithm modifications is carried out. They allow applying the developed algorithm when solving real practical tasks. The developed algorithm with modifications is implemented as software. Software operation is tested at various versions of requirements to the created information security system. The efficiency of structural and parametrical synthesis of an information security system when using the developed software is 84 % at average.

Keywords: structural and parametrical synthesis, information security system, information security, design, security, genetic algorithm.

References

1. InfoWatch: Issledovanie utechek informatsii za pervoe polugodie 2015 goda [InfoWatch. Research of information leakages for the first half of 2015]. Available at: <http://www.infowatch.ru/analytics/reports/16340> (accessed March 30, 2016).
2. Zhukova M., Stefarov A. Development of the protected telecommunication systems. Int. *Siberian Conf. on Control and Communications, SIBCON 2015*. Available at: <http://www.scopus.com/inward/record.url?eid=2-s2.0-84941051058&partnerID=40&md5=ad5536b24acf44c96921ec62626aa70f> (accessed March 30, 2016).
3. InformationSecurity: Tipovaya skhema organizatsii zashchity peredachi i khraneniya informatsii [InformationSecurity. Standard scheme of information transfer and storage protection]. Available at: http://www.itsec.ru/articles2/Inf_security/tipovaya-shema-organizatsii-zaschiti-peredachi-i-hraneniya-informatsii (accessed March 30, 2016).
4. Zhukova M.N. On structural and parametrical synthesis of secured information systems. *Materialy XIX Mezhdunar. nauch. konf. "Reshetnevskie chteniya"* [Proc. 19th Int. Scientific Conf. Reshetnev's Readings]. Part 2. Krasnoyarsk, SibGAU Publ., 2015, pp. 282–284 (in Russ.).
5. Akimov S.V. The analysis of an structure and parametric synthesis automation problem. *Doklady TUSUR* [Proc. of TUSUR Univ.]. 2011, no. 2 (24), pp. 204–211 (in Russ.).
6. Bukhtoyarov V.V., Zhukov V.G., Zhukova M.N., Zolotarev V.V., Kovalev I.V., Popov A.M. *Podderzhka prinyatiya resheniy pri proektirovaniy sistem zashchity informatsii* [Decision-making support when designing information security systems]. Moscow, INFRA-M Publ., 2014, pp. 131 (in Russ.).
7. Akimov S.V. Computer models for an automated structural and parametrical synthesis. *Kompyuternoe modelirovanie 2004: Tr. 5 Mezhdunar. konf.* [Proc. 5th Int. Conf. on Computer Modeling 2004]. Part 1, St. Petersburg, Nestor Publ., 2004, pp. 191–197 (in Russ.).
8. CodeNet: Chto takoe genetichesky algoritm? [CodeNet: What is a genetic algorithm?]. Available at: <http://www.codenet.ru/progr/alg/Smart/Genetic-Algorithms.php> (accessed March 30, 2016).
9. Holland D.K. Genetic Algorithms. *V mire nauki* [In the world of Science]. 1992, no. 9–10, pp. 32–40 (in Russ.).

МАТЕМАТИЧЕСКАЯ МОДЕЛЬ ЗАЩИТЫ КОМПЬЮТЕРНОЙ СЕТИ ОТ ВИРУСОВ

Н.А. Семькина, к.ф.-м.н., доцент, semykina.tversu@yandex.ru

(Тверской государственный университет, ул. Желябова, 33, г. Тверь, 170100, Россия);

И.В. Шавыкина, инженер-программист, inna.shavykina@mail.ru

(НИИ «Центрпрограммсистем», просп. 50 лет Октября, 3а, г. Тверь, 170024, Россия)

Одним из эффективных подходов к анализу процессов функционирования компьютерных систем в условиях активного распространения вредоносных программ является математическое моделирование. Его преимущество заключается в большой скорости моделирования различных сценариев для конкретных параметров компьютерной сети. Численные результаты, получаемые посредством таких моделей, позволяют анализировать поведение вирусов в сетях, состоящих из любого числа элементов.

В статье рассмотрена математическая модель защиты компьютерной сети от вредоносного кода на основании модели MSEIRS. Данная модель является разносторонней и всеобщей, так как учитывает многие факторы, влияющие на состояние сети: временные параметры заражения для конкретных вирусов и лечения сети с помощью антивируса, средняя скорость иммунизации и непроизвольное отключение компьютеров от сети. Выбран критерий управления, характеризующий наилучшее состояние системы, когда большинство хостов невосприимчивы к воздействию сетевых вирусов. Построена дискретная аппроксимация исходной задачи оптимального управления.

Результатом исследования модели является разработанная программа, которая позволяет грамотно оценить защищенность сети и при необходимости применить эффективные методы борьбы с вредоносным кодом. Проведены и проанализированы вычислительные эксперименты. Для этого рассмотрены два случая: прогрессирующее во времени и сети распространение вируса (эпидемия) и отсутствие существенных колебаний численности инфицированных компьютеров.

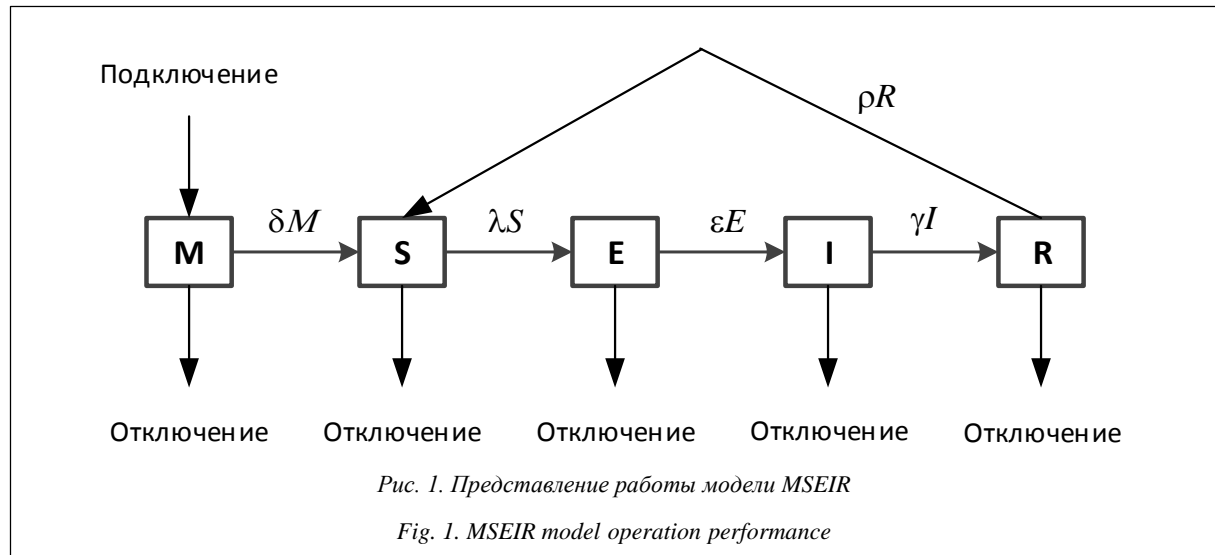
Ключевые слова: компьютерный вирус, математическая модель, модель MSEIRS, нелинейная система, дифференциальные уравнения, оптимальное управление.

В настоящее время для обработки информации, общения и обмена данными повсеместно используются компьютерные сети. Однако благодаря Интернету и большой пропускной способности систем связи распространение вирусов превращается в сетевые эпидемии. Поэтому остро встает вопрос о защите компьютерных сетей и информации. По статистическим данным [1], опубликованным на сайте Лаборатории Касперского, 95 % отечественных организаций как минимум один раз подвергались внешней компьютерной атаке. Россия входит в двадцатку стран, подвергшихся наибольшему риску заражения через Интернет, и занимает одиннадцатое место в мире по уровню зараженности компьютеров. С каждым годом положение в IT-сфере ухудшается: появляется большое количество новых вирусов и, как следствие, растут материальные потери. Важной научной задачей в данной ситуации является построение действенной защиты компьютерной сети. Для этого необходимо исследовать причины и механизмы распространения вирусов. Одним из способов исследований является математическое моделирование распространения вредоносных программ.

Исследования многих авторов показали, что наиболее адекватно распространение вируса описывают модели, основанные на математической теории биологических эпидемий. Большинство исследователей используют для описания компьютерной эпидемии модели типа SI, SIR и SEIR [2–5]. В данной статье рассмотрим модель MSEIRS [6], которая достаточно полно описывает развитие эпи-

демии для популяции людей. Например, модель MSEIRS предусматривает латентный период заболевания, то есть когда объект заражен, но вирус еще не распространяет. Модифицируем эту биологическую модель для наиболее правильного описания деструктивных воздействий вируса на работу компьютерной сети какой-либо организации при условии, что подключаемые новые узлы изначально обеспечены антивирусной защитой. Согласно этой модели, каждый хост компьютерной сети может находиться в одном из пяти состояний: уязвимом, вновь подключенном с установленным антивирусным ПО, инфицированном, латентном (инфицирован, но не распространяет вирус), невосприимчивом к вирусу. Работа этой модели схематично отображена на рисунке 1.

Для описания модели использованы следующие переменные и постоянные величины: $N(t)$ – общее количество машин в сети в момент времени t ; $M(t)$ – количество вновь подключаемых компьютеров в момент времени t ; $S(t)$ – количество уязвимых узлов в момент времени t ; $E(t)$ – количество инфицированных узлов (при этом вирус не наносит никакого вреда зараженному узлу) в момент времени t (латентная стадия); $I(t)$ – количество инфицированных узлов в момент времени t ; $R(t)$ – количество невосприимчивых узлов в момент времени t ; θ – коэффициент, характеризующий скорость прироста новых уязвимых узлов; β – коэффициент, характеризующий скорость заражения уязвимых узлов; μ – коэффициент, характеризующий скорость отключения узлов от сети и не зависящий



от вируса; δ – коэффициент, характеризующий скорость, с которой невосприимчивые хосты вновь становятся уязвимыми; ϵ – коэффициент, характеризующий средний период, когда компьютер является зараженным, но сам вирус неактивен; γ – параметр, характеризующий скорость обновления антивирусных баз.

В любой момент времени будет выполняться условие $M(t) + S(t) + E(t) + I(t) + R(t) = N(t)$.

Процесс деструктивных воздействий вируса на компьютерную сеть описывается системой дифференциальных уравнений с соответствующими начальными условиями:

$$\begin{aligned} \frac{\partial M}{\partial t} &= \theta N - (\delta + \mu)M, \quad M(0) = M_0, \\ \frac{\partial S}{\partial t} &= \delta M - \frac{\beta SI}{N} - \mu S + \delta R, \quad S(0) = S_0, \\ \frac{\partial E}{\partial t} &= \frac{\beta SI}{N} - (\epsilon + \mu)E, \quad E(0) = E_0, \\ \frac{\partial I}{\partial t} &= \epsilon E - (\gamma + \mu)I, \quad I(0) = I_0, \\ \frac{\partial R}{\partial t} &= \gamma I - (\delta + \mu)R, \quad R(0) = R_0. \end{aligned} \quad (1)$$

Непрерывные функции $M(t)$, $S(t)$, $E(t)$, $I(t)$, $R(t)$ будем считать фазовыми переменными. Коэффициент γ будем рассматривать как кусочно-непрерывную функцию управления, удовлетворяющую ограничению

$$0 \leq \gamma(t) \leq Y_{\max} \leq 1, \quad t \in [0, T], \quad (2)$$

где Y_{\max} – максимальная норма управлений, характеризующая технические и экономические возможности организации.

Для оценки защищенности компьютерной сети будем исходить из требования, что в конечный момент времени рассматриваемого периода большинство компьютеров (более 80 %) невосприимчивы к заражению. Для построения функционала используем метод штрафных функций [7], тогда задача

управления будет заключаться в минимизации функционала

$$I(\gamma) = A \max\{(0, 8N(T) - R(T)), 0\}^2, \quad (3)$$

где $N(T) = M(T) + S(T) + E(T) + I(T) + R(T)$ – общее количество компьютеров в сети в конечный момент времени; $A > 0$ – штрафной параметр.

В результате получаем непрерывную задачу оптимального управления (1)–(3). С помощью принципа максимума Понтрягина [8] можно свести исходную задачу (1)–(3) к решению краевой. Однако получение решения краевой задачи в аналитическом виде довольно сложно и поэтому требует применения численных методов.

Для численного решения задачи (1)–(3) построим ее дискретную аппроксимацию, используя явную разностную схему [9, 10]. На отрезке $[0, T]$ введем равномерную сетку $\{t_k = \Delta t k, 0 \leq k \leq L\}$ с шагом $\Delta t = T/L$. Обозначим значения фазовых функций и функции управления в точках разбиения следующим образом: $M(t_k) = M^k$, $S(t_k) = S^k$, $E(t_k) = E^k$, $I(t_k) = I^k$, $R(t_k) = R^k$, $\gamma(t_k) = \gamma^k$, $k = \overline{0, L}$.

Дискретная задача оптимального управления при $k = \overline{0, L-1}$ примет вид

$$\begin{aligned} I &= A[\max\{0, 8N^L - R^L; 0\}]^2 \rightarrow \inf, \\ M^{k+1} &= M^k + \Delta t(\theta N^k - (\delta + \mu)M^k), \\ S^{k+1} &= S^k + \Delta t(\delta M^k - \frac{\beta S^k I^k}{N^k} - \mu S^k + \delta R^k), \\ E^{k+1} &= E^k + \Delta t(\frac{\beta S^k I^k}{N^k} - (\epsilon + \mu)E^k), \\ I^{k+1} &= I^k + \Delta t(\epsilon E^k - (\gamma^k + \mu)I^k), \\ R^{k+1} &= R^k + \Delta t(\gamma^k I^k - (\delta + \mu)R^k), \\ M^0 &= M_0, S^0 = S_0, E^0 = E_0, I^0 = I_0, R^0 = R_0, \\ 0 &\leq \gamma^k \leq Y_{\max} \leq 1. \end{aligned} \quad (4)$$

Для построения приближенного оптимального решения воспользуемся методом проекции градиента [9]. При этом рекуррентные формулы для

определения сопряженных векторов получаем из условий стационарности функции Лагранжа [7, 8].

На основе численных методов решения задач оптимального управления было разработано ПО в среде Lazarus. Пользователь может добавлять, удалять и редактировать параметры сети и антивирусного ПО. Разработанная программа позволяет представлять информацию в удобной для анализа форме в виде таблиц и графиков. Интерфейс главного окна программы (окно ввода данных) и окна «Визуализация решения» представлен по ссылке <http://www.swsys.ru/uploaded/image/2016-4/2016-4-dop/5.jpg>. В окне «Визуализация решения» отображен график обновления антивирусных баз. В главном окне программы расположены поля для ввода параметров сети и антивирусного ПО.

В данной работе приведено исследование нештатной ситуации работы сети (активное развитие эпидемии компьютерного вируса) в течение 24 часов. Рассматриваемый период был разбит на 50 равных отрезков времени. В таблицах 1 и 2 представлены вводимые экспериментальные данные.

Таблица 1

Типы компьютеров сети

Table 1

Network computer types

Состояние компьютеров	Количество
Всего	75 000
Вновь вводимые в эксплуатацию	0
Подвержены вирусу	74 097
Латентные	302
Зараженные	101
С антивирусом	500

Таблица 2

Параметры антивирусного ПО

Table 2

Antivirus software parameters

Параметр ПО	Значение
Коэффициент периода действия антивирусных баз	0,01
Коэффициент распространения вируса	0,06
Коэффициент отключения компьютеров	0,00001
Коэффициент подключения новых компьютеров	0,00001
Коэффициент обновления антивирусных баз (начальное значение)	0,01
Коэффициент обновления антивирусных баз (максимальное значение)	0,4

На рисунках 2 и 3 показаны траектории динамики функций, характеризующие количество компьютеров разных типов, и функции управления.

На приведенных графиках можно заметить, что при большой скорости распространения вредоносного кода за рассматриваемый период удастся погасить эпидемию. При этом выполнено требование,

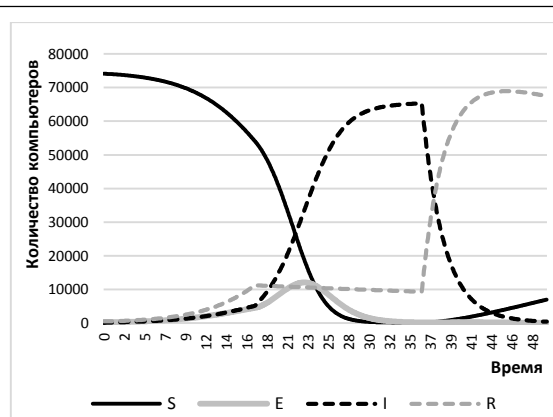
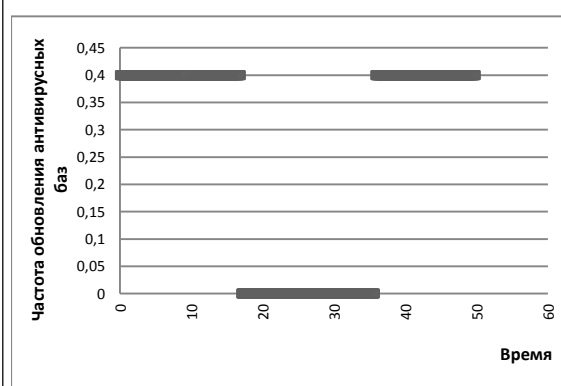


Рис. 2. Графики функций, характеризующие состояние сети

Fig. 2. Function graphs that show network condition

Рис. 3. График функции управления γ Fig. 3. γ control function graph

что в конечный момент времени рассматриваемого периода количество невосприимчивых компьютеров $R(t)$ составляет более 80 % (а именно 89,9 %).

Из графика (рис. 3) можно видеть, что для оптимальной защиты сети необходимая длина временного интервала между загрузками обновления антивирусных баз должна быть равна 19 ед., что в нашем случае составляет чуть больше 9 часов.

Данная программа может быть полезна системным администраторам при формировании структуры компьютерных сетей, максимально защищенных от вирусных атак. При этом работа администратора максимально упрощена: программа высчитывает оптимальное значение частоты обновления антивирусных баз, позволяет моделировать различные ситуации.

Литература

1. Большие угрозы для небольших компаний. URL: <http://www.kaspersky.ru/news?id=207734101> (дата обращения: 13.06.2016).
2. Воронцов В.В., Котенко И.В. Аналитические модели распространения сетевых червей // Тр. СПИИРАН. 2007. Вып. 4. С. 208–224.
3. Kephart J.O., White S.R. Directed-graph epidemiological

models of computer viruses. Proc. 1991 IEEE Comp. Society Symposium on Research in Security and Privacy; Oakland, California, 1991, pp. 343–359.

4. Leveille J. Epidemic spreading in technological networks. Technical Report HPL-2002-287, HP Laboratories Bristol, 2002. URL: <http://www.hpl.hp.com/techreports/2002/HPL-2002-287.pdf> (дата обращения: 13.06.2016).

5. Zhang Ch., Zhao Y., Wu Y. An impulse model for computer viruses. Discrete Dynamics in Nature and Society. 2012. URL: <http://dx.doi.org/10.1155/2013/286209> (дата обращения: 13.06.2016).

6. Hethcote H.W. The basic epidemiology models I & II:

models, expressions for R_0 , parameter estimation, and applications. Master Review, 2005, vol. 9. URL: http://www.ims.nus.edu.sg/Programs/infectiousdiseases/files/hethcote_In.pdf (дата обращения: 13.06.2016).

7. Васильев Ф.П. Численные методы решения экстремальных задач. М.: Наука, 1988. 552 с.

8. Габасов Р., Кириллова Ф.М. Методы оптимизации. Минск: Четыре четверти, 2011. 472 с.

9. Бахвалов Н.С., Жидков Н.П., Кобельков Г.М. Численные методы. М.: Бином. Лаборатория знаний, 2003. 640 с.

10. Самарский А.А. Введение в численные методы. СПб: Лань, 2005. 288 с.

Software & Systems

DOI: 10.15827/0236-235X.116.125-128

Received 14.06.16

2016, vol. 29, no. 4, pp. 125–128

A MATHEMATICAL MODEL OF COMPUTER NETWORK ANTIVIRUS PROTECTION

N.A. Semykina¹, Ph.D. (Physics and Mathematics), Associate Professor, semykina.tversu@yandex.ru

I.V. Shavykina², Software Engineer, inna.shavykina@mail.ru

¹ Tver State University, Zhelyabova St. 33, Tver, 170100, Russian Federation

² R&D Institute Centerprogramsystem, 50 let Oktyabrya Ave. 3a, Tver, 170024, Russian Federation

Abstract. One of the effective approaches to analyzing computer systems in the context of active computer viruses is mathematical modeling. Its advantage is in fast modeling of various scenarios for particular parameters of a computer network. Computational results obtained by these models allow analyzing viruses' behavior in networks consisting of any number of elements.

The article considers a mathematical model of computer network protection from a malicious code based on MSEIRS model. This model is multifaceted and global as it takes into account many factors influencing a network state: timing of infection for specific viruses and network treatment using an antivirus, medium immunization rate and involuntary computer shut down. The selected control criterion characterizes the best state of the system, when most hosts are immune to the impact of network viruses. The paper presents the discrete approximation of the optimal control problem.

The result of the research model is a developed program that allows you to competently evaluate the security of the network and to use effective methods of dealing with malicious code if necessary. The authors conducted and analyzed computational experiments. For this they considered two cases: a progressive virus transmission in time and network (epidemic) and the absence of significant numbers of infected computers' fluctuations.

Keywords: computer virus, mathematical model, MSEIRS model, nonlinear system, differential equations, optimal control.

References

1. Bolshie ugrozy dlya nebolshikh company [Big Threats for Small Companies]. Available at: <http://www.kaspersky.ru/news?id=207734101> (accessed June 13, 2016).

2. Vorontsov V.V., Kotenko I.V. Analytical models of network worm propagation. *Trudy SPIIRAN* [SPIIRAS Proceedings]. St. Petersburg, Nauka Publ., 2007, iss. 4, pp. 208–224 (in Russ.).

3. Kephart J.O., White S.R. Directed-Graph Epidemiological Models of Computer Viruses. *Proc. 1991 IEEE Computer Society Symp. on Research in Security and Privacy*. Oakland, California, 1991, pp. 343–359.

4. Leveille J. Epidemic spreading in technological networks. *Technical Report HPL-2002-287*. HP Laboratories Bristol, 2002. Available at: <http://www.hpl.hp.com/techreports/2002/HPL-2002-287.pdf> (accessed June 13, 2016).

5. Zhang Ch., Zhao Y., Wu Y. An impulse model for computer viruses. *Discrete Dynamics in Nature and Society*. 2012. Available at: <http://dx.doi.org/10.1155/2013/286209> (accessed June 13, 2016).

6. Hethcote H.W. The basic epidemiology models I & II: models, expressions for R_0 , parameter estimation, and applications. *Master Review*. 2005, vol. 9. Available at: http://www.ims.nus.edu.sg/Programs/infectiousdiseases/files/hethcote_In.pdf (accessed June 13, 2016).

7. Vasilev F.P. *Chislennye metody resheniya ekstremalnykh zadach* [Numerical Computation for Extremum Problems]. Moscow, Nauka Publ., 1988, 552 p.

8. Gabasov R., Kirillova F.M. *Metody optimizatsii* [Optimization Methods]. Minsk, Chetyre chetverti Publ., 2011, 472 p.

9. Bakhvalov N.S., Zhidkov N.P., Kobelkov G.M. *Chislennye metody* [Numerical Computation]. Binom Publ., Laboratoriya znaniy, 2003, 640 p.

10. Samarsky A.A. *Vvedenie v chislennye metody* [Introduction to Numerical Computation]. St. Petersburg, Lan Publ., 2005, 288 p.

УДК 004.056.57

DOI: 10.15827/0236-235X.116.129-134

Дата подачи статьи: 30.05.16

2016. Т. 29. № 4. С. 129–134

ОЦЕНКА СТЕПЕНИ ВЛИЯНИЯ АНТИВИРУСНЫХ ПРОГРАММНЫХ СРЕДСТВ НА КАЧЕСТВО ФУНКЦИОНИРОВАНИЯ ИНФОРМАЦИОННО-ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ

*Е.Б. Дроботун, к.т.н., докторант, drobotun@xakep.ru;**Д.В. Козлов, курсовой офицер, kozlov.den.vikt@mail.ru**(Военная академия воздушно-космической обороны им. Маршала Советского Союза Г.К. Жукова, ул. Жигарева, 50, г. Тверь, 170022, Россия)*

Наряду с качеством защиты информационно-вычислительных систем от компьютерных вирусов и вредоносных программ одним из важнейших показателей эффективности антивирусных программных средств является их ресурсоемкость. В идеальном случае антивирусное программное средство не должно оказывать никакого влияния на качество выполнения задач и функций, возложенных на защищаемую информационно-вычислительную систему. Вместе с тем следует отметить, что создать какую-либо систему защиты (в том числе и антивирусное программное средство как систему защиты от компьютерных вирусов и вредоносных программ), которая не оказывала бы никакого отрицательного влияния на защищаемый объект, практически невозможно. Исходя из этого, одним из критериев выбора антивирусного программного средства для защиты какой-либо информационно-вычислительной системы должна быть минимальная степень влияния выбранного антивирусного программного средства на качество выполнения задач и функций, возложенных на защищаемую систему.

В данной статье в качестве показателя, характеризующего степень влияния антивирусных программных средств на информационно-вычислительные системы, предлагается использовать обобщенный показатель снижения производительности защищаемой информационно-вычислительной системы, который, в свою очередь, определяется исходя из частных показателей снижения производительности при решении отдельных задач, возложенных на защищаемую систему, с учетом важности каждой задачи.

Также приведены результаты оценки степени влияния на информационно-вычислительную систему общего назначения семи широко распространенных антивирусных программных средств.

Ключевые слова: антивирусное программное средство, ресурсоемкость антивирусных программных средств, качество функционирования информационно-вычислительных систем, снижение производительности.

При выборе антивирусных программных средств (АВПС) для информационно-вычислительных систем различного назначения, помимо непосредственных характеристик эффективности этих средств по противодействию компьютерным вирусам и вредоносным программам, необходимо учитывать степень их влияния на качество функционирования защищаемых ими информационно-вычислительных систем.

Поскольку при работе все АВПС забирают часть вычислительных ресурсов (процессорное время, оперативную память и др.), влияние этих средств на защищаемую систему выражается в снижении производительности информационно-вычислительной системы, что, в свою очередь, может снизить качество выполнения возложенных на нее задач [1].

Исходя из этого, степень влияния АВПС на качество функционирования информационно-вычислительной системы можно оценить через показатель, характеризующий снижение производительности информационно-вычислительной системы при выполнении каждой из возложенных на нее задач [1, 2].

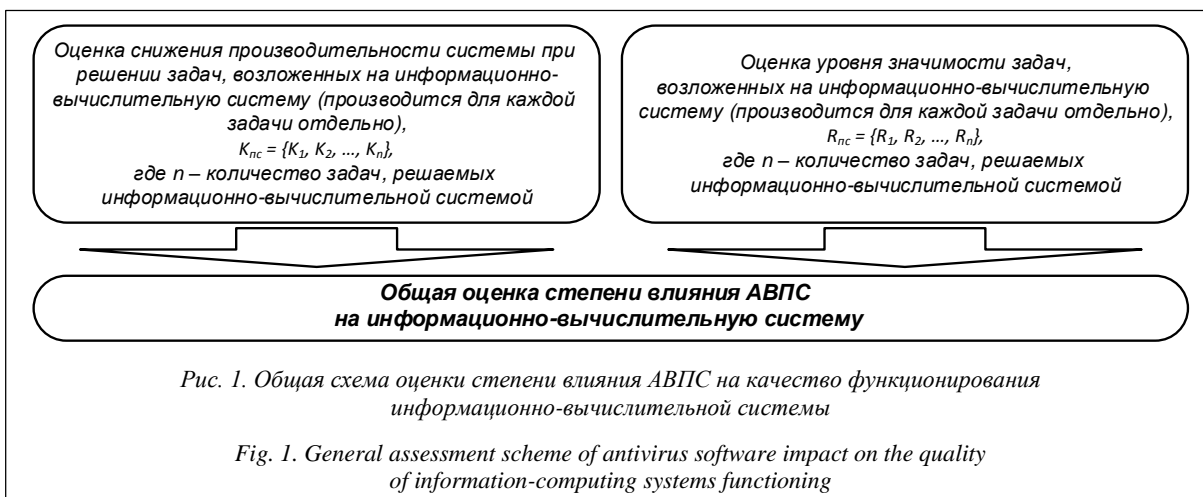
В настоящее время используются два подхода к оценке степени влияния АВПС на защищаемую информационно-вычислительную систему.

Первый подход [3–5] в качестве показателей, характеризующих влияние АВПС на защищаемую систему, использует показатели, характеризующие

расход ресурсов информационно-вычислительной системы:

- использование оперативной памяти в состоянии покоя и во время сканирования АВПС накопителей информации информационно-вычислительной системы;
- загруженность процессора в состоянии покоя и во время сканирования АВПС накопителей информации информационно-вычислительной системы;
- время чтения/записи данных на накопители информации в состоянии покоя и во время сканирования АВПС накопителей информации информационно-вычислительной системы;
- количество дочерних процессов АВПС;
- занимаемый АВПС объем на жестком диске информационно-вычислительной системы после установки;
- пропускная способность сканирования.

Перечисленные показатели достаточно просто оценить с помощью штатных средств оценки производительности, входящих в состав практически всех операционных систем. Однако с их помощью сложно оценить снижение качества выполнения отдельных задач, возложенных на информационно-вычислительную систему. К тому же, поскольку результат оценки представляется совокупностью нескольких разнородных показателей, задача выбора рационального (с точки зрения



потребления ресурсов) АВПС сводится к решению задачи многокритериального выбора, что представляет определенную сложность [6].

Второй подход [7–11] предполагает использование показателей падения производительности информационно-вычислительных систем при выполнении определенных типовых задач информационно-вычислительной системой с установленным АВПС, например:

- загрузка операционной системы;
- запуск каких-либо программ;
- загрузка каких-либо документов в программы, которые их обрабатывают (текстовые и графические редакторы, электронные таблицы и т.п.);
- загрузка данных из сети Интернет и др.

Как правило, при использовании второго подхода в качестве показателя, характеризующего падение производительности, используется время выполнения каждой из задач до установки АВПС в информационно-вычислительную систему и после установки АВПС. Для измерения времени выполнения задач используются различные программные средства (например, для оценки времени загрузки операционной системы – утилита Boot-Racer, а для оценки времени загрузки и запуска программ – утилита AppTimer).

Данный подход, в отличие от первого, позволяет оценить падение производительности информационно-вычислительной системы при выполнении отдельных задач, но также не лишен определенных недостатков. Во-первых, несмотря на то, что в большинстве случаев измеренные оценки однородны и представляют собой значения времени выполнения задач, они могут иметь большой разброс по количественным показателям (от сотых долей секунд до нескольких минут), а во-вторых, как и при использовании первого подхода, задача выбора рационального АВПС сводится к задаче многокритериального выбора.

Для устранения указанных недостатков и ограничений известных и применяемых способов

оценки снижения производительности информационно-вычислительных систем при их защите от компьютерных вирусов и вредоносных программ с помощью АВПС необходимо (рис. 1)

- произвести оценку снижения производительности для каждой из возложенных на информационно-вычислительную систему задач (при этом в качестве показателей снижения производительности использовать безразмерные величины в виде коэффициентов снижения производительности);
- произвести оценку уровня значимости (коэффициента важности) для каждой задачи, выполняемой информационно-вычислительной системой исходя из ее предназначения;
- произвести общую оценку степени влияния АВПС на производительность информационно-вычислительной системы путем свертки полученных коэффициентов снижения производительности для каждой из задач с учетом уровня их значимости (коэффициента важности).

Соответственно, методика оценки степени влияния АВПС на качество функционирования информационно-вычислительной системы будет являться комплексной и включать в себя три частные методики: методику оценки снижения производительности информационно-вычислительной системы при выполнении каждой из задач, возложенных на систему; методику оценки уровня значимости задач, возложенных на информационно-вычислительную систему; методику общей оценки степени влияния АВПС на качество функционирования информационно-вычислительной системы.

Методика оценки снижения производительности информационно-вычислительной системы при выполнении каждой из задач, возложенных на систему. Для оценки снижения уровня производительности информационно-вычислительных систем можно использовать как временные показатели, так и количественные.

Временные показатели характеризуют увеличение времени выполнения какой-либо задачи или

операции информационно-вычислительной системой: $\Delta t_{\text{оп}} = t'_{\text{оп}} - t_{\text{оп}}$, где $\Delta t_{\text{оп}}$ – прирост времени выполнения задачи или операции с учетом влияния АВПС; $t'_{\text{оп}}$ – время выполнения информационно-вычислительной системой задачи или операции с учетом влияния АВПС; $t_{\text{оп}}$ – время выполнения информационно-вычислительной системой задачи или операции без влияния АВПС.

Количественные показатели характеризуют уменьшение количества одновременно выполняемых информационно-вычислительной системой задач или операций за единицу времени: $\Delta k_{\text{оп}} = k_{\text{оп}} - k'_{\text{оп}}$, где $\Delta k_{\text{оп}}$ – снижение числа одновременно выполняемых задач или операций с учетом влияния АВПС; $k_{\text{оп}}$ – число одновременно выполняемых задач или операций до введения в состав информационно-вычислительной системы АВПС; $k'_{\text{оп}}$ – число одновременно выполняемых задач или операций после введения в состав информационно-вычислительной системы АВПС.

Переход к безразмерным коэффициентам снижения производительности ($K_{\text{ПС}}$) производится следующим образом: $K_{\text{ПС}} = \frac{\Delta k_{\text{оп}}}{k_{\text{оп}}}$ для определения

коэффициента снижения производительности по количественным показателям и $K_{\text{ПС}} = \frac{\Delta t_{\text{оп}}}{t_{\text{оп}}}$ для определения коэффициента снижения производительности по временным показателям.

Методика оценки уровня значимости задач, возложенных на информационно-вычислительную систему. Уровень определяется путем опроса экспертов с использованием метода парных сравнений. Данный метод позволяет произвести поочередное сравнение двух элементов, игнорируя все остальные, что значительно облегчает процесс принятия решения [12].

Для этого список всех задач заносится в таблицу парных сравнений (табл. 1), а далее каждый из экспертов принимает решение по поочередной оценке значимости каждой задачи при сравнении ее с остальными путем распределения своего голоса между двумя сравниваемыми задачами

(например, если значимость сравниваемых двух задач, по мнению эксперта, одинакова, в таблицу заносятся числа 0,5 и 0,5, или, если значимость первой задачи, по мнению эксперта, значительно превосходит значимость второй задачи, в таблицу заносятся числа 0,9 и 0,1).

Таблица 1

Пример заполнения экспертом таблицы парных сравнений для пяти задач

Table 1

An example of expert filling a pairwise comparison table for five objectives

Наименование задачи	Задача № 1	Задача № 2	Задача № 3	Задача № 4	Задача № 5
Задача № 1	-	0,7	0,4	0,2	0
Задача № 2	0,3	-	0,5	0,4	0,1
Задача № 3	0,6	0,5	-	0,9	0,3
Задача № 4	0,8	0,6	0,1	-	0,2
Задача № 5	1	0,9	0,7	0,8	-

Каждый эксперт заполняет свою таблицу, после чего формируется общая таблица парных сравнений всех экспертов, которая имеет такой же вид, что и таблица парных сравнений для одного эксперта, при этом значением r_{ij}^* каждой ячейки данной таблицы является сумма значений r_{ij} этой же ячейки в таблицах всех экспертов: $r_{ij}^* = \sum_{k=1}^m r_{ijk}$, где

m – число экспертов; $i = \overline{1, n}$, $j = \overline{1, n}$, n – число задач, возложенных на информационно-вычислительную систему.

Далее определяется суммарный уровень значимости (R_j^*) для j -й задачи (табл. 2): $R_j^* = \sum_{i=1}^n r_{ij}^*$.

Полученные значения нормируются по формуле $R_j = \frac{R_j^*}{\sum_{j=1}^n R_j^*}$ таким образом, чтобы выполнялось условие $\sum_{j=1}^n R_j = 1$.

Методика общей оценки степени влияния АВПС на качество функционирования информа-

Таблица 2

Пример оценки уровня значимости четырьмя экспертами для пяти задач

Table 2

An example of estimating the significance level by four experts for five tasks

Наименование задачи	Задача № 1	Задача № 2	Задача № 3	Задача № 4	Задача № 5	Суммарный уровень значимости j -й задачи (R_j^*)	Итоговое (нормированное) значение уровня значимости j -й задачи (R_j)
Задача №1	-	2,9	1,5	0,8	0,3	2,9+1,5+0,8+0,3=5,5	0,1375
Задача №2	1,1	-	2,1	1,6	1,3	1,1+2,1+1,6+1,3=6,1	0,1525
Задача №3	2,5	1,9	-	3,6	1,2	2,5+1,9+3,6+1,2=9,2	0,23
Задача №4	3,2	2,4	0,4	-	0,9	3,2+2,4+0,4+0,9=6,9	0,1725
Задача №5	3,7	2,7	2,8	3,1	-	3,7+2,7+2,8+3,1=12,3	0,3075

ционно-вычислительной системы. Исходя из того, что для каждого частного показателя на предыдущем этапе определены их коэффициенты важности, которые определяются значимостью задач, возложенных на информационно-вычислительную систему, а также из однородности этих показателей (выражены в виде безразмерных коэффициентов), общую оценку степени влияния АВПС на качество функционирования можно определить с помощью средневзвешенного арифметического обобщенного показателя [12, 13]:

$$Q = \sum_{i=1}^n K_i R_i.$$

С помощью полученных общих оценок для разных АВПС возможен выбор такого из них, которое оказывает наименьшее влияние на производительность информационно-вычислительных систем и, соответственно, на качество выполнения задач, возложенных на защищаемую информационно-вычислительную систему.

Оценка степени влияния наиболее распространенных АВПС на типовую информационно-вычислительную систему с помощью предложенных методик. Под типовой информационно-вычислительной системой в данном случае понимается персональный компьютер, ориентированный на решение общих задач (просмотр и редактирование документов, просмотр интернет-страниц, работа с архивами документов и т.п.). Конфигурация компьютера, использовавшегося при оценке степени влияния АВПС на качество решения задач: процессор – Intel Core i3-2120 CPU 3.30 GHz; объем и тип оперативной памяти – 2,00 Гб, DDR3; видеокарта – Intel HD Graphics Family; объем накопителя на жестком магнитном диске – 250 Гб; операционная система – Microsoft Windows 7 32 bit.

В качестве задач, возложенных на типовую информационно-вычислительную систему, были выбраны следующие:

- загрузка операционной системы;
- загрузка интернет-страницы в веб-браузер (интернет-страница – WebXakep.net, веб-браузер – Opera 12.0);
- загрузка документа в Microsoft Excel 2013 (документ объемом 13 483 байта);
- загрузка документа в Microsoft Word 2013 (документ объемом 11 888 байт, включая таблицы и рисунки);
- архивирование файлов (архивировались 3 файла общим объемом 129 536 байт, тип архива rar, архиватор WinRAR 5.0.1);
- очистка одного из локальных дисков (объем диска 3 Гб).

Всего было оценено семь наиболее распространенных АВПС: Kaspersky Internet Security 16.0.0.614 (KIS), Dr. Web Security Space 11.0, Eset NOD32 Smart Security, Microsoft Security Essentials,

McAfee LifeSafe, Symantec Endpoint Protection 12.1.6, Avast Premier.

Для каждого АВПС проводилась серия измерений по каждой задаче, включающая тридцать измерений. Измерялись время выполнения каждой задачи без установленного АВПС в информационно-вычислительной системе и время выполнения каждой задачи с каждым из вышеуказанных АВПС. После проведения серии измерений для одного АВПС производилось его полное удаление и установка в информационно-вычислительную систему следующего АВПС.

Время загрузки операционной системы измерялось с помощью утилиты BootRacer 4.9 (<http://bootracer.ru.uptodown.com/download/bootracer-4-9-en-win.zip>), время загрузки интернет-страницы в веб-браузер, время загрузки документов в Microsoft Excel и Microsoft Word, время очистки локального диска – с помощью программы AppTimer 1.0 (<http://freesoft.ru/apptimer/download/AppTimer.zip/getpage>), время архивирования файлов – с помощью встроенного в программу архивирования таймера.

На основании работы [14] можно определить точечную оценку времени выполнения задач (\bar{t}) как среднее арифметическое полученных в ходе измерений результатов:

$$\bar{t} = \frac{\sum_{i=1}^{30} t_i}{30}.$$

Полученные точечные оценки среднего времени выполнения каждой задачи и рассчитанные коэффициенты снижения производительности для каждой задачи приведены в таблице 3.

Оценка коэффициента значимости задач типовой информационно-вычислительной системы производилась пятью экспертами. Результаты оценивания и расчетов уровня значимости задач представлены в таблице 4.

Результаты общей оценки уровня влияния семи оцененных АВПС представлены на рисунке 2 в виде гистограммы. Из полученных результатов

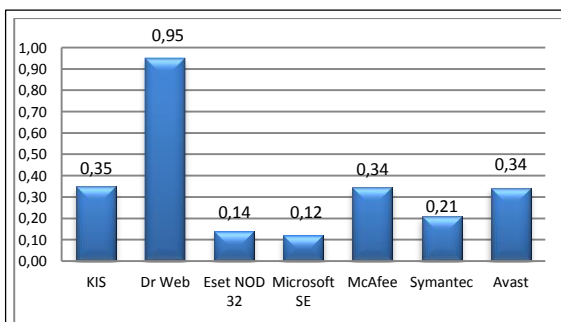


Рис. 2. Общая оценка степени влияния АВПС на типовую информационно-вычислительную систему с учетом значимости решаемых задач

Fig. 2. General assessment of the impact of antivirus software on a typical computing system taking into account the importance of tasks

Таблица 3

**Результаты оценки среднего времени (\bar{t}) и коэффициента (K) влияния АВПС
на задачи типовой информационно-вычислительной системы**

Table 3

**The results of an assessment of anti-virus software average time (\bar{t})
and an influence coefficient (K) on tasks of a typical information system**

Задача	Без антивируса		KIS		Dr Web		Eset NOD 32		Microsoft SE		McAfee		Symantec		Avast	
	\bar{t}	K	\bar{t}	K	\bar{t}	K	\bar{t}	K	\bar{t}	K	\bar{t}	K	\bar{t}	K	\bar{t}	K
Загрузка операционной системы (задача № 1)	22,065	-	36,547	0,656	30,789	0,395	36,533	0,655	42,268	0,915	26,453	0,199	36,403	0,649	42,840	0,942
Загрузка сайта WebXaker.net в веб-браузер Opera (задача № 2)	0,2312	-	0,4022	0,739	1,0601	3,585	0,2654	0,148	0,2381	0,03	0,3501	0,514	0,2597	0,123	0,3317	0,435
Загрузка документа в Microsoft Excel (задача № 3)	5,5574	-	7,0168	0,263	7,8613	0,415	6,0468	0,088	5,6517	0,17	7,4221	0,336	6,5914	0,186	7,0450	0,268
Загрузка документа в Microsoft Word (задача № 4)	6,4398	-	7,7245	0,199	7,6281	0,184	6,7961	0,055	6,6573	0,034	8,7242	0,355	7,6463	0,187	8,6536	0,344
Архивирование файлов (задача № 5)	18,94	-	20,56	0,086	19,23	0,015	20,01	0,057	19,98	0,055	23,89	0,261	22,15	0,169	20,34	0,074
Очистка одного из локальных дисков (задача № 6)	13,6704	-	15,7926	0,155	16,4047	0,2	14,8905	0,089	15,3937	0,126	16,6905	0,221	15,3443	0,122	15,3794	0,125

Таблица 4

Результаты оценки уровня значимости задач типовой информационно-вычислительной системы

Table 4

Results of the assessment of a task importance level of the of typical information systems

Задача	Задача № 1	Задача № 2	Задача № 3	Задача № 4	Задача № 5	Задача № 6	Суммарный уровень значимости j -й задачи (R^*j)	Итоговое (нормированное) значение уровня значимости j -й задачи (R_j)
Задача № 1	-	1,2	0,9	0,8	1,7	2,2	6,8	0,0907
Задача № 2	3,8	-	2,6	2,5	2,9	3,9	15,7	0,2093
Задача № 3	4,1	2,4	-	2,4	4,2	4,5	17,6	0,2347
Задача № 4	4,2	2,5	2,6	-	4,2	4,5	18,0	0,2400
Задача № 5	3,3	1,9	0,7	0,7	-	2,9	9,9	0,1320
Задача № 6	2,8	1,1	0,5	0,5	3,2	-	7,0	0,0933

оценки можно сделать вывод, что наименьшее влияние на информационно-вычислительную систему общего применения оказывают такие АВПС, как Microsoft Security Essential (общая оценка уровня влияния – 0,12) и ESET NOD 32 Smart Security (общая оценка уровня влияния – 0,14), а наибольшее влияние – Dr. Web Security Space 11.0 (общая оценка уровня влияния – 0,95). С полными результатами анализа степени влияния АВПС можно ознакомиться в [15].

Итак, предложенная комплексная методика позволяет проводить оценку степени влияния АВПС на качество функционирования информационно-вычислительных систем различного назначения. Полученные оценки дают возможность при выборе АВПС руководствоваться их эффективностью по противодействию компьютерным вирусам и вредоносным программам, а также учитывать степень их влияния на защищаемую информационно-вычислительную систему. Это, в свою очередь, позволит из всех АВПС, удовлетворяющих требованиям по эффективности противодействия компьютерным вирусам и вредоносным программам, выбрать наиболее рациональный (с точки зрения потребляемых ресурсов защищаемой информационно-вычислительной системы и влияния на качество ее функционирования).

Литература

1. Кустов Ф. Влияние антивирусов на производительность компьютера // Компьютерра. 2007. URL: <http://www.old.computerra.ru/gid/342272/> (дата обращения: 16.05.2016).
2. Дехт А. Влияние антивируса на производительность // IT News. 2012. URL: <http://www.it-bay.ru/?p=5507> (дата обращения: 16.05.2016).
3. Какой антивирус выбрать. 2013. URL: <http://www.itlife.kiev.ua/security/129-kakoj-antivirus-vybrat> (дата обращения: 16.05.2016).
4. Test wydajnościowy (performance test). AVLab. 2014. URL: https://www.avlab.pl/sites/default/files/articles/performance_test_2014.pdf (дата обращения: 16.05.2016).
5. AVLab: Тест антивирусов на быстродействие. 2014. URL: <http://www.comss.ru/page.php?id=1693> (дата обращения: 16.05.2016).
6. Дубов Ю.А., Травкин С.И., Якимец В.Н. Многокритериальные модели формирования и выбора вариантов систем. М.: Наука. Глав. ред. Физматлит, 1986. 296 с.
7. Картавенко М. Тест антивирусов на быстродействие. 2012. URL: http://www.anti-malware.ru/antivirus_test_performance_2012 (дата обращения: 16.05.2016).
8. Картавенко М. Тест корпоративных антивирусов на быстродействие. 2012. URL: http://www.anti-malware.ru/corporate_antivirus_test_performance_2012 (дата обращения: 16.05.2016).
9. Шабанов И. Сравнение скорости работы ведущих антивирусов. 2005. URL: http://www.anti-malware.ru/comparisons/compare_speed_antivirus (дата обращения: 16.05.2016).
10. Дроботун Е.Б. Самый быстрый антивирус // Хакер. 2013. № 7 (171). С. 92–94.
11. Колисниченко Д. X-тестирование: самый скоростной

Internet Security // Хакер. 2015. № 6 (197). С. 84–89.

12. Петровский А.Б. Теория принятия решений: учебник для студ. высш. учеб. заведений. М.: Академия, 2009. 400 с.

13. Демидов Б.А. Теория и методы военно-научных исследований вооружения и военной техники. Харьков: Изд-во ВИРТА ПВО, 1990. 558 с.

14. Гмурман В.Е. Теория вероятностей и математическая статистика: учеб. пособие для вузов. М.: Высш. шк., 2003. 479 с.

15. Результаты анализа степени влияния антивирусных программных средств. 2016. URL: https://yadi.sk/d/WBAC_04oriqM7 (дата обращения: 16.05.2016).

Software & Systems

DOI: 10.15827/0236-235X.116.129-134

Received 30.05.16

2016, vol. 29, no. 4, pp. 129–134

ASSESSMENT OF INFLUENCE OF ANTI-VIRUS SOFTWARE ON QUALITY OF INFORMATION-COMPUTING SYSTEM FUNCTIONING

E.B. Drobotun¹, Ph. D. (Engineering), Doctoral Student, drobotun@xakep.ru

D.V. Kozlov¹, Course Officer, kozlov.den.vikt@mail.ru

¹ Military Academy of the Aerospace Defence, Zhigarev St. 50, Tver, 170022, Russian Federation

Abstract. One of the most important indicators of the antivirus software effectiveness, along with the quality of information systems protection against computer viruses and malware, is resource intensity of these means. Ideally, antivirus software should not have any impact on the quality of tasks and functions assigned to a protected computing system. However, it should be noted that to build any kind of system (including anti-virus software as a protection system against computer viruses and malware) that would not have any adverse effect on a protected object is virtually impossible. Based on this fact, one of the criteria of choosing antivirus software to protect any information system should be a minimum degree of influence of selected antivirus software tools on the quality of the tasks and functions assigned to the protected system.

As an index of the degree of antivirus software influence on a computing system, the article proposes to use a generalized indicator of protected computing system performance degradation. This indicator is determined based on partial indicators of performance degradation when dealing with individual tasks assigned to the protected system, taking into account the importance of each task.

The article also presents the results of evaluation of the degree of seven common antivirus software influence on a general purpose computing system.

Keywords: antivirus software, resource intensity of antivirus software tools, quality of information computer system functioning, performance degradation.

References

1. Kustov F. The impact of antivirus software on PC performance. *Computerra* [Computerra]. 2007. Available at: <http://www.old.computerra.ru/gid/342272/> (accessed May 16, 2016).

2. Deht A. Effect of antivirus on performance. *IT News*. 2012. Available at: <http://www.it-bay.ru/?p=5507> (accessed May 16, 2016).

3. *Which antivirus to choose*. 2013. Available at: <http://www.itlife.kiev.ua/security/129-kakoj-antivirus-vybrat> (accessed May 16, 2016).

4. Test wydajnościowy (performance test). *AVLab*. 2014. Available at: https://www.avlab.pl/sites/default/files/articles/performance_test_2014.pdf (accessed May 16, 2016).

5. *AVLab: Test of antiviruses for speed*. 2014. Available at: <http://www.comss.ru/page.php?id=1693> (accessed May 16, 2016).

6. Dubov Yu.A., Travkin S.I., Yakimets V.N. *Mnogokriterialnye modeli formirovaniya i vybora variantov sistem* [Multicriteria Models of Forming and Choosing System Variants]. Moscow, Nauka Publ., 1986, 296 p.

7. Kartavenko M. *Test of antiviruses for speed*. 2012. Available at: http://www.anti-malware.ru/antivirus_test_performance_2012 (accessed May 16, 2016).

8. Kartavenko M. *Test korporativnykh antivirusov na bystrodeystvie* [Operation Speed Test of Corporate Antiviruses]. 2012. Available at: http://www.anti-malware.ru/corporate_antivirus_test_performance_2012 (accessed May 16, 2016).

9. Shabanov I. *Sravnenie skorosti raboty vedushchikh antivirusov* [Comparison of Leading Antiviruses Operation Speed]. 2005. Available at: http://www.anti-malware.ru/comparisons/compare_speed_antivirus (accessed May 16, 2016).

10. Drobotun E.B. The fastest antivirus. *Haker* [Hacker]. 2013, no. 7 (171), pp. 92–94 (in Russ.).

11. Kolisnichenko D. X-testing: the fastest Internet Security. *Haker* [Hacker]. 2015, no. 6 (197), pp. 84–89 (in Russ.).

12. Petrovsky A.B. *Teoriya prinyatiya resheny* [The Theory of Decision Making]. Textbook for students of the higher educational institutions, Moscow, Academy Publ. center, 2009, 400 p. (in Russ.).

13. Demidov B.A. *Teoriya i metody voenno-nauchnykh issledovaniy vooruzheniya i voennoy tekhniki* [Military Research Theory and Methods for Weapons and Military Equipment]. Kharkov, VIRTA PVO Publ., 1990, 558 p.

14. Gmurman V.E. *Teoriya veroyatnostey i matematicheskaya statistika* [The Probability Theory and Mathematical Statistics]. Textbook for higher educational institutions. Moscow, Vysshaya shkola Publ., 2003, 479 p.

15. *Rezultaty analiza stepeni vliyaniya antivirusnykh programmnykh sredstv* [Analysis Results of Antivirus Software Influence Extent]. 2016. Available at: https://yadi.sk/d/WBAC_04oriqM7 (accessed May 16, 2016).

УДК 510.52, 510.643

DOI: 10.15827/0236-235X.116.135-142

Дата подачи статьи: 06.10.16

2016. Т. 29. № 4. С. 135–142

АЛГОРИТМИЧЕСКАЯ ВЫРАЗИТЕЛЬНОСТЬ НЕКОТОРЫХ ФРАГМЕНТОВ ЯЗЫКА ЛОГИКИ ВЕТВЯЩЕГОСЯ ВРЕМЕНИ

А.В. Духовнева, тренер-консультант, kugusheva_nastya@list.ru

(Школа скороочтения и развития интеллекта IQ007, Тверской просп., 2, г. Тверь, 170100, Россия);

*М.Н. Рыбаков, к.ф.-м.н., доцент, инженер-программист, научный сотрудник,
m_rubakov@mail.ru*

(Тверской государственный университет, ул. Желябова, 33, г. Тверь, 170100, Россия;

НИИ «Центр программных систем», просп. 50 лет Октября, 3а, г. Тверь, 170024, Россия;

Университет Витвоотерсранда, просп. Яна Сметса, 1, г. Йоханнесбург, 2001, ЮАР);

Д.П. Шкатов, к.ф.н., доктор наук, доцент, shkatov@gmail.com

(Университет Витвоотерсранда, просп. Яна Сметса, 1, г. Йоханнесбург, 2001, ЮАР)

В работе рассматривается логика ветвящегося времени CTL и изучается вопрос о сложности проблемы ее разрешения в языке с конечным числом переменных. Приведен полиномиальный алгоритм, решающий задачу принадлежности формул константному фрагменту CTL. Приведен полиномиальный алгоритм, который погружает фрагмент CTL в язык с модальностями AX , EX , AG и EF во фрагмент CTL от одной переменной. Известно, что проблема разрешения фрагмента CTL в языке с модальностями AX , EX , AG и EF является EXPTIME-полной, в частности, не является полиномиально разрешимой. В результате с помощью построенного погружения доказано, что фрагменты CTL от одной и более переменных не являются полиномиально разрешимыми, то есть проблема разрешения фрагмента CTL от одной переменной является EXPTIME-полной. В заключение обсуждаются близкие вопросы, в частности, приводятся примеры других фрагментов логики CTL, имеющих более простую проблему разрешения, чем проблема разрешения CTL.

Ключевые слова: пропозициональные логики, неклассические логики, темпоральные логики, логика ветвящегося времени, проблема разрешения, вычислительная сложность.

Для описания различных свойств сложных систем и для их последующей верификации требуются формальные языки, предполагающие четкость и однозначность формулировок. Широко используемый в математике язык логики предикатов (см., например, [1]) во многих случаях позволяет создавать подобные описания, но в силу теоремы Чёрча–Тьюринга проблемы тождественной истинности, тождественной ложности и выполнимости формул этого языка алгоритмически неразрешимы (см., например, [2]). Это создает существенное препятствие в его использовании для решения соответствующих задач с применением вычислительной техники.

На данный момент уже созданы, изучены и используются различные классы формальных языков, логик и теорий, которые, с одной стороны, позволяют описывать свойства сложных систем, структур, вычислений и т.п., при этом, с другой стороны, многие важные проблемы для них алгоритмически разрешимы.

В данной работе речь пойдет об одном из таких языков – языке логики ветвящегося времени, которая известна в англоязычной литературе как *computational tree logic* и обычно обозначается CTL. Эта логика возникла относительно давно, и сейчас изучены многие ее свойства – алгоритмические, синтаксические, семантические и др.

Особенность языка CTL в том, что, во-первых, вычисления понимаются как пути, образованные состояниями, в которых последовательно оказывается система при выполнении некоторой про-

граммы, а во-вторых, в языке эти состояния не указываются явно: для указания на них используются так называемые темпоральные (временные) модальности типа «в каждом пути когда-нибудь будет, что...», «существует путь, в котором всегда верно, что...» и т.п. Использование подобных средств позволяет описывать различные свойства программных вычислений, в том числе параллельных, что вместе с разрешимостью проблемы выполнимости CTL-формул делает CTL эффективным инструментом как для создания таких описаний, так и для верификации программ (см., например, [3, 4]).

Однако за богатые возможности приходится платить. Но если в случае классической логики предикатов имеет место неразрешимость проблемы выполнимости, то в случае CTL – только высокая степень сложности проблемы разрешения [5, 6]. Известно, что проблема разрешения для CTL является EXPTIME-полной. Это означает, что, во-первых, существует алгоритм, решающий проблему выполнимости CTL-формул за экспоненциальное время от длины тестируемой формулы [7], а во-вторых, любой детерминированный алгоритм, решающий эту проблему, для некоторых формул будет затрачивать время, ограниченное снизу экспонентой от длины этих формул [8]. Другими словами, в классе детерминированных алгоритмов самыми быстрыми решающими проблему выполнимости CTL-формул являются экспоненциальные, и понизить эту оценку, например, до полиномиальной, невозможно.

Нечто похожее справедливо для всех логик, содержащих в себе классическую логику высказываний CL (classical logic): проблема выполнимости CL-формулы является NP-полной, и в предположении, что $P \neq NP$, она тоже не решается быстрее, чем экспоненциально. Поскольку, как правило, известные логики содержат в себе CL в качестве естественного фрагмента, проблема их разрешения не может быть ниже.

Тем не менее CL содержит довольно выразительные фрагменты, которые полиномиально разрешимы. Например, полиномиально разрешима проблема выполнимости CL-формулы, находящейся в совершенной дизъюнктивной или совершенной конъюнктивной нормальной форме (СДНФ, СКНФ): для выполнимости формулы в СДНФ достаточно, чтобы она содержала хотя бы один дизъюнктивный член, а для выполнимости формулы в СКНФ достаточно, чтобы число попарно различных конъюнктивных членов в ней было меньше чем 2^n , где n – число переменных, входящих в эту формулу.

Другой пример – это любой фрагмент CL, состоящий из формул, построенных из переменных, входящих в некоторое фиксированное конечное множество. Действительно, если имеется всего одна переменная, то достаточно проверять истинность формул на двух наборах истинностных значений для нее, если имеются две переменные – на четырех наборах, если три – на восьми наборах и т.д. При этом каждый раз число таких наборов будет фиксированным, следовательно, проблема выполнимости формул из соответствующего фрагмента требует для ее решения лишь полиномиальных (не более чем квадратичных) затрат времени.

Пусть L – некоторая логика, n – неотрицательное целое число. Через $L(n)$ обозначим фрагмент логики L , состоящий из формул от фиксированных n переменных. В частности, $L(0)$ – константный фрагмент L . Тогда, согласно сказанному выше, для всякого целого неотрицательного n фрагмент $L(n)$ полиномиально разрешим. Можно ли утверждать то же самое о CTL(n)? Ответу на этот вопрос, а также его обоснованию и посвящена данная работа.

Ниже будет показано, что фрагмент CTL(0) действительно полиномиально разрешим, и будет приведен соответствующий разрешающий алгоритм. Что касается фрагментов CTL(n) при $n > 0$, то здесь ситуация иная. Для некоторых модальных логик доказано, что проблема разрешения их фрагментов от конечного числа переменных является столь же сложной, как и проблема разрешения для логики в целом [9–13]; аналогичные результаты справедливы и для некоторых других неклассических логик [14, 15]. Покажем, что техника, изложенная в [13, 16], применима к CTL, доказав с ее помощью, что при любом $n > 0$ фрагмент CTL(n) является EXPTIME-полным.

Синтаксис и семантика CTL

Будем считать, что формулы строятся из констант \perp (ложь) и \top (истина), а также счетного множества пропозициональных переменных с помощью связок \wedge (конъюнкция), \vee (дизъюнкция), \rightarrow (импликация), \neg (отрицание), \leftrightarrow (эквивалентность) и модальностей AX , EX , AF , EF , AG , EG , AU , EU . При этом первые шесть модальностей являются одноместными и позволяют из формулы φ получать формулы $AX\varphi$, $EX\varphi$, $AF\varphi$, $EF\varphi$, $AG\varphi$ и $EG\varphi$, а последние две являются двухместными и позволяют из формул φ и ψ получать формулы $(\varphi AU\psi)$ и $(\varphi EU\psi)$. Каждая из этих модальностей состоит из двух частей: первая – это A или E , а вторая – это X , F , G или U . Эти части являются самостоятельными модальностями в более общем языке – языке логики, известной как CTL*, но в CTL они используются только парами. Модальности A и E называются кванторами пути и соответствуют выражениям «для каждого пути верно, что...» (A – от "all") и «существует такой путь, что...» (E – от "exists"). Модальности X , F , G , U называются кванторами состояний. Эти буквы взяты из слов "next", "future", "globally", "until" и соответствуют выражениям «в следующем состоянии данного пути...», «в некотором состоянии данного пути...», «в каждом состоянии данного пути...» и «в состояниях данного пути справедливо..., пока не станет справедливо...». Так, $AF\varphi$ означает, что «в каждом пути существует состояние, в котором верно φ », а формула $(\varphi EU\psi)$ – что «существует путь, в состояниях которого верно φ , пока не станет верным ψ ».

Такое описание смысла CTL-формулы можно назвать интуитивно понятным, но все же оно не является точным. Чтобы сделать его точным, будем использовать семантику С. Крипке.

Шкала Крипке – это набор $\mathcal{F} = \langle W, R \rangle$, где W – непустое множество, элементы которого будем называть состояниями, а R – серийное бинарное отношение на W , называемое отношением достижимости (серийность означает, что для любого состояния $x \in W$ существует такое состояние $y \in W$, что xRy). Если выполнено условие xRy , то говорим, что из x достижимо y .

Путем π в шкале $\mathcal{F} = \langle W, R \rangle$ будем называть бесконечную последовательность $\pi_0, \pi_1, \pi_2, \pi_3, \dots$ состояний (необязательно различных), в которой для каждого k выполнено отношение $\pi_k R \pi_{k+1}$; при этом π_0 будем называть началом пути π . Заметим, что в силу серийности отношения R каждое состояние является началом хотя бы одного пути.

Оценкой в шкале $\mathcal{F} = \langle W, R \rangle$ будем называть функцию v , которая каждой пропозициональной переменной p сопоставляет некоторое подмножество $v(p)$ множества W . Содержательно условие $x \in v(p)$ означает, что переменная p истинна в состоянии x .

Модель Крипке – это набор $\mathcal{M} = \langle W, R, v \rangle$, где $\langle W, R \rangle$ – шкала Крипке, а v – оценка в этой шкале.

Определим отношение истинности CTL-формулы в состояниях модели $\mathcal{M} = \langle W, R, v \rangle$. Тот факт, что формула φ истинна в состоянии x модели \mathcal{M} , будем обозначать как $\mathcal{M}, x \models \varphi$. Определение дадим обычным образом:

- $\mathcal{M}, x \models \perp$;
- $\mathcal{M}, x \models T$;
- $\mathcal{M}, x \models p$, если $x \in v(p)$;
- $\mathcal{M}, x \models \varphi \wedge \psi$, если $\mathcal{M}, x \models \varphi$ и $\mathcal{M}, x \models \psi$;
- $\mathcal{M}, x \models \varphi \vee \psi$, если $\mathcal{M}, x \models \varphi$ или $\mathcal{M}, x \models \psi$;
- $\mathcal{M}, x \models \varphi \rightarrow \psi$, если $\mathcal{M}, x \not\models \varphi$ или $\mathcal{M}, x \models \psi$;
- $\mathcal{M}, x \models \neg \varphi$, если $\mathcal{M}, x \not\models \varphi$;
- $\mathcal{M}, x \models \varphi \leftrightarrow \psi$, если $\mathcal{M}, x \models \varphi$ тогда и только тогда, когда $\mathcal{M}, x \models \psi$;
- $\mathcal{M}, x \models AX\varphi$, если для каждого пути π , начинающегося в x , верно, что $\mathcal{M}, \pi_1 \models \varphi$;
- $\mathcal{M}, x \models AF\varphi$, если в каждом пути π , начинающемся в x , существует такое состояние π_k , что $\mathcal{M}, \pi_k \models \varphi$;
- $\mathcal{M}, x \models AG\varphi$, если в каждом пути π , начинающемся в x , для каждого состояния π_k верно, что $\mathcal{M}, \pi_k \models \varphi$;
- $\mathcal{M}, x \models \varphi AU\psi$, если в каждом пути π , начинающемся в x , существует состояние π_k , для которого $\mathcal{M}, \pi_k \models \psi$, при этом для каждого m , строго меньшего k , верно, что $\mathcal{M}, \pi_m \models \varphi$;
- $\mathcal{M}, x \models EX\varphi$, если существует такой начинающийся в x путь π , что $\mathcal{M}, \pi_1 \models \varphi$;
- $\mathcal{M}, x \models EF\varphi$, если существует такой начинающийся в x путь π , для некоторого состояния π_k которого верно, что $\mathcal{M}, \pi_k \models \varphi$;
- $\mathcal{M}, x \models EG\varphi$, если существует такой начинающийся в x путь π , для каждого состояния π_k которого верно, что $\mathcal{M}, \pi_k \models \varphi$;
- $\mathcal{M}, x \models \varphi EU\psi$, если в некотором начинающемся в x пути π существует состояние π_k , для которого $\mathcal{M}, \pi_k \models \psi$, при этом для каждого m , строго меньшего k , верно, что $\mathcal{M}, \pi_m \models \varphi$.

Формула φ считается истинной в модели, если φ истинна в каждом состоянии этой модели. Формула φ считается истинной в шкале, если φ истинна в каждой модели, определенной на этой шкале.

Под CTL понимаем множество всех CTL-формул, истинных в каждой шкале Крипке.

Разрешающая процедура для константного фрагмента CTL

Опишем несложную синтаксическую процедуру, которая по произвольной константной CTL-формуле дает ответ на вопрос о принадлежности этой формулы логике CTL.

Заметим, что, во-первых, каждая безмодальная константная формула, согласно законам классической логики высказываний (или попросту табли-

цам истинности для булевых связей), эквивалентна в CTL либо константе \perp , либо константе T . Во-вторых, формулы $AX\perp$, $EX\perp$, $AF\perp$, $EF\perp$, $AG\perp$ и $EG\perp$ эквивалентны в CTL формуле \perp , а формулы AXT , EXT , AFT , EFT , AGT и EGT – формуле T . В-третьих, в CTL справедливы следующие эквивалентности:

$$\begin{aligned} (\perp AU \perp) &\leftrightarrow \perp; & (\perp AUT) &\leftrightarrow T; \\ (T AU \perp) &\leftrightarrow \perp; & (T AUT) &\leftrightarrow T; \\ (\perp EU \perp) &\leftrightarrow \perp; & (\perp EUT) &\leftrightarrow T; \\ (T EU \perp) &\leftrightarrow \perp; & (T EUT) &\leftrightarrow T. \end{aligned}$$

Принимая во внимание сказанное, а также то, что для CTL справедлива теорема о замене эквивалентных, получаем следующую процедуру, преобразующую любую константную CTL-формулу либо к формуле \perp , либо к формуле T :

- находим самое внутреннее вхождение логической связки или модальности;
- если его нет, останавливаемся;
- иначе заменяем подформулу, содержащую только это вхождение, на \perp или T в соответствии с эквивалентностями, описанными выше;
- возвращаемся к первому действию.

Эта процедура решает задачу принадлежности константных формул логике CTL: если в результате ее применения получается T , то исходная формула принадлежит CTL, если получается \perp , то не принадлежит.

Заметим, что описанные в процедуре действия придется выполнить столько раз, сколько логических связей и модальностей содержит исходная константная формула. Следовательно, приведенная процедура имеет полиномиальную (не более чем квадратичную) временную сложность.

Предварительная техническая конструкция

Для доказательства того, что аналогичной процедуры не существует для CTL(n) при $n > 0$, проведем небольшую техническую подготовку.

Зафиксируем переменную p . Определим следующие формулы, зависящие от p :

$$\begin{aligned} \delta_1 &= EXAGp, \delta_{k+1} = EX(p \wedge EX(\neg p \wedge \delta_k)); \\ \alpha_k &= p \wedge EX(\neg p \wedge \delta_k \wedge \neg \delta_{k+1} \wedge EFAG\neg p); \\ \beta_k &= EX\alpha_k. \end{aligned}$$

Содержательно δ_1 утверждает, что на следующем шаге вычисления переменная p становится истинной, после чего ее истинность сохраняется навсегда. Формула δ_2 утверждает, что, сделав шаг вычисления, получим p , еще через шаг – $\neg p$, после чего будет δ_1 . В целом δ_k выражает следующее условие: k раз чередуются пары состояний, где сначала истинно p , потом $\neg p$, и после этого возникает состояние, где истинно δ_1 .

Условие, определяемое формулой α_k , сложнее. Но в данном случае нет необходимости описывать его полностью: достаточно будет установить, что в

некотором классе моделей для каждой формулы α_k существует ровно одна модель, ровно в одном из состояний которой α_k истинна.

Определим такой класс моделей.

Для этого рассмотрим шкалу Крипке, образованную двумя путями σ и τ , имеющими два общих состояния r и a , где $r = \sigma_0 = \tau_0$ и $a = \sigma_1 = \tau_1$, при этом все остальные состояния в σ и τ попарно различны. Обозначим эту шкалу \mathcal{F}_{2C} (рис. 1).

Для каждого целого $k > 0$ определим оценку v_k в шкале \mathcal{F}_{2C} : сделаем переменную p истинной в состояниях $\tau_0, \tau_2, \tau_4, \dots, \tau_{2k-2}$ и в каждом состоянии τ_m при $m \geq 2k$. В результате получим модель $\mathcal{M}_k = \langle \mathcal{F}_{2C}, v_k \rangle$ (рис. 2).

Лемма 1. $\mathcal{M}_k, x \models \alpha_m \Leftrightarrow k = m$ и $x = r$.

Доказательство. Пусть $\mathcal{M}_k, x \models \alpha_m$. Тогда в состоянии x истинны переменная p и формула $EX(\neg p \wedge \delta_k \wedge \neg \delta_{k+1} \wedge EFAG\neg p)$. Из истинности в x формулы $EX(\neg p \wedge \delta_k \wedge \neg \delta_{k+1} \wedge EFAG\neg p)$ следует, что из x выходит путь π , в состоянии π_1 которого истинны $\neg p \wedge \delta_m \wedge \neg \delta_{m+1}$ и $EFAG\neg p$. Заметим, что формула $EFAG\neg p$ истинна только в состояниях пути σ . Формула $\neg p \wedge \delta_m \wedge \neg \delta_{m+1}$ истинна только в состоянии $\tau_{2(k-m)+1}$ (при условии, что $k \geq m$). Из сказанного заключаем, что для π_1 имеется лишь одна возможность: $\pi_1 = \sigma_1 = \tau_1 = a$. Это, в частности, означает, что $2(k-m) + 1 = 1$. Но тогда $k = m$ и $x = \pi_0 = \tau_0 = r$.

Пусть теперь $k = m$ и $x = r$. Тогда получаем, что $\mathcal{M}_k, a \models \delta_k \wedge \neg \delta_{k+1}$. Кроме того, $\mathcal{M}_k, a \models \neg p$. Значит, $\mathcal{M}_k, a \models \neg p \wedge \delta_k \wedge \neg \delta_{k+1}$. В каждом состоянии пути σ истинна формула $EFAG\neg p$, следовательно, $\mathcal{M}_k, a \models EFAG\neg p$. Значит, в a истинна формула $\neg p \wedge \delta_k \wedge \neg \delta_{k+1} \wedge EFAG\neg p$, и следовательно, $\mathcal{M}_k, r \models EX(\neg p \wedge \delta_k \wedge \neg \delta_{k+1} \wedge EFAG\neg p)$. Наконец,

учитывая, что $\mathcal{M}_k, r \models p$, получаем, что $\mathcal{M}_k, r \models p \wedge EX(\neg p \wedge \delta_k \wedge \neg \delta_{k+1} \wedge EFAG\neg p)$, то есть $\mathcal{M}_k, r \models \alpha_k$.

Лемма доказана.

Эта лемма нужна для реализации следующей идеи. Предположим, что имеется состояние s , в котором истинны переменные p_1, p_3, p_5 и ложны переменные p_2, p_4, p_6 . Для простоты будем считать, что из s достижимо только само состояние s . Теперь вместо переменных p_1, \dots, p_6 возьмем формулы β_1, \dots, β_6 . Конечно же, эти формулы будут ложны в s . Но, если сделать достижимыми из s корни моделей $\mathcal{M}_1, \mathcal{M}_3, \mathcal{M}_5$, то $\beta_1, \beta_3, \beta_5$ будут истинны в s , а $\beta_2, \beta_4, \beta_6$ останутся ложными. В результате становится возможным моделировать переменную p_k формулой β_k , используя модель \mathcal{M}_k вместо оценки для p_k .

Теперь кажется разумным в произвольной формуле подставить вместо каждой переменной p_k формулу β_k и получить желаемое погружение CTL в CTL(1). Однако такая подстановка не дает требуемого, так как некоторые формулы, не принадлежащие CTL, будут преобразованы в формулы, принадлежащие CTL: например, формула $EF\neg p_1$ не принадлежит CTL, но ее подстановочный пример $EF\neg \beta_1$ логике CTL принадлежит. Покажем, как обойти указанную трудность.

Модификация CTL-формул

Прежде чем выполнить подстановку β_k вместо p_k , модифицируем CTL-формулы. Заметим, что для доказательства EXPTIME-трудности проблемы разрешения CTL достаточно использовать только модальности AX, EX, AG и EF . Это следует из идеи,

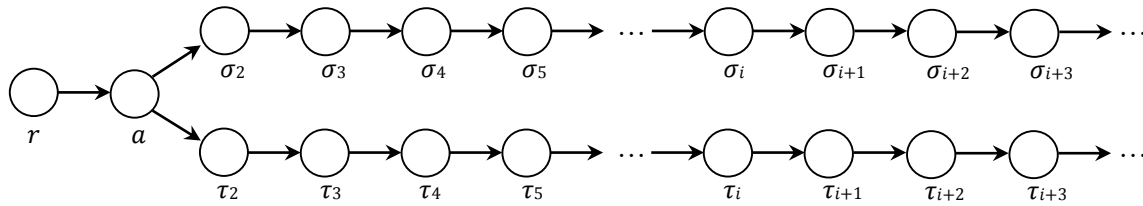


Рис. 1. Шкала \mathcal{F}_{2C}

Fig. 1. Frame \mathcal{F}_{2C}

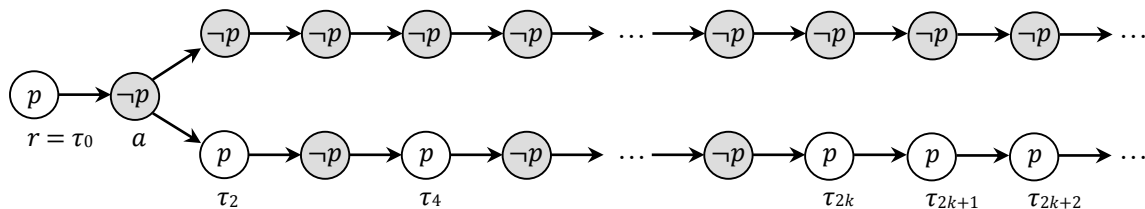


Рис. 2. Модель \mathcal{M}_k

Fig. 2. Model \mathcal{M}_k

изложенной в [8], где доказывалось EXPTIME-трудность другой логики – PDL (propositional dynamic logic), причем для доказательства оказалось достаточно использовать лишь модальности, соответствующие модальностям AX , EX , AG и EF (непринципиальное отличие состоит в том, что в шкалах для PDL не требуется серийность, но она допускается). Итак, пусть дана формула φ , зависящая от n переменных p_1, \dots, p_n и не содержащая модальностей, отличных от AX , EX , AG и EF . Возьмем новую переменную p_{n+1} . Пусть f – преобразование, определенное следующим образом:

$$\begin{aligned} f(\perp) &= \perp; & f(\top) &= \top; & f(p_k) &= p_k; \\ f(\varphi_1 \wedge \varphi_2) &= f(\varphi_1) \wedge f(\varphi_2); \\ f(\varphi_1 \vee \varphi_2) &= f(\varphi_1) \vee f(\varphi_2); \\ f(\varphi_1 \rightarrow \varphi_2) &= f(\varphi_1) \rightarrow f(\varphi_2); \\ f(\varphi_1 \leftrightarrow \varphi_2) &= f(\varphi_1) \leftrightarrow f(\varphi_2); \\ f(\neg\varphi_1) &= \neg f(\varphi_1); \\ f(AX\varphi_1) &= AX(p_{n+1} \rightarrow f(\varphi_1)); \\ f(EX\varphi_1) &= EX(p_{n+1} \wedge f(\varphi_1)); \\ f(AG\varphi_1) &= AG(p_{n+1} \rightarrow f(\varphi_1)); \\ f(EF\varphi_1) &= EF(p_{n+1} \wedge f(\varphi_1)). \end{aligned}$$

Другими словами, f превращает модальности AX , EX , AG и EF в ограниченные модальности, при этом роль «ограничителя» играет переменная p_{n+1} . Наконец, определим преобразование g :

$$g(\varphi) = (p_{n+1} \wedge AG(p_{n+1} \rightarrow EX p_{n+1})) \rightarrow f(\varphi).$$

Формула $g(\varphi)$ и будет той формулой, вместо переменных которой будем подставлять определенные выше формулы от одной переменной.

Лемма 2. Если $\varphi \in \text{CTL}$, то $g(\varphi) \in \text{CTL}$.

Доказательство. Пусть $\varphi \in \text{CTL}$. Предположим, что $g(\varphi) \notin \text{CTL}$. Тогда существует модель $\mathcal{M} = \langle W, R, v \rangle$, в некотором состоянии s которой опровергается формула $g(\varphi)$. Это означает, что $\mathcal{M}, s \models p_{n+1} \wedge AG(p_{n+1} \rightarrow EX p_{n+1})$ и $\mathcal{M}, s \not\models f(\varphi)$.

Пусть $\mathcal{M}' = \langle W', R', v' \rangle$, где W' состоит из тех состояний множества W , которые достижимы за сколько-нибудь шагов из s и в которых переменная p_{n+1} истинна, R' – ограничение отношения R на W' , а v' – ограничение оценки v на W' , то есть $v'(p_k) = v(p_k) \cap W'$. Тогда \mathcal{M}' – модель Крипке: действительно, $W' \neq \emptyset$, так как $\mathcal{M}, s \models p_{n+1}$, а серийность отношения R' обеспечивается тем, что $\mathcal{M}, s \models AG(p_{n+1} \rightarrow EX p_{n+1})$.

Индукцией по построению подформулы ψ формулы φ покажем, что для всякого состояния $x \in W'$ выполняется следующая эквивалентность:

$$\mathcal{M}', x \models \psi \Leftrightarrow \mathcal{M}, x \models f(\psi).$$

Базис индукции (то есть когда $\psi = \perp$, $\psi = \top$ или $\psi = p_k$) выполняется по определению оценки v' . Обоснуем индукционный шаг.

Пусть доказываемое утверждение верно для формул ψ_1 и ψ_2 и пусть $x \in W'$.

Если $\psi = \psi_1 \wedge \psi_2$, то имеем следующую цепочку эквивалентностей:

$$\begin{aligned} \mathcal{M}', x \models \psi &\Leftrightarrow \\ \Leftrightarrow \mathcal{M}', x \models \psi_1 \wedge \psi_2 &\Leftrightarrow \end{aligned}$$

$$\begin{aligned} &\Leftrightarrow \mathcal{M}', x \models \psi_1 \text{ и } \mathcal{M}', x \models \psi_2 \Leftrightarrow \\ &\Leftrightarrow \mathcal{M}, x \models f(\psi_1) \text{ и } \mathcal{M}, x \models f(\psi_2) \Leftrightarrow \\ &\Leftrightarrow \mathcal{M}, x \models f(\psi_1) \wedge f(\psi_2) \Leftrightarrow \\ &\Leftrightarrow \mathcal{M}, x \models f(\psi). \end{aligned}$$

Обоснование тех случаев, когда $\psi = \psi_1 \vee \psi_2$, $\psi = \psi_1 \rightarrow \psi_2$, $\psi = \neg\psi_1$ или $\psi = \psi_1 \leftrightarrow \psi_2$, выполняется аналогично.

Пусть теперь $\psi = EX\psi_1$.

Если $\mathcal{M}', x \models \psi$, то в \mathcal{M}' существует начинающийся в x путь π , для которого $\mathcal{M}', \pi_1 \models \psi_1$. По индукционному предположению для ψ_1 получаем, что $\mathcal{M}, \pi_1 \models f(\psi_1)$. Кроме того, $\pi_1 \in W'$, а значит, $\mathcal{M}, \pi_1 \models p_{n+1}$. Таким образом, $\mathcal{M}, \pi_1 \models p_{n+1} \wedge f(\psi_1)$, следовательно, $\mathcal{M}, \pi_0 \models EX(p_{n+1} \wedge f(\psi_1))$, то есть $\mathcal{M}, x \models f(\psi)$.

Если $\mathcal{M}, x \models f(\psi)$, то в \mathcal{M} существует путь π с началом в x , для которого $\mathcal{M}, \pi_1 \models p_{n+1} \wedge f(\psi_1)$. Поскольку $\mathcal{M}, \pi_1 \models p_{n+1}$, получаем, что $\pi_1 \in W'$. Значит, можем применить индукционное предположение и сделать вывод, что $\mathcal{M}', \pi_1 \models \psi_1$. Возьмем произвольный путь ε в модели \mathcal{M}' , для которого $\varepsilon_0 = x = \pi_0$ и $\varepsilon_1 = \pi_1$. Тогда $\mathcal{M}', \varepsilon_1 \models \psi_1$, а значит, $\mathcal{M}', \varepsilon_0 \models EX\psi_1$, то есть $\mathcal{M}', x \models \psi$.

В тех случаях, когда $\psi = EF\psi_1$, $\psi = AX\psi_1$ или $\psi = AG\psi_1$, обоснование выполняется аналогично.

С учетом того, что $s \in W'$ и $\mathcal{M}, s \not\models f(\varphi)$, из доказанного получаем, что $\mathcal{M}', s \not\models \varphi$. Но тогда $\varphi \notin \text{CTL}$, а это противоречит выбору φ . Значит, сделанное в начале доказательства предположение о том, что $g(\varphi) \notin \text{CTL}$, неверно, следовательно, $g(\varphi) \in \text{CTL}$.

Лемма доказана.

Заметим, что в модели \mathcal{M}' , определенной в доказательстве леммы 2, $v'(p_{n+1}) = W'$. Оформим это наблюдение в виде следствия.

Следствие 1. Если $g(\varphi) \in \text{CTL}$, то существует модель, в которой опровергается $g(\varphi)$ и при этом переменная p_{n+1} истинна в каждом состоянии этой модели.

Лемма 3. Если $g(\varphi) \in \text{CTL}$, то $\varphi \in \text{CTL}$.

Доказательство. Пусть $g(\varphi) \in \text{CTL}$. Пусть h – функция, заменяющая в формулах p_{n+1} на \top . Тогда $h(g(\varphi)) = (\top \wedge AG(\top \rightarrow EX\top)) \rightarrow h(f(\varphi))$. Логика CTL замкнута относительно операции подстановки формул вместо переменных, следовательно, $h(g(\varphi)) \in \text{CTL}$. Несложно убедиться, что формула $\top \wedge AG(\top \rightarrow EX\top)$ истинна во всех шкалах логики CTL, а значит, то же верно и для $h(f(\varphi))$. Следовательно, $h(f(\varphi)) \in \text{CTL}$.

Заметим, что p_{n+1} входит в $f(\varphi)$ только внутри подформулы, имеющих вид $p_{n+1} \rightarrow \psi$ или $p_{n+1} \wedge \psi$. После применения преобразования h эти подформулы будут иметь вид $\top \rightarrow h(\psi)$ и $\top \wedge h(\psi)$. Каждая из формул $\top \rightarrow h(\psi)$ и $\top \wedge h(\psi)$ эквивалентна в CTL формуле $h(\psi)$. Если ψ содержит p_{n+1} , то h заменит p_{n+1} на \top и, как было показано, получившиеся вхождения \top можно будет «удалить». Если же p_{n+1} не входит в ψ , то h не внесет в ψ изменений.

Таким образом, подстановка T вместо p_{n+1} в $f(\varphi)$ равнозначна тому, что p_{n+1} «удаляется» из $f(\varphi)$, то есть формула $h(f(\varphi))$ эквивалентна в CTL исходной формуле φ . Но $h(f(\varphi)) \in \text{CTL}$, а значит, $\varphi \in \text{CTL}$.

Лемма доказана.

Погружение CTL во фрагмент CTL от одной переменной

Необходимые предварительные построения завершены, и теперь можно определить функцию, погружающую фрагмент CTL с модальностями AX , EX , AG и EF во фрагмент CTL от одной переменной. Для произвольной CTL-формулы ψ , зависящей от p_1, \dots, p_{n+1} , определим $t(\psi)$ как результат подстановки в ψ формул $\beta_1, \dots, \beta_{n+1}$ вместо переменных p_1, \dots, p_{n+1} соответственно.

Теорема. $\varphi \in \text{CTL} \Leftrightarrow t(g(\varphi)) \in \text{CTL}$.

Доказательство. Пусть $\varphi \in \text{CTL}$. Тогда, согласно лемме 2, $g(\varphi) \in \text{CTL}$. Поскольку $t(g(\varphi))$ является результатом подстановки формул в $g(\varphi)$, получаем, что $t(g(\varphi)) \in \text{CTL}$.

Пусть $t(g(\varphi)) \in \text{CTL}$. Предположим, $\varphi \notin \text{CTL}$. Тогда из леммы 3 следует, что $g(\varphi) \notin \text{CTL}$. По следствию 1, существует модель $\mathcal{M} = \langle W, R, v \rangle$, в некотором состоянии s которой опровергается $g(\varphi)$ и при этом $v(p_{n+1}) = W$.

Расширим \mathcal{M} до модели $\mathcal{M}^* = \langle W^*, R^*, v^* \rangle$ следующим образом. Возьмем по одной копии каждой из моделей $\mathcal{M}_1, \dots, \mathcal{M}_{n+1}$ и добавим их к \mathcal{M}^* . Если в состоянии x исходной модели \mathcal{M} истинна переменная p_k , то в модели \mathcal{M}^* делаем достижимым из x корень модели \mathcal{M}_k . При этом оценку для p сохраняем в состояниях моделей $\mathcal{M}_1, \dots, \mathcal{M}_{n+1}$, а в состояниях модели \mathcal{M} полагаем p ложной.

Покажем, что в этом случае для всякого $x \in W$ и для всякой подформулы ψ формулы φ выполняется следующая эквивалентность:

$$\mathcal{M}, x \models f(\psi) \Leftrightarrow \mathcal{M}^*, x \models t(f(\psi)).$$

Доказательство проведем индукцией по построению ψ .

Если $\psi = \perp$, то $t(f(\psi)) = t(\perp) = \perp$; если $\psi = \top$, то $t(f(\psi)) = t(\top) = \top$. В обоих случаях требуемая эквивалентность тривиально выполняется.

Если $\psi = p_k$, то $f(\psi) = p_k$, следовательно, $t(f(\psi)) = t(p_k) = \beta_k = EX\alpha_k$. Если $\mathcal{M}, x \models f(\psi)$, то по построению модели \mathcal{M}^* из состояния x достижим корень модели \mathcal{M}_k , и тогда $\mathcal{M}^*, x \models EX\alpha_k$ (в силу леммы 1), то есть $\mathcal{M}^*, x \models t(f(\psi))$. Пусть теперь $\mathcal{M}^*, x \models t(f(\psi))$, то есть $\mathcal{M}^*, x \models EX\alpha_k$. Это возможно, только если из x достижимо состояние u , где истинна формула α_k . Но α_k содержит конъюнктивный член p , а значит, в u истинна переменная p , и следовательно, $u \in W$. Но тогда u – это состояние одной из моделей $\mathcal{M}_1, \dots, \mathcal{M}_{n+1}$, и по лемме 1 это корень модели \mathcal{M}_k . По построению \mathcal{M}^* корень модели \mathcal{M}_k может быть достижим из x ,

только если $\mathcal{M}, x \models p_k$. Поскольку $f(\psi) = p_k$, получаем, что $\mathcal{M}, x \models f(\psi)$.

Пусть доказываемое утверждение верно для формул ψ_1 и ψ_2 . Пусть также $x \in W$.

Если $\psi = \psi_1 \wedge \psi_2$, то имеем следующую цепочку эквивалентностей:

$$\begin{aligned} \mathcal{M}, x \models f(\psi) & \Leftrightarrow \\ \Leftrightarrow \mathcal{M}, x \models f(\psi_1) \wedge f(\psi_2) & \Leftrightarrow \\ \Leftrightarrow \mathcal{M}, x \models f(\psi_1) \text{ и } \mathcal{M}, x \models f(\psi_2) & \Leftrightarrow \\ \Leftrightarrow \mathcal{M}^*, x \models t(f(\psi_1)) \text{ и } \mathcal{M}^*, x \models t(f(\psi_2)) & \Leftrightarrow \\ \Leftrightarrow \mathcal{M}^*, x \models t(f(\psi_1)) \wedge t(f(\psi_2)) & \Leftrightarrow \\ \Leftrightarrow \mathcal{M}^*, x \models t(f(\psi)). & \end{aligned}$$

Обоснование тех случаев, когда $\psi = \psi_1 \vee \psi_2$, $\psi = \psi_1 \rightarrow \psi_2$, $\psi = \neg\psi_1$ или $\psi = \psi_1 \leftrightarrow \psi_2$, выполняется аналогично.

Пусть теперь $\psi = EF\psi_1$. Тогда получаем, что $f(\psi) = EF(p_{n+1} \wedge \psi_1)$ и, согласно определению t , $t(f(\psi)) = EF(\beta_{n+1} \wedge t(f(\psi_1)))$.

Если $\mathcal{M}, x \models f(\psi)$, то в \mathcal{M} существует начинающийся в x путь π , в некотором состоянии π_m которого истинна формула $p_{n+1} \wedge f(\psi_1)$. По построению модели \mathcal{M}^* получаем, что $\mathcal{M}^*, \pi_m \models \beta_{n+1}$, а по индукционному предположению для формулы ψ_1 – что $\mathcal{M}^*, \pi_m \models t(f(\psi_1))$. Значит, в состоянии π_0 истинна формула $EF(\beta_{n+1} \wedge t(f(\psi_1)))$, то есть $\mathcal{M}^*, x \models t(f(\psi))$.

Если $\mathcal{M}^*, x \models t(f(\psi))$, то в \mathcal{M}^* существует начинающийся в x путь π , в некотором состоянии π_m которого истинна формула $\beta_{n+1} \wedge t(f(\psi_1))$. Согласно построению модели \mathcal{M}^* , формула β_{n+1} может быть истинной в \mathcal{M}^* только в состояниях из множества W . Значит, $\pi_m \in W$. Но тогда можно применить предположение индукции, а значит, $\mathcal{M}, \pi_m \models f(\psi_1)$. Учитывая, что в каждом состоянии модели \mathcal{M} истинна переменная p_{n+1} , получаем, что $\mathcal{M}, \pi_0 \models EF(p_{n+1} \wedge f(\psi_1))$, то есть $\mathcal{M}, x \models f(\psi)$.

В тех случаях, когда $\psi = EX\psi_1$, $\psi = AX\psi_1$ или $\psi = AG\psi_1$, обоснование выполняется аналогично.

С учетом того, что $\mathcal{M}, s \models g(\varphi)$, получаем, что $\mathcal{M}, s \models p_{n+1} \wedge AG(p_{n+1} \rightarrow EXp_{n+1})$ и $\mathcal{M}, s \models f(\varphi)$. Из второго условия с учетом доказанного следует, что $\mathcal{M}^*, s \models t(f(\varphi))$. По построению модели \mathcal{M}^* получаем, что $\mathcal{M}, s \models \beta_{n+1} \wedge AG(\beta_{n+1} \rightarrow EX\beta_{n+1})$. Следовательно, $\mathcal{M}^*, s \models t(g(\varphi))$, а это противоречит тому, что $t(g(\varphi)) \in \text{CTL}$. Значит, предположение о том, что $\varphi \notin \text{CTL}$, является неверным, и поэтому $\varphi \in \text{CTL}$.

Теорема доказана.

Сложность фрагментов CTL от одной и более переменных

Заметим, что формула $t(g(\varphi))$ строится по φ полиномиально. Тогда доказанная теорема означает, что существует полиномиальное погружение EXPTIME-полного фрагмента логики CTL в ее фрагмент от одной переменной. Следовательно,

фрагмент CTL от одной переменной является EXPTIME-трудным. В силу принадлежности CTL классу EXPTIME он также принадлежит EXPTIME. Значит, этот фрагмент является EXPTIME-полным.

Следствие 2. Фрагмент логики CTL, образованный формулами от одной переменной, является EXPTIME-полным.

На самом деле доказано более сильное утверждение.

Следствие 3. Фрагмент CTL, образованный формулами от одной переменной, построенными с помощью булевых связок и модальностей AX , EX , AG и EF , является EXPTIME-полным.

Конечно же, оба результата останутся справедливыми, если использовать больше одной переменной: добавление переменных не понижает сложность задачи, при этом не может и повысить ее, так как проблема разрешения CTL находится в классе EXPTIME.

Таким образом, фактически доказано, что всего лишь одной переменной в языке логики CTL достаточно, чтобы проблема разрешения CTL оказалась «практически нерешаемой»: самые быстрые алгоритмы, решающие эту проблему, имеют экспоненциальную оценку снизу для времени их работы. Тем не менее, отметим, что для доказательства EXPTIME-трудности проблемы разрешения CTL были использованы специальные формулы; если же ограничиться, скажем, формулами, содержащими только AX и EX или только AG и EF , то получим PSPACE-полные фрагменты CTL, эквивалентные известным модальным логикам D и S4 соответственно. С помощью описанного метода логики D и S4 также погружаются в собственные фрагменты от одной переменной [10, 11], но обратим внимание на другое. Несмотря на то, что проблема разрешения CTL и даже, казалось бы, очень простых фрагментов CTL алгоритмически весьма трудна, CTL содержит довольно выразительные фрагменты, проблема разрешения которых ниже (например, принадлежит классу PSPACE, как в случае указанных фрагментов, эквивалентных D и S4), а иногда значительно ниже (например, принадлежит классу P, как в случае константного фрагмента), чем для CTL в целом. На данный момент для многих неклассических пропозициональных

логик найдены и описаны некоторые их нетривиальные фрагменты, имеющие относительно невысокую степень сложности проблемы разрешения, и было бы интересно провести аналогичные исследования в отношении CTL, а также найти соответствующие алгоритмы.

Работа выполнена при поддержке РФФИ, гранты №№ 14-06-00298 и 16-07-01272.

Литература

1. Мендельсон Э. Введение в математическую логику. М.: Наука, 1976, 320 с.
2. Булос Дж., Джеффри Р. Вычислимость и логика. М.: Мир, 1994. 396 с.
3. Кларк Э.М., Грамберг О., Пелед Д. Верификация моделей программ: Model checking. М.: Изд-во МЦНМО, 2002. 416 с.
4. Карпов Ю.Г. Model checking: Верификация параллельных и распределенных программных систем. СПб: БХВ-Петербург, 2010. 560 с.
5. Johnson D.S. Catalogue of complexity classes. J. van Leeuwen, ed., Handbook of Theoretical Computer Science, MIT Press, 1994, vol. A, pp. 67–162.
6. Papadimitriou C. Computational complexity. Addison-Wesley, 1994, 540 p.
7. Emerson E.A., Halpern J.Y. Decision procedures and expressiveness in the temporal logic of branching time. Jour. of Comp. and Syst. Sciences, 1985, vol. 30, no. 1, pp. 1–24.
8. Fisher M.J., Ladner R.E. Propositional Dynamic Logic of Regular Programs. Jour. of Comp. and Syst. Sc., 1979, vol. 18, pp. 194–211.
9. Spaan E. Complexity of modal logics. PhD Thesis. Univ. of Amsterdam, 1993, 130 p.
10. Halpern J.Y. The effect of bounding the number of primitive propositions and the depth of nesting on the complexity of modal logic. Artificial Intelligence, 1995, vol. 75, no. 2, pp. 361–372.
11. Chagrov A.V., Rybakov M.N. How many variables one needs to prove PSPACE-hardness of modal logics? Advances in Modal Logic, London, King's College Publ., 2003, vol. 4, pp. 71–82.
12. Švejdar V. The decision problem of provability logic with only one atom. Arch. Math. Logic, 2003, pp. 1–6.
13. Рыбаков М.Н. Сложность константного фрагмента пропозициональной динамической логики // Вестн. Тверского гос. ун-та. Сер.: Прикладная математика. 2007. Вып. 5. № 11 (39). С. 5–17.
14. Rybakov M.N. Complexity of intuitionistic propositional logic and its fragments. Jour. of Applied Non-Classical Logics, 2008, vol. 18, no. 2–3, pp. 267–292.
15. Rybakov M.N. Complexity of intuitionistic and Visser's basic and formal logic in finitely many variables. Advances in Modal Logic, London, King's College Publ., 2006, vol. 6, pp. 393–411.
16. Rybakov M.N. Complexity of finite-variable fragments of EXPTIME-complete logics. Jour. of Applied Non-Classical Logics, 2007, vol. 17, no. 3, pp. 359–382.

Software & Systems

DOI: 10.15827/0236-235X.116.135-142

Received 06.10.16

2016, vol. 29, no. 4, pp. 135–142

ALGORITHMIC POWER OF SOME FRAGMENTS OF COMPUTATIONAL TREE LOGIC

A.V. Dukhovneva¹, Trainer and Consultant, *kugusheva_nastya@list.ru*

M.N. Rybakov^{2, 3, 4}, Ph.D. (Physics and Mathematics), Associate Professor, Software Engineer, Research Fellow, *m_rybakov@mail.ru*

D.P. Shkatov⁴, Ph.D. (Philosophy), Ph.D. (Theoretical Computer Science), Senior Lecturer, *shkatov@gmail.com*

¹ School of Speed Reading and Development of Intellect IQ007, Tverskoy Ave. 2, Tver, 170100, Russian Federation

² Tver State University, Zhelyabov St. 33, Tver, 170100, Russian Federation

³ R&D Institute Centerprogramsystem, 50 let Oktyabrya Ave. 3a, Tver, 170024, Russian Federation

⁴ University of the Witwatersrand, 1 Jan Smuts Ave, Braamfontein, Gauteng, 2001, South Africa, Johannesburg

Abstract. In the paper we consider the Computational Tree Logic CTL and study computational complexity of the decision problem for its finitely-many variable fragments. We give a polynomial-time algorithm solving the decision problem for the variable-free fragment of CTL. We also give a polynomial-time algorithm which embeds the fragment of CTL with the modalities *AX*, *EX*, *AG* and *EF* into its one-variable fragment. The fragment of CTL with the modalities *AX*, *EX*, *AG* and *EF* is known to be EXPTIME-complete, in particular, it is not decidable in polynomial time. As a corollary, using the mentioned embedding we prove that *n*-variable fragment of CTL is not decidable in polynomial time, for any strictly positive integer *n*. More exactly, we prove that the decision problem for the one-variable fragment of CTL is EXPTIME-complete. In the conclusion, we discuss some related questions and, in particular, we give examples of other fragments of CTL with more simple decision problem than the one for CTL.

Keywords: propositional logics, non-classical logics, temporal logics, computational tree logic, decision problem, computational complexity.

Acknowledgements. The work is supported by RFBR, projects 14-06-00298 and 16-07-01272.

References

1. Mendelson E. *Introduction to Mathematical Logic*. D. van Nostrand Comp., Inc., 1964, 300 p.
2. Boolos G.S., Jeffrey R.C. *Computability and Logic*. Cambridge Univ. Press, 1989, 304 p.
3. Clarke E.M., Grumberg O., Peled D. *Model checking*. MIT Press, 1999, 314 p.
4. Karpov Yu.G. *Model checking: Verifikatsiya paralelnykh i raspredelennykh sistem* [Model checking: verification of parallel and distributive systems] St. Petersburg, BHV-Peterburg, 2010, 560 p. (in Russ.).
5. Johnson D.S. Catalogue of complexity classes. *Handbook of theor. comp. sc.*, MIT Press, 1994, vol. A, pp. 67–162.
6. Papadimitriou C. *Computational complexity*. Addison–Wesley, 1994, 540 p.
7. Emerson E.A., Halpern J.Y. Decision procedures and expressiveness in the temporal logic of branching time. *Jour. Comp. and Syst. Sc.*, vol. 30, no. 1, 1985, pp. 1–24.
8. Fisher M.J., Ladner R.E. Propositional dynamic logic of regular programs. *Jour. Comp. and Syst. Sc.*, 1979, vol. 18, pp. 194–211.
9. Spaan E. Complexity of modal logics. *PhD Thesis*. Univ. of Amsterdam, 1993, 130 p.
10. Halpern J.Y. The effect of bounding the number of primitive propositions and the depth of nesting on the complexity of modal logic. *Artificial Intelligence*, 1995, vol. 75, no. 2, pp. 361–372.
11. Chagrov A.V., Rybakov M.N. How many variables one needs to prove PSPACE-hardness of modal logics? *Advances in Modal Logic*, 2003, vol. 4, pp. 71–82.
12. Švejdar V. The decision problem of provability logic with only one atom. *Arch. Math. Logic*, 2003, pp. 1–6.
13. Rybakov M.N. Complexity of variable-free fragment of propositional dynamic logic. *Vestnik TsvGU*. [Bul. Tver State Univ.], 2007, vol. 5, no. 11, pp. 5–17 (in Russ.).
14. Rybakov M.N. Complexity of intuitionistic propositional logic and its fragments. *Jour. Applied Non-Classical Logics*, 2008, vol. 18, no. 2–3, pp. 267–292.
15. Rybakov M.N. Complexity of intuitionistic and Visser's basic and formal logic in finitely many variables. *Advances in Modal Logic*, 2006, vol. 6, pp. 393–411.
16. Rybakov M.N. Complexity of finite-variable fragments of EXPTIME-complete logics. *Jour. Applied Non-Classical Logics*, 2007, vol. 17, no. 3, pp. 359–382.

Примеры библиографического описания статьи

1. Духовнева А.В., Рыбаков М.Н., Шкатов Д.П. Алгоритмическая выразительность некоторых фрагментов языка логики ветвящегося времени // Программные продукты и системы. 2016. Т. 29. № 4. С. 135–142; DOI: 10.15827/0236-235X.116.135-142.

2. Dukhovneva A.V., Rybakov M.N., Shkatov D.P. Algorithmical power of some fragments of computational tree logic. *Programmye produkty i sistemy* [Software & Systems]. 2016, vol. 29, no. 4, pp. 135–142 (in Russ.); DOI: 10.15827/0236-235X.116.135-142.

УДК 519.688

DOI: 10.15827/0236-235X.116.143-148

Дата подачи статьи: 03.10.16

2016. Т. 29. № 4. С. 143–148

ПАРАЛЛЕЛЬНЫЕ ВЫЧИСЛЕНИЯ КАК СРЕДСТВО ПОВЫШЕНИЯ ЭФФЕКТИВНОСТИ РЕШЕНИЯ ЗАДАЧ ВЫЧИСЛИТЕЛЬНОЙ АЭРОДИНАМИКИ

С.А. Буренков, аспирант, burenkovsa@mpei.ru;

О.Ю. Шамаева, к.т.н., доцент, shamayevaou@mpei.ru

(Национальный исследовательский университет «Московский энергетический институт»,
ул. Красноказарменная, 14, 111250, г. Москва, Россия)

Предлагаются программные средства повышения эффективности решения класса аэродинамических задач на высокопроизводительных вычислительных системах кластерного типа. Программные средства реализованы на основе параллельно-последовательного алгоритма решения разреженных систем линейных алгебраических уравнений большой размерности методом обобщенных минимальных невязок.

В статье дается теоретическая оценка трудоемкости разработанного алгоритма, а также выполнено сравнение результатов теоретической оценки и результатов вычислительного эксперимента.

Программные средства реализованы для модели вычислений с распределенной памятью и использованы для решения модельной задачи обтекания профиля воздушным потоком. Приведены результаты вычислительного эксперимента по расчету плотности и давления обтекающего профиль воздушного потока на высокопроизводительном кластере, иллюстрирующие эффект от применения параллельной программы. В статье исследованы характеристики ускорения параллельного решения модельной задачи, позволившие определить порог размерности решаемых систем уравнений, ниже которого организация параллельных вычислений отрицательно сказывается на эффективности использования вычислительных ресурсов. Исследовано также влияние степени разреженности матриц коэффициентов на ускорение параллельной реализации.

Область применения результатов настоящей работы – не только вычислительная гидро- и газодинамика, но и энергетика, биология, химия, фармакология, астрофизика и др.

Ключевые слова: вычислительная аэродинамика, задача обтекания, параллельные вычисления, модель передачи сообщений MPI.

Развитие современного авиастроения требует решения разнообразных сложных задач, таких как обеспечение безопасности авиаперевозок, автоматизация систем управления полетами, создание двигателей, работающих на альтернативном топливе, и многих других. Помимо этого, не перестает расти потребность в увеличении грузоподъемности летательных аппаратов, скорости их полета и тяги силовых установок. Решение большинства из перечисленных проблем можно получить, прямо или косвенно прибегая к помощи аэродинамических механизмов расчета разного рода течений.

Аэродинамические характеристики обтекающего какой-либо объект потока можно определить в ходе натурного эксперимента, который проводится в специальной аэродинамической трубе. Эта процедура достаточно трудоемкая, дорогостоящая, а ее точность существенно зависит от условий проведения. Тем не менее до конца прошлого века аэродинамические испытания проводились исключительно средствами натурного эксперимента.

В настоящее время решение задачи аэродинамического обтекания можно получить с использованием методов математического моделирования. Такой подход позволяет с большой степенью точности определить все необходимые характеристики без использования громоздких стендов и проведения дорогостоящих натурных испытаний. Однако моделирование аэродинамических процессов относится к классу сложных задач [1, 2] в силу

высокой трудоемкости, больших объемов необходимой памяти, а также наличия специфичных для аэродинамики свойств. Например, расчет аэродинамических характеристик турбулентного течения с числом Рейнольдса порядка 10^5 в канале методом прямого численного моделирования на архитектуре производительностью 1 ТФлоп/с займет 122 года [1]. Поэтому процессы моделирования требуют использования специальных численных методов, параллельных технологий и программ, ориентированных на высокопроизводительную вычислительную технику, такую как суперкомпьютеры и кластерные системы.

На сегодняшний день известны несколько подходов к моделированию аэродинамических течений, среди которых метод прямого численного моделирования (Direct Numerical Simulation, DNS), метод моделирования крупных вихрей (Large Eddy Simulation, LES), метод моделирования разделенных вихрей (Detached Eddy Simulation, DES) и др. [1, 2]. Многие модели, такие как LES и DES, основываются на существовании средней скорости потока и среднего отклонения от нее. Поэтому реализация осредненных моделей проигрывает по точности моделирования модели DNS, но требует значительно меньшей вычислительной работы. Высокая вычислительная трудоемкость препятствует использованию DNS при решении комплексных задач аэродинамики, таких как обтекание планера самолета воздушным потоком, даже с использованием высокопроизводительной суперкомпьютер-

ной техники. На основании оценки развития вычислительных ресурсов, выполненной в работе [1], применительно к решению задачи обтекания среднего гражданского самолета или автомобиля расчет методом DNS можно будет производить за приемлемое время к концу XXI века. Более того, данная оценка является неоправданно оптимистичной, так как предполагает рост производительности аппаратных средств в два раза каждые пять лет. Появившиеся не так давно подходы к моделированию турбулентных течений (LES, DES и др.), как правило, менее трудоемки, но их использование увеличивает погрешность получаемого решения. К тому же каждый из них имеет свои преимущества, недостатки и ограниченную область применения [2]. Тем не менее использование DNS оправдано при моделировании обтекания объектов несложной геометрии.

Постановка задачи аэродинамического обтекания

В работе рассматривается задача аэродинамического обтекания профиля воздушным потоком. Предлагаются способы повышения эффективности ее решения на высокопроизводительной вычислительной технике с использованием механизма параллельных вычислений. Требуется определить основные аэродинамические характеристики – скорость, давление, число Маха и др. при обдувании воздушным потоком профиля RAE2822, изображенного на рисунке 1.

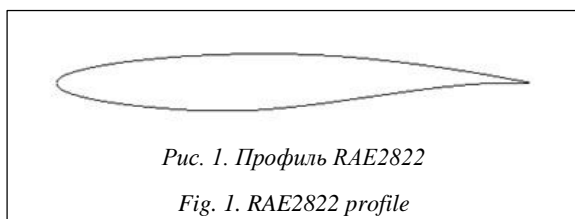


Рис. 1. Профиль RAE2822

Fig. 1. RAE2822 profile

Определить указанные характеристики потока можно, решив систему дифференциальных уравнений Навье–Стокса:

$$\rho \frac{d\mathbf{W}}{dt} = \mathbf{R} - \text{grad}(p) + \mu \Delta \mathbf{W} + \frac{1}{3} \mu \cdot \text{grad}(\text{div} \mathbf{W}). \quad (1)$$

Получить аналитическое решение системы (1) для большинства практических задач не представляется возможным, поэтому систему Навье–Стокса решают численными методами, для чего вокруг обдуваемого профиля строится расчетная область, изображенная на рисунке 2, которая разбивается неравномерной прямоугольной расчетной сеткой, как показано на рисунке 3.

После дискретизации системы (1) дальнейшее решение задачи сводится к системе линейных алгебраических уравнений (СЛАУ) большой размерности, матрица которой обладает свойствами разреженности и нерегулярности портрета, отсутствием диагонального преобладания и симметрии.

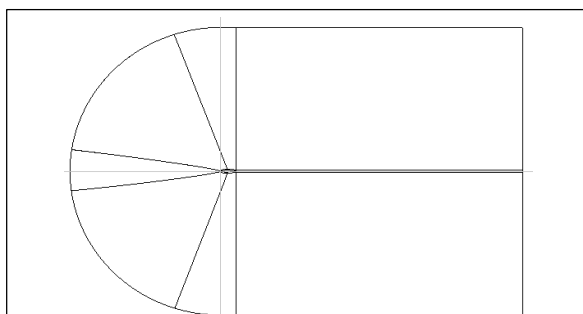


Рис. 2. Расчетная область для профиля RAE2822

Fig. 2. Computational domain for RAE2822 profile

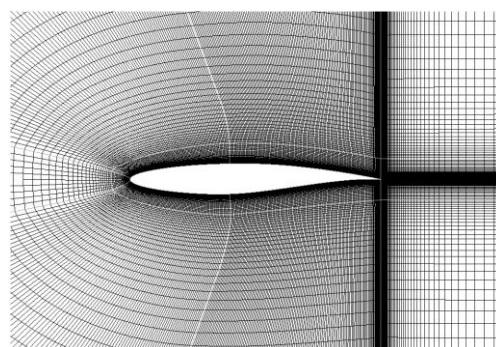


Рис. 3. Фрагмент расчетной сетки профиля RAE2822

Fig. 3. A detail of RAE2822 profile computational grid

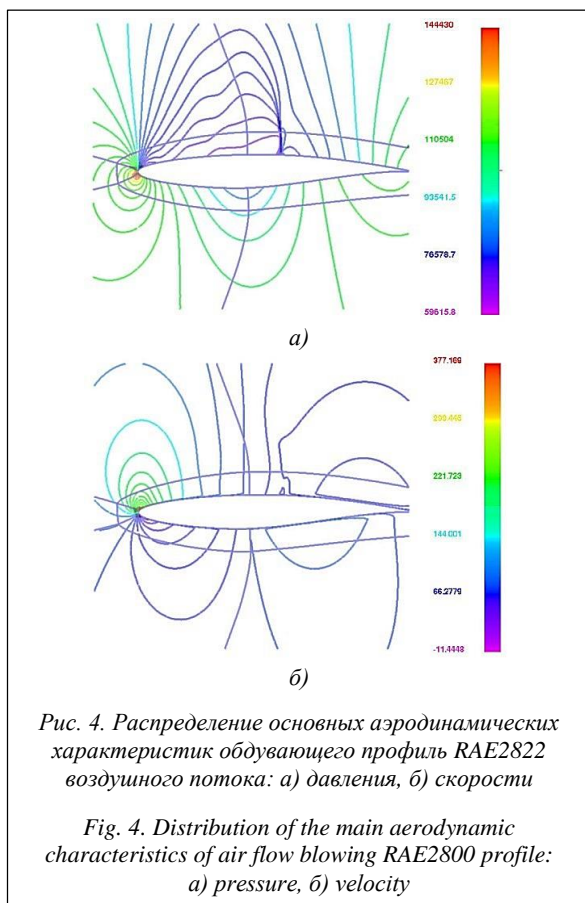
Результаты решения задачи обтекания могут быть представлены графически в виде распределения основных аэродинамических характеристик вокруг обдуваемого профиля. На рисунке 4 изображены распределения давления и скорости в воздушном потоке.

Организация параллельно-последовательных вычислений

На основе анализа методов решения СЛАУ с учетом перечисленных свойств матрицы коэффициентов [3, 4] предпочтение было отдано проекционному методу обобщенных минимальных невязок (General Minimal Residual, GMRES) [5], на основе которого в работе предложена и реализована параллельная модификация GMRES для кластерных архитектур.

Метод обобщенных минимальных невязок относится к классу методов проекционного типа и основывается на проектировании в подпространствах Крылова [5–7].

На рисунке 5 изображена структурная схема параллельного алгоритма решения СЛАУ по методу GMRES, вычисления в которой представлены в виде совокупности взаимодействующих последовательно выполняемых процессов. Для организации взаимодействия используется коммуникационная библиотека передачи сообщений (Message Passing Interface, MPI) [8, 9].



Главный процесс рассылает подчиненным заданные размерность подпространства Крылова m , требуемую точность решения eps , максимальное количество итераций $maxNumIter$, а также соответствующие им части матрицы коэффициентов A и вектора правой части b (блоки 2 и 3). В качестве начального приближения к решению выбирается нулевой вектор. Производится предварительная оценка невязки начального приближения r , и вычисляется ее норма $beta$: i -й процесс производит вычисление фрагмента вектора невязки $r^{(i)}$, после чего фрагменты собираются главным процессом (блок 5).

Далее происходит итерационный процесс вычислений (блоки 6–11), завершающийся при достижении заданной точности eps или при выполнении максимального числа итераций. На каждой итерации процессы совместно формируют базисные вектора w , которые представляются в матричном виде H (блок 8). Полученная матрица H приводится к форме Хессенберга (блок 9) и используется в качестве матрицы коэффициентов треугольной СЛАУ $H \cdot y = g$ (блок 10). Решение полученной треугольной СЛАУ используется при уточнении решения x исходной системы уравнений, полученного на предыдущей итерации. Наконец, по вышеописанному принципу вычисляются новая невязка системы r и ее норма $beta$ (блок 11). Отметим,

что наиболее вычислительно трудоемкие фрагменты алгоритма выполняются параллельно (блоки 5, 8 и 11).

Предложенный алгоритм обладает свойством масштабируемости и предполагает равномерную загрузку всех доступных вычислительных узлов кластера как исходными данными, так и непосредственно вычислительной работой [10].

Оценка трудоемкости параллельного алгоритма

Оценим вычислительную сложность реализованного параллельного алгоритма на множестве вычислительных узлов p при следующих предположениях. Пусть n – размерность решаемой СЛАУ, m – размерность подпространства Крылова, L – количество выполненных итераций, S – длительность аддитивной арифметической операции, P – длительность мультипликативной арифметической операции, d – доля ненулевых элементов в матрице коэффициентов, которая определяется как

$$d = \frac{N - N_0}{N}, \text{ где } N - \text{количество всех коэффициентов матрицы, а } N_0 - \text{количество коэффициентов, равных нулю. Тогда трудоемкость } T \text{ представленного в работе алгоритма может быть оценена с помощью формулы}$$

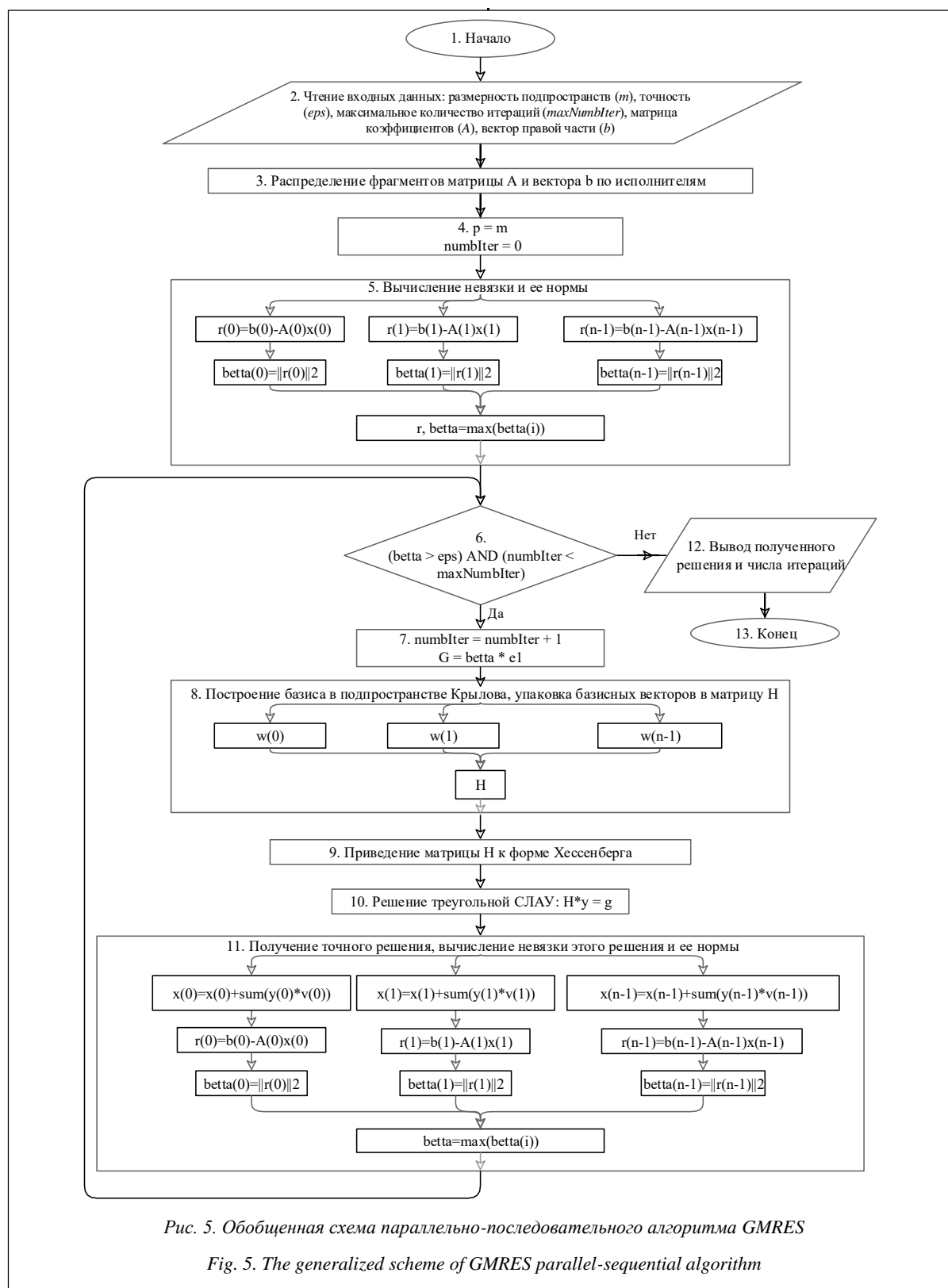
$$T(n, p, d, m, L, S, P) = O\left(\frac{n^3}{p} L \cdot d \cdot [m+1] \cdot S + \frac{n^2}{p} L \cdot d \cdot [m+1] \cdot P\right). \quad (2)$$

В работе выполнено сравнение теоретической оценки трудоемкости алгоритма с экспериментальными данными. На рисунке 6 представлены графики зависимости времени решения СЛАУ с помощью разработанной модификации GMRES от количества используемых вычислительных узлов. Для этой же системы уравнений был проведен вычислительный эксперимент, результаты которого отмечены на графике пунктиром.

Отметим, что время решения СЛАУ в ходе эксперимента немного больше предсказанного оценкой независимо от количества используемых вычислительных узлов. Этот эффект объясняется тем, что оценка (2) не учитывает временные затраты на чтение исходных данных и затраты, необходимые для осуществления коммутационных обменов между исполнителями.

Результаты вычислительного эксперимента

Вычислительный эксперимент проводился на вычислительном кластере из 32 процессоров AMD Opteron 6272 с тактовой частотой 2.1 ГГц, соединенных сетью Gigabit Ethernet, под управлением



ОС Linux Kernel 3.12.13. Для оценки эффективности параллельного решения использовалось понятие ускорения параллельного решения, определяемого как $S(p) = T_1 / T_p$, где T_1 – время последовательного решения; T_p – время параллельного решения с использованием p однородных вычислительных

узлов. На рисунках 7 и 8 приведены зависимости достигаемых ускорений решений от количества вычислительных узлов. При анализе зависимостей (рис. 7) можно отметить, что для разреженной системы уравнений небольшой размерности ($n = 10^3$) ускорение с ростом количества используемых вы-

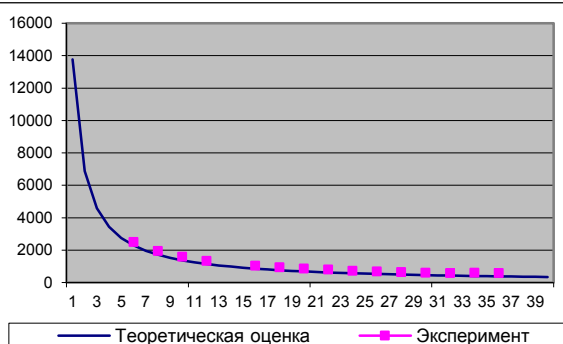


Рис. 6. Сравнение теоретической оценки времени работы разработанной модификации GMRES с экспериментальными данными

Fig. 6. Comparison of theoretical estimates of the time of the developed GMRES modification and the experimental data

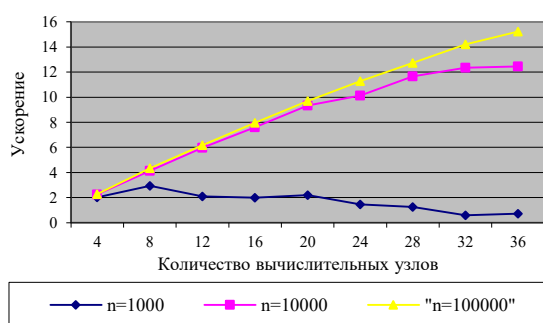


Рис. 7. Зависимости ускорений решения СЛАУ от количества вычислительных узлов для разных размерностей систем

Fig. 7. Dependences of SLAE decision accelerations on the number of computing nodes for different system dimensions

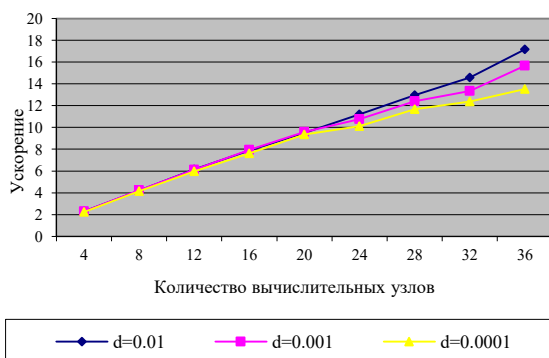


Рис. 8. Зависимости ускорений решения СЛАУ от количества вычислительных узлов для разной степени разреженности матриц коэффициентов

Fig. 8. Dependences of SLAE decision accelerations on the number of computing nodes for systems with different degree of sparsity of coefficient matrixes

числительных узлов убывает. Это явление закономерно и объясняется тем, что выигрыш от приме-

нения распределенных вычислений не компенсирует затраты на их организацию. Для систем уравнений большей размерности ($n > 10^4$) ускорение решения растет с ростом числа вычислительных узлов. Таким образом, применение параллельных технологий не всегда оправдано; в частности, для рассмотренного примера целесообразно использование последовательного варианта метода GMRES при малых размерностях СЛАУ, а уже для систем размерности $n > 10^4$ – параллельного варианта.

На временные характеристики решения СЛАУ, а также на эффект от применения параллельных вычислений оказывает влияние и степень разреженности d матрицы коэффициентов. На рисунке 8 изображена зависимость ускорения решения СЛАУ от количества используемых вычислительных узлов кластера для разной степени разреженности матрицы коэффициентов размерностью $n = 10^5$. Отметим, что степень разреженности матрицы оказывает существенное влияние на ускорение процессов решения задачи на кластерных системах, начиная с числа используемых вычислительных узлов $p \geq 20$: чем более плотная матрица подлежит обработке, тем выше ускорение решения.

Таким образом, чем больше размерность системы и сложнее структура матрицы ее коэффициентов, тем больший эффект можно получить от применения параллельных технологий.

В заключение отметим, что использование высокопроизводительной вычислительной техники без масштабируемых параллельных программ, к сожалению, не приводит к росту эффективности процессов решения сложных задач, а в ряде случаев даже увеличивает время решения. Созданное параллельное программное приложение, использующее механизм параллельных вычислений, позволяет существенно уменьшить временные затраты на проведение вычислительных экспериментов и решать сложные аэродинамические задачи.

Литература

1. Беляев К.В., Гарбарук А.В., Стрелец М.Х., Шур М.Л., Спаларт П.Р. Опыт прямого численного моделирования турбулентности на суперкомпьютерах // Суперкомпьютерные дни в России: сб. тр. конф. молод. ученых. М., 2016. С. 357–364. URL: <http://russianscdays.org/files/pdf16/357.pdf> (дата обращения: 02.10.16).
2. Гарбарук А.В., Стрелец М.Х., Шур М.Л. Моделирование турбулентности в расчетах сложных течений. СПб: Изд-во Политехнич. ун-та, 2012. 88 с.
3. Ортега Дж. Введение в параллельные и векторные методы решения линейных систем. М.: Мир, 1991. 370 с.
4. Писсанецки С. Технология разреженных матриц. М.: Мир, 1988. 398 с.
5. Крукиер Л.А., Пичугина О.А., Чикина Л.Г. Обзор методов подпространства Крылова // Современные проблемы математического моделирования: сб. тр. XIV Междунар. конф. Ростов-н-Д: Изд-во ЮФУ, 2011. 400 с.
6. Баландин М.Ю., Шурина Э.П. Методы решения СЛАУ большой размерности. Новосибирск: Изд-во РГГУ, 2000. 70 с.
7. Жуков В.Т., Новикова Н.Д., Феодоритова О.Б. Сдвиговая стратегия в обобщенном методе минимальных невязок. М.:

Изд-во ИПМ РАН, 2009. 28 с.

8. Воеводин В.В., Воеводин Вл.В. Параллельные вычисления. СПб: БХВ-Петербург, 2002. 608 с.

9. Гергель В.П. Теория и практика параллельных вычислений. М.: БИНОМ. Лаборатория знаний, 2007. 423 с.

10. Буренков С.А., Шамаева О.Ю. Библиотека эффективных функций для решения класса аэродинамических задач на кластерных архитектурах // Актуальные вопросы развития инновационной деятельности в новом тысячелетии: сб. тр. XIV Междунар. науч.-практич. конф. 2015. № 3 (14). С. 8–11.

Software & Systems

DOI: 10.15827/0236-235X.116.143-148

Received 03.10.16

2016, vol. 29, no. 4, pp. 143–148

PARALLEL CALCULATIONS AS A WAY TO INCREASE THE EFFICIENCY OF COMPUTING AERODYNAMIC PROBLEM SOLUTIONS

S.A. Burenkov¹, Postgraduate Student, burenkovsa@mpei.ru

O.Yu. Shamayeva¹, Ph.D. (Engineering), Associate Professor, shamayevaoy@mpei.ru

¹ National Research University "MPEI", Krasnokazarmennaya St. 14, Moscow, 111250, Russian Federation

Abstract. The article proposes software to increase the efficiency of solving aerodynamic problems in high-performance cluster systems. The software is implemented on the basis of the proposed parallel-sequential algorithm for solving sparse systems of linear algebraic equations of high dimensionality using the method of generalized minimal residual.

The article provides a theoretical estimate of the developed algorithm complexity, as well as a comparison of the results of theoretical estimates and computational experiment results.

The software is implemented for a distributed memory computation model and is used to solve a model problem of an aerodynamic flow of a profile with an air flow. There are the results of a computational experiment on calculating the density and pressure of the blowing profile air flow in the high-performance cluster, which illustrate the effect of using a parallel program. The article analyzes the characteristics of model problem parallel solution acceleration, which allow determining a threshold of dimension of systems equations. Below the threshold the organization of parallel computing adversely effects on the efficiency of computing resources. The paper also studies the effect of a sparse coefficient matrix degree on parallel implementation acceleration.

The given results can be applied not only on computational hydrodynamics and gas dynamics, but also energetics, biology, chemistry, pharmacology, astrophysics, and others.

Keywords: computing aerodynamics, air flow problem, parallel computing, Message Passing Interface MPI.

References

1. Belyaev K.V., Garbaruk A.V., Strelets M.Kh., Shur M.L., Spalart P.R. The experience of turbulence direct numerical simulation on supercomputers. *Superkompyuternye dni v Rossii 2016: sb. tr. konf. molod. uchenykh* [Proc. Conf. of Young Scientists "Supercomputer days in Russia 2016"]. Moscow, 2016, pp. 357–364. Available at: <http://russianscdays.org/files/pdf16/357.pdf> (accessed October 2, 2016).
2. Garbaruk A.V., Strelets M.Kh., Shur M.L. *Modelirovanie turbulentnosti v raschetakh slozhnykh techeny* [Turbulence Modeling in Complex Flotation Calculations]. St. Petersburg, Technical Univ. Publ., 2012, 88 p.
3. Ortega J. *Introduction to Parallel and Vector Solution of Linear Systems*. Springer Publ., 1988, 305 p. (Russ. ed.: Moscow, Mir Publ., 1991).
4. Pissanetski S. *Sparse Matrix Technology*. Academic Press, NY, 1984 (Russ. ed.: Moscow, Mir Publ., 1988, 398 p.).
5. Krukier L.A., Pichugina O.A., Chikina L.G. A review of Krylov subspace methods. *XIV Mezhdunar. konf. "Sovremennye problemy matematicheskogo modelirovaniya"* [Proc. 14th Int. Conf. "Modern Problems of Mathematical Modeling"]. Rostov-on-Don, SFedU Publ., 2011, 400 p.
6. Balandin M.Yu., Shurina E.P. *Metody resheniya SLAU bolshoy razmernosti* [Methods of Solving High Dimension SLAE]. Novosibirsk, RGGU Publ., 2000, 70 p.
7. Zhukov V.T., Novikova N.D., Feodoritova O.B. *Sdvigovaya strategiya v obobshchennom metode minimalnykh nevyazok* [A Shearing Strategy in a Generalized Method of Minimal Residual]. Moscow, 2009, 29 p.
8. Voevodin V.V., Voevodin V.I. *Parallelnye vychisleniya* [Parallel Computing]. St. Petersburg, BHV-Peterburg, 2002.
9. Gergel V.P. *Teoriya i praktika parallelnykh vychisleny* [Theory and Practice of Parallel Computing]. Moscow, BINOM, Laboratoriya znany Publ., 2007, 423 p.
10. Burenkov S.A., Shamayeva O.Yu. An effective function library to solve a class of aerodynamic problems on cluster architectures. *XIV Mezhdunar. nauchno-praktich. konf. "Aktualnye voprosy razvitiya innovatsionnoy deyatel'nosti v novom tysyacheletii"* [Proc. 14th Int. Science and Practice Conf. "Important Issues of Innovation Activities Development in a new Millenium"]. 2015, no. 3 (14), pp. 8–11.

УДК 621.391

DOI: 10.15827/0236-235X.116.149-155

Дата подачи статьи: 08.09.16

2016. Т. 29. № 4. С. 149–155

НАПРАВЛЕНИЯ РЕАЛИЗАЦИИ СРЕДСТВ АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ ОСВОЕНИЯ РАДИОЭЛЕКТРОННЫХ КОМПЛЕКСОВ

А.Н. Потапов, к.т.н., доцент, зам. начальника кафедры, potapov_il@mail.ru
(Военный учебно-научный центр Военно-воздушных сил «Военно-воздушная академия
им. проф. Н.Е. Жуковского и Ю.А. Гагарина»,
ул. Старых Большевиков, 54а, г. Воронеж, 394064, Россия);
В.А. Дикарев, д.т.н., профессор, зав. кафедрой, dikva@mail.ru
(Институт математики, информатики и естественных наук
Московского городского педагогического университета);
Р.В. Допира, д.т.н., профессор, зав. отделом, rvdopira@yandex.ru
(НПО РусБИТех, пр-т Калинина, 17, г. Тверь, 170001, Россия);
Ф.Н. Абу-Абед, к.т.н., доцент, aafares@mail.ru;
Д.В. Мартынов, к.т.н., доцент, idpo@tstu.ver.ru
(Тверской государственный технический университет,
наб. Аф. Никитина, 22, г. Тверь, 170026, Россия)

В статье рассмотрены вопросы реализации программных средств автоматизированной системы освоения радиоэлектронных комплексов. Предложена архитектура структурной конфигурации данной системы. Для определения факта наличия конфликтов применения радиоэлектронных комплексов и тренажеров предложено в состав автоматизированной системы освоения этих комплексов включить средство радиомониторинга, позволяющее выявлять наличие признака конфликта применения радиоэлектронных комплексов, и средство оценки адекватности тренажера. Разработаны структуры автоматизированных средств определения содержательных компонентов учебных упражнений и учебных планов.

Предложен профессионально-ориентированный комплекс программ автоматизированной системы освоения радиоэлектронных комплексов.

Проведенные экспериментальные исследования адекватности результатов имитационного моделирования в специализированных тренажерах «Марка – РС» и комплексных тренажерах «Репитер» при управлении воздушным движением в ближней зоне аэродрома в целях корректного определения уровня навыков и рационального планирования тренажной подготовки операторов радиоэлектронных комплексов определили новый уровень возможностей автоматизированных систем подготовки операторов радиоэлектронных комплексов по управлению воздушным движением. Разработанные инструментальные средства в виде моделей, структурных схем, а также математического, алгоритмического и программного обеспечения подсистем, входящих в состав автоматизированных систем освоения радиоэлектронных комплексов, обладают не только научной новизной, но и практической значимостью и реализуемостью.

Ключевые слова: автоматизированная система освоения, конфликтоустойчивость, оператор, радиоэлектронный комплекс, комплекс программ, тренажная подготовка.

В современных условиях достижение конфликтоустойчивости управления *тренажной подготовкой* (ТП) операторов эрготехнических радиоэлектронных комплексов (РЭК) на практике невозможно из-за отсутствия инструментальных средств, позволяющих разрешить конфликт применения РЭК, вызванный четырьмя факторами:

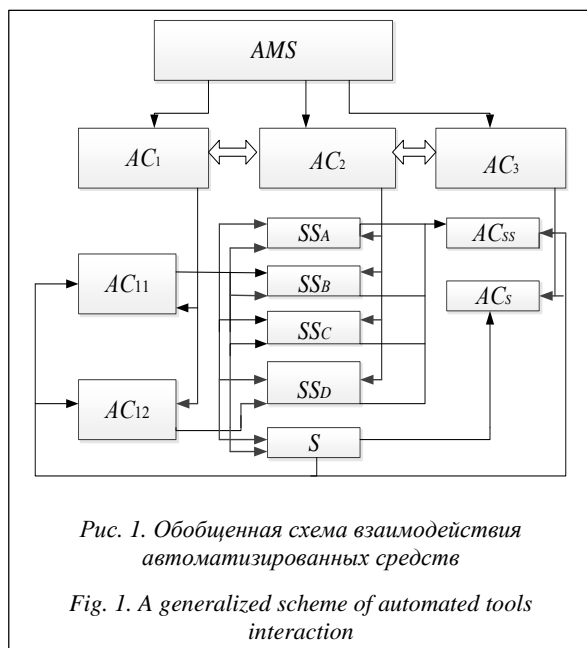
- 1) низкая оперативность корректировки учебных курсов;
- 2) ограниченная адекватность имитируемых процессов;
- 3) низкая достоверность контроля приобретаемых навыков;
- 4) нерациональное использование средств тренажа.

Влияние каждого фактора конфликта можно несколько уменьшить [1, 2]: первого фактора – за счет автоматизации процедур формирования курсов ТП [2], которые обеспечат оперативную корректировку учебных упражнений и учебных планов при изменении условий эксплуатации РЭК с учетом индивидуальных особенностей обучаемых и имею-

щихся в распоряжении средств тренажа; второго – за счет использования тренажеров [3], которые существенно дешевле *штатной техники* (ШТ) и с учетом современного развития вычислительной техники и способов имитационного моделирования могут обеспечить их адекватную адаптацию к изменениям эксплуатации РЭК и уровню подготовленности операторов; третьего – за счет внедрения систем оперативного объективного контроля действий операторов [3]; четвертого – за счет создания *автоматизированной системы* (АС) освоения РЭК [3, 4], в качестве основных элементов которой может использоваться инструментарий, предложенный для разрешения первых трех факторов.

Структура АС освоения РЭК

Реализация представленного инструментария возможна на базе создания автоматизированных средств формирования адаптивных курсов ТП операторов, тренажа операторов и контроля качества подготовленности операторов.

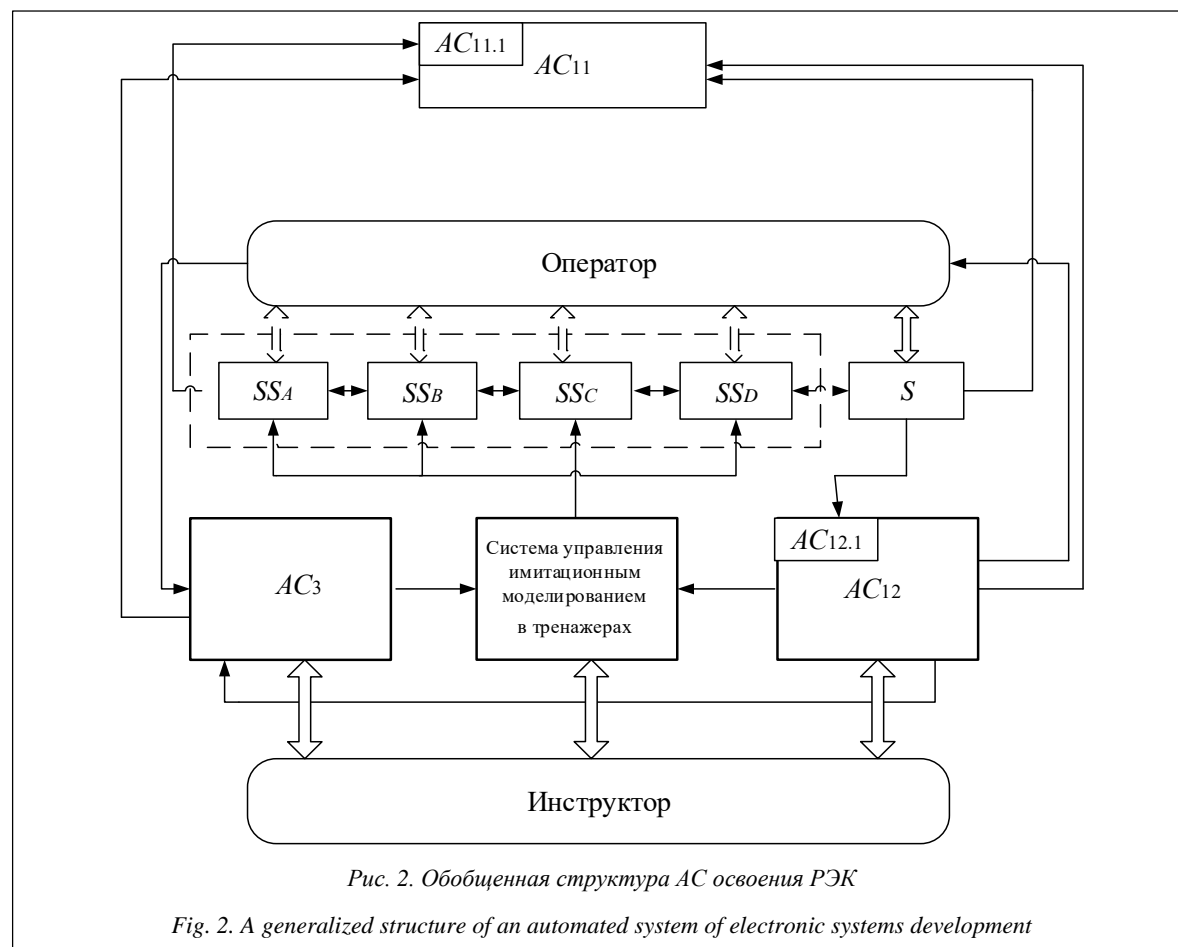


При этом автоматизированные средства тренажа представляют собой множество, состоящее из совокупности различных тренажеров и включающее в свой состав РЭК.

На рисунке 1 изображена обобщенная схема взаимодействия автоматизированных средств в АС

освоения операторами РЭК [5]. На рисунке приняты следующие обозначения: AMS – АС освоения РЭК; AC₁ – автоматизированные средства формирования адаптивных курсов ТП операторов; AC₂ – автоматизированные средства тренажа операторов; AC₃ – автоматизированные средства контроля качества подготовки операторов; S – РЭК; AC₁₁ – автоматизированное средство определения содержательных компонентов учебных упражнений, защищенных от конфликта применения РЭК из-за изменения радиочастотных (РЧ) условий их функционирования; AC₁₂ – автоматизированное средство определения содержательных компонентов учебных планов, защищенных от конфликта применения тренажеров из-за их ограниченной адекватности; AC_{ss} – система оперативного объективного контроля подготовки операторов на тренажерах; AC_s – система оперативного объективного контроля подготовки операторов на РЭК; AMS – АС освоения РЭК; SS_A – процедурные тренажеры; SS_B – специализированные тренажеры; SS_C – комплексные тренажеры; SS_D – тренажные режимы РЭК.

Подобное представление AMS не учитывает такие важные элементы, как обучаемый (оператор) и обучающий (инструктор), а также их взаимосвязи. С учетом этого АС освоения РЭК AMS должна иметь вид, представленный на рисунке 2.



При такой структуре *AMS* оператор непосредственно взаимодействует с автоматизированным средством тренажа операторов, обрабатывая требуемый перечень упражнений с заданной для него последовательностью и продолжительностью [5]. Как показано на рисунке 2, перечень упражнений ТП формируется в блоке AC_{11} с учетом текущего состояния и тенденций изменения РЧ-условий функционирования РЭК.

Сами тренажеры *SS* адаптируются с помощью системы управления имитационным моделированием в тренажерах в соответствии с данными, поступающими из систем AC_{11} и AC_3 . Инструктор опосредованно взаимодействует с оператором и вносит свои коррективы по организации ТП.

Для определения факта наличия конфликтов применения РЭК и тренажеров предлагается в AC_{11} и AC_{12} соответственно включить:

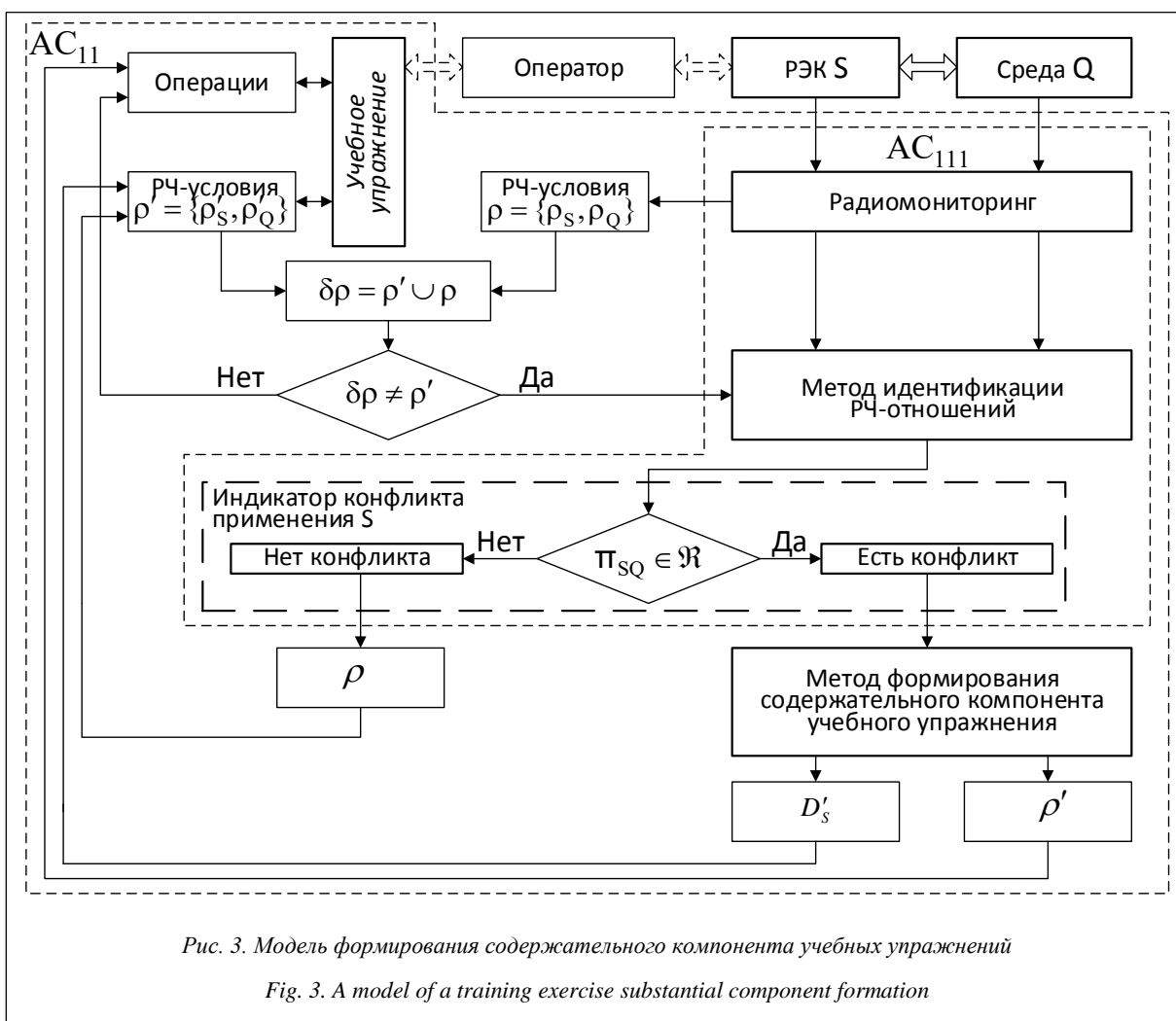
- $AC_{11.1}$ – средство радиомониторинга, позволяющее выявлять наличие признака конфликта применения РЭК;
- $AC_{12.1}$ – средство оценки адекватности тренажера.

В автоматизированном средстве $AC_{12.1}$ реализуется совокупность способов проверки адекватности моделирования в тренажерах, характерной особенностью которых является то, что в них осуществляется не только проверка выполнения нормы годности тренажеров, но и по результатам таких проверок вероятностная оценка степени адекватности имитационного моделирования, используемая для прогноза правильных и ложных навыков. Выбор конкретного способа зависит от условий испытания и сертификации тренажеров.

Структура автоматизированного средства определения содержательных компонентов учебных упражнений

Автоматизированное средство AC_{11} (модель формирования содержательного компонента учебных упражнений) имеет вид, представленный на рисунке 3, и функционирует следующим образом:

- в автоматизированном средстве $AC_{11.1}$ происходит сравнение принятых (учтенных в учебном упражнении) и текущих (определяемых сред-



ствами радиомониторинга) РЧ-условий функционирования РЭК;

- если между ними нет различия, предписанные операции остаются без изменения, в противном случае происходит идентификация РЧ-воздействий [6];

- если РЧ-воздействия не принадлежат к категории \mathcal{R} (определяющей снижение функции полезности РЭК), предписанные операции остаются без изменения, а принятые РЧ-условия подстраиваются под текущие, в противном случае происходит индикация *конфликт* [6];

- при срабатывании индикатора *конфликт* на основании способа формирования содержательных компонентов учебных упражнений осуществляются выбор бесконфликтных операций и уточнение РЧ-условий;

- по выбранной операции и уточненным РЧ-условиям функционирования РЭК происходит коррекция соответствующих содержательных компонентов учебного упражнения.

На рисунке 3 приняты следующие обозначения: ρ – множество входов и выходов радиосигналов РЭК в стационарном режиме; ρ' – множество входов и выходов радиосигналов РЭК с учетом динамики изменения РЧ-условий его функционирования; $\delta\rho$ – относительная разность изменения РЧ-условий функционирования РЭК в сравнении со стационарным режимом; $\pi SQ \in \mathcal{R}$ – РЧ-воздействия на РЭК S среды Q ; D'_s – множество операций (действий оператора) с учетом динамики изменения РЧ-условий.

Структура АС определения содержательных компонентов учебных планов

Структура модели формирования содержательного компонента учебных планов AC_{12} имеет вид, изображенный на рисунке 4. Функционирование автоматизированного средства заключается в следующем.

- В системе $AC_{12.1}$ в зависимости от условий (объективных или субъективных) испытаний и сертификации производится выбор способа вероятностной оценки степени адекватности имитационного моделирования в тренажере. Для объективных испытаний: если структура тренажера и РЭК известна и является иерархической, то используется разработанный в [6] метод *белого ящика* (метод оценки адекватности имитационного моделирования в тренажерах, имеющих иерархическую структуру построения), в противном случае – метод *черного ящика* [6] (метод оценки адекватности имитационного моделирования в тренажерах по их информационному полю). Для субъективных: если имеется достаточное количество экспертов, используется метод *экспертизы* (метод оценки адек-

ватности имитационного моделирования с привлечением экспертов) [6], в противном случае – метод *анкетирования обучаемых* (метод оценки адекватности имитационного моделирования с использованием анкетирования обучаемых).

- Если закон распределения невязок между результатами имитационного моделирования в тренажере и откликами РЭК является нормальным, то в выбранном методе оценка адекватности производится по обобщенному критерию Фишера–Хи-квадрат, в противном случае – по обобщенному закону распределения [3]. Если ошибка оценки адекватности меньше заданной, производятся дополнительные испытания.

- Если оценка адекватности равна единице (что маловероятно), учебный план остается без изменений, в противном случае происходит индикация *конфликт*.

- При срабатывании индикатора *конфликт* осуществляются выбор варианта ТП (средств тренажа) и определение количества тренировок.

- По выбранным средствам тренажа и определенному на каждом средстве количеству тренировок происходит коррекция соответствующих содержательных компонентов учебного плана.

Профессионально-ориентированный комплекс программ АС освоения РЭК

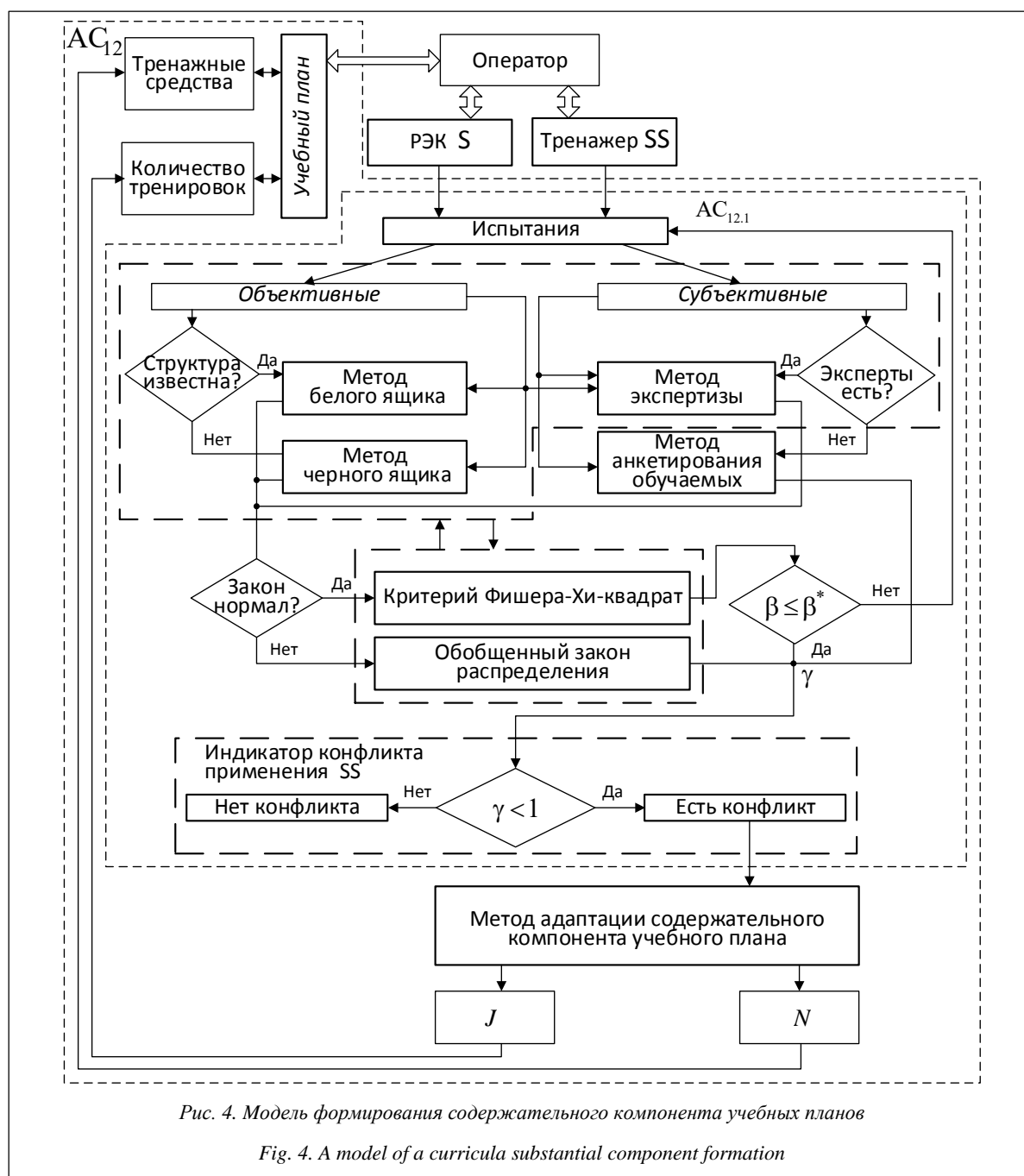
В рамках реализации требований к структурной конфигурации элементов АС освоения РЭК разработан комплекс программ для ЭВМ, в состав которого входят следующие элементы.

Программа определения структуры системных отношений (Strukture SA) [7], предназначенная для автоматизации анализа условий применения систем различного назначения; в программе реализованы алгоритмы, основанные на выявлении ядер конфликта, сотрудничества и безразличия взаимодействия элементов системы и позволяющие определять свойства этих отношений.

Программа оценки адекватности автоматизированной информационной системы освоения эргономического комплекса, имеющего иерархическую структуру построения «адекватность АИС и ЭТК-иерархия» [8], предназначенная для оценки адекватности автоматизированных информационных систем подготовки операторов РЭК, имеющих иерархическую структуру.

Программно-ориентированный комплекс оценки адекватности автоматизированных информационных систем освоения эргономических объектов управления воздушным движением [9], выполняющий оценку адекватности имитационного моделирования в тренажерах операторов *управления воздушным движением* (УВД) по информационному полю.

Программа определения содержательных компонентов учебных упражнений и учебных планов,



разработанная для автоматизации коррекции по выбранным операциям и уточненным РЧ-условиям функционирования РЭС соответствующих содержательных компонентов учебного упражнения, а по выбранным средствам тренажа и определенному на каждом тренажном средстве количеству тренировок – коррекции содержательных компонентов учебного плана в процессе подготовки.

Программа управления планированием ТП специалистов эрготехнических систем («Информационная конфликтоустойчивая автоматизированная система рационального планирования практической подготовки операторов радиоэлектронных объектов УВД»), обеспечивающая автоматизацию

управления планирования ТП операторов с использованием компьютерных систем тренажа. Интерфейс этой программы представлен на сайте (см. <http://www.swsys.ru/uploaded/image/2016-4/2016-4-dop/2.jpg>). В программе реализованы новые алгоритмы, основанные на минимизации стоимости ТП с учетом начального и конечного уровней обученности операторов, интенсивности приобретения ими навыков, адекватности компьютерных систем тренажа и позволяющие определить рациональные учебные планы с устранением возможности приобретения обучаемыми ложных навыков.

Программа «Система оперативного объективного контроля действий операторов информации

онных комплексов УВД», предназначенная для оперативного объективного контроля действий операторов УВД на основе учета оценки адекватности автоматизированных информационных систем подготовки операторов УВД (лиц группы руководства полетами) в целях разрешения конфликтов использования ШТ и тренажеров в процессе практической подготовки операторов УВД.

В целях корректного определения уровня навыков и рационального планирования ТП лиц группы руководства полетами на базе 4 Государственного центра подготовки авиационного персонала и войсковых испытаний МО РФ проведены экспериментальные исследования адекватности результатов имитационного моделирования в специализированных тренажерах «Марка – РС» и комплексных тренажерах «Репитер» при УВД в ближней зоне аэродрома с использованием разработанного профессионально-ориентированного комплекса программ АС освоения РЭК «Конфликтоустойчивость АСУ ОРЭС».

Полученные результаты, а также реализация разработанного профессионально-ориентированного комплекса программ АС освоения РЭК «Конфликтоустойчивость АСУ ОРЭС» при организации полетов воздушных судов в ближней зоне аэродрома показали новый уровень возможностей АС подготовки операторов радиоэлектронных средств по УВД. Все изложенное позволяет констатировать следующее: разработанные инструментальные средства в виде моделей, структурных схем, а также математического, алгоритмического и программного обеспечения подсистем, входящих в состав конфликтоустойчивой АС управления подготовкой операторов РЭС, обладают не только науч-

ной новизной, но и практической значимостью и реализуемостью.

Литература

1. Дудоров А.Д. Методика оценки тренажеров лиц группы руководства полетами в интересах повышения качества их подготовки // Современные проблемы и перспективные направления развития авиационных комплексов и систем военного назначения, формы и способы их боевого применения: сб. стат. Всерос. науч.-практич. конф. Воронеж: Изд-во ВАИУ, 2011. С. 92–98.
2. Сысоев Е.С. Оптимизация тренажерной подготовки операторов сложных информационных радиоэлектронных систем управления воздушным движением // Информатика: проблемы, методология, технологии: матер. 11 Междунар. науч.-методич. конф. Воронеж: Изд-во ВГУ, 2011. Т. 2. С. 216–222.
3. Романов А.В. Теоретико-множественный подход к описанию функционирования эрготехнических радиоэлектронных средств // Проблемы эффективности и безопасности функционирования сложных технических и информационных систем: тр. 32 Всерос. науч.-технич. конф. Серпухов: Изд-во филиала ВА ВВСН им. Петра Великого, 2013. С. 118–122.
4. Помазуев О.Н., Платонов А.Ю., Миронов А.М., Абу-Абед Ф.Н., Рюмшин А.Р. Методика определения показателей качества обнаружения радиолокационных станций // Программные продукты и системы. 2014. № 2. С. 118–124.
5. Агапов Е.А. Проектирование автоматизированных информационных систем освоения эрготехнических комплексов // Эффективность автоматизированных систем управления авиацией, систем связи и РТО ВВС: науч.-технич. сб. 2012. С. 112–118.
6. Пономаренко В.А. Психология жизни и труда летчика. М.: Воениздат, 1992. 58 с.
7. Дикарев В.А. Программа определения структуры системных отношений (Strukture SA): 2001610938 Рос. Федерация. № 2001610680; заявл. 01.06.01, зарегистр. 30.07.01.
8. Потапов А.Н., Овчаров В.В., Кинчя О.Т. Адекватность АИС О ЭТК-иерархия: свид. 2013616624 Рос. Федерация. № 2013614737; заявл. 04.06.2013, зарегистр. 12.07.2013.
9. Назаров Т.И. Конфликтоустойчивость АСУ ОРЭС: свид. № 2015610377 Рос. Федерация. № 2014661423; заявл. 05.11.2014, зарегистр. 12.01.2015.

Software & Systems

DOI: 10.15827/0236-235X.116.149-155

Received 08.09.16

2016, vol. 29, no. 4, pp. 149–155

THE DIRECTIONS OF IMPLEMENTATION OF AUTOMATED SYSTEM MEANS FOR ELECTRONIC SYSTEMS DEVELOPMENT

A.N. Potapov¹, Ph.D. (Engineering), Associate Professor, Deputy Head of Chair, potapov_il@mail.ru

V.A. Dikarev², Dr.Sc. (Engineering), Professor, Head of Chair, dikva@mail.ru

R.V. Dopira³, Dr.Sc. (Engineering), Professor, Head of Department, rvdopira@yandex.ru

F.N. Abu-Abed⁴, Ph.D. (Engineering), Associate Professor, aafares@mail.ru

D.V. Martynov⁴, Ph.D. (Engineering), Associate Professor, idpo@tstu.tver.ru

¹ Military Scholastic-Scientific Centre of the Air Forces "Zhukovsky and Gagarin Air Forces Academy", Sarykh Bolshevikov St. 54a, Voronezh, 394064, Russian Federation

² Institute of Mathematics, Informatics and Natural Sciences, Moscow City Pedagogical University

³ NPO RusBITex, Kalinina Ave. 17, Tver, 170001, Russian Federation

⁴ Tver State Technical University, Nikitin Quay 22, Tver, 170026, Russian Federation

Abstract. The article considers the issues implementing software tools of an automated system of electronic systems development. The authors propose an architecture of the automated system's structural configuration. To determine the existence of conflicts when applying radio-electronic complexes (REC) and simulators it is proposed to add radio-monitoring means and means of assessing simulator adequacy to the automated system of electronic systems development. Radio-monitoring means can identify the signs of a conflict when applying electronic systems. The paper shows the developed structures of automated means for defining the substantial components of training exercises and curricula.

The authors offer a professionally-oriented program complex of the automated system of electronic systems development. Experimental studies of the adequacy of simulation results in specialized simulators “Marka – RS” and integrated simulators “Repeater” in the air traffic control (ATC) in the near zone from an aerodrome in order to correct determine the level of skills and rational planning of operators’ training identified a new level of automated systems of electronic system operators’ training in the air traffic control. The developed tools in the form of models, block diagrams, mathematical, algorithmic support and software for subsystems in the automated system of electronic systems development have not only a scientific novelty, but also practical significance and feasibility.

Keywords: automated system development, conflict-stability, operator, electronic systems, software training.

References

1. Dudorov A.D. The methodology of the assessment of equipment entities of flight group management to improve the quality of their training. *Sovremennye problemy i perspektivnye napravleniya razvitiya aviatsionnykh kompleksov i sistem voennogo naznacheniya, formy i sposoby ikh boevogo primeneniya: sb. stat. Vseros. nauch.-praktich. konf.* [Proc. All-Russian Science and Practice Conf. “Modern Problems and Prospects of Aircraft and Military Systems, Forms and Methods of Their Combat Use”]. 2011, Voronezh, VAI O Publ., pp. 92–98 (in Russ.).
2. Sysoev E.S. Optimization of simulator training of operators of complex information electronic systems for air traffic control. *Informatika: problemy, metodologiya, tekhnologii: mater. 11 Mezhdunar. nauch.-metodich. konf.* [Proc. 11th Int. Scientific Conf. “Informatics: Problems, Methodology, Technologies”]. 2011, Voronezh, VSU Publ., vol. 2, pp. 216–222 (in Russ.).
3. Romanov A.V. Theoretic approach to the description of ergotechnical electronic equipment. *Problemy effektivnosti i bezopasnosti funktsionirovaniya slozhnykh tekhnicheskikh i informatsionnykh sistem: tr. 32 Vseros. nauch.-tekhnich. konf.* [Proc. 32 All-Russian Science and Practice Conf. “Problems of Efficiency and Safety of Complex Technical and Information Systems”]. Serpukhov, Branch of VA RVS N.a. Peter the Great Publ., 2013, pp. 118–122 (in Russ.).
4. Pomazuev O.N., Platonov A.Yu., Mironov A.M., Abu-Abed F.N., Ryumshin A.R. Method for determining the quality of detection of radar. *Programmnye produkty i sistemy* [Software & Systems]. 2014, no. 2, pp. 118–124 (in Russ.).
5. Agapov E.A. Design of automated information systems of ergotechnical systems development. *Effektivnost avtomatizirovannykh sistem upravleniya aviatsiey, sistem svyazi i RTO VVS: nauch.-tekhnich. sb.* [Proc. Efficiency of Automated Aircraft Control Systems, Communication Aystems, and Air Force Radio-Technical Support]. Noginsk, 2012, pp. 112–118 (in Russ.).
6. Ponomarenko V.A. *Psikhologiya zhizni i truda letchika* [Psychology of Pilot’s Life and Work]. Moscow, Voenizdat Publ., 1992, 58 p.
7. Dikarev V.A. *Programma opredeleniya struktury sistemnykh otnosheniy (Struktura SA)* [The Program of Defining a Structure of System Relationships (Struktura SA)]. The State certificate of computer program registration no. 2001610938 of the Russian Federation. 2001.
8. Potapov A.N., Ovcharov V.V., Kinchaya O.T. *Adekvatnost AIS O ETK-ierarkhiya* [The Adequacy of AIS On ETK-hierarchy]. The State certificate of computer program registration no. 2013616624 of the Russian Federation. 2013.
9. Nazarov T.I. *Konfliktoustoychivost ASU ORES* [Conflict Stability ASU ORES]. The State certificate of computer program registration no. 2014661423 of the Russian Federation. 2014.

Примеры библиографического описания статьи

1. Потапов А.Н., Дикарев В.А., Допира Р.В., Абу-Абед Ф.Н., Мартынов Д.В. Направления реализации средств автоматизированной системы освоения радиоэлектронных комплексов // Программные продукты и системы. 2016. Т. 29. № 4. С. 149–155; DOI: 10.15827/0236-235X.116.149-155.

2. Potapov A.N., Dikarev V.A., Dopira R.V., Abu-Abed F.N., Martynov D.V. The directions of implementation of automated system means for electronic systems development. *Programmnye produkty i sistemy* [Software & Systems]. 2016, vol. 29, no. 4, pp. 149–155 (in Russ.); DOI: 10.15827/0236-235X.116.149-155.

УДК 004.021

DOI: 10.15827/0236-235X.116.156-164

Дата подачи статьи: 02.03.16

2016. Т. 29. № 4. С. 156–164

ПОСТАНОВКА ЗАДАЧИ ИССЛЕДОВАНИЯ ДИФФУЗИОННОГО ПЕРЕХОДА ЧЕРЕЗ ГРАНИЦУ ШЛАК-МЕТАЛЛ В КОЛОННОМ РЕАКТОРЕ И АЛГОРИТМ ЕЕ РЕШЕНИЯ

П.А. Сеченов, аспирант, pavesa89@mail.ru;

В.П. Цымбал, д.т.н., профессор, tsumbal33@mail.ru;

А.А. Оленников, к.т.н., доцент, tgsu-sibsin@mail.ru

*(Сибирский государственный индустриальный университет,
ул. Кирова, 42, г. Новокузнецк, 654007, Россия)*

Предметом данного исследования являются задачи диффузионного перехода через границу шлак-металл и обезуглероживание частиц металла при прохождении через слой шлака. Авторами дана математическая постановка задач перехода через границу шлак-металл в рамках задачи имитационной модели гравитационного сепаратора струйно-эмульсионного реактора. Проведено сравнение объектно-ориентированных языков программирования (Delphi, Visual C#, ActionScript 3.0), выделены целевые задачи имитационной модели. Сделан сравнительный анализ имитационных моделей, созданных за последние 10 лет, в качестве критерия сравнения взята мерность пространства. Приведена также схема взаимодействия между классами имитационной модели и основным модулем. Особое внимание уделено классам шлака внизу, металла внизу, расчетов и основному модулю, так как именно в них реализуется поставленная задача. Для класса шлака описаны свойства (проценты массовых составляющих компонентов шлака, переменная для взаимодействия с классом расчетов) и методы (инициализация переменной класса, функции приращения и изменения состава шлака). В классе расчетов показаны функции определения балансного содержания FeO и скорости восстановления на границе шлак-металл, а затем алгоритм обезуглероживания железа шлака и его взаимодействие с другими классами программы. Для реализации задачи выбран метод имитационного моделирования (Монте-Карло), позволяющий при знании механизмов изучаемого объекта на основе разыгрывания случайности построить алгоритм. В результате реализации второй схемы – обезуглероживания частиц металла при прохождении через слой шлака – на имитационной модели были проведены опыты с одинаковыми начальными условиями и достаточным временем моделирования, равным 10 минутам.

Проведенные модельные исследования показали, что при реализации только первого механизма содержание углерода в конечном продукте составляет 4 %, а при реализации двух отмеченных выше схем – 3,5 %, что свидетельствует о возможности более гибкого управления процессом обезуглероживания.

Ключевые слова: *диффузионный переход, имитационная модель, ActionScript 3.0, гравитационный сепаратор, взаимодействие частиц, алгоритм, программная реализация, струйно-эмульсионный реактор, метод Монте-Карло.*

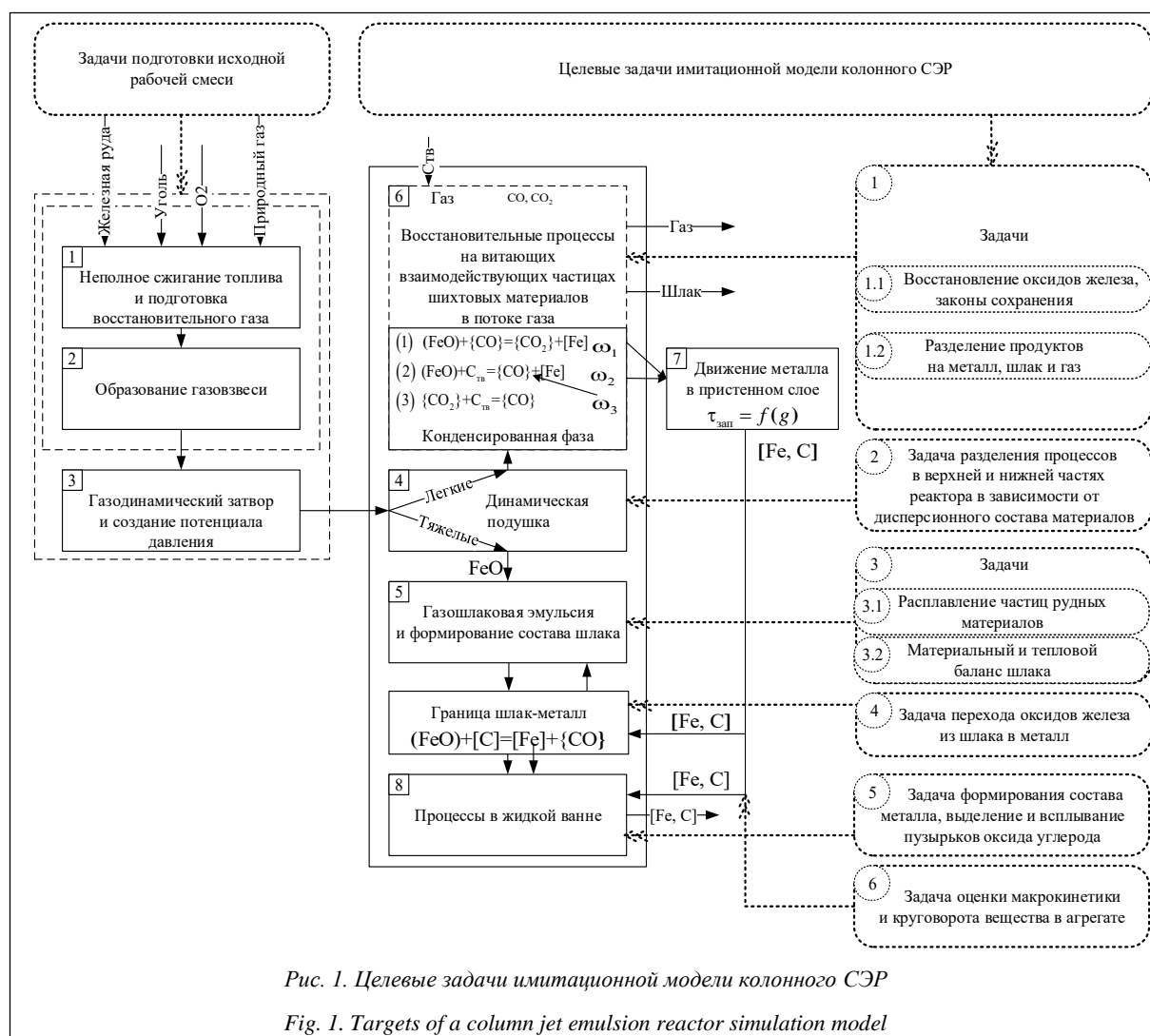
Целью данной работы является математическое описание процессов на границе шлак-металл в колонном *струйно-эмульсионном реакторе* (СЭР) по двум схемам (диффузионный переход на границе шлак-металл и обезуглероживание частиц железа в слое шлака) и создание алгоритмов для двух схем в рамках имитационной модели гравитационного сепаратора СЭР [1]. Данные механизмы описывают окислительные процессы, происходящие в нижней части колонного реактора и влияющие на содержание углерода в конечном продукте – металле.

Для описания механизмов в агрегате СЭР был применен принцип мозаичной неравновесной термодинамики, состоящий в выделении диссипативных структур в колонном реакторе, которые существуют только при условии протекания процесса в определенном кинетическом режиме. Целевые задачи имитационной модели СЭР представлены на рисунке 1.

Для решения задачи был выбран метод имитационного моделирования (Монте-Карло). Первоначально он использовался в основном для решения задач нейтронной физики, где традиционные численные методы оказались малоприменимыми, однако не утратил актуальности и на сегодняшний

день [2]. Большинство методов реализуют генерацию случайности по одной координате [3], реже встречаются задачи двухмерного [4] и трехмерного моделирования [5]. Анализ готовых средств имитационного моделирования [6, 7] показывает, что для их использования требуется знание инструментальной среды, а для гибкого взаимодействия – знание языка среды. Использование объектно-ориентированных языков программирования позволит создать более сложную систему имитационного моделирования. Авторы рассматривали объектно-ориентированные языки программирования Delphi [8], Visual C# [9] и ActionScript 3.0 [10, 11]. Язык ActionScript [12, 13] известен благодаря Flash-поликам и Flash-играм, в которых используется большое количество взаимодействующих объектов на сцене, поэтому он и был выбран для анимации движения частиц в колонном реакторе.

Одной из важнейших диссипативных структур, более подробно рассмотренных в [14, 15], является гравитационный сепаратор металла, шлака и газа, образующийся в верхней половине колонного реактора (зоны 6 и 7 на рисунке 1). Из-за влияния гравитационной составляющей и близкого к параболическому распределению вектора скорости дви-



жения эмульсии в поперечном сечении колонного реактора имеет место сепарация частиц в зависимости от их плотности. Вследствие этого явления частицы восстановленного железа, имеющие плотность, в два-три раза большую, чем частицы оксидов железа, «скатываются» на периферию потока и образуют пристенный слой с отрицательной (обратной) скоростью (зона 7). Это явление визуально наблюдалось авторами на физической модели.

Таким образом, с учетом противодействия гравитационных и аэродинамических сил в верхней половине колонного реактора (зона 6) происходит сепарация металла и шлака, а также образуется неравномерное (по плотности, газосодержанию и химсоставу) распределение параметров по высоте агрегата. Распределенность содержания оксидов железа по высоте, что играет важную роль в управлении соотношением восстановительных и окислительных процессов, обуславливается также термодинамически неравновесным характером процессов, в том числе интенсивными потоками вещества и энергии, поступающими снизу через соединительный канал 3 из реактора-осциллятора 2.

Учитывая важную, по существу центральную, роль отмеченной диссипативной структуры, связывающей все остальные зоны и обуславливающей круговорот веществ в агрегате, для ее изучения создана имитационная модель [14, 15], в которой в качестве «первых принципов» – элементов нижнего уровня иерархии – выступают дисперсные частицы шихты с реально заданным гранулометрическим распределением. Ядром этой модели является задача обтекания единичной частицы вертикальным потоком несущего реакционного газа.

В процессе плавления частицы железной руды образуются частицы металла, шлака и газа. Частицы металла из-за гравитационных сил попадают в копильник (зона 8). Частицы шлака, не повстречавшие частицы газа, попадают на верхнюю границу металла и образуют слой шлака (зона 5). Данный слой шлака будет накапливаться до тех пор, пока высота шлака не достигнет соединительного канала (зона 3), который будет «сдувать» частицы шлака.

В данной статье основное внимание уделено алгоритмам двух следующих моделей:

- диффузионный переход на границе шлак-металл (зоны 5–8);
- процессы, протекающие при прохождении частицы металла через шлак.

Стоит также отметить, что ранее в имитационной модели был задействован только первый механизм [14].

Математическая постановка задачи диффузионного перехода на границе шлак-металл

На рисунке 2 показаны процессы, происходящие в зонах 5–8 колонного СЭР.

Сначала рассмотрим *задачу диффузионного перехода на границе шлак-металл*.

1. Крупные частицы железной руды из-за плотности и сил, действующих на них, плавятся в слое шлака. Диффузионный переход на границе шлак-металл определяется следующим соотношением:

$$V_{\text{FeO}}^{\text{д}} = \beta_{\text{ШМ}} [(\text{FeO})_{\text{Ш}} - (\text{FeO})_{\text{РШМ}}], \quad (1)$$

где $(\text{FeO})_{\text{Ш}}$ – содержание FeO в шлаке; $(\text{FeO})_{\text{РШМ}}$ – содержание FeO, равновесное с металлом; $\beta_{\text{ШМ}}$ – коэффициент массоотдачи из шлака в металл.

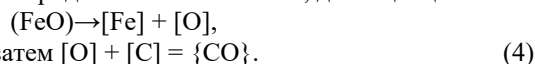
$$(\text{FeO})_{\text{РШМ}} = \frac{[\text{O}]_{\text{РМ}}}{f(\text{FeO})L} = \frac{[\text{O}]_{\text{РМ}}}{0,0023}, \quad (2)$$

где L – коэффициент распределения между шлаком и металлом; $[\text{O}]_{\text{РМ}}$ – равновесное содержание кислорода в металле.

$$[\text{O}]_{\text{РМ}} = \frac{0,0025 \cdot P_{\text{CO}}}{[\text{C}]_{\text{М}}}, \quad (3)$$

где P_{CO} – давление в пузырьках CO; $[\text{C}]_{\text{М}}$ – содержание углерода в металле.

2. В результате диффузионной силы, определенной по формуле (1), осуществляется переход кислорода из шлака в металл, диссоциация FeO:



а затем $[\text{O}] + [\text{C}] = \{\text{CO}\}$.

Объемы компонентов углерода, железа и оксида углерода, образовавшиеся в результате диссоциации FeO, рассчитываются по формулам

$$V_{\text{C}}^{\circ} = \frac{12}{72} V_{\text{FeO}}^{\circ}, \quad V_{\text{Fe}}^{\circ} = \frac{56}{72} V_{\text{FeO}}^{\circ}, \quad V_{\text{CO}}^{\circ} = \frac{22,4}{72} V_{\text{FeO}}^{\circ} \frac{M_{\text{М}}}{100},$$

где $M_{\text{М}}$ – масса металла.

3. Осуществляется подъем газовых пузырей CO, образовавшихся в результате диссоциации FeO по формуле (4).

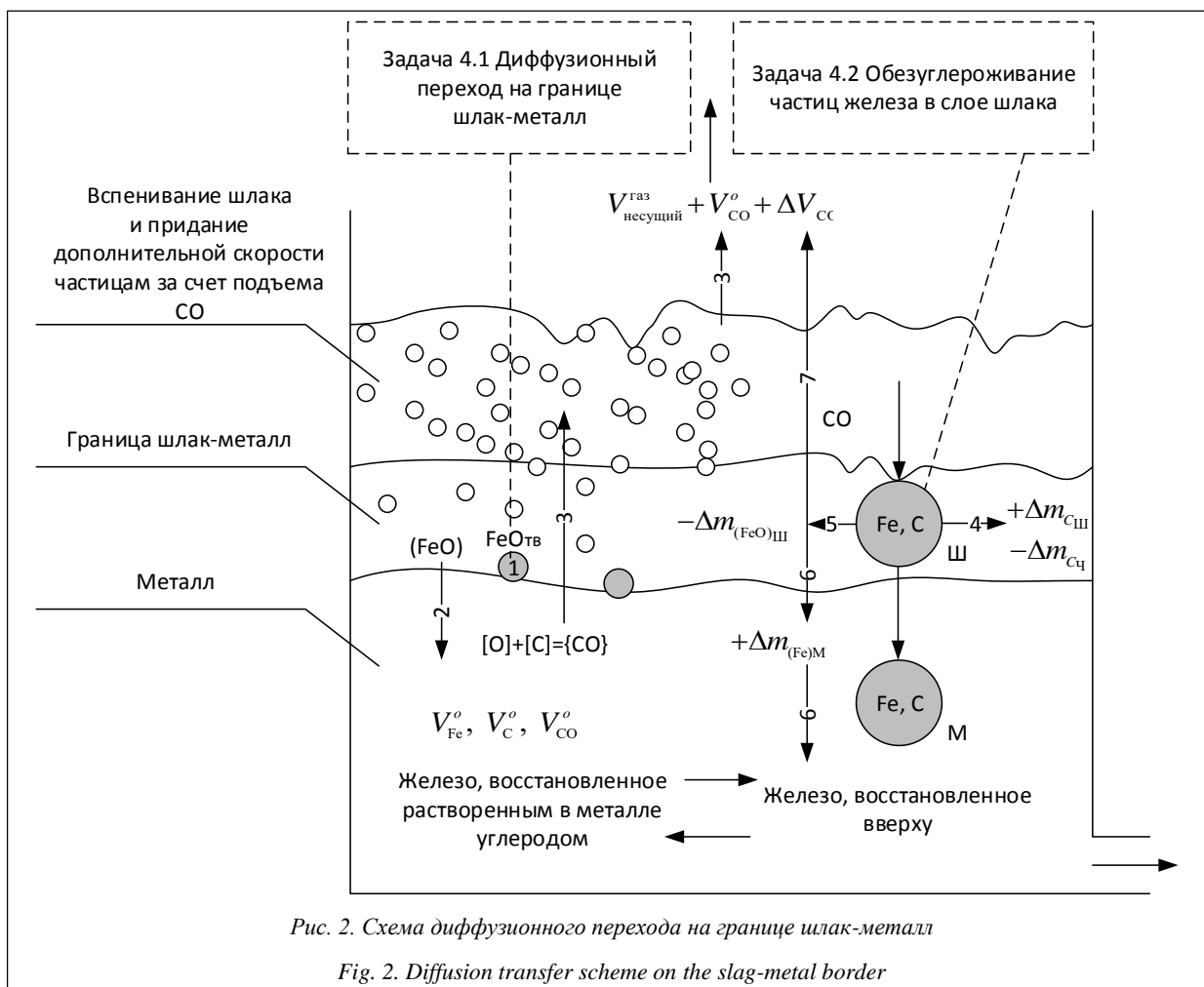
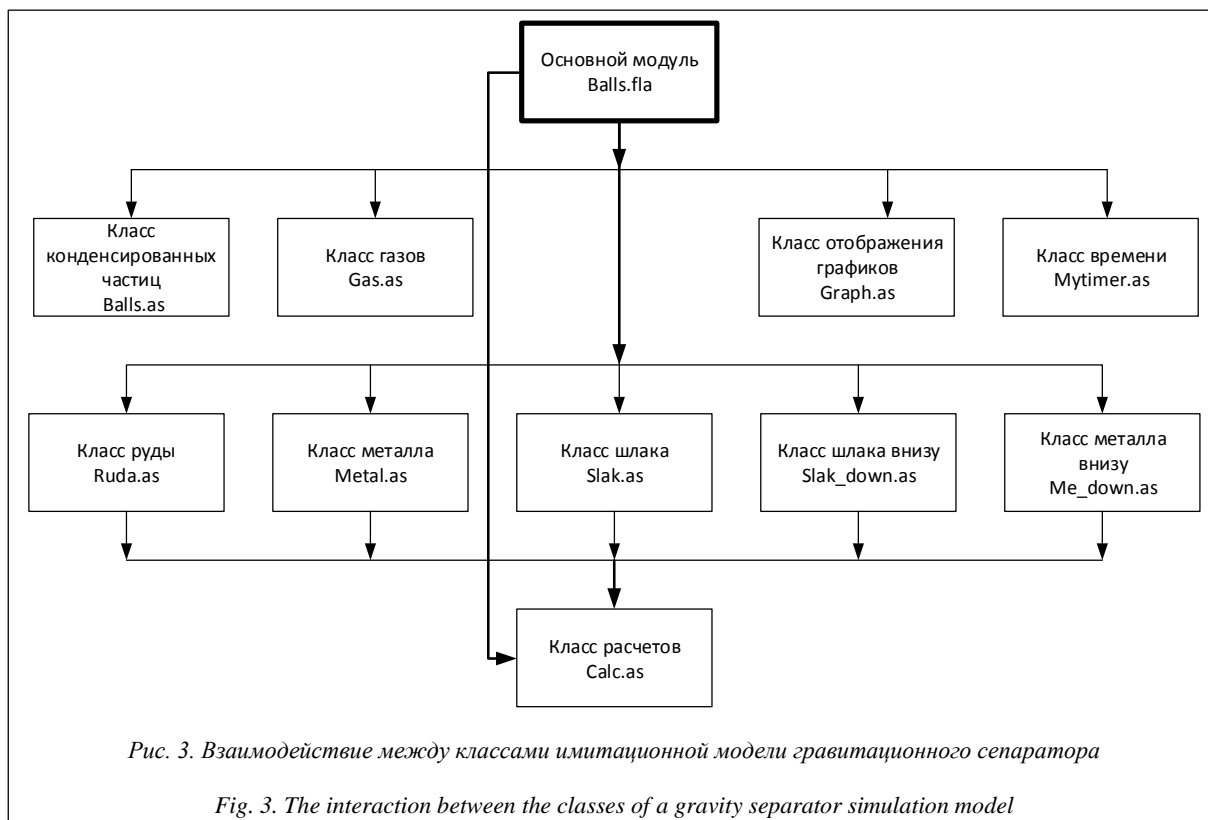


Рис. 2. Схема диффузионного перехода на границе шлак-металл

Fig. 2. Diffusion transfer scheme on the slag-metal border



Задача обезуглероживания частиц железа в слое шлака.

4. Из частицы железа, проходящей через слой шлака, выгорает углерод: $\Delta m_c = m_k \Delta C$, где m_k – масса капли; ΔC – процент обезуглероживания; $\Delta C = W \cdot \tau_{пр}$, где W – скорость обезуглероживания; $\tau_{пр}$ – время пребывания капли в слое шлака.

5. При выгорании углерода в шлак попадает часть оксида железа пропорционально массе сгоревшего углерода: $-\Delta m_{FeO} = \frac{72}{12} \Delta m_c$.

6. Осуществляется переход железа из шлака в металл:

$$\Delta m_{Fe} = \frac{56}{12} \Delta m_c. \quad (5)$$

7. Образуется газовый пузырь в результате сгорания углерода: $\Delta V_{CO} = \frac{22,4}{12} \Delta m_c$.

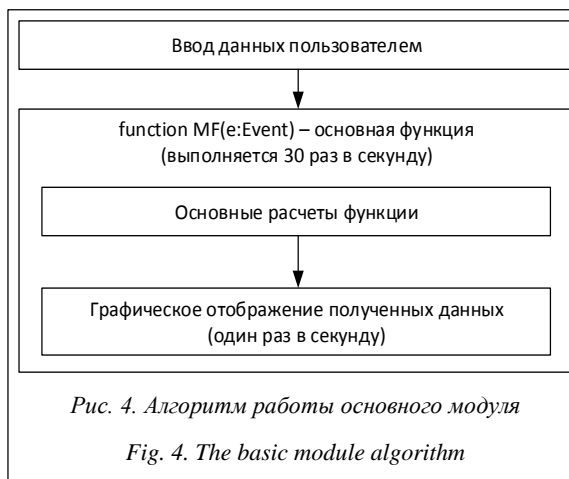
Структура имитационной модели гравитационного сепаратора

Прежде чем переходить к алгоритмам на границе шлак-металл, рассмотрим общую структуру имитационной модели (рис. 3).

Как видно из рисунка 3, основной модуль взаимодействует со всеми классами. Классы руды, металла, шлака, металла внизу и шлака внизу вызывают класс расчетов. При этом визуализация частиц и статистических данных происходит в основном модуле, как и взаимодействие с пользова-

телем, а отображение графиков – в соответствующем классе.

Стоит также отметить, что в основном модуле и классах конденсированных частиц и газов перерасчеты происходят 30 раз в секунду (например, определение нового положения частицы в колонном реакторе). На рисунке 4 показан алгоритм работы основного модуля, в котором выделены два этапа: ввод и инициализация данных, а также вызов основной функции MF.



В основных расчетах функции MF() выполняется ряд задач:

- инициализация переменных и массивов;
- добавление новых частиц с заданной дискретностью;

- расчет плотности по газу;
- расчет изменений состава шлака и металла при диффузионном переходе на границе шлак-металл;
- образование частиц шлака, если высота шлака достигла входного отверстия потока газозвеси;
- расплавление частицы и образование металла, шлака и газа;
- учет изменений металла и шлака при достижении частицей металла нижней границы шлака;
- удаление частиц, достигших выходного отверстия;
- расчеты упругих и неупругих ударов между частицами;
- определение плотности слоев по высоте канала;
- накопление статистических данных об исследуемом объекте и их отображение.

Алгоритмы процессов на границе шлак-металл

Прежде чем обратиться к алгоритму взаимодействия на границе шлак-металл, рассмотрим структуру класса шлака внизу. Данный класс создан для учета изменения материального баланса шлакового слоя, методы и функции которого представлены на рисунке 5.

В классе учитываются массовые доли входящих компонентов, масса и плотность шлака. Переменная для вызова класса расчетов является скрытой и может вызываться только экземпляром данного класса.

При инициализации данного класса все значения по умолчанию приравниваются к 0. Частица шлака, образованная в результате расплавления частицы металла, попадает в нижний слой шлака. Вызывается функция приращения шлака, в которую передаются массовые проценты и масса частицы шлака, затем определяется новый состав шлака. Если это первая частица, которая попадает в шлаковый слой, то состав шлакового слоя будет идентичен составу частицы. Постепенно происходит накопление массы шлака внизу до тех пор, пока высота шлакового слоя не достигнет входного канала. В этом случае частицы шлака определенной крупности будут отрываться от шлакового слоя, соответственно, в функцию изменения состава шлака будет передаваться масса со знаком минус.

Как видно из рисунка 5, в функции определения нового состава шлака идет вызов функции класса расчетов, в котором по входным данным (массам и массовым долям заданного компонента) рассчитывается новая массовая доля в результате прихода или ухода какого-то вещества. Для шлака вызов данной функции происходит 12 раз, так как в шлаке учитываются 12 компонентов.

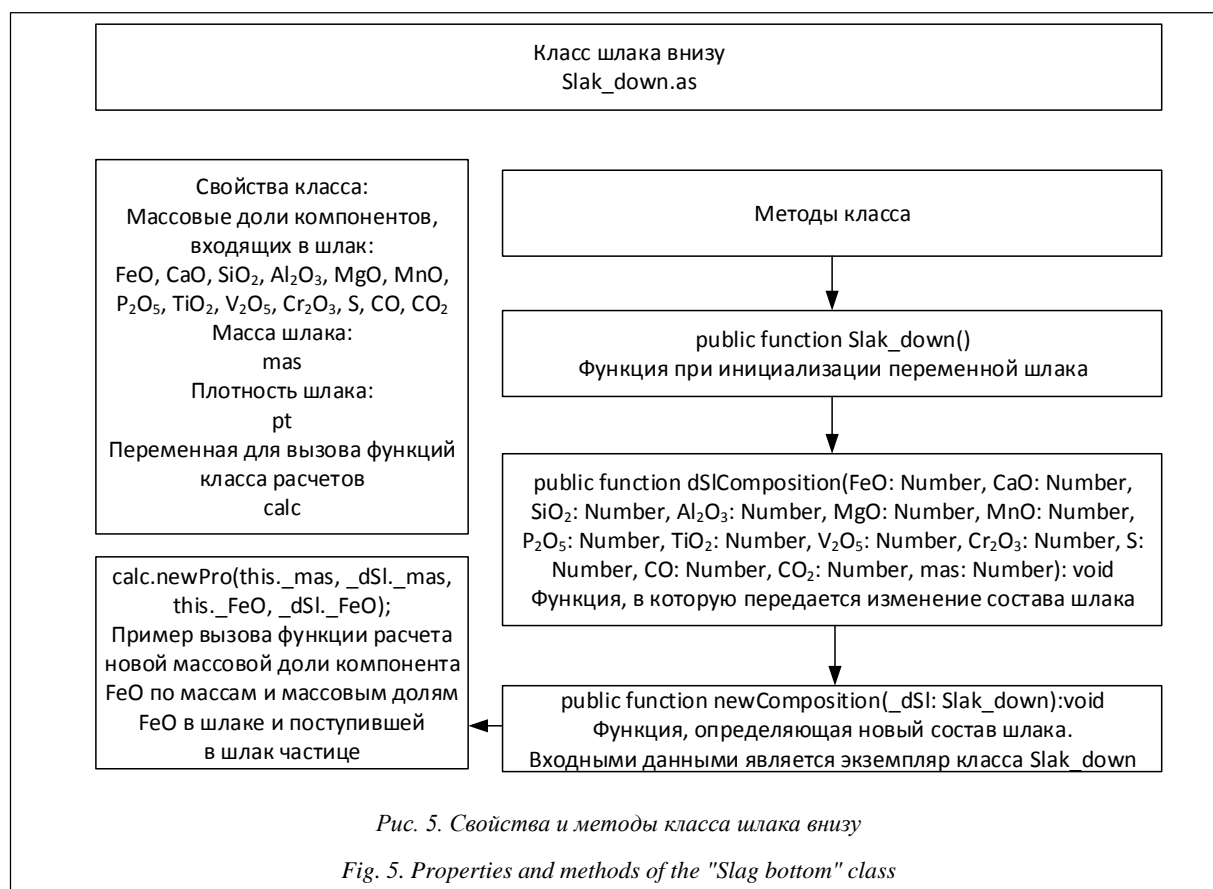


Рис. 5. Свойства и методы класса шлака внизу

Fig. 5. Properties and methods of the "Slag bottom" class

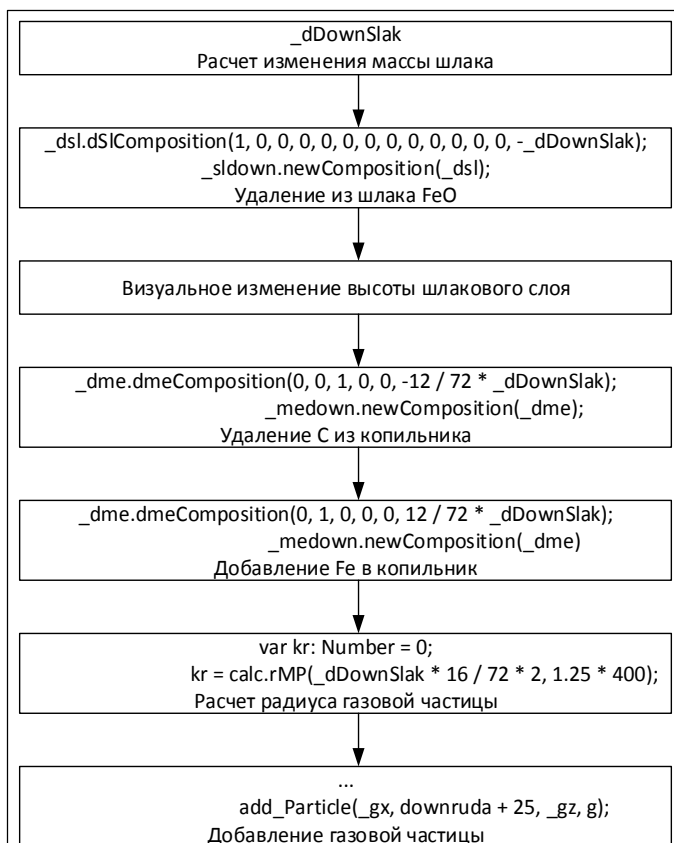


Рис. 6. Алгоритм диффузионного перехода на границе шлак-металл

Fig. 6. The diffusion transition algorithm on the slag-metal border

Устройство класса металла внизу аналогично классу шлака внизу, за исключением компонентов, входящих в металл (S, Fe, C, Mn, P).

Алгоритм диффузионного перехода показан на рисунке 6.

Алгоритм диффузионного перехода на границе шлак-металл (рис. 6) состоит из определения массы шлака, удаления FeO из шлака, удаления C из копильника, добавления Fe в копильник, образования газового пузыря CO, визуального отображения изменения высоты шлакового слоя и отображения образовавшейся газовой частицы.

Рассмотрим более подробно алгоритм расчета изменения массы шлака (рис. 7). Для расчета массы шлака, прореагировавшего в результате диффузионного перехода, необходимо знать балансное содержание FeO (по формуле (2)) и скорость восстановления на границе шлак-металл (по формуле (1)). Эти две функции расположены в классе расчетов. Для знания балансового содержания FeO требуется передать в функцию содержание углерода. А для расчета скорости обезуглероживания передаются содержание FeO в шлаке и балансное FeO (формула (3)), рассчитываемое в соответствующей функции.

Перейдем к рассмотрению второго алгоритма обезуглероживания частицы железа в слое шлака. Напомним, что частица железа образуется в результате расплавления железной руды и под действием силы тяжести проходит через слой шлака. Алгоритм прохождения представлен на рисунке 8.

Алгоритм обезуглероживания частицы железа в слое шлака состоит из следующих пунктов:

- 1) определение массы частицы железа в килограммах;

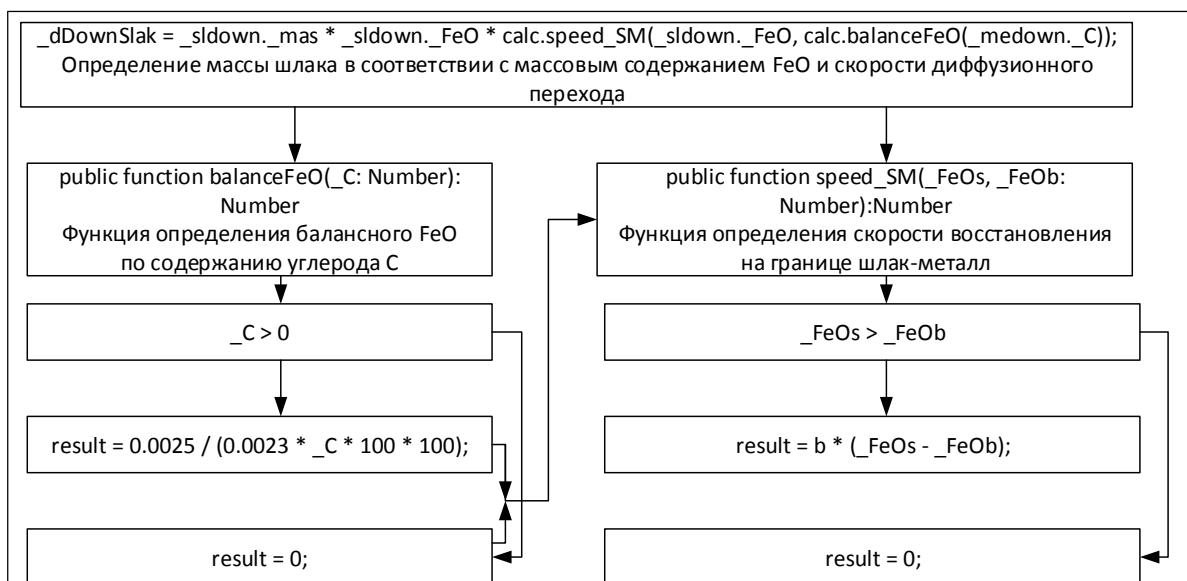
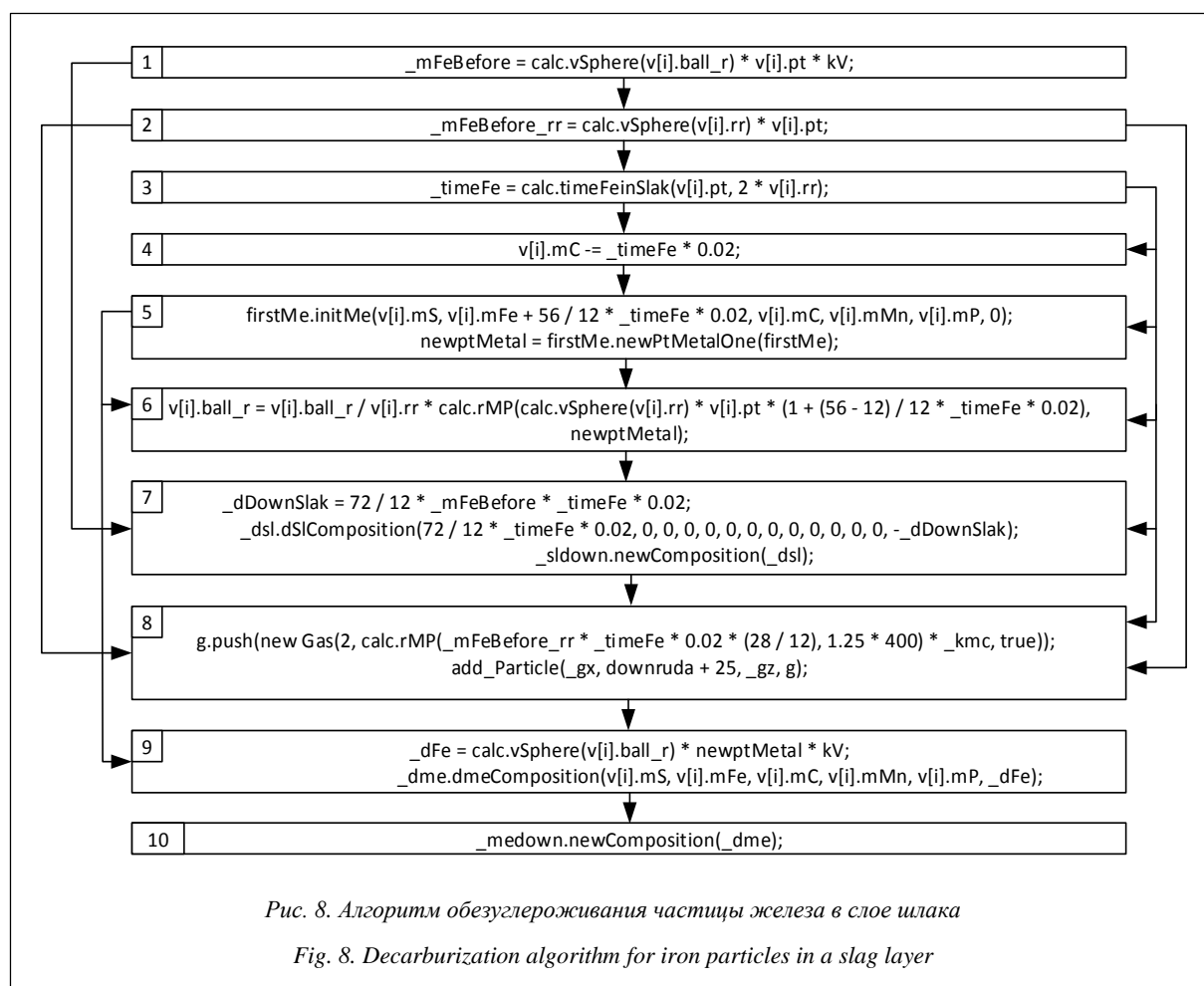


Рис. 7. Алгоритм определения массы прореагировавшего шлака на границе шлак-металл

Fig. 7. The mass determining algorithm for unreacted slag in the slag-metal border



2) определение массы частицы в модельных единицах измерения;

3) определение времени пребывания частицы в шлаке по плотности и радиусу;

4) удаление углерода из частицы железа с учетом скорости выгорания углерода и времени пребывания частицы в железе;

5) инициализация выделившегося из шлака металла (формула (5)), вычисление новой плотности металла;

б) определение нового состава шлака с учетом того, что из него выделился FeO ;

7) создание газовой частицы CO;

8) учет прироста массы железа, пришедшей из шлака;

9) определение нового состава металла в копильнике.

Как видно из рисунка 7, в алгоритме используются данные о частицах железа, которые берутся из класса конденсированных частиц (пп. 1–6, 9). Принадлежность к частицам железа определяется по идентификатору.

В пп. 1–3 происходит обращение к классу расчетов, в котором рассчитываются масса частицы по радиусу и плотности (пп. 1 и 2) и время нахождения частицы металла в шлаке (п. 3).

Время растворения влияет на выгорание углерода из частицы металла (п. 4), определение выделившегося из шлака металла (п. 5), радиуса частицы металла (п. 6), состава шлака (п. 7) и объема частицы газа, образовавшейся в результате реакции (п. 8).

С учетом пришедшего из шлака металла (п. 9) и частицы металла определяется состав металла в копильнике (п. 10).

Алгоритм обезуглероживания частицы металла в шлаке взаимодействует с другими классами программы: классом расчетов (пп. 1–3, 6, 8, 9), классом конденсированных частиц (пп. 1–6, 9), классом газовых частиц (п. 8), классом металла (пп. 5, 9), классом шлака внизу (п. 7), классом металла в копильнике (п. 10).

Таким образом, в статье рассмотрены математическая постановка задачи и алгоритмы на границе шлак-металл. Особенностью алгоритма является применение объектно-ориентированного подхода – разбиение сложного объекта на более простые в виде классов: конденсированных частиц, газовых частиц, шлака внизу, металла внизу. В каждом из классов выделяются необходимые для изучения методы. При этом количество экземпляров класса, например частиц, ограничивается вычислительной

мощностью компьютера. Такой подход обладает гибкостью, так как разработанные классы (или некоторые функции классов) могут быть использованы при разработке других задач. В результате проведенных модельных исследований показано, что при реализации только первого механизма содержание углерода в конечном продукте составляет 4 %, а при реализации двух отмеченных выше схем – 3,5 %, что свидетельствует о возможности более гибкого управления процессом обезуглероживания.

Литература

1. Цымбал В.П., Мочалов С.П., Рыбенко И.А. и др. Процесс СЭР – металлургический струйно-эмульсионный реактор. М.: Металлургиздат, 2014. 488 с.
2. Kroese D.P., Brereton T., Taimre T., Botev Z.I. Why the Monte Carlo method is so important today. *WIREs Comput Stat*, 2014, no. 6, pp. 386–392.
3. Милованов М.М. Разработка инструментальной системы имитационного моделирования бизнес-процессов предприятия // Моделирование, программное обеспечение и наукоемкие технологии в металлургии: сб. докл. III Всерос. науч.-практич. конф. Новокузнецк: Изд-во СибГИУ, 2011. С. 262–266.
4. Меликов А.В. Имитационная модель оценки эффективности стрельбы подразделений РСЗО // NB: Кибернетика и программирование. 2014. № 3. С. 91–102.
5. Гришко М.С., Безносюк С.А., Жуковский М.С. Компьютерное моделирование методом Монте-Карло процессов формирования кластеров никеля в нанопорах материалов // Изв. Алтайского гос. ун-та. 2007. № 3. С. 71–75.
6. Карпов Ю.Г. Имитационное моделирование систем. Введение в моделирование с AnyLogic 5. СПб: БХВ-Петербург, 2009. 400 с.
7. Лычкина Н.Н. Современные технологии имитационного моделирования и их применение в информационных бизнес-системах и системах поддержки принятия решений // Имитационное моделирование. Теория и практика (ИММОД-2005): сб. докл. II Всерос. науч.-практич. конф. СПб: Изд-во ЦНИИТС, 2005. Т. 1. С. 25–31.
8. Кожмяченко В.И. Программирование в среде Delphi. В 2 ч. Новокузнецк: Изд-во СибГИУ, 2015. Ч. 1. 32 с.
9. Кожмяченко В.И. Программирование в среде Visual C#. В 2 ч. Новокузнецк: Изд-во СибГИУ, 2012. Ч. 2. 54 с.
10. Изучение ActionScript 3.0. URL: http://help.adobe.com/ru_RU/as3/learn/as3_learning.pdf (дата обращения: 21.02.2016).
11. Справочник по ActionScript 3.0 для платформы Adobe Flash. URL: http://help.adobe.com/ru_RU/FlashPlatform/reference/actionscript/3/ (дата обращения: 21.02.2016).
12. Лотт Дж., Шалл Д., Питерс К. ActionScript 3.0. Сборник рецептов. СПб: Символ-Плюс, 2007. 608 с.
13. Мук К. ActionScript 3.0 для Flash. Подробное руководство. СПб: Питер, 2009. 992 с.
14. Сеченов П.А. Алгоритм и программная реализация имитационной модели гравитационного сепаратора колонного струйно-эмульсионного реактора // Программные продукты и системы. 2015. № 3 (111). С. 214–219.
15. Цымбал В.П., Сеченов П.А., Оленников А.А., Падалко А.Г. Имитационное моделирование гетерогенного потока и проблема уравнения Навье-Стокса // Современ. науч. достижения металлург. теплотехники и их реализация в пром-ти: сб. докл. Междунар. научн.-практ. конф. Екатеринбург: Изд-во УрФУ, 2015. С. 357–365.

Software & Systems

DOI: 10.15827/0236-235X.116.156-164

Received 02.03.16

2016, vol. 29, no. 4, pp. 156–164

STATEMENT OF THE RESEARCH PROBLEM OF THE DIFFUSION TRANSITION ACROSS THE SLAG-METAL BORDER IN A COLUMN REACTOR AND ITS SOLUTION ALGORITHM

P.A. Sechenov¹, *Postgraduate Student, pavesa89@mail.ru*

V.P. Tsybmal¹, *Ph.D. (Engineering), Professor, tsumbal33@mail.ru*

A.A. Olennikov¹, *Ph.D. (Engineering), Associate Professor, tgs-sibs@mail.ru*

¹ *Siberian State Industrial University, Kirov St. 42, Novokuznechsk, 654007, Russian Federation*

Abstract. The subject of the research is the problems of the diffusion transition across a slag-metal border and metal particles decarbonization when passing through a slag layer. The authors give a mathematical problem statement through the slag-metal border problems within a simulation model of a jet emulsion reactor gravity separator. The article compares the object-oriented programming languages (Delphi, Visual C #, ActionScript 3.0). It allocates simulation model targets. There is a comparative analysis of simulation models created over the past 10 years. Dimension space is taken as a comparison test (one-, two- and three-dimensional). The authors give the scheme of interaction between simulation model classes and the main unit. Particular focus is on the classes: “slag at the botto”, “metal at the bottom”, calculations and the main unit since they implement the assigned task. The paper describes the properties of the slag class (percentages of slag components’ weight and the variable for interaction with the settlement class) and methods (initialization of the class variable, increment function and slag composition change). The calculating class contains the functions determining FeO balanced content and the recovery rate on the slag-metal border. This article shows an algorithm of slag iron decarburization and its interaction with other program classes. To implement the task the authors chose the simulation method (Monte-Carlo). It allows constructing an algorithm with the knowledge of object mechanisms, based on random playing. As a result of the second scheme (metal particles decarbonization when passing through a slag layer) the authors conducted some experiments on the simulation model with the same initial conditions and sufficient modeling time (10 minutes). The simulation studies showed that when implementing only

the first mechanism carbon content in the final product is 4 %, and when implementing two above-mentioned circuits it is 3,5 %. It shows the possibility of more flexible management of a decarburization process.

Keywords: diffusion transfer, simulation model, ActionScript 3.0, gravity separator, the interaction of particles, algorithm, software implementation, jet-emulsion reactor, Monte-Carlo method.

References

1. Tsymbal V.P., Mochalov S.P., Rybenko I.A. *Protsess SER – metallurgichesky struyno-emulsionny reactor* [JER Process – Metallurgical Jet Emulsion Reactor]. Moscow, Metallurgizdat Publ., 2014, 488 p.
2. Kroese D.P., Brereton T., Taimre T., Botev Z.I. Why the Monte-Carlo method is so important today. *WIREs Comput Stat.* 2014, no. 6, pp. 386–392 (in Russ.).
3. Milovanov M.M. Development tool system simulation of business processes. *Sb. dokladov III Vseross. nauch.-praktich. konf. "Modelirovanie, programmnoe obespechenie i naukoemkie tekhnologii v metallurgii"* [Simulation Software and High Technologies in Metallurgy: Proc. 3rd All-Russian Science and Practice Conf.]. Novokuznetsk, SibSIU Publ., 2011, pp. 262–266 (in Russ.).
4. Melikov A.V. Simulation model of evaluating efficiency of Multiple Launch Rocket System units firing. *Kibernetika i programirovanie* [Cybernetics and Programming]. 2014, no. 3, pp. 91–102 (in Russ.).
5. Grishko M.S., Beznosyuk S.A., Zhukovsky M.S. Computer simulation using Monte-Carlo method for nickel clusters formation processes in material nano-pores. *Izvestiya Altayskogo gosudarstvennogo universiteta* [The News of Altai State University]. 2007, no. 3, pp. 71–75 (in Russ.).
6. Karpov Yu.G. *Imitatsionnoe modelirovanie sistem. Vvedenie v modelirovanie s AnyLogic 5* [Simulation systems. Introduction to Modeling with AnyLogic 5]. St. Petersburg, BHV-Peterburg Publ., 2009, 400 p.
7. Lychkina N.N. Modern technologies of simulation modeling and their application in information business systems and decision-making systems. *Sb. dokl. II Vseross. nauch.-praktich. konf. "Imitatsionnoe modelirovanie. Teoriya i praktika" (IMMOD-2005)* [Simulation. Theory and Practice: Proc. 2nd All-Russian Science and Practice Conf. IMMOD 2005]. St. Petersburg, TsNIITS Publ., 2005, vol. 1, pp. 25–31 (in Russ.).
8. Kozhemyachenko V.I. *Programirovanie v srede Delphi* [Programming in Delphi]. 2 parts. Part 1. SibSIU Publ., 2015, 32 p.
9. Kozhemyachenko V.I., Krasnoperov S.Yu. *Programirovanie v srede Visual C#* [Programming in Visual C#]. In 2 parts. Part 2. Novokuznetsk Univ. SibSIU Publ., 2012, 54 p.
10. *Izuchenie ActionScript 3.0* [Learning ActionScript 3.0]. Available at: http://help.adobe.com/ru_RU/as3/learn/as3_learning.pdf (accessed February 21, 2016).
11. *Spravochnik po ActionScript 3.0 dlya platformy AdobeFlash* [ActionScript 3.0 Reference for AdobeFlash Platform]. Available at: http://help.adobe.com/ru_RU/FlashPlatform/reference/actionsript/3/ (accessed February 21, 2016).
12. Lott J., Schall D., Piters K. *ActionScript 3.0. Cookbook*. O'Reilly, 2006 (Russ.ed.: St. Petersburg, Symbol plus Publ., 2007, 608 p.).
13. Moock C. *ActionScript for Flash MX: The Definitive Guide*. O'Reilly, 2002 (Russ.ed.: St. Petersburg, 2009, 992 p.).
14. Sechenov P.A. The algorithm and software implementation of a simulation model of the gravitational separator column jet emulsion reactor. *Programmnye produkty i sistemy* [Software & Systems]. 2015, no. 3, pp. 214–219 (in Russ.).
15. Tsymbal V.P., Sechenov P.A., Olennikov A.A., Padalko A.G. Simulation modeling of heterogeneous flow and the problem of the Navier-Stokes equations. *Sb. dokl. Mezhd. nauch.-praktich. konf. "Sovremennye nauchnye dostizheniya metallurgicheskoy teplotekhniki i ikh realizatsiya v promyshlennosti"* [Modern scientific achievements of metallurgical heat engineering and their industrial implementation: Proc. Int. Science and Practice Conf.]. Ekaterinburg, UrFU Publ., 2015, pp. 357–365 (in Russ.).

Примеры библиографического описания статьи

1. Сеченов П.А., Цымбал В.П., Оленников А.А. Постановка задачи исследования диффузионного перехода через границу шлак-металл в колонном реакторе и алгоритм ее решения // Программные продукты и системы. 2016. Т. 29. № 4. С. 156–164; DOI: 10.15827/0236-235X.116.156-164.

2. Sechenov P.A., Tsymbal V.P., Olennikov A.A. Statement of the research problem of the diffusion transition across the slag-metal border in a column reactor and its solution algorithm. *Programmnye produkty i sistemy* [Software & Systems]. 2016, vol. 29, no. 4, pp. 156–164 (in Russ.); DOI: 10.15827/0236-235X.116.156-164.

УДК 004.932

DOI: 10.15827/0236-235X.116.165-170

Дата подачи статьи: 14.06.16

2016. Т. 29. № 4. С. 165–170

АВТОМАТИЗИРОВАННАЯ ЦИФРОВАЯ ОБРАБОТКА ИЗОБРАЖЕНИЙ ПРИ РЕШЕНИИ ЗАДАЧИ МАГНИТНОЙ ДЕФЕКТΟΣКОПИИ

А.Г. Коробейников, д.т.н., профессор, korobeynikov_a_g@mail.ru;

В.И. Поляков, к.т.н., доцент, v_i_polyakov@mail.ru;

М.Е. Федосовский, к.т.н., профессор, stts@diakont.com;

С.А. Алексанин, аспирант, stts@diakont.com

*(Санкт-Петербургский национальный исследовательский университет
информационных технологий, механики и оптики (Университет ИТМО),
Кронверкский просп., 49, г. Санкт-Петербург, 197101, Россия)*

Методы, базирующиеся на анализе рассеяния магнитных полей возле дефектов после намагничивания изделий, позволяют установить наличие в поверхностных слоях стальных деталей дефектов, например трещин. В областях, где нарушена сплошность, будет происходить изменение магнитного потока. Одним из наиболее известных методов магнитной дефектоскопии является метод магнитного порошка: на поверхность намагниченной детали наносят магнитный порошок (сухой метод) или магнитную суспензию (мокрый метод). При использовании люминесцентных порошков или суспензий на изображениях исследуемых деталей дефекты видны значительно лучше, поэтому появляется возможность автоматизированной обработки таких изображений. В работе представлена автоматизированная процедура выбора методов обработки изображений, на которых присутствуют дефекты. На первом этапе выбранными методами производится обработка изображения с целью улучшения его качества. В основном это методы фильтрации. Затем в автоматизированном режиме определяются границы всех объектов на изображении. На следующем этапе выполняется морфологическая обработка изображения. На заключительном этапе, после вычисления компоненты связности и диаметров объектов, делается вывод о наличии дефектов в изделии. Программная реализация данной процедуры осуществлена в системе MATLAB. Приведен пример обработки фотографии приваренного к трубе фланца, обработанного люминесцентной суспензией Magnaglo 14HF, создавшей люминесцентный зеленый индикаторный рисунок, наблюдаемый в ультрафиолетовом свете при длине волны 365 нм. Обозначены направления и названы методы, которые можно включить в процедуру для ее улучшения.

Ключевые слова: магнитная дефектоскопия, обработка изображений, улучшение изображений, морфологическая обработка изображений.

При решении задачи поиска дефектов в поверхностных слоях стальных деталей достаточно часто применяют методы, базирующиеся на анализе рассеяния магнитных полей возле дефектов после намагничивания этих изделий [1]. В локальной области, где нарушена сплошность, происходит изменение магнитного потока. Картина распределения магнитного поля будет определяться глубиной залегания дефекта, его величиной и формой. Например, у поверхностных трещин, ориентированных перпендикулярно магнитному потоку, наблюдается резко выраженное рассеяние магнитного поля. А у дефектов, ориентированных вдоль магнитного потока, поля рассеяния практически не наблюдаются [2].

Одним из наиболее известных методов магнитной дефектоскопии является метод магнитного порошка (ISO 9934-1:2001): на поверхность намагниченной детали высыпают магнитный порошок (сухой метод) или наносят магнитную суспензию (мокрый метод) [3]; магнитные частицы порошка или суспензии, попавшие в зоны магнитных полей рассеяния, осядут на поверхности детали в области нахождения дефектов.

При использовании магнитной дефектоскопии основным намагничиванием является циркулярное. Продольное намагничивание применяют обычно тогда, когда предполагается, что в исследуемом изделии дефекты строго поперечные, или ис-

пользование циркулярного намагничивания неприемлемо [4].

При использовании люминесцентных порошков или суспензий на изображениях исследуемых деталей дефекты видны значительно лучше [2], поэтому появляется возможность автоматизированной обработки таких изображений [5, 6]. В данной работе представлена автоматизированная процедура выбора методов обработки изображений с целью обнаружения дефектов.

Выбор метода улучшения изображения

Решение задачи автоматизированного выбора методов обработки изображений начинается с ввода цифровой фотографии исследуемой детали [7, 8]. В настоящее время обычно используют цветные фото [9].

Изображение принято определять через функцию $f(x, y)$, где x и y являются пространственными (плоскостными) координатами, а амплитуду f для каждой пары координат x и y называют интенсивностью или яркостью изображения в точке с этими координатами. Словосочетание *уровень серого* часто используют для обозначения яркости монохромного изображения. Цветные изображения формируются комбинацией нескольких монохромных изображений. Например, в цветовой системе RGB цветное изображение строится из трех от-

дельных монохромных компонент (красной, зеленой и синей). По этой причине многие методы и приемы, разработанные для монохромных изображений, могут быть распространены на цветные изображения путем последовательной обработки трех монохромных компонент.

В качестве примера изображения в данной работе будет обрабатываться фотография приваренного к трубе фланца (рис. 1). Изделие было обработано люминесцентной суспензией Magnaglo 14HF, создавшей люминесцентный зеленый индикаторный рисунок, который наблюдается в ультрафиолетовом свете при длине волны 365 нм.

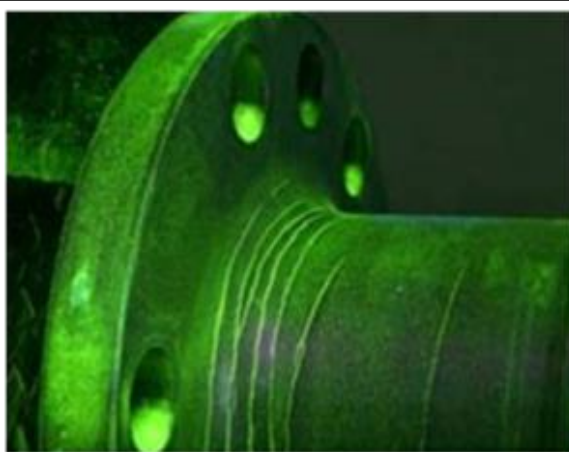


Рис. 1. Исходная фотография

Fig. 1. The original photo

После загрузки изображения необходимо принять решение об улучшении его качества [9]. Это связано с тем, что надежная идентификация дефектов напрямую зависит от качества анализируемого изображения [10]. Низкое качество может привести к неправильной идентификации или к недопустимому определению характеристик дефектов [11]. В связи с этим надежное качество улучшения по-

добных изображений является важной задачей, которую необходимо решать в условиях слабой контрастности и низкой резкости изображений [12].

Общей теории улучшения изображений не существует. Если изображение обрабатывается с целью визуальной интерпретации, то оценку работы конкретного метода дает в конечном счете оператор-исследователь. Методы улучшения настолько разнообразны и используют так много различных подходов к обработке изображения, что трудно собрать осмысленную совокупность подходящих для улучшения методов в одной статье, не проводя отдельное обширное исследование [13–16].

Достаточно часто для улучшения изображений используют фильтрацию. При этом под этим термином наравне с удалением или компенсацией шумов и помех понимается и прямое получение из изображения информации о характеристиках локально неоднородных объектов [17].

Необходимо учитывать, что применение методов цифровой обработки изображений (ЦОИ) для задачи улучшения изображений достаточно часто дает искажение информации об объектах, присутствующих на изображении. Например, следствием повышения контраста и усиления краев достаточно часто является искажение формы и размера дефектоскопического объекта, что крайне нежелательно.

Один из возможных вариантов решения этой проблемы состоит в применении не методов ЦОИ для улучшения изображений, а методов фильтрации полезного сигнала. Но в этом случае под полезным сигналом следует понимать изображения дефектов. Информация от дефектов может быть интерпретирована как локальные неоднородности двумерного конечного нестационарного стохастического сигнала. В этом случае можно решать задачу фильтрации локальных неоднородностей изображений на фоне помех [18].

На рисунке 2 представлена схема автоматизированного выбора алгоритма улучшения изображения.

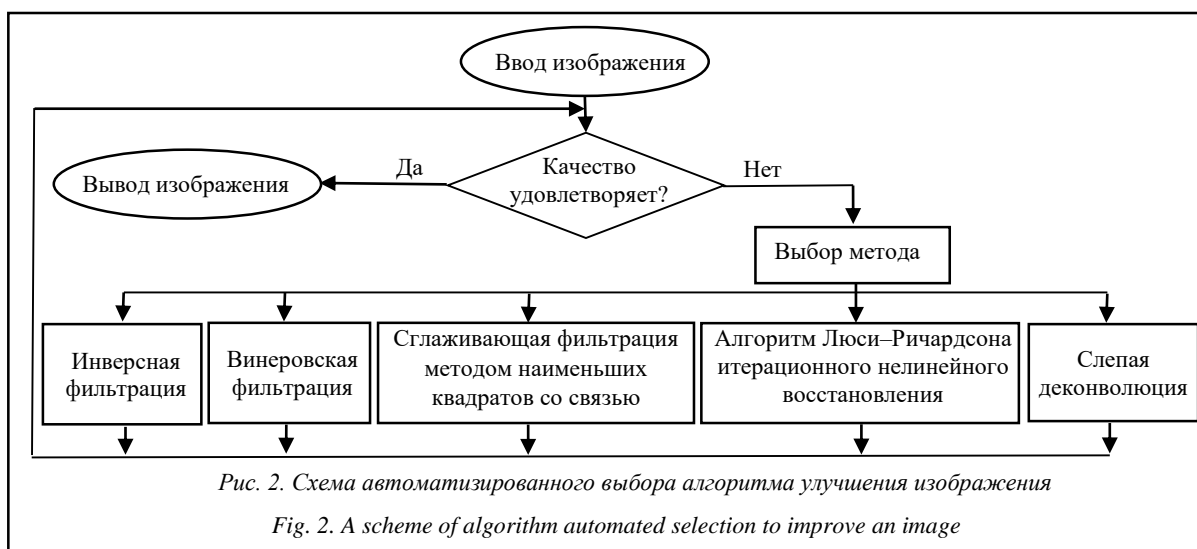


Рис. 2. Схема автоматизированного выбора алгоритма улучшения изображения

Fig. 2. A scheme of algorithm automated selection to improve an image

Как видно из рисунка, в данной методике реализованы следующие методы: инверсная фильтрация, винеровская фильтрация, сглаживающая фильтрация методом наименьших квадратов со связью, алгоритм Люси–Ричардсона итерационного нелинейного восстановления, слепая деконволюция.

Можно отметить, что в последнее время методы слепой деконволюции приобретают все большую популярность.

Для рассматриваемого примера наиболее предпочтительной, на взгляд авторов, оказалась обработка изображения методом Люси–Ричардсона (рис. 3) [19].

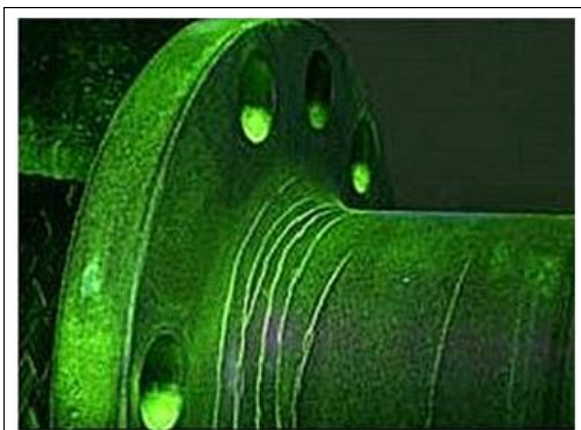


Рис. 3. Изображение, обработанное методом Люси–Ричардсона

Fig. 3. The image processed by Lucy–Richardson

Выделение границ объектов на изображении

Задача выделения границ объектов при ЦОИ возникает достаточно часто, например, выделение границ у дефектов. Дело в том, что многие методы распознавания изображений базируются на определении краев. Это связано с тем, что анализ на краях малочувствителен к общей освещенности. Обнаружение краев выделяет контраст (градиент). По определению, контраст – разная яркость. А ввиду того, что контраст между объектами и задним фоном происходит на границах, это обстоятельство и лежит в основе выделения границы объектов на изображении. Можно отметить, что и человеческое зрение выделяет границы объектов таким же образом.

Существует достаточно много математических методов выделения границ, но все они базируются на операции дифференцирования (в непрерывном случае) или замене конечными разностями в дискретном случае. Кроме того, практически все методы можно разделить на два семейства, которые базируются на поиске максимумов или нулей.

Те методы, которые базируются на поиске максимумов, определяют границы при помощи вычис-

ления силы края, например, вычисление значений первой производной. После этого производится поиск локальных максимумов силы края с использованием информации о предполагаемом направлении границы, которое достаточно часто перпендикулярно вектору градиента.

Так как интенсивность изображения задается функцией от двух переменных $f(x, y)$, ее градиент показывает направление максимального изменения интенсивности и вычисляется следующим обра-

$$\text{зом: } \nabla f = \left[\frac{\partial f(x, y)}{\partial x}, \frac{\partial f(x, y)}{\partial y} \right].$$

Направление градиента (угол) вычисляют по формуле

$$\theta(x, y) = \arctan \left(\frac{\frac{\partial f(x, y)}{\partial y}}{\frac{\partial f(x, y)}{\partial x}} \right) = \arctan \left(\frac{\nabla f_y}{\nabla f_x} \right).$$

Сила края задается длиной градиента:

$$\|\nabla f\| = \sqrt{\left(\frac{\partial f}{\partial x} \right)^2 + \left(\frac{\partial f}{\partial y} \right)^2}.$$

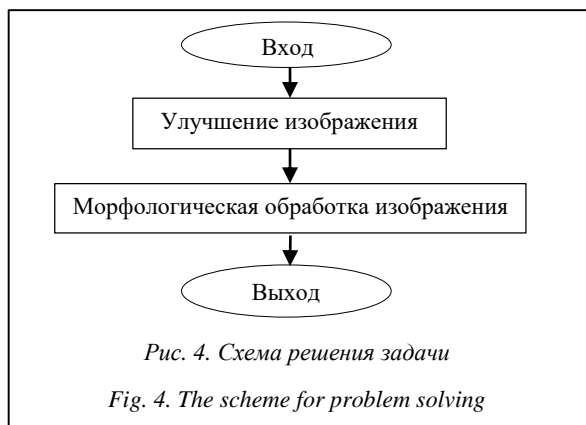
С помощью методов поиска нулей вычисляются пересечение оси абсцисс с выражением для второй производной. Достаточно часто это нули лапласиана или нули нелинейного дифференциального выражения.

Начальным шагом при решении задачи выделения границ достаточно часто служит сглаживание изображения, обычно это фильтр Гаусса.

Методы выделения границ различаются используемыми фильтрами сглаживания и способами расчета силы края. Большинство методов выделения границ базируются на вычислениях градиента изображения, все они различаются по типу применяемых фильтров, которые служат для вычисления градиентов в направлениях x и y .

Обозначенную проблему можно решать при помощи алгоритмов выделения контуров. Наиболее часто применяются методы Собеля, Превитта и Робертса, использующие базовое свойство сигнала яркости – разрывность. Самый общий алгоритм нахождения разрывов состоит в использовании скользящей маски, которую также называют фильтром, окном, ядром или шаблоном. По сути маска – это квадратная матрица, которая поставлена в соответствие заданному множеству пикселей в исходном изображении. Элемент матрицы называют коэффициентом. Математические преобразования для локальных областей изображения с использованием этой матрицы называют пространственной фильтрацией, или просто фильтрацией.

Процесс пространственной фильтрации базируется на линейном перемещении маски фильтра от одного пикселя изображения к другому – в каждом месте (x, y) отклик фильтра рассчитывается с учетом заранее определенных зависимостей. Для

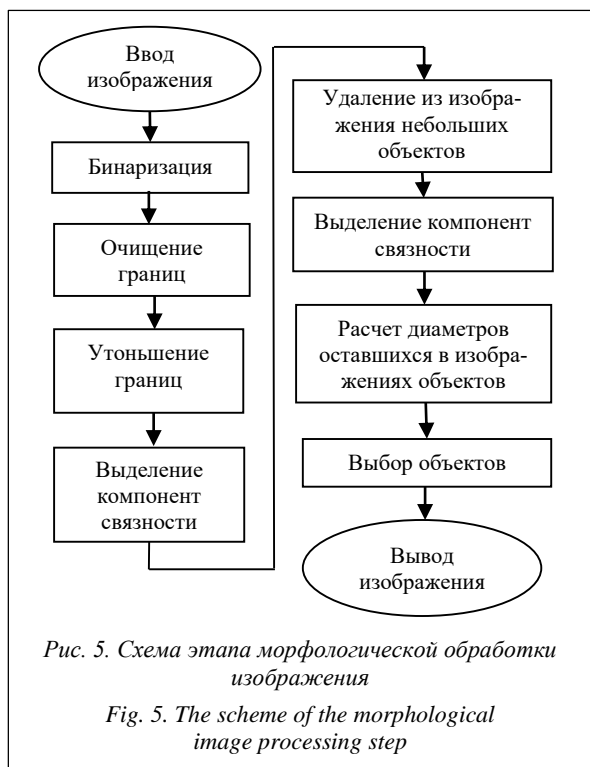


случая линейной пространственной фильтрации отклик задают при помощи суммы произведения коэффициентов фильтра на соответствующие значения пикселей в области, покрытой маской фильтра.

В задаче обнаружения перепадов яркости используют дискретные аналоги производных первого и второго порядков.

Морфологическая обработка изображений

После получения изображения удовлетворительного для исследователя качества проводится морфологическая обработка этого изображения. Такая обработка нужна для извлечения определенных компонентов изображения, другими словами, для обнаружения дефектов, то есть для решения поставленной задачи. На рисунке 4 в виде



схемы представлен процесс решения задачи, а на рисунке 5 – этап морфологической обработки изображения.

Продemonстрируем работу этого этапа на обработке изображения, приведенного на рисунке 3.

Для начала необходимо провести бинаризацию изображения, то есть значения пикселей изображения приравнять к 1 или 0 (рис. 6). При помощи морфологической реконструкции очистим изображение от объектов, соприкасающихся с границей (рис. 7), и произведем утоньшение границ объектов на изображении. Результат представлен на рисунке 8.

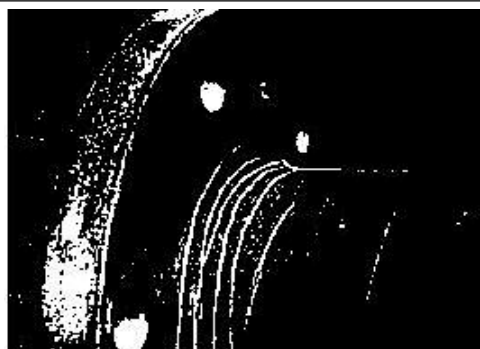


Рис. 6. Бинаризация изображения

Fig. 6. Image binarization

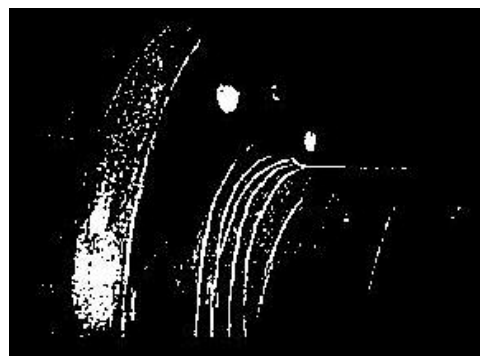


Рис. 7. Результат морфологической реконструкции

Fig. 7. The result of morphological reconstruction

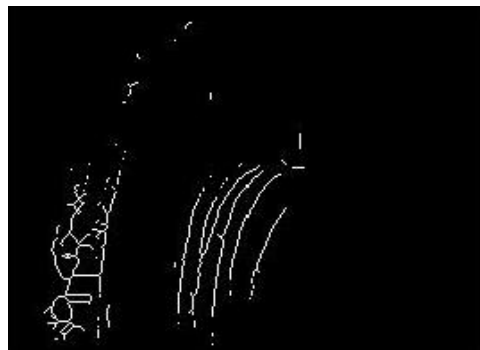


Рис. 8. Утоньшение границ изображения

Fig. 8. Thinning of image borders

Далее на полученном изображении выделим компоненты связности. На рисунке 8 имеются 56 компонент связности. Это означает, что на изображении присутствуют 56 объектов. При помощи операции морфологического замыкания с использованием компонент связности на изображении удалим небольшие объекты. Результат представлен на рисунке 9.

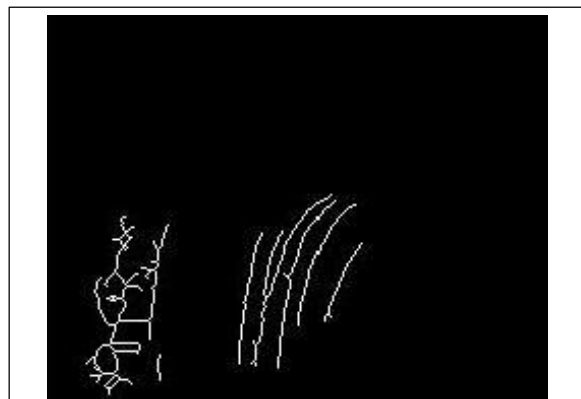


Рис. 9. Результат морфологического замыкания

Fig. 9. The result of morphological short circuit

Для полученного изображения определим компоненты связности. Их осталось семь. Используя эти данные, рассчитаем диаметры для оставшихся объектов.

Объект с наименьшим диаметром можно удалить.

Далее, анализируя рисунки 9 и 3, можно сделать вывод о том, что два левых объекта на рисунке 9 не являются результатом сварочных работ (отверстие и грани фланца). Поэтому они не учитываются при выработке решения о наличии дефектов во фланце.

В результате работы алгоритма специалист получает информацию, на основании которой делает вывод о наличии или отсутствии дефектов в исследуемом изделии. В рассмотренном примере видно, что линии (объекты справа) непрерывны и не имеют резкой смены направлений. Поэтому можно сделать вывод об отсутствии нарушений сплошности (дефектов) в изделии.

Следует заметить, что бинарные изображения по желанию исследователя можно инвертировать.

Инструментарий, применяемый при обработке изображений, может быть совершенно различным [20]. Авторы в ходе представленных исследований использовали MATLAB [21].

Необходимо отметить, что данная процедура является открытой, то есть к ней можно (и даже нужно) подключать различные методы для работы с цифровыми изображениями, например, для реконструкции смазанных изображений. В этом случае необходимо использовать интегральное уравнение Фредгольма I рода, решение которого представляет собой достаточно сложную задачу.

Поэтому привлечение различных методов, например тихоновской регуляризации, является необходимым условием.

Литература

1. Неразрушающий контроль; [под общ. ред. В.В. Клюева]. Т. 1. Кн. 1: Соснин Ф.Р. Визуальный и измерительный контроль. М.: Машиностроение, 2008. 323 с.
2. Исаев М.А., Круглов И.А. Дефекты сварных соединений. Фотоальбом: практич. пособие. М.: Спектр, 2013. 84 с.
3. Шелихов Г.С., Глазков Ю.А. Магнитопорошковый контроль: учеб. пособие. М.: Спектр, 2011. 183 с.
4. Шелихов Г.С. Магнитопорошковая дефектоскопия М.: Спектр, 2010. 336 с.
5. Gonzalez R.C., Woods R.E. Digital image processing. Pearson Intern. Edition prepared by Pearson Education, 2008, 954 p.
6. Конушин А., Барина О., Конушин В., Якубенко А., Велижев А. Введение в компьютерное зрение. 2013. URL: http://courses.graphicon.ru/files/courses/vision/2010/cv_2010_01.pdf (дата обращения: 13.06.16).
7. Фисенко В.Т., Фисенко Т.Ю. Компьютерная обработка и распознавание изображений: учеб. пособие. СПб: Изд-во СПбГУ ИТМО, 2008. 192 с.
8. Gatchin Y.A., Zharinov I.O., Korobeynikov A.G., Zharinov O.O. Theoretical estimation of grassmann's transformation resolution in avionics color coding systems. Modern Applied Science, 2015, vol. 9, no. 5, pp. 197–210.
9. Грищенко А.Ю., Коробейников А.Г. Методы и модели цифровой обработки изображений. СПб: Изд-во Политехн. ун-та, 2014. 190 с.
10. Красильников Н.Н. Реставрация изображений с учетом их структуры // Оптический журнал. 2009. Т. 76. № 2. С. 7–12.
11. Красильников Н.И. Цифровая обработка 2D- и 3D-изображений: учеб. пособие. СПб: БХВ-Петербург, 2011. 608 с.
12. Jahne B. Digital Image Processing. Springer Sc. & Business Media, 2005, 608 p.
13. Tschumperle D. Fast anisotropic smoothing of multi-valued images using curvature-preserving. PDE's Init'l Jour. on Comp. Vision, 2006, no. 68 (1), pp. 65–82.
14. Богатырев В.А., Богатырев А.В. Функциональная надежность систем реального времени // Науч.-технич. вестн. информ. технологий, механики и оптики. 2013. № 4. С. 150–151.
15. Коробейников А.Г., Кудрин П.А., Сидоркина И.Г. Алгоритм распознавания трехмерных изображений с высокой детализацией // Вестн. Поволж. гос. технологич. ун-та. Сер.: Радиотехнич. и инфокоммуникационные системы. 2010. № 2. С. 91–98.
16. Коробейников А.Г., Алексанин С.А. Методы автоматизированной обработки изображений при решении задачи магнитной дефектоскопии // Кибернетика и программирование. 2015. № 4. С. 49–61; DOI: 10.7256/2306-4196.2015.4.16320; URL: http://e-notabene.ru/kp/article_16320.html (дата обращения: 13.06.16).
17. Liu C., Szeliski R., Sing Bing Kang, Zitnick C.L., Freeman W.T. Automatic estimation and removal of noise from a single image // IEEE Transactions on Pattern Analysis and Machine Intelligence, 2008, vol. 30.1, no. 2, pp. 299–314.
18. Voskoboinikov Yu.E. A combined nonlinear contrast image reconstruction algorithm under inexact point-spread function. Optoelectronics, Instrumentation and Data Processing, 2007, no. 6, pp. 489–499.
19. Кирьянов К.А. Инструментальная реализация алгоритмов реконструкции искаженных изображений // GraphiCon-2010: сб. тр. 20-й Междунар. конф. СПб: Изд-во СПбГУ ИТМО, 2010. С. 188–191.
20. Визильтер Ю.В., Желтов С.Ю., Князь В.А. и др. Обработка и анализ цифровых изображений с примерами на LabVIEW IMAQ Vision. М.: ДМК Пресс, 2007. 464 с.
21. Gonzalez R.C., Woods R.E., Eddins S.L. Digital image processing using MATLAB. Gatesmark Publ. a Division of Gatesmark, LLC, 2009, 827 p.

AUTOMATED DIGITAL PROCESSING OF IMAGES IN SOLVING THE PROBLEM OF MAGNETIC DEFECTOSCOPY

A.G. Korobeynikov¹, Dr.Sc. (Engineering), Professor, korobeynikov_a_g@mail.ru

V.I. Polyakov¹, Ph.D. (Engineering), Associate Professor, v_i_polyakov@mail.ru

M.E. Fedosovsky¹, Ph.D. (Engineering), Professor, stts@diakont.com

S.A. Aleksanin¹, Postgraduate Student, aleksanin@diakont.com

¹ The National Research University of Information Technologies, Mechanics and Optics, Kronverksky Ave. 49, St. Petersburg, 197101, Russian Federation

Abstract. The methods that are based on the analysis of dispersion of magnetic fields near defects after magnetization of these products allow determining the existence of defects in a surface coat of steel details, for example cracks. In areas with continuity violation there will be a change of a magnetic flux. One of the most known methods of magnetic fault detection is the method of magnetic powder. In this case we apply magnetic powder (a dry method) or magnetic suspension (a wet method) on a surface of a magnetized detail. When using luminescent powders or suspensions, defects are more visible on the images of the studied details. Therefore, there is a possibility of automated image processing. The paper presents an automated procedure of choosing image processing methods. It also provides an example of processing a steel detail image aiming to detect its defects by means of shining lines, which appear after applying a wet method. Software implementation of the proposed approach is in MATLAB. The authors determine ways and methods that might be included in the abovementioned procedure to improve it.

Keywords: magnetic fault detection, image processing, improvement of images, morphological processing of images.

References

1. *Nerazrushayushchy kontrol* [Non-destructive Control]. V.V. Klyuev (Ed.). Sosnin F.R. *Vizualny i izmeritelny kontrol* [Visual and Measurement Control]. Moscow, Mashinostroenie Publ., 2008, vol. 1, book 1, 323 p.
2. Isaev M.A., Kruglov I.A. *Defekty svarykh soedineniy. Fotoalbum* [Welding Defects. Photoalbum]. Practical guide. SPEKTR, 2013, 84 p.
3. Shelikhov G.S., Glazkov Yu.A. *Magnitoporoshkovaya defektoskopiya* [Magnetic Powder Inspection]. Study guide. SPEKTR Publ., 2011, 183 p.
4. Shelikhov G.S. *Magnitoporoshkovaya defektoskopiya* [Magnetic Powder Inspection]. SPEKTR Publ., 2010, 336 p.
5. Gonzalez R.C., Woods R.E. *Digital Image Processing*. 3rd ed. Pearson Int. Edition prepared by Pearson Education, 2008, 954 p.
6. Konushin A., Barinova O., Konushin V., Yakubenko A., Velizhev A. *Vvedenie v kompyuternoe zrenie* [Introduction to Computer Vision]. MGU VMK Publ., Graphics&Media Lab, 2013. Available at: http://courses.graphicon.ru/files/courses/vision/2010/cv_2010_01.pdf (accessed June 13, 2016).
7. Fisenko V.T., Fisenko T.Yu. *Kompyuternaya obrabotka i raspoznavanie izobrazheniy* [Computer Processing and Image Recognition]. Study guide. St. Petersburg, SPbGU ITMO Publ., 2008, 192 p.
8. Gatchin Y.A., Zharinov I.O., Korobeynikov A.G., Zharinov O.O. Theoretical estimation of Grassmann's transformation resolution in avionics color coding systems. *Modern Applied Science*. 2015, vol. 9, no. 5, pp. 197–210.
9. Grishentsev A.Yu., Korobeynikov A.G. *Metody i modeli tsifrovoy obrabotki izobrazheniy* [Methods and Models of Digital Image Processing]. St. Petersburg, Politekhicheskyy Univ. Publ., 2014, 190 p.
10. Krasilnikov N.N. Image restoration considering their structure. *Optichesky zhurnal* [Optical Journal]. 2009, vol. 76, no. 2, pp. 7–12 (in Russ.).
11. Krasilnikov N.I. *Tsifrovaya obrabotka 2D- i 3D-izobrazheniy* [2D and 3D Image Digital Processing]. Study guide. St. Petersburg, BHV-Peterburg Publ., 2011, 608 p.
12. Jahne B. *Digital Image Processing*. 6th ed. Springer Science & Business Media Publ., 2005, 608 p.
13. Tschumperle D. Fast anisotropic smoothing of multi-valued images using curvature-preserving. *PDE's Init'l Jour. on Computer Vision*. 2006, no. 68 (1), pp. 65–82.
14. Bogatyrev V.A., Bogatyrev A.V. Functional reliability of real-time systems. *Nauchno-tekhnichesky vestnik ITMO* [Scientific and Technical Journal of Information Technologies, Mechanics and Optics]. 2013, no. 4, pp. 150–151 (in Russ.).
15. Korobeynikov A.G., Kudrin P.A., Sidorkina I.G. 3D images recognition algorithm with high resolution. *Vestnik Povolzhskogo gos. tekhnologicheskogo univ. Seriya: Radiotekhnicheskie i infokommunikatsionnye sistemy* [Vestnik of Volga State University of Technology. Series: Radioengineering and Infocommunication Systems]. 2010, no. 2, pp. 91–98 (in Russ.).
16. Korobeynikov A.G., Aleksanin S.A. Methods of automated image processing when solving a magnetic flaw detection problem. *Cybernetics and programming*, 2015, no. 4, pp. 49–61; DOI: 10.7256/2306-4196.2015.4.16320; URL: http://e-notabene.ru/kp/article_16320.html (accessed June 13, 2016).
17. Liu C., Szelski R., Kang S.B., Zitnick C.L., Freeman W.T. Automatic Estimation and Removal of Noise from a Single Image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2008, vol. 30.1, no. 2, pp. 299–314.
18. Voskoboinikov Yu.E. A Combined Nonlinear Contrast Image Reconstruction Algorithm under Inexact Point-Spread Function. *Optoelectronics, Instrumentation and Data Processing*. 2007, no. 6, pp. 489–499.
19. Kiryanov K.A. Tool implementation of image restoration algorithms. *Trudy 20 Mezhdunar. konf. "GraphiCon-2010"* [Proc. 20th Int. Conf. "GraphiCon-2010"]. St. Petersburg, SPbGU ITMO Publ., 2010, pp. 188–191 (in Russ.).
20. Vizilter Yu.V., Zheltov S.Yu., Knyaz V.A. *Obrabotka i analiz tsifrovyykh izobrazheniy s primerami na LabVIEW IMAQ Vision* [Digital Image Processing and Analysis with examples in LabVIEW IMAQ Vision]. Moscow, DMK Press, 2007, 464 p.
21. Gonzalez R.C., Woods R.E., Eddins S.L. *Digital Image Processing Using MATLAB*. 2nd ed., Gatesmark Publ., A Division of Gatesmark, LLC, 2009, 827 p.

УДК 536.24: 621.396

DOI: 10.15827/0236-235X.116.171-175

Дата подачи статьи: 16.09.16

2016. Т. 29. № 4. С. 171–175

ПРОГРАММНЫЙ КОМПЛЕКС МОДЕЛИРОВАНИЯ СТОХАСТИЧЕСКИХ ТЕМПЕРАТУРНЫХ ПОЛЕЙ В ТЕХНИЧЕСКИХ СИСТЕМАХ STF-ElectronMod

П.И. Кандалов, научный сотрудник, petrki87@gmail.ru

*(Центр визуализации и спутниковых информационных технологий НИИСИ РАН,
Нахимовский просп., 36, корп. 1, г. Москва, 117218, Россия)*

В статье рассматривается программный комплекс STF-ElectronMod для моделирования стационарных стохастических трехмерных температурных полей в технических системах. Практика показывает, что реальные температурные поля технических систем носят неопределенный интервальный характер. Это обусловлено интервальным характером факторов, определяющих тепловой режим технической системы, а именно: параметрами конструкции технической системы, обладающими статистическим технологическим разбросом изготовления; факторами, возникающими при функционировании технической системы (потребляемые мощности, тепловые потоки, параметры внутренней среды); факторами окружающей среды (температуры среды, хладоносителей, скорости потоков и пр.). Вместе с тем моделирование температурных полей технических систем в настоящее время проводится в предположении, что все параметры теплового режима однозначно и абсолютно точно известны. В статье описывается структура программного комплекса STF-ElectronMod, предназначенного для моделирования интервальных стохастических температурных полей в технических системах при неопределенности входных определяющих данных. Метод и алгоритм компьютерного моделирования основаны на авторских разработках по матрично-топологическому методу, программному комплексу моделирования детерминированных трехмерных температурных полей технических систем, а также на методе статистических испытаний. Применение разработанного метода и алгоритма для моделирования интервально стохастических температурных полей рассмотрено на примере реальной электронной системы.

Ключевые слова: программный комплекс, математическое моделирование, температурные поля, стохастический, интервальный.

Системы теплового проектирования можно создавать только с применением современных информационных технологий компьютерного и суперкомпьютерного моделирования, являющихся инструментом проектирования и отработки новых систем. Многофункциональные высокоэффективные программные комплексы позволяют еще на этапе проектирования с высокой степенью достоверности и адекватности прогнозировать поведение, эксплуатационные характеристики и параметры будущих сложных технических систем. Решение данной проблемы особенно актуально при создании высоконадежных технических систем, предназначенных для эксплуатации в экстремальных условиях окружающей среды, при тепловых, механических, химических, радиационных и космических воздействиях. Разработка многофункционального программного комплекса теплового проектирования технических систем является критически важной для создания новой конкурентоспособной техники.

Математические модели, положенные в основу современных программных комплексов теплового проектирования (Beta Soft, TGM, Thermal Analysis, Thermal Solution, ANSYS и др.), исходят из допущения, что факторы, определяющие тепловые режимы и, как следствие, температурные поля в технических системах, носят детерминированный характер. Это означает, что все исходные данные, определяющие характер протекания теплового процесса в системе, являются однозначно определенными и полностью известными. В действитель-

ности факторы, которые определяют тепловые процессы в технических системах, в частности в электронных системах, будучи интервальными [1–3], представляют собой случайные величины, принимающие значения внутри своих интервалов изменения, подчиняющиеся некоторым усеченным (ограниченными интервалами) законам распределения вероятностей. Интервальную неопределенность тепловых процессов обуславливают следующие факторы:

- статистические технологические разбросы при изготовлении технической системы и ее элементов;
- интервально стохастические параметры окружающей среды;
- стохастические факторы, возникающие в процессе функционирования технической системы.

Таким образом, для построения высокоэффективного программного комплекса компьютерного моделирования необходимо, чтобы математические модели и последующие алгоритмы, положенные в основу программы, учитывали случайный характер факторов, определяющих тепловые режимы технической системы. Такие математические модели обладают высоким уровнем достоверности и адекватности отражения реальности.

Для решения поставленной задачи были разработаны стохастическая математическая модель анализа интервально *стохастических температурных распределений* (СТР-модель) [4–6] и алгоритм компьютерного расчета распределений ста-

математических мер (KPCM-алгоритм) стационарных интервально стохастических температурных распределений. Полученные математическая модель и алгоритм реализованы в программном комплексе STF-ElectronMod (Simulation of Temperature Fields of Electronic Modules) [7].

В статье рассматриваются структура разработанного программного комплекса STF-ElectronMod, а также процесс моделирования температурных распределений электронного модуля с этапа задания исходных данных до этапа визуализации результатов расчета. Приводится пример моделирования электронного модуля.

Разработанный программный комплекс STF-ElectronMod применяется для расчетов интервально стохастических распределений температуры при разработке и создании новых электронных систем различного назначения. Тепловое проектирование и моделирование электронных систем с помощью данного программного комплекса показали адекватные реальности результаты. На сегодняшний день как в России, так и за рубежом аналогов данному программному комплексу не существует.

Структура программного комплекса STF-ElectronMod

Программный комплекс STF-ElectronMod предназначен для компьютерного моделирования детерминированных и интервально стохастических трехмерных температурных полей в электронных модулях, содержащих многослойную печатную плату (МПП), установленные на ее поверхности интегральные микросхемы (МС) и электрорадиоэлементы (ЭРЭ), конструктивные элементы теплоотвода, охлаждения и крепления, с учетом особенностей конструкции, реальных условий эксплуатации и монтажа МС и ЭРЭ, многослойности МПП, конструкций корпусов МС, при различных условиях теплоотвода и охлаждения.

Программный комплекс реализован с применением объектно-ориентированного программирования (ООП) на языке высокого уровня Pascal с использованием программной среды Embarcadero RAD Studio для Microsoft Windows. Применение ООП обеспечивает комплексу высокую степень модульности благодаря таким свойствам, как инкапсуляция, полиморфизм и позднее связывание. Модульность архитектуры соответствует основному базовому принципу открытости программного комплекса, что обеспечивает создание надежного и эффективного комплекса моделирования, который в случае необходимости позволяет наращивать и модернизировать отдельные модули системы, при этом производимые действия оказывают минимальное влияние на работоспособность системы в целом.

В структуру программного комплекса входят следующие элементы:

- подсистема GUI (Graphic User Interface);
- подсистема математического моделирования;
- подсистема STF Kernel;
- подсистема информационного обеспечения.

Подсистема GUI – это удобная разветвленная пользовательская оболочка, в функции которой входит предоставление пользователю гибких и удобных механизмов:

- задания исходных данных;
- графического отображения конструкции электронного модуля в соответствии с заданными исходными данными;
- визуализации рассчитанных статистических мер (математических ожиданий (МО), дисперсий (Д), среднеквадратических отклонений (СКО)) интервально стохастических трехмерных температурных полей электронных модулей в виде цветных изотерм;
- визуализации рассчитанных детерминированных трехмерных температурных полей электронных модулей в виде цветных изотерм;
- сохранения и загрузки исходных и рассчитанных данных в едином формате хранения модельных данных;
- интерактивного отображения рассчитанных данных в графическом и цифровом видах;
- представления конструкций МС, ЭРЭ и теплоотводов системы охлаждения с помощью многополюсного направленного графа тепловой схемы;
- формирования отчета и протокола результатов моделирования в удобном для пользователя виде.

Подсистема математического моделирования реализует программный модуль с алгоритмами математических моделей. В данной подсистеме реализован ряд алгоритмов, обеспечивающих расчеты:

- интервально стохастических трехмерных температурных полей в электронных модулях [4];
- детерминированных температурных полей в электронных модулях [8];
- статистических мер (МО, Д, СКО) интервально стохастических трехмерных температурных полей электронных модулей [4, 5];
- границ интервалов изменения реальных значений стохастических температур в различных точках электронного модуля [4];
- конструкций МС, ЭРЭ и теплоотводов системы охлаждения с помощью многополюсного направленного графа тепловой схемы [9, 10].

В основу подсистемы расчетного ядра STF Kernel программного комплекса STF-ElectronMod положена детерминированная стационарная модель уравнений теплопроводности [7]. STF Kernel

предоставляет общий *Application Programming Interface* (API) для подсистемы математического моделирования. На основе функции, классов и методов, предоставляемых API, реализуются все алгоритмы в подсистеме математического моделирования программного комплекса. API имеет унифицированный интерфейс и может быть предоставлен внешним приложением либо посредством DLL (Dynamic Link Library), либо в качестве COM-объекта (Component Object Model), что делает подсистему STF Kernel универсальной по отношению к сторонним приложениям.

Подсистема информационного обеспечения программного комплекса состоит из двух файловых БД: материалов и тепловых схем.

БД материалов хранится единым файлом в формате XML (eXtensible Markup Language) и содержит знания о теплопроводности материала и его названии, которые используются при задании исходных данных теплопроводностей слоев МПП, зазоров между элементами и МПП, выводов и т.д.

БД тепловых схем представляет собой совокупность файлов формата XML, каждый из которых содержит описание шаблона тепловой схемы одного элемента в виде направленного графа [10].

Хранение информации о материалах и шаблонов тепловых схем в виде файлов формата XML обеспечивает легкий доступ к операциям добавления/редактирования/удаления и не требует дополнительных манипуляций на машине пользователя, связанных с настройками полноценных СУБД. Помимо этого, файлы БД могут быть перенесены с одной машины на другую путем простого копирования на внешние носители информации.

Практическое применение разработанного метода и алгоритма моделирования

Рассмотрим пример применения СТР-модели для численного моделирования электронной системы, содержащей девять однотипных электронных модулей, образующих восемь вертикальных каналов, через которые нагнетается воздух. Электронные модули зажимаются по краям специальными фиксирующими устройствами. Расчеты теплового и аэродинамического режимов системы показали, что коэффициент теплоотдачи принудительного воздушного охлаждения через каналы, образованные электронными модулями, составляет $53 \text{ Вт/м}^2\text{К}$. Статистический разброс мощностей источников теплоты (МС, установленных в электронных модулях) составляет $\pm 10^\circ\text{C}$ от номинального значения для МС (рис. 1) с номерами 1–7, 10, 14–17, 22–25, 27, 28, 31, 36, 38–48 и $\pm 15^\circ\text{C}$ для остальных элементов. Математическое ожидание температуры воздуха в каналах между электронными модулями и ее статистический разброс составляют $71,3^\circ\text{C} \pm 3\%$. На крышке корпуса МС 1 (рис. 1) через эластичную теплопроводную

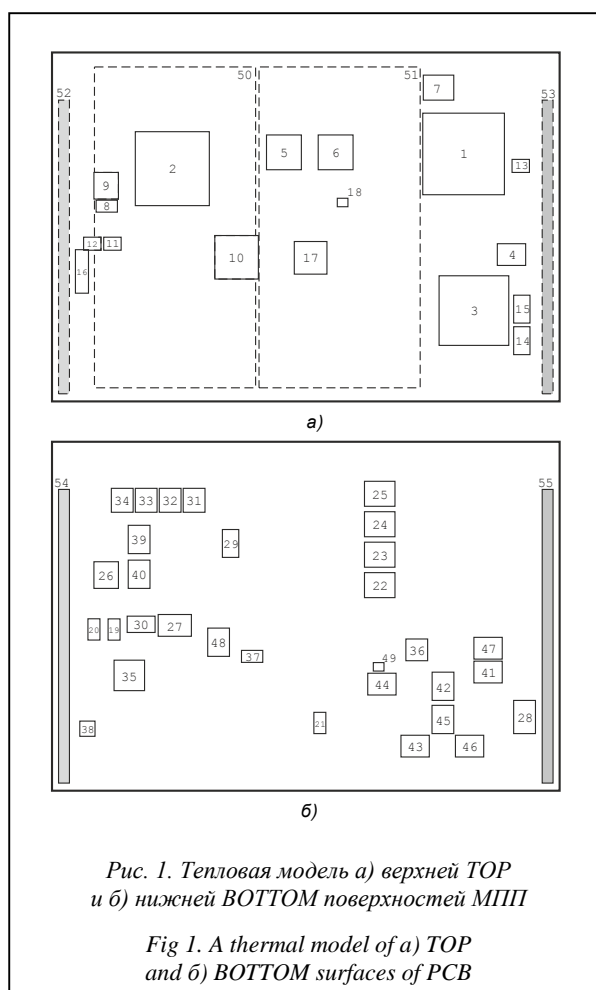


Рис. 1. Тепловая модель а) верхней TOP и б) нижней BOTTOM поверхностей МПП

Fig 1. A thermal model of a) TOP and б) BOTTOM surfaces of PCB

прокладку установлен штыревой радиатор, выполненный из меди. Многослойная структура МПП электронного модуля состоит из 31 слоя, содержащего сигнальные слои, слои полиимида, слои заземления и распределения питания.

В конструкцию электронного модуля входят также два дополнительных мезонинных электронных модуля, электрические разъемы и металлические планки, которые прижимаются и фиксируются в направляющих системы.

Моделирование по разработанному программному комплексу показывает, что реальные значения температуры корпусов электронного модуля не являются однозначно определенными и детерминированными, а представляют собой интервально стохастические температуры, изменяющиеся в пределах своих интервалов. Гистограммы, рассчитанные для интервальных распределений температуры кристаллов и корпусов микросхемы МС 1, приведены на рисунке 2.

Максимальному разогреву подвержена МС 36: МО температуры корпуса – $105,98^\circ\text{C}$, СКО – $2,83^\circ\text{C}$, интервал изменения – $99,33\text{--}112,74^\circ\text{C}$. Таким образом, разброс реальной температуры МС № 36 может достигать $\pm 7\%$ относительно ее МО ($= 105,98^\circ\text{C}$).

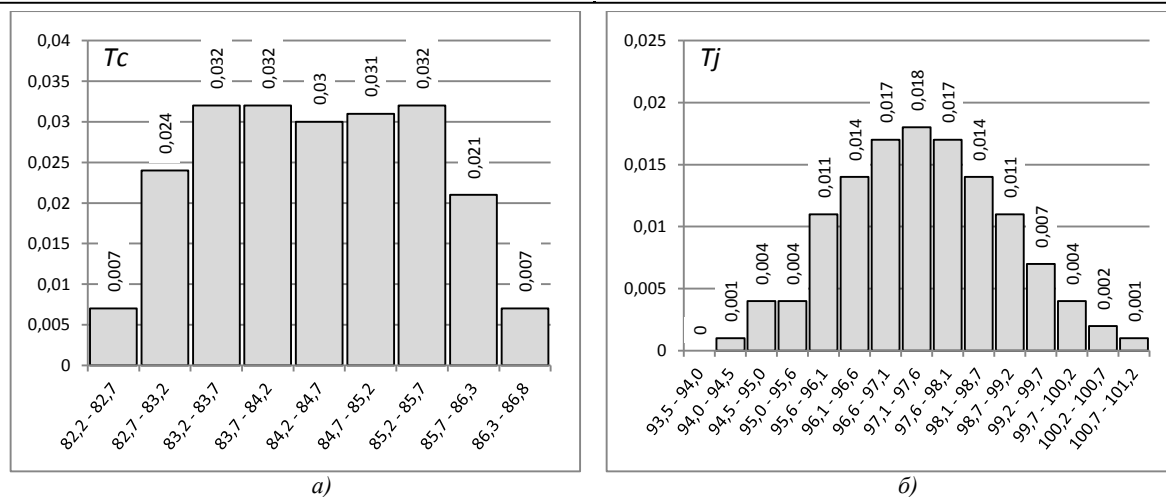


Рис. 2. Гистограммы, рассчитанные для интервальных распределений температур а) корпуса и б) кристалла для микросхемы МС 1

Fig. 2. Histograms for interval temperature distribution of a case (a) and MC 1 silicon (b)

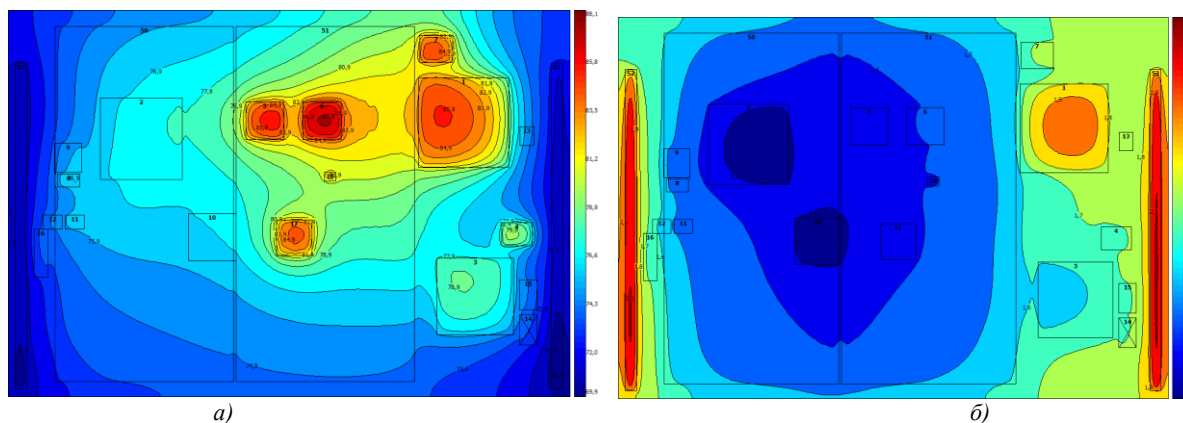


Рис. 3. Распределение температурного поля а) МО и б) СКО на верхней TOP поверхности МПП ЭМ

Fig. 3. Expected values (a) and standard deviation (b) temperature fields on TOP surface of PCB

Результаты моделирования статистических мер (МО и СКО) температурных полей на плате электронного модуля в условиях конвективного теплообмена для TOP поверхности платы приведены на рисунке 3. Минимальное значение температуры для распределения МО температурного поля электронного модуля составляет 69,96 °C, максимальное – 88,13 °C.

Разработанная математическая СТР-модель и алгоритм КРСМ стационарных интервально стохастических температурных распределений реализованы в программном комплексе STF-ElectronMod и применяются для теплового проектирования сложных электронных систем. Программный комплекс предоставляет эргономичный пользовательский интерфейс, который позволяет эффективно производить оценку мест наибольшего нагрева конструкции путем визуализации результатов моделирования в виде полей МО и СКО, а также построением гистограмм интервальных распределений

температур. Как показывает практика моделирования электронного модуля, заложенные в программный комплекс методы позволяют моделировать трехмерные температурные поля электронного модуля с высокой эффективностью.

Литература

1. Исаев И.П., Иньков Ю.М., Маричев М.А. Вероятностные методы расчета полупроводниковых преобразователей. М.: Энергоиздат, 1983. 96 с.
2. Чеканов А.Н. Расчеты и обеспечение надежности электронной аппаратуры: учеб. пособие. М.: КНОРУС, 2012. 440 с.
3. Keller C.J., Antonetti V.W. Statistical thermal design for computer electronics. Electronic Packaging and Production, vol. 19, no. 3, pp. 55–62.
4. Мадера А.Г., Кандалов П.И. Анализ интервально стохастических температурных полей технических систем // Программные продукты и системы. 2014. № 4. С. 41–45.
5. Кандалов П.И., Мадера А.Г. Моделирование температурных полей технических систем в условиях интервально стохастической неопределенности // VI Рос. нац. конф. по теплообмену: сб. тр. М.: Изд-во МЭИ, 2014. Т. 1. С. 191–192.

6. Решетников В.Н. Космические телекоммуникации. Системы спутниковой связи и навигации. СПб: Ленинградское изд-во, 2010. 134 с.

7. Кандалов П.И., Мадера А.Г. Прогр. для ЭВМ: Simulation of Temperature Fields of Electronic Modules 2014 (STF-ElectronMod 2014); Свид. 2015619835. Рос. Федерация. № 2015613488; заявл. 28.04.15; опубл. 15.09.15.

8. Кандалов П.И., Мадера А.Г. Моделирование температурных полей в многослойных структурах // Программные про-

дукты и системы. 2008. № 4. С. 46 – 49.

9. Мадера А.Г., Кандалов П.И. Матрично-топологический метод математического и компьютерного моделирования температурных полей в электронных модулях: программный комплекс STF-ElectronMod // Программные продукты и системы. 2012. № 4. С. 160–164.

10. Мадера А.Г. Моделирование теплообмена в технических системах. М.: Изд-во НФ «Первая исслед. лаб. им. акад. В.А. Мельникова», 2005. 208 с.

Software & Systems
DOI: 10.15827/0236-235X.116.171-175

Received 16.09.16
2016, vol. 29, no. 4, pp. 171–175

SOFTWARE MODELING OF STOCHASTIC TEMPERATURE FIELDS IN STF-ELECTRON MOD TECHNICAL SYSTEMS

P.I. Kandalov¹, Research Associate, petrki87@gmail.ru

¹ Center of Visualization and Satellite Information Technologies SRISA, Nakhimovsky Ave. 36/1, Moscow, 117218, Russian Federation

Abstract. The article presents the software package “STF-ElectronMod” to simulate a steady-state stochastic 3D temperature fields in technical systems. Experience has shown that the actual temperature fields of technical systems have uncertainty interval nature. This is due to the interval factors determining the thermal regime of the technical system. These factors include: the design parameters of technical systems with statistical technological manufacture spread; factors arising in the operation of technical systems (power consumption, heat flows, internal environment parameters); environmental factors (temperature, medium, refrigerant, flows velocity, etc.). However, now modeling of temperature fields of technical systems is being conducted under the assumption of determinacy. This means that all thermal condition parameters are considered accurately known. This article describes a structure of the designed software system STF-ElectronMod for modelling of interval stochastic temperature fields in technical systems under uncertainty of input determining data. The method and algorithm are based on the author's developments according to a matrix-topological method, the software complex three-dimensional deterministic modeling of temperature fields of technical systems, as well as Monte-Carlo method. The application of the method and algorithm for modeling interval stochastic temperature fields is considered on the example of a complex electronic system.

Keywords: software, mathematical modeling, temperature fields, stochastic, interval.

References

1. Isaev I.P., Inkov Yu.M., Marichev M.A. *Veroyatnostnye metody rascheta poluprovodnikovyykh preobrazovateley* [Probabilistic Methods of Calculating Semiconductor Converters]. Moscow, Energoizdat Publ., 1983, 96 p.
2. Chekanov A.N. *Raschety i obespechenie nadezhnosti elektronnoy apparatury* [Calculating and Reliability Control of Electronic Equipment]. Study guide. Moscow, KNORUS Publ., 2012, 440 p.
3. Keller C.J., Antonetti V.W. Statistical thermal design for computer electronics. *Electronic Packaging and Production*. 1979, vol. 19, no. 3, pp. 55–62.
4. Madera A.G., Kandalov P.I. The analysis of interval stochastic temperature fields of technical systems. *Programmnye produkty i sistemy* [Software & Systems]. 2014, no. 4, pp. 41–45 (in Russ.).
5. Kandalov P.I., Madera A.G. Modeling temperature fields of technical systems under interval stochastic uncertainty. *6 Ross. nats. konf. po teploobmenu* [Proc. 6th Russian National Conf. on Heat Exchange]. Moscow, MEI Publ., 2006, vol. 1, pp. 191–192 (in Russ.).
6. Reshetnikov V.N. *Kosmicheskie telekommunikatsii. Sistemy sputnikovoy svyazi i navigatsii* [Cosmic Telecommunications. Satellite Communication and Navigation Systems]. St. Petersburg, Leningradskoe izdatelstvo Publ., 2010, 132 p.
7. Kandalov P.I., Madera A.G. *Simulation of Temperature Fields of Electronic Modules 2014 (STF-ElectronMod 2014)*. Software State Registration Certificate no. 2015613488. 2015.
8. Kandalov P.I., Madera A.G. Modeling temperature fields in multi-layer structures. *Programmnye produkty i sistemy* [Software & Systems]. 2008, no. 4, pp. 46–49 (in Russ.).
9. Madera A.G., Kandalov P.I. Matrix-topological method for mathematical and computer modeling of temperature fields in electronic modules: software complex STF-ElectronMod. *Programmnye produkty i sistemy* [Software & Systems]. 2012, no. 4, pp. 160–164 (in Russ.).
10. Madera A.G. *Modelirovanie teploobmena v tekhnicheskikh sistemakh* [Heat Exchange Modeling in Engineering Systems]. Moscow, Pervaya issled. lab. im. ak. V.A. Melnikova Publ., 2005, 208 p.

УДК 004.42; 519.85

DOI: 10.15827/0236-235X.116.176-182

Дата подачи статьи: 11.01.16

2016. Т. 29. № 4. С. 176–182

ПОСТРОЕНИЕ АРХИТЕКТУРЫ САПР ОДНОШНЕКОВЫХ ЭКСТРУДЕРОВ С ПРИМЕНЕНИЕМ ЭЛЕМЕНТОВ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

Т.М. Зубкова, д.т.н., профессор, bars87@mail.ru;

Н.А. Мустюков, аспирант, 256nail@mail.ru;

М.А. Токарева, к.т.н., доцент, tokareva@mail.osu.ru

(Оренбургский государственный университет, просп. Победы, 13, г. Оренбург, 460018, Россия)

Конкуренция на рынке экструзионного оборудования заставляет его производителей ежегодно предлагать усовершенствованные и новые технологии, рассчитанные на широкий круг выпускаемой продукции. Поэтому производство должно обладать гибкостью, перенастраиваясь на различные конфигурации изделий, типы сырья и производительность в зависимости от текущих потребностей рынка. Сложность процессов, которыми характеризуется экструзия, не позволяет рассчитать их обычными методами без использования САПР. Принятие решения о выборе маршрута проектирования требует высокой профессиональной подготовки пользователя. Выходом из данной ситуации является интеллектуализация проектирования. Указанный подход позволяет осуществлять автоматизацию процедур оценки свойств проекта, планирование маршрута проектирования и диалог с пользователем САПР. В силу данных причин все большее развитие получают интеллектуальные САПР с перестраиваемой архитектурой. Таким образом, снижение трудоемкости процесса проектирования шнековых экструдеров на основе создания перестраиваемой архитектуры САПР является актуальной проблемой.

Авторами данной статьи разработана методика построения интеллектуальной САПР шнековых экструдеров, основанная на интеграции и конфигурировании его подсистем. Создана программная система, которая является ядром САПР и включает в себя БЗ, БД, набор программных компонентов, интеграцию компонентов (сторонние CAD/CAE-системы, математические модели проектируемых процессов, компоненты пользовательского интерфейса, инструментальные средства разработки), управление данными и оптимизацию технологических и геометрических параметров шнековых экструдеров.

Описано построение архитектуры САПР на основе теории искусственного интеллекта, которое заключается в выборе набора компонентов для решения проектных задач таким образом, чтобы обеспечить необходимые свойства в системе при минимизации времени проектирования.

Разработанная интегрированная среда САПР шнековых экструдеров позволяет создавать конструкции экструдеров, автоматизировать коррекцию геометрических параметров и подготовку данных для проведения анализа конструкции, моделировать процесс экструдирования, оптимизировать конструкции и технологические режимы. Предложенная методика конфигурирования САПР на основе методов искусственного интеллекта позволяет снизить трудоемкость проектирования шнековых экструдеров.

Ключевые слова: экструзионный процесс, математическое моделирование, интегрированная среда САПР, CAD/CAE-системы, шнек, матрица, векторная оптимизация, программная система, вычислительный эксперимент.

Широкое использование экструдеров в различных областях производства объясняется стабильностью их работы, обеспечением высокого качества продукции, совмещением нескольких технологических операций, возможностью компоновки в поточные линии с высокой степенью автоматизации.

Сложность процессов, которыми характеризуется экструзия, не позволяет рассчитать их обычными методами без использования САПР, поэтому задача разработки алгоритмов построения и реконфигурирования архитектуры комплексной САПР шнековых экструдеров является актуальной. Необходимость формирования перестраиваемых архитектур САПР объясняется ростом элементной базы, специализацией программных средств проектирования экструдеров, быстрой сменой технологических и конструктивных требований к проектируемым объектам, внедрением CALS-технологий и сложностью создания прикладного ПО.

Построением САПР, математическим моделированием, разработкой ПО занимались многие исследователи [1–9]. На основе их трудов делается

вывод о существовании нескольких направлений построения САПР с перестраиваемой архитектурой. Среди них выделена априорная адаптация архитектуры САПР к особенностям объекта проектирования (рис. 1).

Согласно данной концепции, адаптация заключается в использовании наиболее подходящего программного компонента САПР, при котором обеспечивается оптимальное проектирование объекта с учетом изначально неизвестных и меняющихся условий работы. Комбинирование и отбор проектирующих модулей осуществляются с помо-

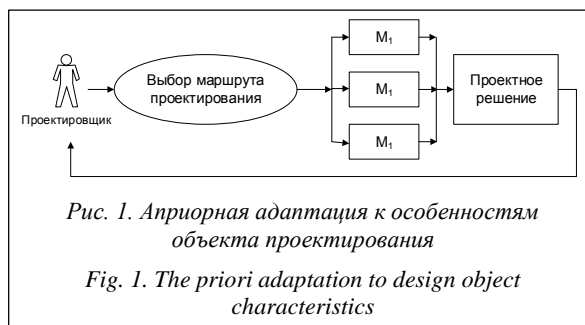


Рис. 1. Априорная адаптация к особенностям объекта проектирования

Fig. 1. The priori adaptation to design object characteristics

щью выявления общих свойств объекта проектирования, разделения компонентов на отдельные классы, варьирования критериев оценки качества проекта. Указанные действия реализуются отдельным программным модулем – модулем управления (принятия решения).

Математическое описание построения САПР

Задача построения архитектуры САПР на основе теории искусственного интеллекта заключается в выборе набора компонентов R для решения проектных задач Z таким образом, чтобы обеспечить необходимые свойства F в системе S при минимизации времени проектирования t .

Целевая функция конфигурирования системы [10] в случае применения аддитивного критерия

примет вид:
$$\sum_{j=1}^J \left(c_j \frac{Q_j(F_j)}{Q_j^0} - v_j \frac{t_j(F_j)}{t_j^0} \right) \rightarrow \max,$$
 где

F_j – свойство альтернативной подсистемы для решения j -й задачи проектирования (управляемый параметр), при этом $F_j \in \bigcup_{p=1}^{P_{A_j}} \{F_{jp}\}$; Q_j – функция

оценки качества системных свойств для j -й задачи проектирования; t_j – функция оценки времени выполнения проектной задачи; P_{A_j} – набор альтернативных компонентов для решения j -й задачи; Q_j^0, t_j^0 – j -й нормирующий делитель для качественных и временных характеристик соответственно; J – количество проектных задач; c_j, v_j – весовые коэффициенты j -го частного критерия, причем $\frac{c_j}{v_j} \rightarrow \infty$.

При этом управляемыми параметрами являются свойства системы (F) и совокупность компонентов (R), обеспечивающих данные свойства:

$$U = \begin{cases} F, \\ R. \end{cases} \quad (1)$$

В качестве ограничений выступают следующие:

- проектная задача: синтез (Z_S), анализ (Z_A), оценка (Z_E), конвертирование (Z_C), визуализация (Z_V), принятие проектных решений (Z_D);
- конструктивные особенности изделия: количество секций (K_I), наличие компрессионных затворов (K_Z), количество шнеков (K_S), форма фильер матрицы (K_F), количество фильер (K_{FI}), наличие оттока жидкой фазы (K_O), наличие пристенного слоя (K_P), форма канала шнека (K_{FS});
- технологические требования производства: свойства обрабатываемого материала (коэффициент консистенции в винтовом канале шнека (T_μ), индекс течения прессуемого материала (T_n)), напряженное состояние в различных участках экструдера (T_σ), температура обработки (T_i);

– набор начальных данных (возможность применения компонента): конструктивные (U_K), геометрические (U_G), кинематические (U_P) и реологические (U_R) параметры.

Система ограничений примет вид:

$$C = \begin{cases} Z, \text{ где } Z \in \{Z_S, Z_A, Z_E, Z_C, Z_V, Z_D\}, \\ K, \text{ где } K \subseteq \{K_I, K_Z, K_S, K_F, K_{FI}, \\ K_O, K_P, K_{FS}\}, \\ T, \text{ где } T \subseteq \{T_n, T_\mu, T_\sigma, T_i\}, \\ U, \text{ где } U \subseteq \{U_K, U_G, U_P, U_R\}. \end{cases} \quad (2)$$

Решить представленную оптимизационную задачу с помощью теории графов невозможно, поскольку нет полной информации о связях между компонентами по воздействию, по управлению, по информации, по размещению.

Принятие решения в большинстве случаев заключается в генерации возможных альтернативных решений, их оценке и выборе лучшего варианта. При выборе варианта приходится учитывать большое число неопределенных и противоречивых факторов. Неопределенность – неотъемлемая часть процессов принятия решений.

Выходом являются системы, основанные на мягких вычислениях, которые используют вероятностные вычисления, нечеткую логику, обучение и адаптацию, оптимизацию на основе случайного поиска и эволюции (генетические вычисления).

Выделяют несколько моделей нечеткого вывода (модель Мамдани, Ларсена, Сугено и др.), которые различаются видом логических операций и используемых правил.

Метод Мамдани, использующий минимаксную композицию нечетких множеств, является наиболее распространенным способом логического вывода. Применительно к задаче реконфигурирования математической модели одношнековых экструдеров данный метод включает следующие действия.

1. Формирование базы правил следующего вида: если <условие 1> и <условие 2> ... и <условие n >, то <вывод>.

В качестве условий указывается соответствие входных параметров X_i ($i \in [1, ..., n]$) предъявляемым требованиям (например, количество секций больше 2).

На основе входных параметров, а также по оценочному мнению <условие k > принимает значение в интервале $[0, ..., 1]$.

«Вывод» соответствует выбору использования того компонента, для которого составлено правило (например, использовать математическую модель экструдера для биополимера).

2. Фаззификация входных переменных. Этот этап часто называют приведением к нечеткости.

На вход поступают сформированная база правил и массив входных данных $A = \{a_1, ..., a_m\}$, где m – количество входных переменных. В этом мас-

сиве содержится информация о конструктивных особенностях экструдера и технологических требованиях производства. Целью этапа является получение значений истинности для всех подусловий из базы правил. Это происходит следующим образом: для каждого из подусловий находится значение $b_i = \lambda_i(a_j)$, где λ – функция принадлежности, которая всем значениям входных переменных ставит в соответствие конкретные значения степени истинности; $j = 1, \dots, m$; $i = 1, \dots, k$, где k – общее число подусловий в базе правил. Таким образом, получается множество значений b_i .

3. Агрегирование подусловий – определение степени истинности условий в каждом правиле: $c_i = \min\{b_i\}$.

4. Активизация подзаключений – переход от условий к подзаключениям. Для каждого подзаключения находится степень истинности $d_i = c_i F_i$, где $i = 1, \dots, q$, q – общее число подзаключений в базе правил; F – весовые коэффициенты, означающие степень уверенности в истинности получаемого подзаключения. Затем каждому i -му подзаключению сопоставляется множество D_i с новой функцией принадлежности. Ее значение определяется как минимум из d_i и значения функции принадлежности компонента из подзаключения. Этот метод называется min-активизацией и формально записывается так: $\lambda'(x) = \min\{d_i, \lambda_i(x)\}$.

5. Аккумуляция заключений – получение нечеткого множества (или их объединения) для каждой из выходных переменных. Выполняется следующим образом: i -й выходной переменной сопоставляется объединение множеств $E_i = \cup D_j$, где j – номера подзаключений, в которых участвует i -я выходная переменная ($i = 1, \dots, s$). Объединением двух нечетких множеств является третье нечеткое множество со следующей функцией принадлежности: $\lambda'(x) = \max\{\lambda_1(x), \lambda_2(x)\}$, где $\lambda_1(x)$, $\lambda_2(x)$ – функции принадлежности объединяемых множеств.

6. Дефазификация выходных переменных. На данном этапе определяется количественное значение (crisp value) для каждой из выходных лингвистических переменных. Рассматриваются i -я выходная переменная и относящееся к ней множество E_i ($i = 1, \dots, s$). Затем при помощи метода дефазификации находится итоговое количественное значение выходной переменной. В данной реализации алгоритма используется метод центра тяжести, в котором значение i -й выходной переменной рассчитывается по формуле

$$y_i = \frac{\int_{\min}^{\max} x \cdot \lambda_i(x) dx}{\int_{\min}^{\max} \lambda_i(x) dx}, \quad (3)$$

где $\lambda_i(x)$ – функция принадлежности соответствующего нечеткого множества E_i ; \min и \max – границы

универсума нечетких переменных; y_i – результат дефазификации.

Рассмотрим реализацию данного алгоритма на примере выбора математической модели процесса экструдирования согласно заданным начальным условиям. Для проведения эксперимента необходимо составить БЗ. В данном примере в качестве правил возьмем соответствие свойств обрабатываемого материала и используемых математических моделей.

Пример заполнения БЗ приведен в таблице 1. Показателями, по которым будет производиться отбор, являются температура материала (t , °C), индекс течения прессуемого материала (n), коэффициент консистенции (μ').

Таблица 1

БЗ для выбора математической модели

Table 1

Knowledge base to select a mathematical model

Условие 1	Условие 2	Условие 3	Вывод
$t = 40$	$n = 0,18$	$\mu' = 0,052$	Используется математическая модель для биополимеров
$t = 40$	$n = 0,28$	$\mu' = 0,0343$	Используется математическая модель для сырья с оттоком жидкой фазы

Начальными данными являются следующие свойства обрабатываемого материала: $t = 40$ °C, $n = 0,21$, $\mu' = 0,049$.

Найдем степень истинности начальных данных каждому правилу из БЗ и реализуем метод Мамдани. Результаты приведены в таблице 2.

Таблица 2

Результаты алгоритма

Table 2

The algorithm results

Условие 1	Условие 2	Условие 3	min
1	0,83	0,94	0,83
1	0,75	0,57	0,57
max			0,83

На основании полученных результатов можно сделать вывод, что математическая модель для биополимеров предпочтительна при проведении эксперимента в соответствии с начальными условиями. Преимуществом метода является возможность учета безграничного числа условий и составления правил различных форм. Точность результатов зависит от объема БЗ.

Таким образом, по результату этапа дефазификации можно судить о необходимости использования определенного программного компонента из имеющихся альтернатив.

На основе описанного алгоритма разработана методика управления конфигурацией САПР (рис. 2). В результате ее реализации определяется набор интегрированных программных подсистем,

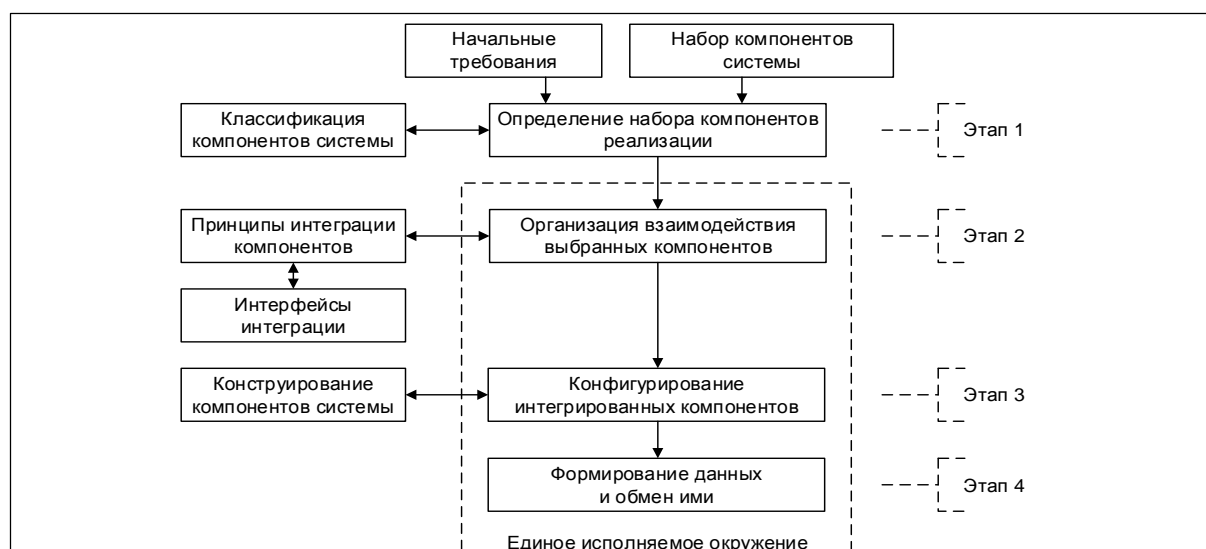


Рис. 2. Схема функционирования САПР

Fig. 2. CAD operation scheme

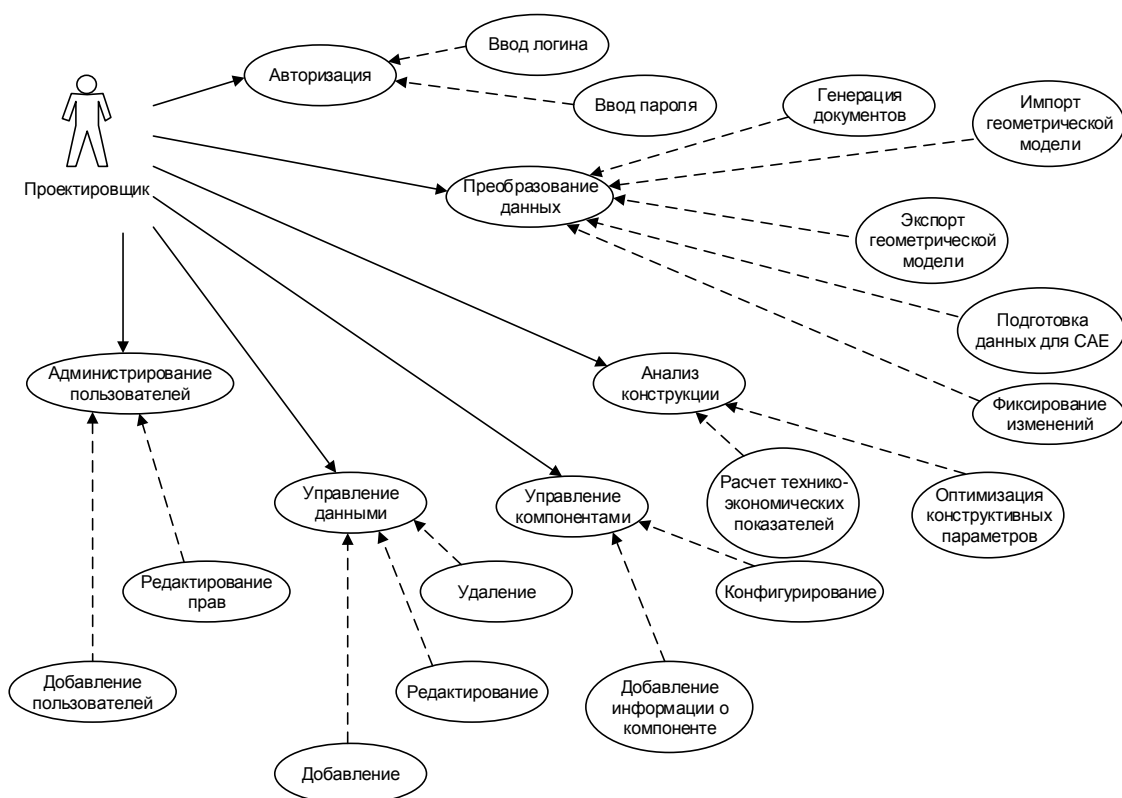


Рис. 3. Диаграмма вариантов использования

Fig. 3. A use case diagram

с помощью которых решается поставленная задача проектирования [11].

Реализация построения САПР

Диаграмма вариантов использования САПР показана на рисунке 3.

Работа с программной системой осуществляется следующим образом.

При добавлении новой конструкции создается ее проект.

Параметры конструкции вносятся через форму ввода либо определяются автоматически при загрузке чертежа конструкции (рис. 4).

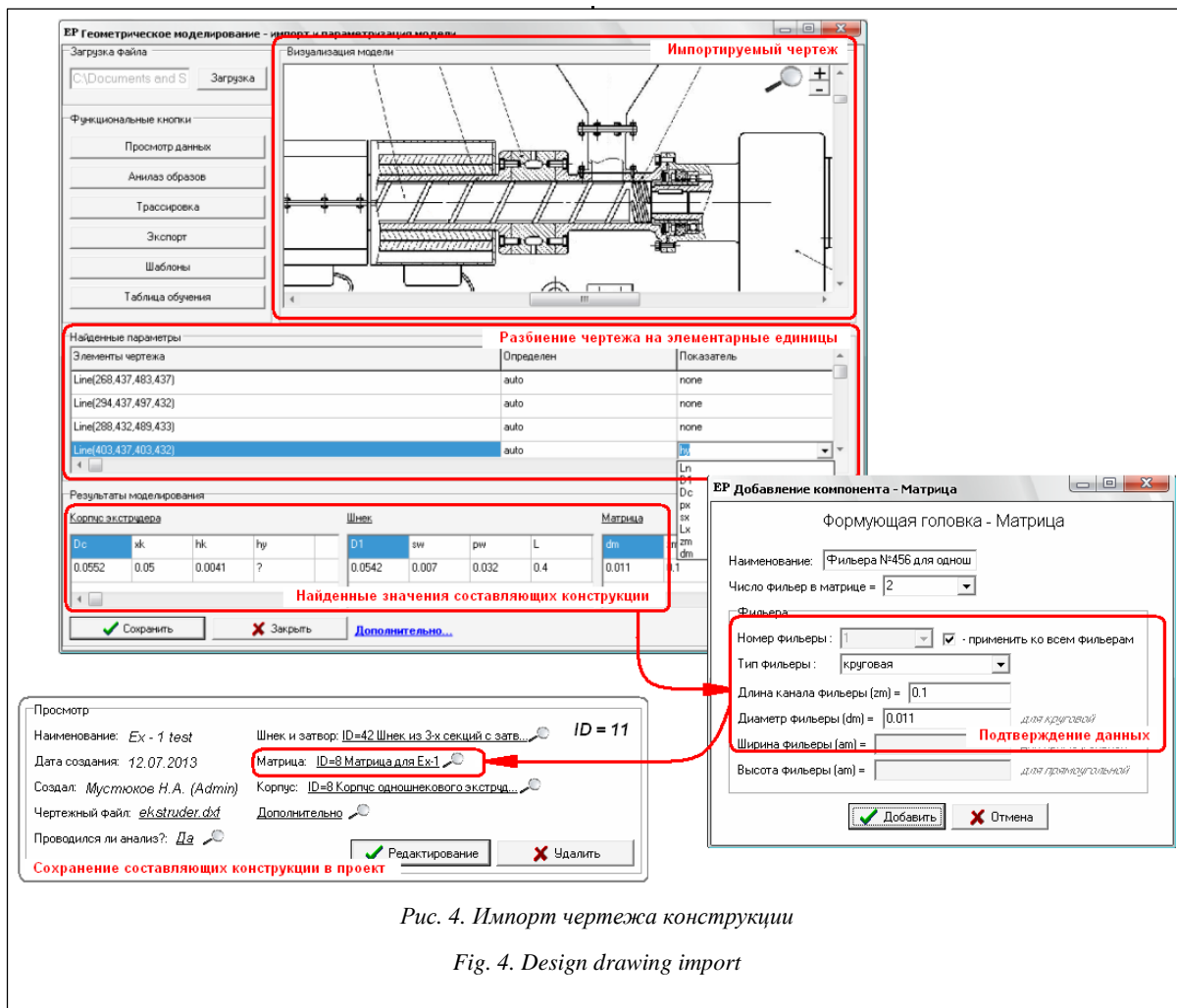


Рис. 4. Импорт чертежа конструкции

Fig. 4. Design drawing import

После создания проекта конструкции в программной системе доступны следующие функции: редактирование конструкции, анализ конструкции с помощью математической модели расчета технико-экономических показателей процесса (производительности, мощности сил полезного сопротивления, времени процесса, эффективности процесса), оптимизация конструкции, подготовка данных для применения их во внешних системах.

Анализ процесса экструдирования реализован в три этапа: выбор конструкции, задание параметров процесса экструдирования, моделирование (см. рис. 5). Пользователю предоставляется возможность выбора одного или нескольких вычисляемых параметров, при этом подключение и использование математических моделей осуществляются автоматически на основе БЗ и описанного алгоритма конфигурирования.

В приведенном примере требуется определить ряд технико-экономических показателей. По результатам конфигурирования для расчетов был выбран модуль «Модель расчета технико-экономических показателей экструдера для биополимеров». Отследить этапы расчета возможно в блоке «Трассировка и детализация».

Для оптимизации выбранной конструкции необходимо указать условия оптимизации (целевой параметр, ограничения), корректируемые характеристики и критерий останковки алгоритма. Результатом расчета являются значения параметров оптимизации и технико-экономические характеристики полученной конструкции. На рисунке 6 представлена оптимизация параметров шнека конструкции при условии максимизации коэффициента полезного действия.

В заключение отметим достоинства предлагаемой методики, основанной на конфигурировании интеллектуальной САПР шнековых экструдеров.

Благодаря увеличению числа просматриваемых вариантов и детальности проработки каждого из них повышается качество проектируемых объектов. За счет более широкого использования средств имитационного моделирования и численных методов уменьшается время проектирования. Повышается эффективность инженерного анализа. Благодаря планированию вычислений и контролю результатов ускоряются проектные расчеты.

Кроме того, снижается трудоемкость процесса проектирования шнековых экструдеров в среднем на 24 % и уменьшается срок внедрения САПР.

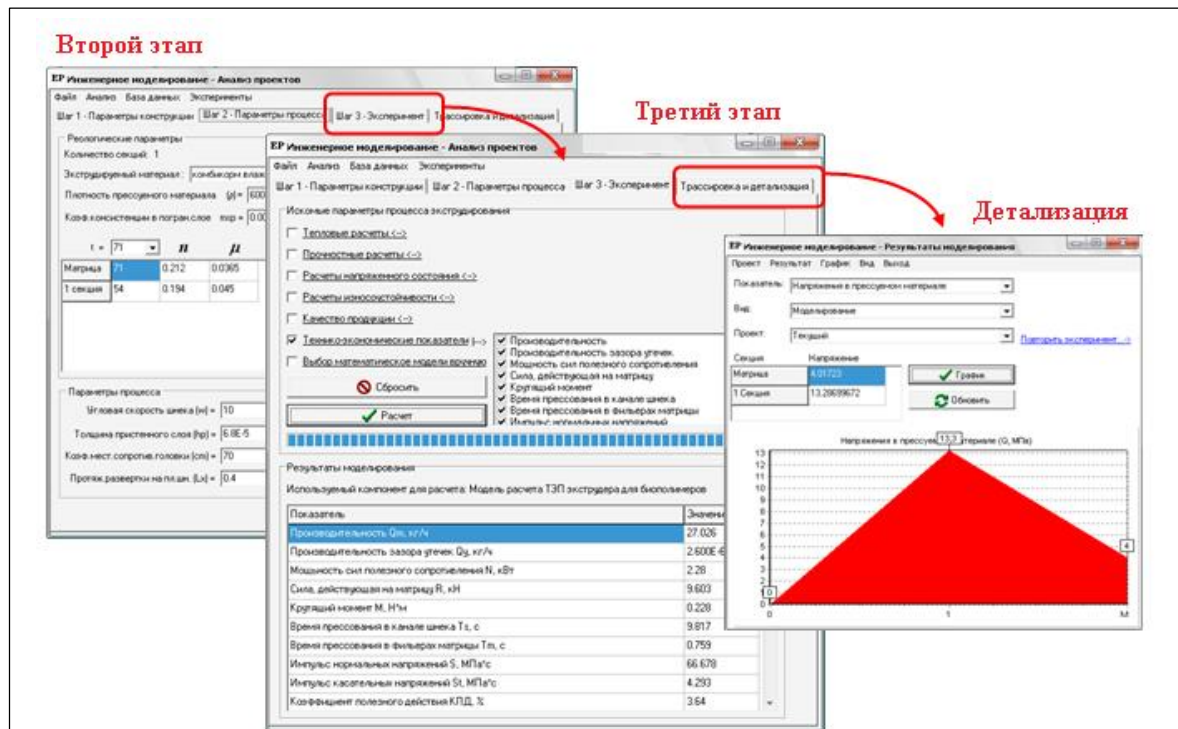


Рис. 5. Анализ процесса экструдирования

Fig. 5. Extrusion process analysis

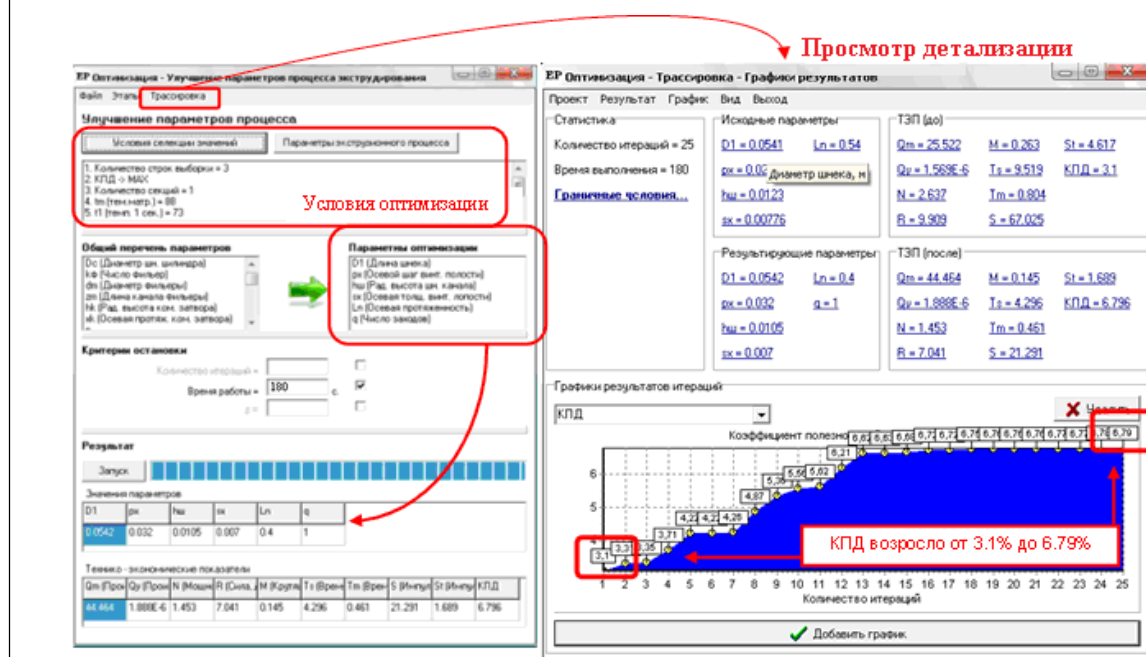


Рис. 6. Оптимизация конструкции

Fig. 6. Design optimization

Литература

1. Шевченко М.Н. Математическое обеспечение САПР шнековых экструдеров // Вестн. Оренбургского гос. ун-та. 2010. № 5 (111). С. 145–146.
2. Кузнецов С.А. Разработка специализированных САПР – главное направление при создании высокоэффективных про-

граммных продуктов // Автоматизация. Современные технологии. 2011. № 10. С. 29–34.

3. Остриков А.Н., Абрамов О.В. Разработка САПР шнековых рабочих органов экструдеров // Вестн. машиностроения. 2009. № 10. С. 27–30.
4. Пищухин А.М., Шевченко М.Н. Архитектура системы автоматизации проектирования шнековых экструдеров // Вестн.

Самарского гос. технич. ун-та. Сер.: Технич. науки. 2010. № 2. С. 128–134.

5. Барабанов В.В. Роль интегрированных информационных систем управления производством в решении проблемы повышения качества и конкурентоспособности продукции промышленных предприятий // Информационные технологии в проектировании и производстве. 2000. № 4. С. 3–8.

6. Волкова Г.Д. Концептуальное моделирование при создании систем автоматизации проектирования // Техника машиностроения. 2000. № 2. С. 92–102.

7. Гаврилова Т.А., Хорошевский В.Ф. Базы знаний интеллектуальных систем. СПб: Питер, 2001. 384 с.

8. Кравченко Ю.А. Перспективы развития гибридных ин-

теллектуальных систем // Перспективные информационные технологии и интеллектуальные системы. 2002. № 3. С. 34–38.

9. Курейчик В.В. Перспективные архитектуры генетического поиска // Перспективные информационные технологии и интеллектуальные системы. 2000. № 1. С. 58–60.

10. Зубкова Т.М., Мустюков Н.А., Колобов А.Н. Реконфигурирование САПР для проектирования одношнековых экструдеров на основе модели нечеткого вывода Мамдани // Вестн. ОГУ. 2013. № 1. С. 176–181.

11. Мустюков Н.А., Зубкова Т.М. Применение генетического алгоритма для проведения параметрического синтеза конструкции экструдера // Науч.-технич. вестн. ИТМО. 2013. № 4. С. 114–118.

Software & Systems

DOI: 10.15827/0236-235X.116.176-182

Received 11.01.16

2016, vol. 29, no. 4, pp. 176–182

SINGLE SCREW EXTRUDER ENGINEERING WITH ARTIFICIAL INTELLIGENCE ELEMENTS

T.M. Zubkova¹, Dr.Sc. (Engineering), Professor, bars87@mail.ru

N.A. Mustyukov¹, Postgraduate Student, 256nail@mail.ru

M.A. Tokareva¹, Ph.D. (Engineering), Associate Professor, tokareva@mail.osu.ru

¹ Orenburg State University, Pobedy Ave. 13, Orenburg, 460018, Russian Federation

Abstract. Market fierce competition makes the extrusion equipment manufacturers offer advanced and new technologies for a wide range of products every year. Therefore, production must be flexible, adjusting to different configurations of products, raw materials and types of productivity depending on current needs of the market. The complexity of the extrusion process does not allow figuring it out by conventional methods without using automation design (SAPR). The reduction of engineering process complexity of single screw extruders based on flexible SAPR organization is a current issue.

The study developed a method of building SAPR screw extruders based on the integration and configuration of its subsystems. The created software system is the SAPR core and includes the integration of the components (a third-party CAD/CAE-systems, designed processes mathematical models, user interface components, engineering tools), data management, native software modules connection, designed to analyze the process of extrusion and optimization of technological and geometrical parameters of screw extruders.

SAPR framework for screw extruders allows engineering the extruders; automating the adjustment of geometrical parameters and data preparation for analysing the structure; to carry out a simulation extrusion process; to optimize design and technological regimes; to manage extruder design data, extrusion process and simulation results. The developed software system helped to create a new press extruder and improve the homogenization of the material being processed. The constructions are protected by patents of the Russian Federation. The proposed method of geometrical parameters optimization of the extruder-based adaptive methods enhances the search for new and improvement of existing designs. The article reflects the results of the grant № 14-08-97031 performed with the financial support of the Government of the Orenburg Region and Russian Fundamental Research Fund.

Keywords: extrusion process, mathematical modeling, integrated environment SAPR, CAD/CAE-systems, screw, matrix, vector optimization, software system, computer experiment.

References

1. Shevchenko M.N. SAPR Screw extruders software. *Vestn. Orenburgskogo gos. univ.* [The Herald of the Orenburg State Univ.]. 2010, no. 5 (111), pp. 145–146 (in Russ.).
2. Kuznetsov S.A. Specialized SAPR engineering is the high performance software mainstream. *Avtomatizatsiya. Sovremennye tekhnologii* [Automation. Modern technologies]. 2011, no. 10, pp. 29–34 (in Russ.).
3. Ostrikov A.N., Abramov O.V. Extruder working body SAPR engineering. *Vestn. mashinostroeniya* [Engineering Herald]. 2009, no. 10, pp. 27–30 (in Russ.).
4. Pishchukhin A.M., Shevchenko M.N. Screw extruders system architecture design engineering. *Vestn. Samarskogo gos. tekhnich. un-ta. Ser.: Tekhnich. nauki* [The Herald Samara State Technical Univ. Series: Technical sciences]. 2010, no. 2, pp. 128–134 (in Russ.).
5. Barabanov V.V. The role of integrated information systems for production management in addressing the problem of increasing the quality and competitiveness of industrial enterprises. *Informatsionnye tekhnologii v proektirovanii i proizvodstve* [Information Technologies in Engineering and Manufacture]. 2000, no. 4, pp. 3–8 (in Russ.).
6. Volkova G.D. Conceptual modeling while automation systems engineering. *Tekhnika mashinostroeniya* [Mechanical Engineering]. Moscow, SPE Virazh-centre Publ., 2000, no. 2, pp. 92–102 (in Russ.).
7. Gavrilova T.A., Khoroshevsky V.F. *Bazy znany intellektualnykh sistem* [Intellectual Systems Databases]. St. Petersburg, Piter Publ., 2001, 384 p.
8. Kravchenko Yu.A. Hybrid intelligent systems development prospects. *Perspektivnye informatsionnye tekhnologii i intellektualnye sistemy* [Advanced Information Technologies and Intelligent Systems]. 2002, no. 3, pp. 34–38 (in Russ.).
9. Kureychik V.V. Genetic search prospective architecture. *Perspektivnye informatsionnye tekhnologii i intellektualnye sistemy* [Advanced Information Technologies and Intelligent Systems]. 2000, no. 1, pp. 58–60 (in Russ.).
10. Zubkova T.M., Mustyukov N.A., Kolobov A.N. SAPR reconfiguring for single-screw extruders engi-neering, Mamdani model-basing fuzzy inference. *Vestn. OGU* [OSU Herald]. 2013, no. 1, pp. 176–181 (in Russ.).
11. Mustyukov N.A., Zubkova T.M. Application of genetic algorithms for parametric synthesis design extruder. *Nauch.-tekhnich. vestn. ITMO* [Scientific and Technical Journal of Information Technologies, Mechanics and Optics]. 2013, no. 4, pp. 114–118 (in Russ.).

УДК 004.9, 519.65

DOI: 10.15827/0236-235X.116.183-188

Дата подачи статьи: 11.07.16

2016. Т. 29. № 4. С. 183–188

СИСТЕМЫ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ В УПРАВЛЕНИИ ВУЗОМ: ВЫЧИСЛИТЕЛЬНЫЕ МОДЕЛИ ДИФФЕРЕНЦИАЦИИ УЧЕБНОЙ НАГРУЗКИ

*Е.Ю. Шахова, к.т.н., доцент, koroleva0602@mail.ru
(Братский государственный университет,
ул. Макаренко, 40, г. Братск, Иркутская область, 665709, Россия)*

В настоящее время во многих вузах идет переработка локальных нормативных актов, регламентирующих труд педагогических работников, отнесенных к профессорско-преподавательскому составу. Требование дифференциации верхних пределов учебной нагрузки по должностям профессорско-преподавательского состава является обязательным. Построены и проанализированы линейные и нелинейные модели распределения нормативной учебной нагрузки. Применяемые методы: аппроксимация по методу наименьших квадратов линейной функцией и кубическим полиномом, сплайн-интерполяция. Моделирование проводилось с помощью системы MathCAD. Определены наиболее эффективные модели при различной структуре численности профессорско-преподавательского состава. Критерий эффективности – наименьший объем сверхнормативной учебной нагрузки.

Оценка эффективности построенных линейных и нелинейных моделей проводилась на основе трех вариантов структуры численности профессорско-преподавательского состава структурного подразделения вуза (института) с высоким, средним и пороговым уровнями профессоров и доцентов в общей численности.

Полученные результаты позволяют сделать вывод, что линейная дифференциация верхних пределов учебной нагрузки не всегда является наилучшим решением. Линейные модели в данном исследовании оказались наиболее эффективными только для модели института с высоким уровнем профессоров и доцентов в общей численности профессорско-преподавательского состава; для моделей численности со средним и пороговыми уровнями наиболее эффективной оказалась нелинейная модель, построенная по методу сплайн-интерполяции (кривая – кубический сплайн с параболическими конечными точками). Для всех рассмотренных моделей дифференциации при структуре численности института с высоким уровнем профессоров и доцентов в общей численности профессорско-преподавательского состава минимальная сверхнормативная нагрузка превышает максимальную сверхнормативную нагрузку для институтов с пороговым и средним уровнями.

Ключевые слова: моделирование, линейная модель, нелинейная модель, MathCAD, норма учебной нагрузки.

В настоящее время во многих вузах осуществляется переработка локальных нормативных актов, касающихся нормирования рабочего времени преподавателей. Это связано как с появлением новых нормативных документов, так и с изменением функций преподавателя в условиях реформирования высшей школы [1].

Одним из острых является вопрос дифференциации учебной нагрузки – нормируемой части рабочего времени педагогических работников вуза, замещающих должности профессорско-преподавательского состава (ППС). Рассматриваемая задача, хотя и является локальной, относится к нескольким областям исследований: экономика и организация труда, правовое регулирование труда, управление в организационных системах, в том числе принятие управленческих решений, и к достаточно новой области – применение математических моделей в организационных системах, активно разрабатываемых как отечественными, так и зарубежными исследователями.

Как отмечают авторы работы [2], компьютерное моделирование в организационных системах, которое является мощной методологией для продвижения теории и исследования поведения сложных систем, более медленно развивалось в управлении, чем в некоторых других, связанных с ним областях социальных наук. Вместе с тем математические модели организационных систем довольно разнообраз-

ны [3]: это и вычислительные модели (эмуляция, экспертные системы, численный анализ), и математические (формальная логика, алгебра матриц, анализ сети, дискретные и непрерывные уравнения). Методологии моделирования базируются на теории искусственного интеллекта и теории сложных систем, в том числе на применении методологии исследования активных систем (многоагентных систем), теории расписаний, теории игр, методологий процессного подхода [4–9] и т.д.

Проблема управления в сфере высшего образования, в частности, оценка показателей эффективности работы преподавателей, актуальна и для зарубежных систем высшего образования [10–12]. В настоящей статье представлены результаты построения в среде MathCAD и исследования достаточно простых и наглядных вычислительных моделей распределения норм рабочего времени. Данный подход – моделирование и исследование моделей социальных (экономических) систем средствами системы MathCAD – описан в [13], где отмечается, что документы MathCAD, отображающие построенные модели и реализующие алгоритмы оптимизации, являются наглядными, хорошо читаемыми специалистами различных областей, живыми – позволяющими вносить изменения как в саму модель, так и в метод расчета, а также оперативно анализировать результаты.

В рамках проводимого исследования будут рассматриваться должности педагогических работников, отнесенных к ППС в соответствии с Постановлением Правительства РФ от 08.08.2013 № 678 (табл. 1).

**Перечень должностей ППС
и уровней квалификации**

Таблица 1

Table 1

The list of faculty positions (PPP) and skill levels

№	Должность	Уровень квалификации	Подуровень квалификации
1	Директор института	Не указан	Не указан
2	Декан факультета	Не указан	Не указан
3	Заведующий кафедрой	Не указан	Не указан
4	Профессор	8	7.3–8.3
5	Доцент	8	7.2–8.1
6	Старший преподаватель	7	6.2–7.1
7	Преподаватель		6.2–7.1
8	Ассистент		6.2

В профессиональном стандарте педагога указаны уровень и подуровень квалификации по должностям ППС (см. табл. 1).

Определить наиболее эффективные модели. Критерий эффективности – наименьший объем (количество часов) сверхнормативной учебной нагрузки.

Исходные данные:

- количество должностей ППС – 8 (табл. 1);
- максимальный верхний предел учебной нагрузки – 900 часов;
- минимальный предел учебной нагрузки – 450 часов.

Результаты моделирования

Линейная дифференциация. В таблице 2 приведены характеристики построенных моделей. Метод: линейная регрессия – аппроксимация линейной функцией по методу наименьших квадратов. Обработка результатов: округление до ближайшего значения, кратного 10 (функция Round).

Модель 1.1 предполагает обычную линейную дифференциацию в соответствии с занимаемой должностью. Модель 1.2 имеет две промежуточные узловые точки – желаемые промежуточные значения для должностей профессора и старшего преподавателя. Модель 1.3. имеет *ступеньку* – уравнивание нагрузки по должностям заведующего кафедрой и профессора, а также шаг в 100 ча-

Таблица 2

Линейная дифференциация. Характеристики моделей

Table 2

Linear differentiation. Model specifications

№ модели	Учебная нагрузка		Количество точек	Исходные константы для моделирования	Фиксированная должностная нормативная нагрузка (узловые точки)
	мин.	макс.			
1.1	450	900	2	D1=450 D8=900	Директор института (D1) Ассистент (D8)
1.2	450	900	4	D1=450 D4=600 D6=750 D8=900	Директор института (D1) Профессор (D4) Старший преподаватель (D6) Ассистент (D8)
1.3	450	900	6	D1=450 D3=600 D4=600 D5=700 D6=800 D8=900	Директор института (D1) Заведующий кафедрой (D3) Профессор (D4) Доцент (D5) Старший преподаватель (D6) Ассистент (D8)

Согласно Приказу Минобрнауки РФ от 22.12.2014 № 1601, требование дифференциации верхних пределов учебной нагрузки (нормативной нагрузки) в зависимости от занимаемой должности и уровня квалификации преподавателей является обязательным.

Постановка задачи

Построить и проанализировать модели дифференциации учебной нагрузки ППС: а) линейные, б) нелинейные.

сов между должностями профессор–доцент–старший преподаватель. Приведем документ MathCAD для расчета модели 1.3, где вектор DD – нормативная нагрузка, дифференцированная по должностям, рассчитанная с фиксированным шагом, вектор RDD – результат округления, шаг дифференциации – 60–70 часов:

ORIGIN := 1 Модель 1.3

minn := 450 maxn := 900
 DX₁ := 1 DY₁ := minn
 DX₂ := 3 DY₂ := 600 зав. кафедрой
 DX₃ := 4 DY₃ := 600 профессор

$DX_4 := 5$ $DY_4 := 700$ доцент
 $DX_5 := 6$ $DY_5 := 800$ старший преподаватель
 $DX_6 := 8$ $DY_6 := \max$

$x := 1..8$ $y(x) := \text{slope}(DX, DY) \cdot x + \text{intercept}(DX, DY)$
 $i := 1..8$ $DD_i := y(1)$ $RDD := \text{Round}(DD, 10)$

$DD^T = (446.61 \ 511.86 \ 577.12 \ 642.37 \ 707.63 \ 772.88 \ 838.14 \ 903.39)$
 $RDD^T = (450 \ 510 \ 580 \ 640 \ 710 \ 770 \ 840 \ 900)$

На рисунке 1 показан график, построенный в MathCAD: узловые точки и аппроксимирующая прямая (модель 1.3). Результаты моделирования по моделям 1.1–1.3 приведены в таблице 3.

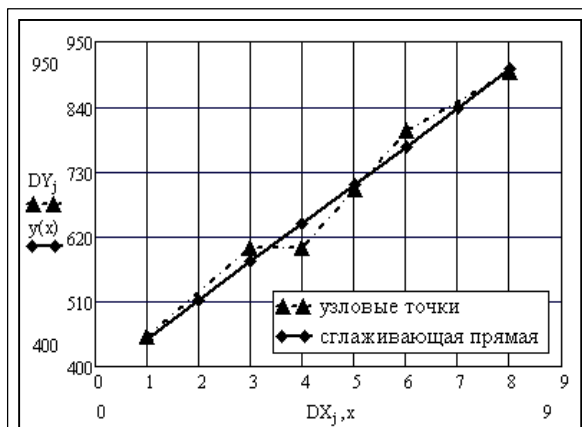


Рис. 1. Линейная модель 1.3. Аппроксимация линейной функции по 6 точкам

Fig. 1. The linear model 1.3.
Linear function approximation by 6 points

Таблица 3

Результаты моделирования.
Нормативы учебной нагрузки

Table 3

Simulation results. Teaching load standards

Долж- ность	Линейная дифференци- ация			Нелинейная дифференциация					
	Номер модели								
	1.1	1.2	1.3	2.1	2.2	2.3	3.1	3.2	3.3
Директор инсти- тута	450	430	450	450	450	450	450	450	450
Декан факуль- тета	510	500	510	510	550	560	480	490	490
Заведую- щий ка- федрой	580	560	580	570	600	600	540	540	540
Профес- сор	640	630	640	640	600	600	600	600	600
Доцент	710	690	710	710	700	700	670	670	670
Старший препода- ватель	770	760	770	780	800	800	750	750	750
Препода- ватель	840	820	840	850	860	870	830	830	830
Асси- стент	900	880	900	900	900	900	900	900	900

Следует отметить, что результаты моделирования по моделям 1.1 и 1.3 дают идентичные результаты. Неокругленные значения нормативной учебной нагрузки в данных моделях различны, округление свело на нет различия между моделями. Для вариантов со сложным распределением промежуточных узловых точек (модель 1.3) необходимо перейти к нелинейным моделям.

Нелинейная дифференциация. В таблице 4 приведено описание методов построения аппроксимирующей (интерполирующей) функции. Обработка результатов: округление до ближайшего значения, кратного 10 (функция Round).

Таблица 4

Нелинейная дифференциация.
Методы аппроксимации (интерполяции)

Table 4

Nonlinear differentiation.
Approximation methods (interpolation)

№ модели	Исходные данные	Метод	Вид кривой
2.1	Модель 1.3 (см. табл. 2)	Построение аппроксимирующей кривой по методу наименьших квадратов	Кубический полином
2.2	Модель 1.3 (см. табл. 2)	Сплайн-интерполяция	Кубический сплайн с линейными конечными точками
2.3	Модель 1.3 (см. табл. 2)	Сплайн-интерполяция	Кубический сплайн с параболическими конечными точками

Приведем фрагмент документа MathCAD для расчета моделей 2.1–2.3:

Модели 2.1–2.3

$x := 1..1..8$ $i := 1..8$ $j := 1..6$
 $z := \text{regress}(DX, DY, 3)$ $z1 := \text{lspline}(DX, DY)$ $z2 := \text{pspline}(DX, DY)$
 $y(x) := \text{interp}(z, DX, DY, x)$ $yl(x) := \text{intep}(z1, DX, DY, x)$ $y2(x) := \text{interp}(z2, DX, DY, x)$

$DD_i := y(1)$ $DD1_i := yl(1)$ D $DD2_i := y2(1)$
 $RDD := \text{Round}(DD, 10)$ $RDD1 := \text{Round}(DD1, 10)$ $RDD2 := \text{Round}(DD2, 10)$

Результаты:

$DD^T = (454.84 \ 505.975 \ 567.726 \ 636.33 \ 708.025 \ 779.048 \ 845.638 \ 904.031)$
 $DD1^T = (450 \ 551.471 \ 600 \ 600 \ 700 \ 800 \ 861.029 \ 900)$
 $DD2^T = (450 \ 564.298 \ 600 \ 600 \ 700 \ 800 \ 866.472 \ 900)$
 $RDD^T = (450 \ 510 \ 570 \ 640 \ 710 \ 780 \ 850 \ 900)$
 $RDD1^T = (450 \ 550 \ 600 \ 600 \ 700 \ 800 \ 860 \ 900)$
 $RDD2^T = (450 \ 560 \ 600 \ 600 \ 700 \ 800 \ 870 \ 900)$

Результаты моделирования отражены в таблице 3. На рисунке 2 показаны построенные кривые.

Аналогично проведена аппроксимация (интерполяция) по узловым точкам модели 1.2 (4 точки, см. табл. 2) нелинейными методами согласно таблице 4. Результаты моделирования представлены в таблице 3 (модели 3.1–3.3).

Оценка эффективности моделей. Оценка эффективности моделей 1.1–3.3 проводилась на основании примерной численности педагогических работников структурного подразделения вуза – института (табл. 5). Для моделирования ППС принимались следующие условия:

- количество преподавателей, имеющих ученую степень, – не менее 60 % от общей численности ППС (процент оспепенности);
- количество преподавателей, имеющих ученую степень доктора наук, – не менее 10 % от общей численности ППС (процент докторов наук).

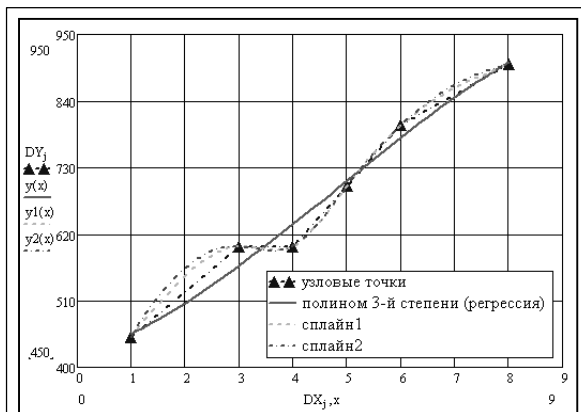


Рис. 2. Аппроксимирующие (интерполирующие) кривые (модели 2.1–2.3)

Fig. 2. Approximation (interpolation) curves (models 2.1–2.3)

Критерий эффективности – наименьший объем учебной нагрузки (количество часов), переходящей в сверхнормативную.

Таблица 5

Характеристики ППС (структура численности)

Table 5

Characteristics of the teaching staff composition (number structure)

Исходные данные	Вариант 1	Вариант 2	Вариант 3
Средняя учебная нагрузка (час.)	900	900	900
Общая численность ППС (чел.)	112	112	112
Процент оспепенности	72,3	91,1	61,6
Процент докторов наук	12,5	18,8	10,7
Распределение численности ППС по должностям (112 чел.)			
Директор института	1	1	1
Декан факультета	3	3	3
Заведующий кафедрой	15	15	15
Профессор	14	21	12
Доцент	48	62	38
Старший преподаватель	12	7	23
Преподаватель	10	0	5
Ассистент	9	3	15

Рассчитывались и анализировались следующие характеристики:

- объем сверхнормативной учебной нагрузки (час.);
- процентное соотношение объема сверхнормативной учебной нагрузки к общему количеству нагрузки (таблица 6 – процент сверхнормативной нагрузки);
- среднее количество часов сверхнормативной нагрузки в расчете на каждого преподавателя (таблица 6 – среднее значение сверхнормативной нагрузки).

Расчет проводился по округленным дискретным значениям рассчитанной нормативной нагрузки (табл. 3). Результаты представлены в таблице 6; ячейки с наименьшими значениями для наглядности выделены серым цветом.

Анализ результатов (табл. 6) показывает следующее:

- для варианта 1 (остепененность 72,3 %) наиболее эффективными являются модели 2.1 (нелинейная дифференциация: аппроксимация кубическим полиномом по методу наименьших квадратов по 6 точкам) и 2.3 (нелинейная дифференциация: сплайн-интерполяция кубическим сплайном с параболическими конечными точками по 6 точкам);
- для варианта 2 (высокая оспепенность 91,1 %) наиболее эффективными являются модели 1.1 (линейная дифференциация: аппроксимация по методу наименьших квадратов линейной функцией по двум точкам) и 1.3 (линейная дифференциация: аппроксимация по методу наименьших квадратов линейной функцией по 6 точкам);
- для варианта 3 (пороговая оспепенность 61,6 %) наиболее эффективной является модель 2.3 (нелинейная дифференциация: сплайн-интерполяция кубическим сплайном с параболическими конечными точками по 6 точкам);
- среднее значение сверхнормативной нагрузки, полученное по наиболее эффективным моделям, различается в зависимости от структуры ППС: наименьшее значение соответствует структуре с пороговой оспепенностью (175,2 час.), наибольшее – структуре с высокой оспепенностью (219,4 час.).

Представленные результаты позволяют сделать вывод, что не всегда линейная дифференциация верхних пределов учебной нагрузки является наилучшим решением. Линейные модели в данном исследовании оказались наиболее эффективными только для модели вуза с высоким показателем оспепенности (вариант 2); вместе с тем для подобной структуры численности минимальная сверхнормативная нагрузка по различным моделям дифференциации превышает максимальную сверхнормативную нагрузку для вуза с пороговым и средним значениями оспепенности.

Следует отметить, что построенные по методу сплайн-интерполяции модели 2.2 и 2.3 (ступенька) не совсем соответствуют требованию дифференци-

Таблица 6

Рассчитанные объемы сверхнормативной нагрузки

Table 6

Calculated volumes of excess teaching load

Структура численности ППС (№ варианта)	Номер модели								
	1.1	1.2	1.3	2.1	2.2	2.3	3.1	3.2	3.3
<i>Сверхнормативная нагрузка (%)</i>									
1	21.2	23.1	21.2	21.1	21.2	21.1	24.7	24.6	24.6
2	24.4	26.3	24.4	24.5	25.2	25.2	28.5	28.5	28.5
3	19.9	21.7	19.9	19.8	19.5	19.5	23.1	23.0	23.0
<i>Среднее значение сверхнормативной нагрузки (час.)</i>									
1	190.5	207.9	190.5	189.9	191.1	189.9	221.9	221.6	221.6
2	219.4	236.6	219.4	220.1	226.8	226.5	256.4	256.2	256.2
3	179.0	195.6	179.0	177.9	175.9	175.2	207.6	207.3	207.3

ации: исходные узловые точки по должностям заведующего кафедрой и профессора уравнены по ординате. Вопрос дифференциации по данным должностям требует дальнейшего обсуждения. Однако не стоит отвергать модели с неравномерным шагом и даже со ступенькой: для вузов со средним и пороговым вариантами оспетченности (варианты 1, 3) модель 2.3 (сплайн-интерполяция) дает лучшие результаты, чем линейная модель 1.1.

Таким образом, в зависимости от различной структуры численности ППС наиболее эффективными оказываются различные модели дифференциации учебной нагрузки; наиболее распространенные линейные модели дифференциации учебной нагрузки не являются наиболее эффективными в общем случае.

Для определения наиболее предпочтительной модели дифференциации учебной нагрузки для конкретного вуза с применением разработанных моделей необходимо, в первую очередь, ориентироваться на фактическую структуру численности ППС (распределение по должностям, количество преподавателей с ученой степенью) – базовые параметры; далее в процессе обсуждения с руководителями и представителями различных подразделений вуза необходимо определить минимальный и максимальный пределы учебной нагрузки, узловые точки – данные параметры могут варьироваться в процессе проработки вариантов. Критерий эффективности, применяемый при оценке полученных моделей, – наименьший объем учебной нагрузки (количество часов), переходящей в сверхнормативную, – остается неизменным.

Кроме простоты, наглядности и достаточно легкой воспроизводимости, разработанные вычислительные модели отличаются тем, что хорошо вписываются в контур системы поддержки принятия решений даже в «ручном» варианте, без разработанной автоматизированной системы поддержки принятия управленческих решений на уровне вуза. Вместе с тем они позволяют оперативно получать

варианты оптимальной дифференциации учебной нагрузки и представляют собой основу для дальнейшей автоматизации.

Литература

1. Курбатова М.В., Донов И.В. Эффекты внешнего контроля деятельности преподавателей российских вузов // Вестн. Омского ун-та. Сер.: Экономика. 2015. № 2. С. 17–27.
2. Harrison J.R., Lin Z., Carroll G.R., Carley K.M. Simulation modeling in organizational and management research. Academy of Management Review, 2007, vol. 32, no. 4, pp. 1229–1245.
3. Carley K.M. Computational and mathematical organization theory: Perspective and directions. Computational & Mathematical Organization Theory, 1995, vol. 1, no. 1, pp. 39–56.
4. Виноградов Г.П., Бурдо Г.Б., Исаев А.А. Согласованное принятие решений в производственных системах изготовления наукоемких изделий // Программные продукты и системы. 2015. № 2. С. 75–82.
5. Виноградов Г.П., Шматов Г.П., Борзов Д.А. Формирование представлений агента о предметной области в ситуации выбора // Программные продукты и системы. 2015. № 2. С. 83–94.
6. Бурдо Г.Б., Федотова А.В. Алгоритмы и модели АСУ технологическими процессами технического обслуживания // Программные продукты и системы. 2015. № 4. С. 237–243.
7. Завгородний В.Н. Моделирование процессов принятия решений в сложных организационно-технических системах // Программные продукты и системы. 2014. № 1. С. 147–150.
8. Беляева М.А., Бурляева О.К., Сырова И.В. Формирование мультимодельной системы для принятия оптимальных управленческих решений на предприятии // Программные продукты и системы. 2014. № 2. С. 181–187.
9. Шведенко В.Н., Веселова Н.С. Моделирование информационных ресурсов при процессной организации системы управления предприятием // Программные продукты и системы. 2014. № 4. С. 260–264.
10. e Costa C.A.B., Oliveira M.D. A multicriteria decision analysis model for faculty evaluation. Omega, 2012, vol. 40, no. 4, pp. 424–436.
11. Bai S., Hussain S., Rajput Q., Khoja S.A. Faculty performance evaluation system: An ontological approach. Proc. 11th Intern. Conf. on Computer Systems and Applications (AICCSA). IEEE, 2014, pp. 117–124.
12. Collan M., Stoklasa J., Talasova J. On academic faculty evaluation systems—more than just simple benchmarking. Intern. Jour. of Process Management and Benchmarking. 2014, vol. 4, no. 4, pp. 437–455.
13. Охорзин В.А. Оптимизация экономических систем. Примеры и алгоритмы в среде MathCAD: учеб. пособие. М.: Финансы и статистика, 2005. 144 с.

**SCIENTIFIC AND TECHNICAL COMPUTING SYSTEM MATHCAD TO DESIGN
AND ANALYZE DIFFERENTIATED TEACHING LOAD MODELS**

E.Yu. Shakhova¹, Ph.D. (Engineering), Associate Professor, koroleva0602@mail.ru

¹ Bratsk State University, Makarenko St. 40, Bratsk, 665709, Russian Federation

Abstract. Nowadays, many universities have processing of local regulations governing the work of academic teaching staff. There is a strong requirement to differentiate teaching load upper limits according to academic teaching staff positions. The paper considers and analyzes linear and non-linear models of standard teaching load distribution. The methods used are the following: least-square approximation by a linear function and third-degree polynomial, spline interpolation. The simulation was performed using MathCAD. The authors defined the most effective models for different population structure of professors. The criterion of efficiency is the smallest amount of excess teaching load.

Evaluating of the constructed linear and nonlinear models' effectiveness was based on three options of an academic teaching staff population structure in a university (institute) structural unit with high, medium and threshold levels of professors and associate professors in a total number.

The obtained results show that the linear differentiation of teaching load upper limits is not always the best solution. Linear models in this study proved to be most effective only in case of the institute model with a high level of professors and associate professors in the total number of teaching staff. The most effective non-linear model for models with middle and threshold levels of academic teaching staff is a spline interpolation model (a curve that is cubic spline with parabolic endpoints).

The paper also shows that for all the above mentioned differentiation models with a high level professors and associate professors in the total number of institute teaching staff the minimum excess teaching load exceeds the maximum excess teaching load for the institutions with a threshold and average level.

Keywords: modeling, linear model, nonlinear model, MathCAD, standard teaching load.

References

1. Kurbatova M.V., Donova I.V. Effects of the external control of lecturers' activities of the Russian universities. *Vestnik Omskogo univ. Seriya: Ekonomika* [Herald of Omsk Univ. Series Economics]. 2015, no. 2, pp. 17–27 (in Russ.).
2. Harrison J.R., Lin Z., Carroll G.R., Carley K.M. Simulation modeling in organizational and management research. *Academy of Management Review*. 2007, vol. 32, no. 4, pp. 1229–1245.
3. Carley K.M. Computational and mathematical organization theory: Perspective and directions. *Computational & mathematical organization theory*. 1995, vol. 1, no. 1, pp. 39–56.
4. Vinogradov G.P., Burdo G.B., Isaev A.A. Decision-making in high-tech products production systems. *Programmnye produkty i sistemy* [Software & Systems]. 2015, no. 2 (110), pp. 75–82 (in Russ.).
5. Vinogradov G.P., Shmatov G.P., Borzov D.A. Formation of agent's representations of the domain in a situation of choice. *Programmnye produkty i sistemy* [Software & Systems]. 2015, no. 2 (110), pp. 83–94 (in Russ.).
6. Burdo G.B., Fedotova A.V. Algorithms and models for maintenance process automated control systems. *Programmnye produkty i sistemy* [Software & Systems]. 2015, no. 4, pp. 237–243 (in Russ.).
7. Zavgorodny V.N. Modeling of decision-making processes in complex organizational and technical systems. *Programmnye produkty i sistemy* [Software & Systems]. 2014, no. 1, pp. 147–150 (in Russ.).
8. Belyaeva M.A., Burlyaeva O.K., Syrova I.V. Creating a multi-model system to make optimal management decisions in the company. *Programmnye produkty i sistemy* [Software & Systems]. 2014, no. 2, pp. 181–187 (in Russ.).
9. Shvedenko V.N., Veselova N.S. Enterprise information resources modeling for process organization of an enterprise management system. *Programmnye produkty i sistemy* [Software & Systems]. 2014, no. 4, pp. 260–264 (in Russ.).
10. e Costa C.A.B., Oliveira M.D. A multicriteria decision analysis model for faculty evaluation. *Omega*. 2012, vol. 40, no. 4, pp. 424–436.
11. Bai S., Hussain S., Rajput Q., Khoja S.A. Faculty performance evaluation system: An ontological approach. *IEEE/ACS 11th Int. Conf. on Computer Systems and Applications (AICCSA)*. 2014, pp. 117–124.
12. Collan M., Stoklasa J., Talasova J. On academic faculty evaluation systems – more than just simple benchmarking. *Int. Jour. of Process Management and Benchmarking*. 2014, vol. 4, no. 4, pp. 437–455.
13. Okhorzin V.A. *Optimizatsiya ekonomicheskikh sistem. Primery i algoritmy v srede MathCAD* [Optimization of economic systems. Examples and Algorithms in MathCAD]. Study guide. Moscow, 2005.

УДК 004.891

DOI: 10.15827/0236-235X.116.189-193

Дата подачи статьи: 19.01.16

2016. Т. 29. № 4. С. 189–193

МЕТОДИКА ОЦЕНКИ КАЧЕСТВА ОБУЧЕНИЯ СТУДЕНТОВ ВУЗА С ИСПОЛЬЗОВАНИЕМ НЕЙРО-НЕЧЕТКОГО ПОДХОДА

А.Р. Айдинян, к.т.н., доцент, andstyle@mail.ru;

О.Л. Цветкова, к.т.н., доцент, olga_cvetkova@mail.ru

*(Донской государственный технический университет,
пл. Гагарина, 1, г. Ростов-на-Дону, 344000, Россия)*

В настоящее время в соответствии с Болонским процессом для обеспечения успешного функционирования высших учебных заведений и повышения их конкурентоспособности на международном рынке внедряются системы оценки качества обучения. На точность определения эффективности работы вуза в большой степени влияет правильный выбор критериев оценки факторов. Оценка качества образования затрудняется тем, что значение этого показателя обусловлено множеством факторов, возможно, с неизвестным характером влияния, а также тем, что при проведении педагогических измерений необходимо работать с информацией, имеющей нечисловой характер.

Для решения поставленных задач был проведен обзор факторов, оказывающих наибольшее влияние на процесс обучения студентов. Традиционно среди них выделяют следующие: качество взаимодействия с рынком труда при формировании содержания образования, качество абитуриентов и студентов, технологий обучения, учебно-методического и материально-технического обеспечения образовательного процесса, профессорско-преподавательского состава.

Предложенная авторами работы методика оценки качества обучения студентов вуза основана на использовании двухуровневой системы, построенной на базе адаптивной системы нейро-нечеткого вывода ANFIS, реализованной в пакете Fuzzy Logic Toolbox системы MatLab и искусственных нейронных сетей.

Для оценки четырех выделенных групп факторов используются четыре модуля, реализованные с помощью искусственных нейронных сетей. Число входов каждой из этих сетей обусловлено числом факторов, входящих в соответствующую группу. На входы искусственных нейронных сетей подаются значения показателей в виде величин в диапазоне от 0 до 1. На выходе каждой ИНС формируются значения оценки соответствующей группы факторов также в виде величин в диапазоне от 0 до 1.

Полученные в итоге значения оценки групп факторов подаются на вход сети ANFIS, которая представляет собой второй уровень системы оценки качества обучения студентов вуза. А выходная переменная системы ANFIS представляет собой численную оценку качества обучения студентов вуза.

Двухуровневое вычисление упрощает сбор данных для обучения искусственных нейронных сетей, а также процесс формирования экспертных оценок для обучения и формирования входов искусственных нейронных сетей и сети ANFIS.

Предложенная система оценки качества обучения студентов вуза позволяет получить числовую оценку, представляющую собой суммарный показатель, характеризующий результаты деятельности образовательного учреждения и показывающий, насколько эффективно проходит процесс обучения.

Ключевые слова: качество образования, искусственные нейронные сети, адаптивные сети нейро-нечеткого вывода, экспертные оценки.

При проведении педагогических исследований неизбежна работа с информацией, имеющей нечисловой, слабоформализуемый характер. Вследствие этого возникает необходимость построения методики количественного описания процессов и предметов, связанных с педагогикой [1, 2]. Особое значение имеет качество образования, под которым понимают некоторый суммарный показатель, отражающий результат деятельности образовательного учреждения, а также соответствие потребностям и ожиданиям общества (различных социальных групп) в формировании различных компетенций личности.

Оценка качества образования затрудняется тем, что значение этого показателя зависит от множества факторов, возможно, с неизвестным характером влияния. Также в данном случае имеет место специфичность «продукта» образования – выпускник учебного заведения, который должен рассматриваться как сложная система. Существуют различные методы и алгоритмы оценки качества обра-

зования [2, 3], в данной статье предлагается методика оценки качества обучения студентов вуза на основе нейро-нечеткого подхода.

Предварительно необходимо определить факторы, оказывающие наибольшее влияние на процесс обучения студентов. Традиционно среди них выделяют следующие [4].

1. Качество взаимодействия с рынком труда при формировании содержания образования:

- уровень привлечения работодателей к формированию образовательных программ;
- наличие системы целевой подготовки специалистов по заказам предприятий;
- наличие соглашений с зарубежными университетами о совместной реализации образовательных программ;
- процент иностранных студентов по отношению ко всем студентам;
- уровень развития системы анализа и содействия трудоустройству выпускников.

2. Качество абитуриентов и студентов:
 - уровень усвоения текущего учебного материала;
 - комплексная оценка остаточных знаний;
 - уровень умения решать прикладные задачи;
 - уровень итоговой аттестации.
3. Качество технологий обучения, учебно-методического и материально-технического обеспечения образовательного процесса:
 - уровень системы непрерывного контроля успеваемости студентов в течение семестра и технологии рейтингования студентов;
 - уровень учебно-методического материала;
 - уровень книгообеспеченности учебного процесса;
 - уровень обеспеченности специализированными лабораториями;
 - уровень организации поддержки обучения студентов по индивидуальным планам.
4. Качество профессорско-преподавательского состава:
 - уровень соответствия базового образования преподавателя с учетом стажа педагогической работы и специализации;
 - уровень квалификации преподавателя (научная степень, звание, должность);
 - уровень социально-психологических отношений преподавателя со студентами;
 - внедрение практики приглашения специалистов различных отраслей промышленности для проведения учебных занятий.

Построение нейро-нечеткой системы оценки качества обучения студентов вуза

Методику оценки качества обучения студентов вуза предлагается построить с использованием методов и средств искусственного интеллекта, реализованных в пакете Fuzzy Logic Toolbox системы MatLab в виде адаптивной системы нейро-нечеткого вывода ANFIS (Adaptive Neuro-Fuzzy Inference System) [5].

Гибридная система ANFIS представляет собой сочетание нейро-нечеткого метода вывода Сугено с возможностью обучения пятислойной *искусственной нейронной сети* (ИНС) прямого распространения с одним выходом и несколькими входами, которые представляют собой нечеткие лингвистические переменные.

В качестве входных переменных системы ANFIS используем оценки четырех групп факторов V_i ($i=1, \dots, 4$), которые, соответственно, определяют качество: V_1 – взаимодействия с рынком труда при формировании содержания образования, V_2 – абитуриентов и студентов, V_3 – технологий обучения, учебно-методического и материально-технического обеспечения образовательного процесса, V_4 – профессорско-преподавательского состава.

Выходная переменная системы ANFIS представляет собой численную оценку качества обучения студентов вуза и определяется как функция $y = f(V_1, V_2, V_3, V_4)$.

Слой 1 системы ANFIS для лингвистической оценки входных параметров использует терм-множества $A_{V_i} = \{\text{«удовлетворительное»}, \text{«хорошее»}, \text{«отличное»}\}$.

Оценка «неудовлетворительное» не рассматривается, поскольку при выставлении оценки «неудовлетворительно» по любой из входных переменных выходная переменная также принимает значение «неудовлетворительно».

В символическом виде запишем: $A_{V_i} = \{УД<i>, ХО<i>, ОТ<i>\}$.

Терм-множество выходной лингвистической переменной y составляет множество значений оценок качества обучения студентов вуза: $T_y = \{\text{«удовлетворительное»}, \text{«хорошее»}, \text{«отличное»}\}$ или в символическом виде: $T_y = \{УД, ХО, ОТ\}$.

Выходы узлов слоя 1 представляют собой значения функций принадлежности при конкретных значениях входных переменных.

Слой 2 является неадаптивным и определяет предпосылки нечетких продукционных правил. Каждый узел данного слоя соединен с теми узлами слоя 1, которые формируют предпосылки соответствующего правила.

Для решения рассматриваемой задачи сформулированы три нечетких продукционных правила: $P = \{p_1, p_2, p_3\}$.

В соответствии с особенностями сети ANFIS необходимо, чтобы число правил сети соответствовало размерности терм-множества выходной переменной y .

Узлы выполняют нечеткую логическую операцию «И» (min). Выходами узлов данного слоя являются степени истинности (выполнения) предпосылок каждого из трех нечетких продукционных правил, вычисленные по формулам:

$$\begin{cases} w_1 = \min(\mu_{УД1}(V_1), \mu_{УД2}(V_2), \mu_{УД3}(V_3)), \\ w_2 = \min(\mu_{ХО1}(V_1), \mu_{ХО2}(V_2), \mu_{ХО3}(V_3)), \\ w_3 = \min(\mu_{ОТ1}(V_1), \mu_{ОТ2}(V_2), \mu_{ОТ3}(V_3)). \end{cases}$$

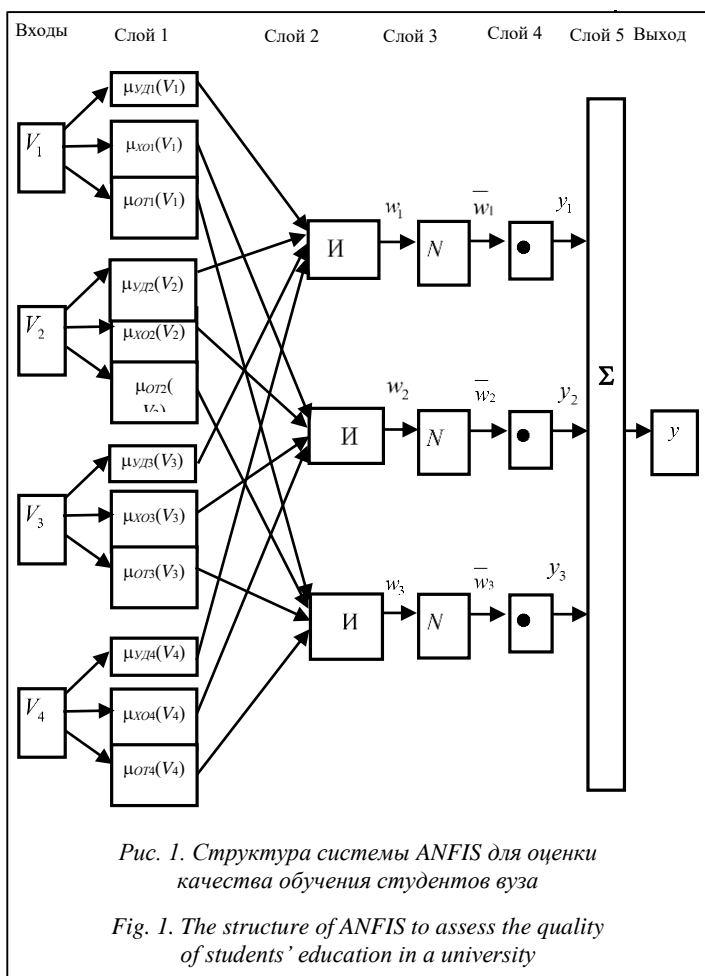
Слой 3 проводит нормализацию степеней выполнения каждого из нечетких продукционных правил (вычисление относительной степени выполнения правил) следующим образом:

$$\bar{w}_h = w_h / \sum_{i=1}^3 w_i, \text{ где } h = 1, \dots, 3 - \text{номер продукционного правила.}$$

На слое 4 вычисляется вклад (заключение правил) каждого нечеткого продукционного правила в выход сети по формуле

$$y_h(v, V) = \bar{w}_h (v_h^{(0)} + v_h^{(1)}V_1 + v_h^{(2)}V_2 + v_h^{(3)}V_3 + v_h^{(4)}V_4),$$

где $v_h^{(i)}$ – коэффициенты выходной функции ($i = 0, \dots, 4$).



Слой 5 обобщает вклады

всех правил: $y = \sum_{i=1}^3 y_i$.

Структура системы ANFIS для оценки качества обучения студентов вуза показана на рисунке 1.

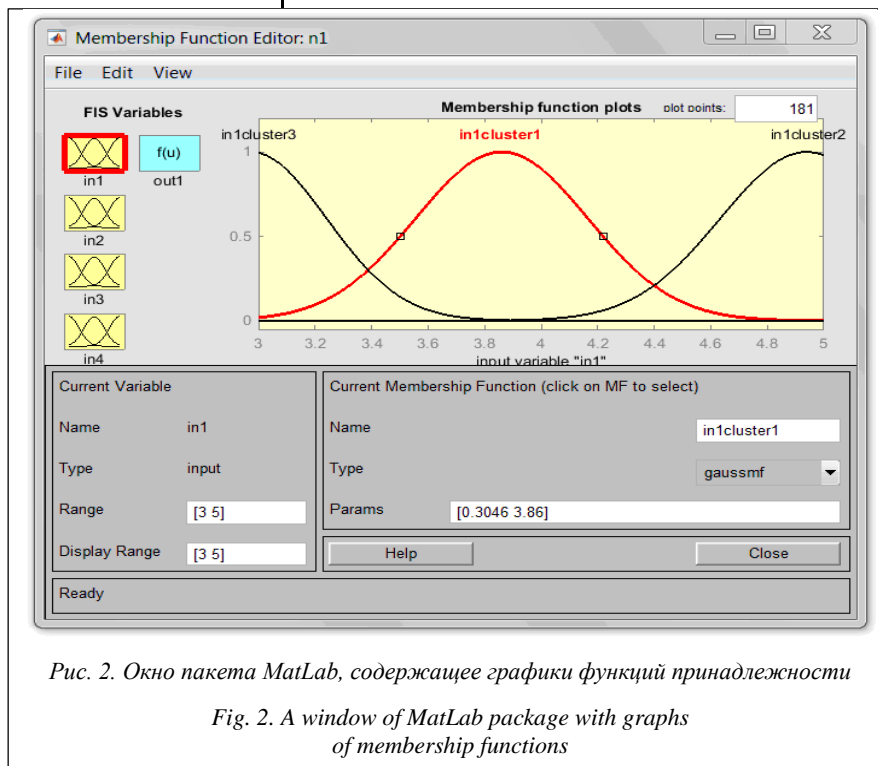
Обучение сети ANFIS осуществлялось за 24 эпохи гибридным методом. При обучении подбираются вид функций принадлежности, вид выходной функции и их коэффициенты. В результате обучения нечеткой сети для трех правил в качестве функций принадлежности были приняты функции Гаусса, а в качестве выходной функции – линейная функция. В результате обучения также были получены функции принадлежности и их коэффициенты, часть которых представлена на рисунке 2.

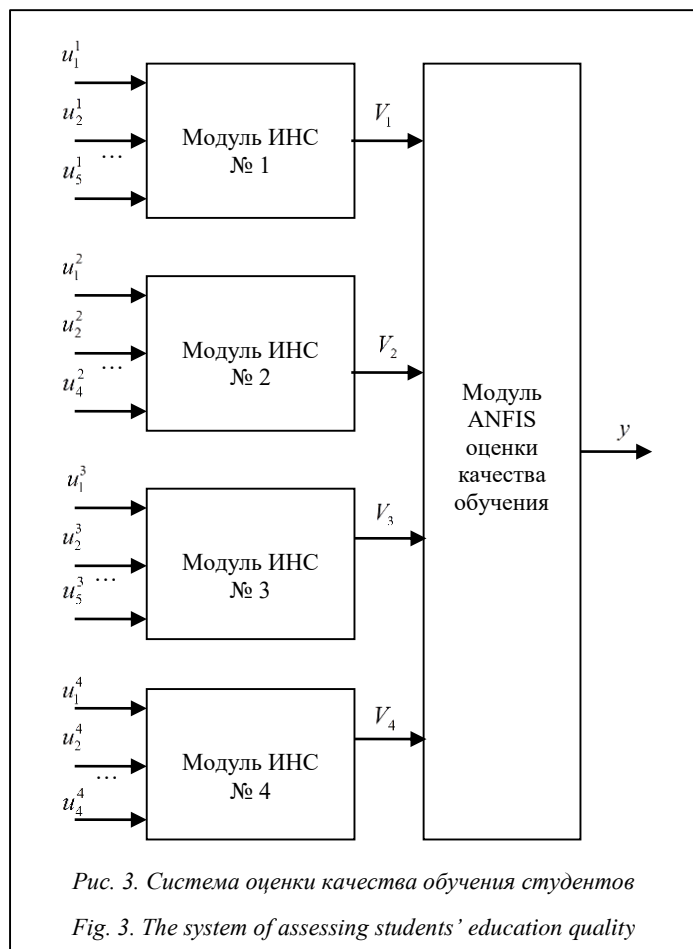
Нейросетевые модули оценки групп факторов, влияющих на процесс обучения

Для оценки каждой из четырех групп факторов, оказывающих влияние на качество процесса обучения студентов, используются четыре модуля, реализованные с помощью ИНС, предназначенные для учета качества взаимодействия с рынком труда при формировании содержания образования (модуль № 1), качества абитуриентов и студентов (модуль № 2), технологий обучения, учебно-методического и материально-технического обеспечения образовательного процесса (модуль № 3), профессорско-преподавательского состава (модуль № 4).

В зависимости от имеющихся характеристик вуза или конкретного направления обучения будут формироваться входные сигналы этих модулей.

Для реализации модулей используются ИНС, показавшие высокую эффективность при решении различных задач [6–8]. Число входов каждой ИНС обусловлено числом факторов, входящих в соответствующую группу. На входы ИНС подаются значения показателей в виде величин в диапазоне от 0 до 1. На выходе каждой ИНС формируются значения оценки соответствующей группы факторов также в виде величин в диапазоне от 0 до 1.





При выборе структуры сети необходимо тщательно проанализировать закономерности обучающей выборки [9–11]. Все используемые ИНС являются сетями прямого распространения с одним скрытым слоем с количеством сигмоидальных нейронов $2v+1$, где v – количество входов этой ИНС. Данное количество искусственных нейронов в скрытом слое связано с теоремой Колмогорова и позволяет с достаточной степенью точности аппроксимировать требуемую функцию при правильном обучении [9, 12]. Разработанная система оценки качества обучения студентов вуза изображена на рисунке 3.

Обучение осуществлялось методом оптимизации Левенберга–Маркара (TRAINLM) в течение 17 эпох. Ошибка обучения определялась с помощью среднеквадратического отклонения и составила менее $2 \cdot 10^{-14}$.

Значения оценки групп факторов подаются на вход сети ANFIS, которая представляет собой второй уровень системы оценки качества обучения студентов вуза.

Двухуровневое вычисление упрощает сбор данных для обучения ИНС, а также процесс формирования экспертных оценок для обучения и формирования входов ИНС и сети ANFIS.

Для анализа оценок, полученных от экспертов, применяются различные методы математической статистики, которые могут комбинироваться в зависимости от типа задачи и необходимого результата.

Предложенная система оценки качества обучения студентов вуза позволяет получить числовую оценку, представляющую собой суммарный показатель, характеризующий результаты деятельности образовательного учреждения и показывающий, насколько эффективно проходит процесс обучения.

Литература

1. Селиванов Е.И., Панюшкина Е.В. Краеугольные камни квалиметрии компетенций на соответствие требованиям федеральных государственных образовательных стандартов среднего профессионального образования // *Фундаментальные и прикладные исследования в современном мире*. 2014. Т. 3. № 5. С. 87–91.
2. Ворсина Е.В., Снигирева Т.А. Квалиметрия учебной компетентности студентов на основе анализа учебных текстов // *Вестн. Костромского гос. ун-та им. Н.А. Некрасова*. 2012. Т. 18. № 3. С. 124–127.
3. Лисецкий Ю.М. Алгоритм сравнения методов комплексной количественной оценки качества сложных систем // *Программные продукты и системы*. 2012. № 4. С. 153–156.
4. Бартасевич И.Г. Система критериев качества обучения студентов в высшем учебном заведении // *Вестн. Астраханского гос. технич. ун-та*. 2008. № 3. С. 217–220.
5. Штовба С.Д. Проектирование нечетких систем средствами MatLab. М.: Горячая линия–Телеком, 2007. 288 с.
6. Колесников А.А., Маршаков Д.В., Айдинян А.Р. Комплексное применение синергетического подхода и нейросетевых структур к проблеме синтеза интеллектуальной системы управления электроприводом // *Вестн. Донского гос. технич. ун-та*. 2014. Т. 14. № 4 (79). С. 60–71.
7. Цветкова О.Л., Айдинян А.Р. Интеллектуальная система оценки информационной безопасности предприятия от внутренних угроз // *Вестн. компьютерных и информационных технологий*. 2014. № 8 (122). С. 48–53.
8. Галушка В.В., Фатхи В.А. Формирование обучающей выборки при использовании искусственных нейронных сетей в задачах поиска ошибок баз данных // *Инженер. вестн. Дона*. 2013. № 2. URL: ivdon.ru/ru/magazine/archive/n2y2013/1597/ (дата обращения: 20.08.2015).
9. Осовский С. Нейронные сети для обработки информации. М.: Финансы и статистика, 2002. 344 с.
10. Анохин М.Н., Березняк С.А., Бланко Л.М.Л. Организация пространственной группировки сигналов в ассоциативных процессах искусственных нейронных сетей // *Инженер. вестн. Дона*. 2014. Т. 28. № 1. URL: ivdon.ru/ru/magazine/archive/n1y2014/2230/ (дата обращения: 20.08.2015).
11. Вагин В.Н., Ганишев В.А. Кластеризация пользователей по голосу с помощью улучшенных самоорганизующихся растущих нейронных сетей. // *Программные продукты и системы*. 2015. № 3 (111). С. 136–142. URL: <http://elibrary.ru/item.asp?id=24131141> (дата обращения: 22.12.2015).
12. Белявский Г.И., Пучков Е.В., Лиля В.Б. Алгоритм и программная реализация гибридного метода обучения искусственных нейронных сетей // *Программные продукты и системы*. 2012. № 4. С. 96–100.

METHODS OF ASSESSING THE QUALITY OF UNIVERSITY STUDENTS' EDUCATION
USING A NEURO-FUZZY APPROACHA.R. Aydinyan¹, Ph.D. (Engineering), Associate Professor, andstyle@mail.ruO.L. Tsvetkova¹, Ph.D. (Engineering), Associate Professor, olga_cvetkova@mail.ru¹ Don State Technical University, Gagarin Sq. 1, Rostov-on-Don, 344000, Russian Federation

Abstract. Nowadays according to the Bologna process the system of education quality assessment are introduced for successful functioning of higher education institutions and enhancing their competitive in the international market. The accuracy of determining the university efficiency to a large extent depends on the correct choice of criteria to assess the factors that determine education quality. Education quality assessment is hampered by the fact that the value of this indicator depends on many factors, as well as the fact that during the pedagogical dimensions it is inevitable to work with non-numeric information.

In order to achieve the objectives the authors have reviewed factors that have the greatest impact on student learning. Traditionally among these factors there are the following: the quality of interaction with a labor market during the formation of educational content, the quality of applicants and students, the quality of teaching technologies, teaching and logistical support of an educational process, the quality of teaching staff.

The proposed methods of assessing the quality of students education is based on a two-level system. It is built on the basis of an adaptive neuro-fuzzy inference system (ANFIS) which is implemented in the Fuzzy Logic Toolbox package of MatLab and artificial neural networks.

To evaluate each of the four groups of factors that influence the quality of a students' learning process, there are four modules implemented using artificial neural networks. The number of inputs of each artificial neural network is caused by a number of factors included in an appropriate group. Factor group values go to the input of ANFIS, which is the second level of a quality assessment system for university students' education. The output variable of ANFIS is a numerical evaluation of students' learning quality. Two-level calculation simplifies data collection for artificial neural networks training and the expert assessments formation for training and formation of inputs of artificial neural networks and ANFIS.

The proposed system of students education quality assessment provides a numerical estimate, which is an aggregate measure of an educational institution performance, showing learning process efficiency.

Keywords: education quality, artificial neural networks, adaptive neuro-fuzzy inference systems, expert assessments.

References

1. Selivanov E.I., Panyushkina E.V. The foundation stones of qualimetry of competencies to satisfy the requirements of the federal state educational standards for secondary vocational education. *Fundamentalnye i prikladnye issledovaniya v sovremennoy mire* [Basic and Applied Research in a Modern World]. 2014, vol. 3, no. 5, pp. 87–91 (in Russ.).
2. Vorsina E.V., Snigireva T.A. Qualimetry of educational competence of students based on the analysis of educational texts. *Vestnik Kostromskogo gos. univ. im. N.A. Nekrasova* [Bull. Kostroma State Univ.]. 2012, vol. 18, no. 3, pp. 124–127 (in Russ.).
3. Lisetsky Yu.M. A comparison algorithm of methods of complex quantitative quality of complex system evaluation. *Programmnye produkty i sistemy* [Software & Systems]. 2012, no. 4, pp. 153–156 (in Russ.).
4. Bartasevich I.G. The system of quality factors of students education in high school. *Vestnik Astrakhanskogo gos. tekhnich. univ.* [Vestn. Astrakhan State Tech. Univ.]. 2008, no. 3, pp. 217–220 (in Russ.).
5. Shtovba S.D. *Proektirovaniye nechetkikh sistem sredstvami MatLab* [Fuzzy systems design of by means of MatLab]. Moscow, Goryachaya Liniya–Telekom Publ., 2007, 288 p. (in Russ.).
6. Kolesnikov A.A., Marshakov D.V., Aydinyan A.R. Integrated application of synergetic approach and neural network structures to the problem of electric drive intellectual control system synthesis. *Vestnik Donskogo gos. tekhnich. univ.* [Vestn. DSTU]. 2014, vol. 14, no. 4, pp. 60–71 (in Russ.).
7. Tsvetkova O.L., Aydinyan A.R. Intelligent system evaluation information security of the enterprise from internal threats. *Vestnik kompyuternykh i informatsionnykh technology* [Her. Comp. and Inform. Technol.]. 2014, no. 8, pp. 48–53 (in Russ.).
8. Galushka V.V., Fatkhi V.A. Train set forming for using artificial neural networks to database errors search. *Inzhenernyy vestnik Dona* [Eng. J. Don]. 2013, no. 2. Available at: <http://ivdon.ru/magazine/archive/n2y2013/1597/> (accessed August 20, 2015) (in Russ.).
9. Osovsky S. *Neyronnye seti dlya obrabotki informatsii* [Neural networks for information processing]. Moscow, Finansy i statistika Publ., 2002, 344 p.
10. Anokhin M.N., Bereznyak S.A., Blanko L.M.L. Spatial organization grouping signals in associative processes of artificial neural networks. *Inzhenernyy vestnik Dona* [Eng. J. Don]. 2014, vol. 28, no. 1. Available at: <http://ivdon.ru/magazine/archive/n1y2014/2230/> (accessed August 20, 2015) (in Russ.).
11. Vagin V.N., Ganishev V.A. Speaker clustering using enhanced self-organizing incremental neural networks. *Programmnye produkty i sistemy* [Software & Systems]. 2015, no. 3, pp. 136–142 (in Russ.). Available at: <http://elibrary.ru/item.asp?id=24131141> (accessed December 22, 2015).
12. Belyavsky G.I., Puchkov E.V., Lila V.B. Algorithm and software realization of hybrid methods of training artificial neural network. *Programmnye produkty i sistemy* [Software & Systems]. 2012, no. 4, pp. 96–100 (in Russ.).

СОДЕРЖАНИЕ

Сухобоков А.А., Сухобоков А.В., Лахвич Д.С., Тихонов И.В. Проблемы и перспективы создания APS для мультинациональных компаний	5
Болодурина И.П., Парфенов Д.И., Решетников В.Н. Моделирование размещения сервис-ориентированных приложений в программно-управляемой инфраструктуре виртуального центра обработки данных	15
Сидоров И.А. Архитектура пользовательского интерфейса для взаимодействия с мультиагентной средой	23
Лукинова О.В. Методологические аспекты управления жизненным циклом информационной системы на основе инструментов функциональной стандартизации	27
Атаева О.М. Информационная модель семантической библиотеки LibMeta	36
Батура Т.В. Семантический анализ и способы представления смысла текста в компьютерной лингвистике	45
Сулейманов Р.С. Извлечение метаданных из полнотекстовых электронных русскоязычных изданий при помощи Томита-парсера	58
Mutovkina N.Yu., Klyushin A.Yu., Semenov N.A. Models of heterogeneous agents' behavior in a conflict and their implementation algorithms.....	63
Гиацингов А.М., Мамросенко К.А. Отображение трехмерных объектов с использованием кластерной визуализации.....	69
Щекочихин О.В., Шведенко П.В. Анализ уровней интеграции компонентов гетерогенных информационных систем.....	73
Ключев А.О., Антонов А.А. Измерение производительности компонентов подсистемы памяти для гетерогенных систем на кристалле.....	78
Беляев С.А., Васильев А.В., Кудряков С.А. Архитектура системы мониторинга информационных трендов на основе свободного программного обеспечения	85
Парамонов А.В. Графическое моделирование технологического процесса как вспомогательное средство составления алгоритма управления	89
Горелова А.Е., Корнилова Н.Л., Григорьева З.Р., Васильев Д.А., Корнилова Л.А. Метод получения развертки деталей одежды с учетом деформационной способности материала	94
Зыков А.Г., Кочетков И.В., Поляков В.И. Применение системы KLEE для автоматизации тестирования программ на языках C/C++	101
Ткешелашвили Н.М., Клименков С.В., Дергачев А.М. Метод распознавания структуры таблицы в электронных табличных документах.....	107

Рябов С.Ю., Лютов А.Г., Рябов Ю.В., Вавилова А.В. Метод обмена информацией между программными системами автоматизации технологических и производственных процессов.....	113
Жуков В.Г., Жукова М.Н., Тимохович А.С., Волков Д.С. Программная система структурно-параметрического синтеза системы защиты информации	118
Семькина Н.А., Шавыкина И.В. Математическая модель защиты компьютерной сети от вирусов	125
Дроботун Е.Б., Козлов Д.В. Оценка степени влияния антивирусных программных средств на качество функционирования информационно-вычислительных систем.....	129
Духовнева А.В., Рыбаков М.Н., Шкатов Д.П. Алгоритмическая выразительность некоторых фрагментов языка логики ветвящегося времени.....	135
Буренков С.А., Шамаева О.Ю. Параллельные вычисления как средство повышения эффективности решения задач вычислительной аэродинамики.....	143
Потапов А.Н., Дикарев В.А., Допира Р.В., Абу-Абед Ф.Н., Мартынов Д.В. Направления реализации средств автоматизированной системы освоения радиоэлектронных комплексов.....	149
Сеченов П.А., Цымбал В.П., Оленников А.А. Постановка задачи исследования диффузионного перехода через границу шлак-металл в колонном реакторе и алгоритм ее решения.....	156
Коробейников А.Г., Поляков В.И., Федосовский М.Е., Алексанин С.А. Автоматизированная цифровая обработка изображений при решении задачи магнитной дефектоскопии.....	165
Кандалов П.И. Программный комплекс моделирования стохастических температурных полей в технических системах STF-ElectronMod	171
Зубкова Т.М., Мустюков Н.А., Токарева М.А. Построение архитектуры САПР одношнековых экструдеров с применением элементов искусственного интеллекта.....	176
Шахова Е.Ю. Системы поддержки принятия решений в управлении вузом: вычислительные модели дифференциации учебной нагрузки	183
Айдинян А.Р., Цветкова О.Л. Методика оценки качества обучения студентов вуза с использованием нейро-нечеткого подхода.....	189

C O N T E N T

Sukhobokov A.A., Sukhobokov A.V., Lakhvich D.S., Tikhonov I.V.	
APS for multinational companies: problems and prospects of development	5
Bolodurina I.P., Parfenov D.I., Reshetnikov V.N.	
Modeling of placing service-oriented applications in a software-defined infrastructure of the virtual data center	15
Sidorov I.A.	
User interface architecture for interaction with multi-agent environment	23
Lukinova O.V.	
Methodological aspects of information system life cycle management based on functional standardization tools	27
Ataeva O.M.	
An information model of LibMeta semantic library	36
Batura T.V.	
Semantic analysis and methods of text meaning representation in computer linguistics	45
Suleymanov R.S.	
Extraction of metadata from the full-text electronic materials written in russian using Tomita-parser	58
Mutovkina N.Yu., Klyushin A.Yu., Semenov N.A.	
Models of heterogeneous agents' behavior in a conflict and their implementation algorithms	63
Giatsintov A.M., Mamrosenko K.A.	
3D objects rendering using clustered shading	69
Schekochikhin O.V., Shvedenko P.V.	
Analysis of integration levels of heterogeneous information system components	73
Klyuchev A.O., Antonov A.A.	
Memory subsystem components performance evaluation for heterogeneous Systems-on-Chip	78
Belyaev S.A., Vasilev A.V., Kudryakov S.A.	
The monitoring of information trends system's architecture based on the free software	85
Paramonov A.V.	
Graphical modeling of technological process as a support tool for designing of control algorithm	89
Gorelova A.E., Kornilova N.L., Grigoreva Z.R., Vasilev D.A., Kornilova L.A.	
The method of obtaining a scan of clothing details considering deformational aptitude of a material	94
Zykov A.G., Kochetkov I.V., Polyakov V.I.	
KLEE for automatic testing programs in C/C++	101
Tkeshelashvili N.M., Klimenkov S.V., Dergachev A.M.	
Table structure recognition method for spreadsheets data	107
Ryabov S.Yu., Lyutov A.G., Ryabov Yu.V., Vavilova A.V.	
Method of information exchange between software systems of automation of technological and production processes	113
Zhukov V.G., Zhukova M.N., Timokhovich A.S., Volkov D.S.	
Program system of structural and parametrical synthesis of an information security system	118
Semykina N.A., Shavykina I.V.	
A mathematical model of computer network antivirus protection	125
Drobotun E.B., Kozlov D.V.	
Assessment of influence of anti-virus software on quality of information-computing system functioning	129
Dukhovneva A.V., Rybakov M.N., Shkatov D.P.	
Algorithmical power of some fragments of computational tree logic	135
Burenkov S.A., Shamayeva O.Yu.	
Parallel calculations as a way to increase the efficiency of computing aerodynamic problem solutions	143
Potapov A.N., Dikarev V.A., Dopira R.V., Abu-Abed F.N., Martynov D.V.	
The directions of implementation of automated system means for electronic systems development	149
Sechenov P.A., Tsymbal V.P., Olennikov A.A.	
Statement of the research problem of the diffusion transition across the slag-metal border in a column reactor and its solution algorithm	156
Korobeynikov A.G., Polyakov V.I., Fedosovsky M.E., Aleksanin S.A.	
Automated digital processing of images in solving the problem of magnetic defectoscopy	165
Kandalov P.I.	
Software modeling of stochastic temperature fields in stf-electron mod technical systems	171
Zubkova T.M., Mustyukov N.A., Tokareva M.A.	
Single screw extruder engineering with artificial intelligence elements	176
Shakhova E.Yu.	
Scientific and technical computing system mathcad to design and analyze differentiated teaching load models	183
Aydinyan A.R., Tsvetkova O.L.	
Methods of assessing the quality of university students' education using a neuro-fuzzy approach	189