

Software & Systems  
 Programmnye produkty i sistemy  
 2017, vol. 30, no. 3

Print ISSN 0236-235X

Online ISSN 2311-2735

# Программные продукты и системы

DOI 10.15827/0236-235X

DOI 10.15827/0236-235X

/ Проблемы отладки  
многопроцессных  
систем/ Воспроизведение  
видеоданных  
в виртуальной  
трехмерной среде

## / О системе стереозрения для мобильного робота

/ Программный комплекс  
теплового  
проектирования  
электронных систем



№ 3  
ТОМ 30  
2017

## ***Уважаемые читатели!***

Перед вами очередной номер нашего журнала. По уже сложившейся традиции в него включена большая подборка статей сотрудников Научно-исследовательского института системных исследований Российской Академии наук (НИИСИ РАН).

Статьи посвящены результатам научных исследований и разработкам института, развитию прикладного программного обеспечения и созданию новых элементов вычислительной техники. В них описываются новые подходы и методы, программы и комплексы, создаваемые в НИИСИ РАН. Следует отметить, что предложенные работы дают представление лишь о некоторых направлениях научных исследований института.

Большое внимание в этом номере уделяется визуализации, верификации моделей систем, созданию виртуальных моделей сложных технических изделий, обеспечению безопасной подготовки персонала, концептуальному моделированию.

Кроме того, в данном выпуске журнала опубликованы работы по проблемам автоматизации проектирования, исследований в области искусственного интеллекта, программного обеспечения, тренажерных и обучающих систем, а также касающиеся других актуальных вопросов.

Приятного вам общения с нашим журналом. Ждем отзывов и пожеланий.

*Научные редакторы*

*В.Н. Решетников, д.ф-м.н, профессор  
Н.А. Семенов, д.т.н, профессор*

---

Научно-исследовательский институт  
«Центрпрограммсистем»

---

# Программные продукты и системы

МЕЖДУНАРОДНЫЙ НАУЧНО-ПРАКТИЧЕСКИЙ ЖУРНАЛ

2017, том 30, № 3  
(год издания тридцатый)

Главный редактор  
С.В. ЕМЕЛЬЯНОВ, академик РАН

Тверь

## SOFTWARE & SYSTEMS

(PROGRAMMNYE PRODUKTY I SISTEMY)

International research and practice journal

2017, vol. 30, no. 3

Editor-in-Chief  
S.V. EMELYANOV, Academician of the Russian Academy of Sciences

Tver  
Russian Federation

---

Research Institute CENTERPROGRAMSYSTEM

---

## © ПРОГРАММНЫЕ ПРОДУКТЫ И СИСТЕМЫ

Международный научно-практический журнал

2017. Т. 30. № 3

DOI: 10.15827/0236-235X.119

### Главный редактор

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

### Научные редакторы:

**В.Н. РЕШЕТНИКОВ**, д.ф.-м.н., профессор МАИ  
(г. Москва, Россия)

**Н.А. СЕМЕНОВ**, д.т.н., профессор ТвГТУ  
(г. Тверь, 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)

**Нгуен 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 15.08.2017 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 117.

Год издания двадцать девятый. Формат 60×84 1/8. Объем 216 стр.

Заказ № 72. Тираж 1000 экз. Цена 330,00 руб.

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



**Editor-in-chief**

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

**Science editors:**

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

**N.A. Semenov**, Dr.Sc. (Engineering), Professor TvSTU  
(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 26<sup>th</sup> 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)

**Afanasiev 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 15.08.2017

Printed in printing-office "Faktor i K"

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

Published quarterly. 30th year of publication

Format 60×84 1/8. Circulation 1000 copies

Prod. order № 72. Wordage 216 pages. Price 330,00 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](http://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 (Committee on Publication Ethics – COPE).

УДК 621.382.2/3

DOI: 10.15827/0236-235X.119.345-352

Дата подачи статьи: 03.05.17

2017. Т. 30. № 3. С. 345–352

## **ВЫСОКОПРОИЗВОДИТЕЛЬНЫЙ МИКРОПРОЦЕССОР 1890ВМ118 С АРХИТЕКТУРОЙ КОМДИВ ДЛЯ СОЗДАНИЯ ДОВЕРЕННЫХ СИСТЕМ**

*С.И. Аряшев, к.т.н., зав. отделением, aserg@cs.niisi.ras.ru;*

*С.Г. Бобков, директор, bobkov@cs.niisi.ras.ru;*

*П.С. Зубковский, зав. отделом, zubkovsky@niisi.ras.ru;*

*С.А. Морев, младший научный сотрудник, morev@cs.niisi.ras.ru;*

*Б.Ю. Рогаткин, научный сотрудник, boris240@cs.niisi.ras.ru*

*(Федеральный научный центр Научно-исследовательский институт системных исследований  
РАН (ФНЦ НИИСИ РАН), Нахимовский просп., 36, корп. 1, г. Москва, 117218, Россия)*

В статье рассматриваются особенности разработки высокопроизводительного микропроцессора для создания доверенных систем. Производительность микропроцессора определяется производительностью ядра или количеством одновременно выполняющихся операций и временем доступа к памяти. Возможность использования микропроцессора для создания доверенных систем основывается на использовании в его составе блоков и узлов собственной разработки.

Производительность ядра микропроцессора определяется в основном тремя характеристиками: тактовой частотой микропроцессора, частотой следования инструкций и количеством операций, выполняемых одной инструкцией. Для микропроцессора 1890ВМ118 эти характеристики в большинстве случаев оптимизируются по параметру соотношения производительности и потребляемой мощности. Повышение тактовой частоты достигается за счет использования заказного проектирования критичных для быстродействия блоков и оптимизации длины конвейера. Частота следования инструкций увеличивается путем использования таких аппаратных решений, как суперскалярное исполнение инструкций, предсказание переходов и предварительная подкачка данных в кэш-памяти. Реализация арифметического сопроцессора, ориентированного на задачи цифровой обработки сигналов, позволяет повысить число операций, выполняемых одной командой. Повышение производительности подсистемы памяти рассматривается в статье с точки зрения симметричного доступа к памяти для двухъядерного микропроцессора. Описан подход к реализации когерентности в кэш-памяти процессорных ядер.

Особое внимание уделено средствам повышения безопасности в микропроцессоре, которые предназначены для создания на его основе доверенных систем. Рассмотрены аппаратные решения для доверенной загрузки операционной системы и изолированного доступа к памяти. Для обеспечения доверенной загрузки предлагается использовать накрестальное постоянное запоминающее устройство и однократно программируемую память, содержащие безопасный начальный загрузчик, а также ключи для проверки подписей операционной системы. К средствам изолированного доступа относится рассмотренный в статье контроллер доступа к памяти, реализованный в микропроцессоре. Предложены перспективные направления повышения безопасности систем на кристалле для создания доверенных систем на основе микропроцессоров разработки ФНЦ НИИСИ РАН.

**Ключевые слова:** доверенные системы, система на кристалле, архитектура микропроцессора, сопроцессор.

Одной из основных проблем функционирования современных высокопроизводительных вычислительных систем является защита от несанкционированного доступа. Несанкционированный доступ к вычислительной системе возможен из-за наличия в микропроцессоре неописанных аппаратных средств – закладок, которые могут быть заложены как при проектировании, так и в процессе производства. К несанкционированному доступу могут привести отсутствие защиты программы загрузчика, а также различного рода атаки на ПО, не контролируемые аппаратурой. Растущие требования к увеличению производительности ведут к усложнению микропроцессоров, а в результате к многократному усложнению поиска возможных ошибок или закладок в чужом проекте. Решение проблемы – в использовании микропроцессоров и ПО собственной разработки со встроенными средствами повышения безопасности. Из-за сложности проектов возникают самые различные ошибки в современных микропроцессорах, включая микропроцессоры мировых лидеров. Собственное ПО и пол-

ное владение проектом микропроцессора в большинстве случаев позволяют парировать эти ошибки программными средствами.

Особенно остро вопрос безопасности стоит в банковской сфере, промышленности и на транспорте. Хакерские атаки могут привести к существенным денежным потерям, экологическим бедствиям или к прекращению штатного функционирования транспорта и промышленных предприятий. Для такого рода применений создаются доверенные вычислительные системы. В США критерий оценки доверенных компьютерных систем определен стандартом Министерства обороны США (*Department of Defense Trusted Computer System Evaluation Criteria, TCSEC, DoD 5200.28-STD, December 26, 1985*), более известным как «Оранжевая книга». Данный стандарт получил международное признание и оказал сильное влияние на последующие разработки в области информационной безопасности. Невозможно создать абсолютно безопасную от проникновения (доверенную) систему, поэтому в стандарте предложено оценивать

лишь ее степень доверия. Под доверенной системой понимается система, использующая аппаратные и программные средства для обеспечения одновременной обработки информации разных категорий секретности группой пользователей без нарушения прав доступа. В стандарте заложены уровень гарантированности, подотчетность, доверенная вычислительная база, монитор обращений, ядро безопасности, периметр безопасности. Безопасность и доверенность оцениваются по уровню управления доступом к информации и обеспечению конфиденциальности и целостности.

В России в настоящее время отсутствуют критерии оценки доверенных компьютерных систем. Доверенные вычислительные системы предполагают использование только отечественных компонентов, включая элементную базу и ПО. Постановлением правительства РФ от 09.08.2016 г. № 764 введены понятия интегральной схемы первого уровня, когда микросхема разработана и произведена налоговым резидентом стран-членов Евразийского экономического союза, и интегральной схемы второго уровня, когда проектирование и разработка, а также испытания интегральной схемы произведены этим же резидентом. При отсутствии других критериев оценки доверенности ее степень можно также определять по степени «отечественности».

Основным компонентом доверенной системы является микропроцессор. В ФНЦ НИИСИ РАН разработка микропроцессоров является одним из основных направлений, в институте разрабатываются и серийно выпускаются 64-разрядные микропроцессоры с архитектурой КОМДИВ64, подобной архитектуре MIPS64 [1].

Производительность микропроцессора определяется производительностью ядра или количеством одновременно выполняющихся операций и временем доступа к памяти. В настоящее время доступ к памяти для большинства задач становится узким местом и в наибольшей степени определяет производительность. Существуют два способа повышения производительности:

- архитектурные решения, позволяющие увеличить количество одновременно выполняемых инструкций и скрыть простои, связанные с промахами в кэш-памяти;
- оптимизация подсистемы памяти.

В свою очередь, для первого способа используются следующие пути увеличения производительности:

- внеочередное исполнение инструкций (*Out-of-Order*, *OoO*);
- мультитредовая технология (*Multithreading*);
- использование многоядерности.

В современных микропроцессорах развивается технология переупорядочения не зависящих друг от друга инструкций для максимального повышения эффективности распараллеливания. Такие про-

цессоры называются процессорами с внеочередным исполнением инструкций (*out-of-order processors*). Техника переупорядочения инструкций замечательна тем, что резко ослабляет негативные эффекты от медленной оперативной памяти и от наличия зависимых цепочек инструкций. Для таких процессоров рост производительности при наличии двух одновременно выполняемых инструкций составляет 1,8–1,9 раза, для трех – 2,5–2,8 раза. Однако такие процессоры существенно усложняются и резко возрастает потребление ядра.

Другим решением повышения производительности является использование мультитредовой технологии, аппаратно поддерживающей эффективное выполнение нескольких тредов (*thread*) для каждого ядра. Тред – это минимальный аппаратный функциональный блок микропроцессора, поддерживаемый операционной системой и использующий общие ресурсы с другими тредями (кэш-память, регистры).

Для встраиваемых применений более эффективным по показателю «производительность/потребляемая мощность» является увеличение числа ядер.

С точки зрения оптимизации подсистемы памяти используются многоуровневые кэш-памяти, симметричный доступ к общей памяти, механизмы предвыборки и многоуровневая буферизация [2].

В разрабатываемом высокопроизводительном микропроцессоре 1890BM118 для повышения производительности и возможности создания доверенных систем применена многоядерность и оптимизирована подсистема памяти.

Данный процессор является двухъядерным и создан по проектной норме 28 нм. Оптимизация архитектуры ядра микропроцессоров осуществлялась по параметру «производительность/потребляемая мощность».

Важнейшим параметром является производительность вычислительных систем. В работе [3]

производительность представлена как 
$$\frac{N_{\text{инст}}}{\text{Программа}}$$

$$\times \frac{N_{\text{цикл}}}{N_{\text{инст}}} \times \frac{T_{\text{прог}}}{N_{\text{цикл}}} = \frac{T_{\text{прог}}}{\text{Программа}} = \text{время работы CPU,}$$

где  $N_{\text{инст}}$  – число выполненных инструкций за время работы микропроцессора (CPU);  $N_{\text{цикл}}$  – число циклов микропроцессора при выполнении программы;  $T_{\text{прог}}$  – время выполнения программы.

Как видим, производительность зависит от трех характеристик – тактовой частоты, частоты следования инструкций и количества операций, выполняемых за одну инструкцию. Более того, время работы CPU в равной степени зависит от каждой из них: увеличение на 10 % одной дает общий прирост также на 10 %.

Частота микропроцессора зависит от технологии изготовления, количество инструкций за такт – от архитектуры микропроцессора, а количество

операций в инструкции – от системы команд и технологии компиляции. Каждый из этих параметров сложно изменить изолированно от остальных, поскольку основные технологии, определяющие каждую характеристику, взаимозависимы: длина конвейера влияет на частоту, количество инструкций, выполняемых за такт, влияет на длину конвейера. Рассмотрим увеличение значений всех трех компонент на примере микропроцессора 1890BM118.

**Тактовая частота.** Как было сказано выше, частота во многом определяется технологией производства. Расчет частоты функционирования микропроцессора 1890BM118 при нормальной температуре окружающей среды для технологии КМОП 28 нм процесс НРС+ показал возможность достижения частоты 1,5 ГГц. Однако для худшего случая (температура – 60 °C или +125 °C), минимального напряжения питания и худшего процесса без принятия специальных мер частота микропроцессора падает до 700 МГц. Для повышения частоты разработана методика, позволившая почти удвоить частоту. На первом шаге определяются наиболее критичные к задержкам блоки. В ядре микропроцессора это блок трансляции виртуальных адресов в физические (TLB), регистровый файл и кэш-память первого уровня. Соответственно блоки TLB, регистровые файлы и кэш-память первого уровня спроектированы полностью заказным образом. Общепризнанной методикой повышения частоты микропроцессора является удлинение конвейера. В простейших микроконтроллерах глубина конвейера составляла 1–3 стадии. Для некоторых современных микропроцессоров глубина конвейера достигает 40. Однако при прерываниях и переходах конвейер разрушается и производительность соответственно падает. Особенно это критично для встраиваемых применений. Именно поэтому в ядрах КОМДИВ64 используется минимально возможная глубина конвейера. Для рассматриваемого ядра 1890BM118 глубина конвейера составляет 7 стадий. Сбалансированный 7-стадийный конвейер позволяет достичь частоты функционирования ядра 1,5 ГГц.

С точки зрения подсистемы памяти на частоту микропроцессора оказывает влияние кэш-память второго уровня. Большие площади памяти приводят к заметным паразитным значениям и соответствующим задержкам. Оптимальным вариантом с точки зрения времени доступа и размера для 28 нм КМОП технологического процесса определен размер 512 Кб стандартной кэш-памяти второго уровня для 7-стадийного конвейера, не влияющий на быстродействие микропроцессора.

Следующей мерой повышения частоты микропроцессора является использование методики проектирования, рассмотренной в работах [4, 5]. Синтез микросхемы начинается с начального этапа разработки модели микросхемы, что позволяет на раннем этапе определять наиболее узкие места

проекта, выделять блоки разного быстродействия и оптимизировать модель микросхемы. Создаются специальные скрипты САПР для обнаружения наиболее критичных путей, разрабатываются дополнительные библиотечные элементы к существующей библиотеке, позволяющие увеличить быстродействие проекта. При необходимости осуществляется ручная расстановка и/или разводка наиболее критичных по быстродействию блоков в соответствии с разработанной методикой.

**Частота следования инструкций.** Микропроцессор 1890BM118 является суперскалярным RISC-микропроцессором. Суперскалярность – это способность выполнения нескольких инструкций за один такт. В микропроцессоре 1890BM118 возможно одновременное выполнение двух инструкций. Исследования показали, что дальнейшее увеличение числа одновременно выполняемых инструкций приводит к заметному усложнению и увеличению потребляемой мощности, в то же время производительность растет нелинейно.

Для двух одновременно выполняемых инструкций производительность ядра 1890BM118 растет в 1,2–1,5 раза в зависимости от задачи, для трех инструкций – в 1,5–1,8 раза. Более эффективным является создание многоядерной системы на кристалле.

Возможность одновременного выполнения инструкций во многом поддерживается механизмом предсказания переходов и спекулятивным выполнением инструкций [6]. Суть метода заключается в прогнозировании направления ветвления программы и начале выполнения предполагаемых операций до подтверждения условия перехода. Повышение эффективности данного метода актуально для любого современного RISC-микропроцессора. Это объясняется тем, что в обычных программах порядка 20 % инструкций являются командами ветвления, которые в простейших случаях реализации блока выборки инструкций приводят к остановке конвейера. Базовые подходы к механизму предсказания ветвления одинаковые для всех RISC-микропроцессоров – это статическое и динамическое предсказания ветвления. Точность предсказания переходов в современных процессорах превышает 90 %. В ранних микропроцессорах КОМДИВ64 использовалось статическое предсказание ветвления. Исследования показали, что оптимальным с точки зрения сложности (площади и потребляемой мощности) и достижения вероятности предсказания (порядка 90 %) является динамическое предсказание с ограничением очереди предсказаний четырьмя записями и алгоритмом выбора предсказания по двухуровневой схеме. Этот вариант реализован в ядре 1890BM118.

**Количество операций, выполняемых одной инструкцией.** Существенное повышение производительности выделенных задач достигается за счет создания специализированных сопроцессоров.

В микропроцессор 1890BM118 встроен сопроцессор CPV, ориентированный на обработку изображений и сигналов [7]. Вычислительный блок векторного сопроцессора позволяет выполнять операции одинарной и двойной точности над векторами шириной 128 разрядов. Пиковая производительность достигается при использовании команд комплексного умножения с прибавлением и вычитанием третьего операнда («бабочка Фурье») и составляет за такт 10 вещественных операций двойной точности или 20 вещественных операций одинарной точности. Сопроцессор CPV может программироваться на языках Си и ассемблера.

Проведенные исследования производительности прототипа микропроцессора 1890BM118 показали, что по производительности он приближается к современному Out-of-Order микропроцессору XLP316 фирмы Broadcom.

Для оценки использовались тесты CoreMark и SciMark2. Тест CoreMark был разработан консорциумом EEMBC (Embedded Microprocessor Benchmark Consortium). Тест CoreMark вычисляет производительность ядра микропроцессора, выполняя относительно простой код, однако этот код использует общие практически для всех приложений структуры данных и алгоритмы. Для теста выбрана такая реализация, чтобы все вычисления (и их итоговые значения) производились непосредственно в процессе выполнения программы, что предотвращает исключение кода при оптимизации в процессе компиляции. Результаты работы всех тестов сверяются с эталонными.

В состав теста CoreMark входят различные вычислительные алгоритмы, характерные для микропроцессоров для встраиваемых систем, в том числе

- *операции с матрицами*, такие как умножение матрицы на константу, вектор или другую матрицу; также производится проверка данных в итоговой матрице;

- *сортировка списков*; применяется для определения того, как быстро процессор может получить данные из памяти для сканирования списка; если списки больше доступной кэш-памяти, то при их обработке проверяется и эффективность подсистемы кэш-памяти;

- *операции переключения конечного автомата*; конечный автомат тестирует входную строку, чтобы определить, являются ли входные данные числом, и принимает состояние invalid, если это не число; для проверки правильности операции в тесте CoreMark подсчитывается, сколько раз было пройдено каждое состояние;

- *подсчет контрольных сумм*; осуществляется подсчет 16-битного циклического избыточного кода (CRC) на основе данных, содержащихся в элементах списка.

Результат теста составил 3 309 итераций в секунду для микропроцессора XLP316 и 2 705 итераций в секунду для прототипа микропроцессора

1890BM118. Оба микропроцессора функционировали на частоте 1 ГГц.

Еще одним репрезентативным набором тестов производительности микропроцессоров является тест SciMark2. Это мощный однопоточный тест (интенсивные научные вычисления), показывающий производительность одного ядра микропроцессора в мегафлопсах (Mflops). Тесты написаны на языке C и входят в состав Phoronix Test Suite.

Тесты производительности SciMark2 разработаны Национальным институтом стандартизации США (NIST) для измерения производительности математических вычислений и являются общепризнанным пакетом.

Пакет состоит из пяти вычислительных ядер.

- *Fast Fourier Transform (FFT)*, выполняющее одномерное прямое преобразование Фурье для  $2^{12}$  ( $2^{20}$ ) комплексных чисел. Это ядро использует комплексную арифметику, перестановки битов, неконстантные ссылки на память и тригонометрические функции. Первая часть ядра производит обращение битов (без измерения), вторая – требуемые  $N \log(N)$  вычислений.

- *Jacobi Successive Over-relaxation (SOR)*. Это метод последовательной сверхрелаксации Якоби, реализующий типичные паттерны доступа к памяти, свойственные для конечно-разностных методов, таких как решение двухмерного уравнения Лапласа с ограничениями Дирихле. Ядро работает на сетках размером  $100 \times 100$  и  $1\,000 \times 1\,000$  соответственно.

- *Monte Carlo integration*, вычисляющее приближенное значение числа Пи оценкой площади четверти круга методом Монте-Карло. Алгоритм выбирает случайные точки на квадрате и проверяет их принадлежность кругу радиуса 1. Ядро использует генераторы случайных чисел, синхронизированные методы, вкладывание функций.

- *Sparse matrix multiply* – умножение разреженных матриц, использующее неструктурированные матрицы в сжатом формате. Это ядро использует косвенную адресацию и нерегулярный доступ к памяти.

- *Dense LU matrix factorization*, находящее LU-разложение плотной матрицы. Алгоритм использует линейные ядра (BLAS) и операции над плотными матрицами.

Численные значения производительности, полученные после запуска пяти вычислительных тестов, усредняются, в итоге выдается одна оценка производительности, которая называется Composite Score.

Результат теста для микропроцессора XLP316 составил 181,7 Мфлопса, для прототипа микропроцессора 1890BM118 – 149,5 Мфлопса. Сравнение производилось при частоте функционирования 1 ГГц.

Микропроцессор 1890BM118 является системой на кристалле. Интеграция функций микропро-



цессора и системного контроллера на одном кристалле и создание систем на кристалле приводят не только к повышению надежности системы и уменьшению ее габаритов в силу сокращения числа компонент, но и к увеличению производительности всей системы.

Следующим фактором повышения производительности вычислительных систем является организация подсистемы памяти в целом. Особенно важна задача повышения скорости обмена данными с ОЗУ для задач с большими объемами данных, не позволяющими эффективно использовать кэш-память. В таких случаях ускорение возможно прежде всего за счет введения предвыборок, увеличения частоты и ширины памяти. В двухъядерном микропроцессоре 1890BM118 реализуется 64-разрядный канал памяти с частотой до 1000/2000 МГц. Два процессорных ядра и общий системный контроллер объединены по схеме с симметричным доступом к общей памяти ОЗУ (архитектура SMP). Отключение средств когерентного доступа к памяти дает возможность использовать двухъядерный микропроцессор в режиме асимметричного доступа к памяти (архитектура AMP) с разделением ее между ядрами. Это позволяет расширить область применения микропроцессора: при некотором снижении надежности добиться максимального увеличения производительности или повышения надежности вычислительной системы в целом за счет аппаратного разделения памяти.

Архитектура SMP предъявляет ряд требований к разрабатываемой системе, направленных на обеспечение целостности данных при симметричном доступе к памяти в многоядерной системе и взаимодействие процессорных ядер. К таким требованиям можно отнести когерентность кэш-памяти процессорных ядер, поддержку атомарных операций чтение-модификация-запись, межпроцессорные прерывания.

В разрабатываемой системе на кристалле общий системный контроллер обеспечивает доступ к ОЗУ для каждого микропроцессорного ядра, а также подключение периферийных устройств. Для обмена данными между ОЗУ, микропроцессорными ядрами и контроллером периферийных устройств используется шина AMBA AXI.

Проблема обеспечения целостности данных при обмене данными внутри системы на кристалле может возникнуть в том случае, когда один из участников обмена данными получает доступ к не самой последней копии данных. Причиной этому могут стать кэшируемые обращения для записи и чтения от обоих ядер к данным в ОЗУ по одинаковому адресу (к одной строке кэш-памяти), а также обращение от периферийного устройства к области памяти, копия которой уже содержится в кэш-памяти одного или обоих ядер.

Таким образом, разработку протокола когерентности для системы на кристалле можно условно

разделить на две задачи – реализация средств обеспечения когерентного обмена данными между ядрами и реализация когерентного ввода/вывода для периферийных устройств.

В качестве протокола когерентности для межпроцессорного обмена широко используется протокол MOESI. По сравнению с базовым протоколом MSI он позволяет уменьшить число обращений в кэш-память другого ядра. Кроме того, передача модифицированной строки кэш-памяти может осуществляться напрямую между ядрами, а не путем записи/чтения строки в ОЗУ. Обеспечение когерентного ввода/вывода для периферийных устройств достигается за счет проверки наличия строки с адресом, по которому происходит обращение от периферийного устройства, в кэш-памяти одного или обоих ядер. Запрос на прослушивание (snooping) кэш-памяти содержит физический адрес обращения, признак записи/чтения и признак того, что объем записываемых данных меньше ширины строки кэш-памяти. Ответ на прослушивание формируется по окончании соответствующего действия. При промахе запроса на прослушивание в кэш ответ периферийному устройству выдается без выполнения дополнительных действий.

Существенным недостатком прослушивания ядер при обращении в ОЗУ от периферийного устройства является потеря производительности как для ядер – из-за постоянных обращений в кэш-память, так и для обмена данными между устройством и ОЗУ из-за ожидания ответа из кэш-памяти для каждого сегмента данных размером со строку кэш-памяти.

Решением этой проблемы может стать наличие в системном контроллере блока, называемого директорией и содержащего информацию о наличии в кэш-памяти каждого ядра строки с адресом обращения. Любой запрос строки от периферийного устройства на запись или чтение поступает в директорию. Физический адрес запроса проверяется на совпадение с адресами, хранящимися в директории. При выявлении совпадения запрос на прослушивание отправляется в то ядро, в кэш-памяти которого обнаружен адрес запроса от периферийного устройства. Если совпадение не обнаружено, запрос на прослушивание в кэш-память ядер не отправляется, а периферийное устройство получает разрешение на продолжение передачи данных. Такая директория снижает нагрузку на кэш-память ядра, поскольку в нее попадают только запросы на прослушивание по адресам, наличие которых подтверждено директорией. Кроме того, ответ об отсутствии строки с адресом обращения от директории приходит в периферийное устройство быстрее, чем от кэш-памяти, сокращая таким образом время, затрачиваемое на прослушивание.

Важным свойством разрабатываемых микропроцессоров является возможность создания доверенных систем на их основе.

Отечественные микропроцессоры для ответственного применения должны поддерживать организацию доверенной среды исполнения, гарантирующей заложенную устойчивость к попыткам взлома всей системы. Если принять критерий оценки эффективности такой защищенности системы за время и количество условных единиц, которые готов потратить злоумышленник на преодоление защиты, то ресурс системы можно считать условно надежным при условии, что система готова противостоять взлому достаточное количество времени, в течение которого злоумышленник откажется от попыток взлома. При проектировании микропроцессора 1890BM118 особое внимание было уделено повышению безопасности, в основу чего легли архитектурные решения и принимаемые еще на стадии проектирования, и используемые в ПО. На основе микропроцессора 1890BM118 становится возможным комплексный подход к построению системы, устойчивой к взлому, позволяющей развернуть программно-аппаратный комплекс доверенной среды исполнения, эффективность организации которой ограничивается лишь расчетным списком уязвимостей и потенциальных атак, противостоять которым способна система. В микропроцессорах реализованы такие аппаратные решения для повышения безопасности систем, как организация доверенной загрузки операционной системы, использование второго ядра для контроля безопасности, политика изолированного доступа к памяти. Рассмотрим эти аппаратные решения.

Доверенная загрузка микропроцессора становится возможной при использовании накристалльного ПЗУ, в которое разработчиком аппаратуры записываются флаги конфигурации доверенной загрузки и хэш-ключи проверки загружаемой операционной системы. В накристалльное ПЗУ на этапе производства микросхемы записывается загрузчик первого уровня, который проверяет цифровую подпись загрузчика второго уровня, находящегося в основной памяти, и дает разрешение на его загрузку. Схема загрузки операционной системы при этом примерно следующая.

После включения питания или по сигналу сброса (Reset) микросхемы микропроцессор начинает исполнять инструкции по адресам, отображаемым на накристалльное ПЗУ.

Из накристалльного ПЗУ запускается загрузчик первого уровня, который копирует необходимый код программы в ОЗУ микропроцессора.

Код, скопированный в ОЗУ, выполняет проверку цифровой подписи загрузчика второго уровня, в случае успешной проверки загружается операционная система.

Накристалльное ПЗУ (ROM) представляет собой классическое энергонезависимое ПЗУ, которое создается на производстве в процессе выращивания основного кристалла. Данные в ПЗУ также записы-

ваются на стадии производства и после этого не могут быть перезаписаны. В процессе работы системы на кристалле ПЗУ доступно только для чтения данных. Ввиду того, что данные в память записываются при производстве, несанкционированная запись в нее невозможна. В отличие от отдельных микросхем ПЗУ, которые интегрируются на системную плату отдельно, невозможен и подлог накристалльного ПЗУ, что гарантирует достоверность содержащихся в ПЗУ данных.

В совокупности с накристалльным ПЗУ для доверительной загрузки операционной системы может использоваться однократно программируемая память XPM (Extra Permanent Memory).

Память XPM, построенная в базе стандартной КМОП-логики, является однократно записываемой. Незапрограммированная память содержит в себе логические «0». В основе физики процесса записи в ячейку памяти логической «1» лежит принцип пробоя подзатворного окисла транзистора высоким напряжением, получаемым схемой накачки напряжения или отдельной линией, выведенной на высокое напряжение. Запатентованный NVM (non-volatile memory) механизм гарантирует полную защищенность данных. В связи с этим данная память может находить применение в области защиты информации. В системах на кристалле она может использоваться для хранения важных данных, прошивок и т.п.

Второе ядро микропроцессора может использоваться как монитор безопасности (гипервизор), осуществлять перехват подозрительной активности на основе метрик (временных, поведенческих), обнаружение недопустимой конфигурации устройств, прочистку кэш-памяти и сохранение точек восстановления после сбоя.

Изолированный доступ к памяти для периферийных устройств гибко настраивается с помощью блока защиты совместного доступа к ней. Данный блок (MPU) позволяет на аппаратном уровне реализовать политику изолированного доступа к памяти системы. Использование MPU в микропроцессоре 1890BM118 делает их устойчивыми, надежными и в некоторых случаях более безопасными, препятствуя задачам приложения получить нежелательный доступ к стеку или области памяти, или повредить стек и память данных, используемые другими задачами. MPU позволяет избежать распространенных атак, направленных на порчу конфигурации или критических данных, принадлежащих другому процессу.

Блок имеет следующие возможности:

- определение границ окна доступа устройства к системной памяти;
- определение типа операции доступа к окну, чтение и/или запись;
- регистры для контроля ошибок доступа устройств к памяти;
- формирование прерывания при ошибке доступа к окну.

За каждым «мастер»-устройством закреплено свое окно доступа в системную память, окно доступно по чтению и/или по записи. Если в конфигурационном регистре окна биты доступа не установлены, адресное окно недоступно. При обращении к памяти проверяются начальный и конечный адреса окна, тип операции (чтение/запись). Если адрес обращения не принадлежит границам окна или тип операции запрещен, выставляется флаг ошибки окна, адрес и атрибуты обращения записываются в управляющие регистры и вызывается прерывание. После конфигурации управляющих регистров MPU возможна однократная запись бита, запрещающего последующие изменения конфигурационных регистров MPU.

Можно выделить два основных перспективных направления повышения безопасности систем на кристалле на базе отечественных микропроцессоров: на основе средств виртуализации и на основе доверенной среды исполнения. Концепция безопасности на основе виртуализированной системы базируется на основе минимизации потенциально уязвимых участков кода путем изоляции недоверенной операционной системы в виртуальной машине с помощью программно-аппаратных средств. Кроме свойств изолированности доверенной среды, виртуализация позволяет осуществлять клонирование, резервное копирование и откат виртуальной машины, это повышает надежность системы, что также актуально в серверных приложениях. Оба направления взаимосвязаны и находятся в плоскости разработки аппаратно-программных средств. Разработка двух направлений имеет пересекающиеся части в области аппаратных средств защиты памяти и ядра микропроцессора, так, большинство современных архитектур микропроцессоров поддерживают режим виртуализации операционной системы. Аппаратная поддержка ин-

фраструктуры доверенного режима исполнения приложений без поддержки виртуализации позволяет относительно упростить архитектуру защищенной системы, однако не избавляет от программно-аппаратной части, что в универсальном случае представляет собой монитор событий, реализованный на основном или отдельном микропроцессорном ядре.

Все описанные аппаратные решения для повышения производительности и безопасности применяются в блоках микропроцессоров, разрабатываемых в ФНЦ НИИСИ РАН, что позволяет строить на их основе доверенные высокопроизводительные системы.

### Литература

1. Бобков С.Г. Импортзамещение элементной базы вычислительных систем // Вестн. РАН. 2014. Т. 84. № 11. С. 1010–1016.
2. Аряшев С.И., Бычков К.С. Оптимизация механизма предварительного считывания в кэш-памяти второго уровня. Проблемы разработки перспективных микро- и наноэлектронных систем // Сб. науч. тр. М.: Изд-во ИППМ РАН, 2016. Ч. II. С. 274–279.
3. Patterson D.A., Hennessy J.L. Computer architecture: a quantitative approach. 4th ed. The Morgan Kaufmann Series in Computer Architecture and Design. Elsevier Sc. Kindle Ed., 2011, 676 p.
4. Бобков С.Г. Высокопроизводительные вычислительные системы. М.: Изд-во НИИСИ РАН, 2014. 299 с.
5. Бобков С.Г., Евлампиев Б.Е. Анализ САПР микроэлектроники и выделение методов достижения предельного быстродействия для СБИС со сложной структурой класса микросхем графического контроллера // Инструментальные средства программирования: сб. стат.; [под ред. акад. РАН В.Б. Бетелина]. М.: Изд-во НИИСИ РАН, 2006. С. 112–128.
6. Барских М.Е., Бобков С.Г. Исследование влияния динамического предсказания ветвлений на производительность перспективных микропроцессоров НИИСИ РАН // Информационные технологии. 2015. № 10. С. 736–742.
7. Бобков С.Г., Аряшев С.И., Барских М.Е., Зубковский П.С., Ивасюк Е.В. Высокопроизводительные расширения архитектуры универсальных микропроцессоров для ускорения инженерных расчетов // Информационные технологии. 2014. № 6. С. 27–37.

Software & Systems

DOI: 10.15827/0236-235X.119.345-352

Received 03.05.17

2017, vol. 30, no. 3, pp. 345–352

### HIGH-PERFORMANCE MICROPROCESSOR 1890BM118 FOR TRUSTED COMPUTING SYSTEMS

**S.I. Aryashev**<sup>1</sup>, Ph.D. (Engineering), Branch Manager, aserg@cs.niisi.ras.ru,

**S.G. Bobkov**<sup>1</sup>, Director, bobkov@cs.niisi.ras.ru

**P.S. Zubkovsky**<sup>1</sup>, Head of Department, zubkovsky@niisi.ras.ru

**S.A. Morev**<sup>1</sup>, Junior Researcher, morev@cs.niisi.ras.ru

**B.Yu. Rogatkin**<sup>1</sup>, Research Associate, boris240@cs.niisi.ras.ru

<sup>1</sup> Federal State Institution "Scientific Research Institute for System Analysis of the Russian Academy of Sciences" (SRISA RAS), Nakhimovsky Ave. 36/1, Moscow, 117218, Russian Federation

**Abstract.** This article considers the problems of developing a high-performance microprocessor for trusted computing systems.

Microprocessor performance is determined by core capacity or a number of stages, which may be in execution simultaneously, and memory access time. Microprocessor applicability to create trusted systems is based on using self-made blocks and nodes.

Microprocessor core performance is determined by three characteristics: clock frequency, operations per instruction, instruction rate. For 1890VM118 microprocessor these characteristics were optimized by performance/power parameter. Clock frequency increasing is achieved using custom development of timing critical blocks and pipeline length optimization. Hardware solutions, such as superscalar instruction execution, branch prediction and preliminary data load in cache memory, increase instruction rate. Implementation of the arithmetic co-processor focused on digital signal processing tasks allows increasing the number of operations per instruction. The paper considers increasing memory subsystem performance in terms of symmetric memory access for a dual-core microprocessor. It also describes the approach to implementing cache coherence in processor cores.

The authors pay special attention to increasing the security level in microprocessor for trusted computing systems. They consider hardware solutions for operating system trusted boot and isolated memory access. To ensure trusted boot the authors suggest using on-chip ROM and one-time-programmable memory containing a secured bootloader and the keys to verify the signatures of an operating system. Isolated access solutions include a memory access controller discussed in the article and implemented inside a microprocessor. The paper proposes prospective solutions for creating trusted systems based on microprocessors by NIISI RAS.

**Keywords:** trusted systems, System-on-a-Chip, microprocessor architecture, co-processor.

### References

1. Bobkov S.G. Import phaseout of computing system hardware components. *Vestn. RAN* [Herald of the Russian Academy of Sciences]. 2014, vol. 84, no. 11, pp. 1010–1016 (in Russ.).
2. Aryashev S.I., Bychkov K.S. Optimization of a pre-read mechanism in second-level cache. *Problemy razrabotki perspektivnykh mikro- i nanoelektronnykh sistem. Sb. nauch. tr.* [Design Problems of Advanced Micro and Nanoelectronic Systems. Collected Papers]. Moscow, IPPM RAN Publ., 2016, part II, pp. 274–279 (in Russ.).
3. Patterson D.A., Hennessy J.L. *Computer architecture: a quantitative approach*. 4th ed. The Morgan Kaufmann Series in Computer Architecture and Design. Elsevier Sc. Kindle Ed., 2011, 676 p.
4. Bobkov S.G. *Vysokoproizvoditelnye vychislitelnye sistemy* [High-performance Computing Systems]. Moscow, NIISI RAN Publ., 2014, 299 p.
5. Bobkov S.G., Evlampiev B.E. Microelectronics CAD analysis and selecting methods of maximum highspeed response for SBIS with a complex structure of a graphics controller chip class. *Instrumentalnye sredstva programmirovaniya: sb. stat.* [Programming Tools. Collected Papers]. V.B. Betelin (Ed.). Moscow, NIISI RAN Publ., 2006, pp. 112–128 (in Russ.).
6. Barskikh M.E., Bobkov S.G. Investigation of the influence of dynamic branch prediction to perspective microprocessors performance from NIISI RAS. *Informatsionnye tekhnologii* [Information Technologies]. 2015, no. 10, pp. 736–742 (in Russ.).
7. Bobkov S.G., Aryashev S.I., Barskikh M.E., Zubkovsky P.S., Ivasyuk E.V. High-performance Extensions of Microprocessor Architecture for Speeding-up of Scientific and Engineering Calculations. *Informatsionnye tekhnologii* [Information Technologies]. 2014, no. 6, pp. 27–37 (in Russ.).

### Примеры библиографического описания статьи

1. Аряшев С.И., Бобков С.Г., Зубковский П.С., Морев С.А., Рогаткин Б.Ю. Высокопроизводительный микропроцессор 1890BM118 с архитектурой КОМДИВ для создания доверенных систем // Программные продукты и системы. 2017. Т. 30. № 3. С. 345–352. DOI: 10.15827/0236-235X.119.345-352.
2. Aryashev S.I., Bobkov S.G., Zubkovsky P.S., Morev S.A., Rogatkin B.Yu. High-performance microprocessor 1890BM118 for trusted computing systems. *Programmnye produkty i sistemy* [Software & Systems]. 2017, vol. 30, no. 3, pp. 345–352 (in Russ.). DOI: 10.15827/0236-235X.119.345-352.

УДК 004.921

DOI: 10.15827/0236-235X.119.353-358

Дата подачи статьи: 21.03.17

2017. Т. 30. № 3. С. 353–358

## **ВОСПРОИЗВЕДЕНИЕ ВИДЕОДАННЫХ ВЫСОКОЙ ЧЕТКОСТИ В ВИРТУАЛЬНОЙ ТРЕХМЕРНОЙ СРЕДЕ ИМИТАЦИОННО-ТРЕНАЖЕРНЫХ СИСТЕМ**

*А.М. Гуацингов, научный сотрудник, algts@inbox.ru;*

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

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

В статье приводятся разработанные методы воспроизведения видеоматериалов высокой четкости в подсистеме визуализации тренажерно-обучающих систем.

Тренажерно-обучающие системы, как правило, содержат значительное количество разнородных информационных ресурсов. Особый интерес представляют такие виды аудиовизуальной учебной информации, как динамические графики развития процессов, иллюстративные материалы изучаемых объектов, трехмерные модели объектов и их частей, результаты работы моделирующих комплексов в форме видеообразов с сохранением управляемости приложения, видеоматериалы. Подсистема визуализации обеспечивает отображение результатов моделирования внешней среды и объекта управления с помощью устройств отображения информации. Отображение видеоматериалов в виртуальной трехмерной сцене является одним из требований к тренажерно-обучающим системам.

Воспроизведение видеоматериалов внутри виртуальной трехмерной сцены является сложной задачей, так как необходимо учитывать такие факторы, как производительность подсистемы визуализации и производительность видеокарты. Подсистема визуализации должна обеспечивать отображение трехмерной сцены с приемлемой частотой кадров (не менее 25 кадров в секунду) и при этом быть способной реагировать на внешние воздействия, в том числе на изменения параметров трехмерной сцены или загрузку дополнительных объектов.

Для отображения нескольких видеоматериалов высокой четкости в виртуальной трехмерной сцене авторами была разработана и реализована архитектура декодера кадров видео. Архитектура включает декодер видео, в котором происходит декодирование видео- и аудиопакетов; подсистему воспроизведения декодированного звука; управляющую структуру, необходимую для запуска видео, паузы воспроизведения, выставления громкости воспроизводимого видео и т.д.; интерфейс взаимодействия с движком, необходимый для обновления видеокадров.

**Ключевые слова:** *тренажерно-обучающие системы, система визуализации, видеоматериалы, тренажер, рендеринг, видео, декодер.*

Увеличение числа технических систем и повышение их сложности приводят к необходимости создания новых видов технических средств обучения. Одним из таких средств являются *тренажерно-обучающие системы* (ТОС). Подобные системы незаменимы в тех отраслях, где ошибки при обучении на реальных объектах могут привести к негативным последствиям, а их устранение – к значительным финансовым затратам: в военном деле, медицине, атомной энергетике, авиации и космосе. Из этого следует актуальность задачи построения современных ТОС.

Под ТОС оператора *сложной технической системы* (СТС) будем понимать техническое средство для подготовки операторов в едином информационном окружении, отвечающее требованиям методик подготовки, создающее условия для получения знаний, навыков и умений, реализующее модель таких систем и обеспечивающее контроль над действиями обучаемого, а также для проведения исследований [1, 2].

Как правило, ТОС содержат значительное количество информационных ресурсов. Особый интерес представляют такие виды аудиовизуальной учебной информации, как динамические графики развития процессов, иллюстративные материалы изучаемых объектов, трехмерные модели объектов и их частей, результаты работы моделирую-

щих комплексов в форме видеообразов с сохранением управляемости приложения, видеоматериалы [3].

Подсистема визуализации обеспечивает отображение результатов моделирования внешней среды и объекта управления с помощью устройств отображения информации. Отображение разнородных видеоматериалов в виртуальной трехмерной сцене – одно из требований к ТОС.

Воспроизведение видеоматериалов внутри виртуальной трехмерной сцены является сложной задачей, так как необходимо учитывать такие факторы, как производительность подсистемы визуализации и производительность видеокарты. Подсистема визуализации должна обеспечивать отображение трехмерной сцены с приемлемой частотой кадров (не менее 25 кадров в секунду) и при этом быть способной реагировать на внешние воздействия, в том числе на изменения параметров трехмерной сцены или загрузку дополнительных объектов. Для отображения нескольких видеоматериалов высокой четкости в виртуальной трехмерной сцене авторами была разработана и реализована архитектура декодера кадров видео, а для обеспечения требуемой производительности подсистемы визуализации при отображении видеоматериалов высокой четкости – методы передачи данных в видеопамять.

### Декодер видеоданных

Структура видеофайла – контейнер с общей информацией о его содержимом (количество видео- и аудиопотоков, наличие субтитров, информация о разделах, метаданные – название, год создания, автор и т.д.) и различных потоках информации (видео, аудио, субтитры и т.д.). Некоторые контейнеры допускают использование только определенных кодеров для представления информации. Наиболее популярными являются AVI, OGV, MP4, MKV, MOV, WMV.

В большинстве контейнеров данные представляются в виде пакетов (в некоторых форматах – куски (chunks) или сегменты (segments)). Пакеты хранят закодированную информацию определенного типа – видео, аудио или субтитры и т.д. В большинстве видеофайлов количество аудиопакетов значительно меньше количества видеопакетов – приблизительно 75 % видеопакетов и 25 % аудиопакетов. В зависимости от типа контейнера и используемого кодера расположение пакетов внутри контейнера может быть разным. Зачастую пакеты расположены в виде последовательности очередей пакетов разного типа, например, около 20 видеопакетов, расположенных подряд, затем несколько аудиопакетов, после которых снова подряд расположены около 20 видеопакетов.

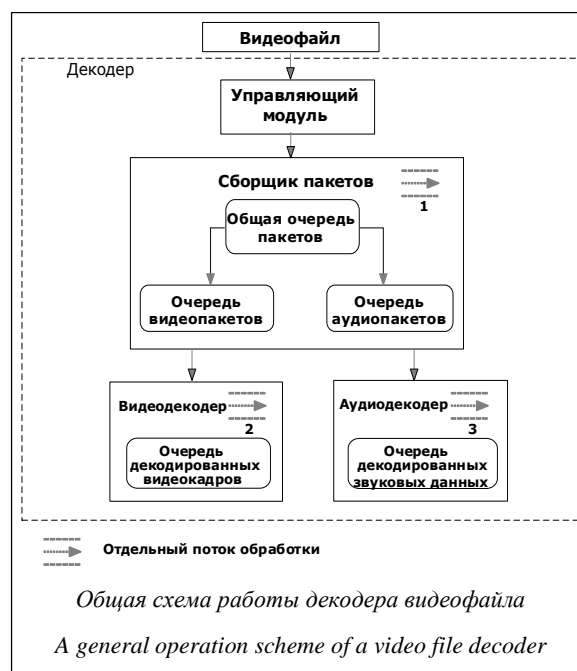
Пакеты хранят в себе определенный сегмент аудио- или видеoinформации. В большинстве случаев в видеопакете хранится целый видеокادر, но в некоторых форматах один видеокادر может состоять из нескольких пакетов. Это обусловлено тем, как кодер сохраняет информацию о видеокadre. Например, в формате MPEG 2 присутствуют три типа кадров: независимо сжатые кадры (I-кадры), кадры, сжатые с использованием предсказания движения в одном направлении (Р-кадры), и кадры, сжатые с использованием предсказания движения в двух направлениях (В-кадры). Соответствующие группы кадров от одного I-кадра до другого образуют GOP (group of pictures) группу кадров [4].

В пакетах, кроме закодированных данных и информации о размере и типе данных данного пакета, присутствует информация о времени, когда необходимо декодировать данный пакет (последовательности декодирования пакетов), и времени, когда декодированная информация должна быть отображена или воспроизведена.

Аудиоинформация видеофайла может быть закодирована в различных форматах сжатия, таких как MP3, AAC, AC3, WMA и т.д. Для корректного воспроизведения аудиоинформации необходимо знать количество каналов (моно-, стерео-, 5.1, 7.1 и т.д.) и частоту дискретизации. После декодирования звукового пакета декодированные данные представляются в формате PCM (pulse code modulation, импульсно-кодовая модуляция) и передаются звуковой карте для воспроизведения звука.

Разработанная архитектура графической подсистемы позволяет декодировать и отображать одновременно несколько видео высокой четкости в трехмерной сцене. Состав архитектуры: декодер видео, в котором происходит декодирование видео- и аудиопакетов; подсистема воспроизведения декодированного звука; управляющая структура, необходимая для запуска видео, паузы воспроизведения, выставления громкости воспроизводимого видео и т.д.; интерфейс взаимодействия с движком, необходимый для обновления видеокadres; интерфейс управления графической подсистемой.

Общая схема работы декодера представлена на рисунке.



Декодер состоит из четырех независимых частей:

- блок управления, осуществляющий запуск, остановку, а также обработку ошибок в случае их возникновения;
- блок получения пакетов из файла или внешнего источника, их демультимплексирование (при необходимости) и сортировка видео- и аудиопакетов по соответствующим очередям;
- блок декодирования видеопакетов;
- блок декодирования аудиопакетов.

Для каждого отображаемого видеоматериала создается свой экземпляр декодера. Во время инициализации декодера считываются все необходимые параметры о видеофайле, такие как кодеки, используемые для сжатия видео- и аудиоинформации, количество звуковых и видеопотоков в файле, ширина и высота (разрешение) видео, количество каналов звука и т.д.

Размещение пакетов в файле может различаться в зависимости от формата видеофайла, следовательно, сложно предсказать, в каком порядке будут



следовать видео- и аудиопакеты. Это может привести к ситуации, когда очередь видеопакетов заполнена, а очередь аудиопакетов пуста. Узнать тип следующего пакета невозможно до его чтения. Если следующий видеопакет окажется закодированным видеокадром, необходимо либо поместить этот пакет в очередь видеопакетов, то есть превысить заданные рамки очереди, либо удалить его, что приведет к артефактам на конечном изображении или пропущенным кадрам и рывкам воспроизводимого видео. Для решения этой проблемы была создана третья очередь пакетов, в которую помещаются пакеты различного типа из видеофайла, а затем сортируются по типу пакета (аудио- или видеопакет) в одну из очередей. Сортировка осуществляется по методу FIFO («первый вошел, первый вышел»).

Видеодекoder проверяет наличие пакетов в очереди видеопакетов и при обнаружении забирает пакет из начала очереди. Процесс декодирования видеопакета состоит из двух этапов – декодирование пакета и преобразование декодированного кадра в необходимую цветовую модель. После декодирования и преобразования в другую цветовую модель кадр помещается в очередь декодированных кадров. Аналогично аудиодекoder проверяет наличие пакетов в очереди и при обнаружении декодирует первый пакет из очереди. После декодирования звуковые данные в формате PCM помещаются в очередь декодированных звуковых данных.

Так как подсистема визуализации принимает изображения только в цветовой модели BGRA, после декодирования видеокадр будет преобразован в эту цветовую модель. Цветовая модель BGRA выбрана неслучайно: большинство форматов изображений и большая часть оборудования хранят изображения в формате BGRA. При использовании других цветковых моделей на центральном процессоре осуществляется преобразование используемой цветовой модели в модель BGRA. Так как любое лишнее преобразование снижает производительность графической подсистемы, использование цветовой модели BGRA является оптимальным.

Процесс декодирования ресурсоемкий. Например, декодирование одного видеопакета с разрешением 3 860×2 160 (4K) на процессоре Intel Core i7 3 ГГц занимает от 10 до 20 мс без использования специализированных аппаратных средств, таких как Intel QuickSync [5]. Декодирование пакета звуковых данных занимает значительно меньше времени (0,5–2 мс), но, если одновременно декодируется звук для нескольких воспроизводимых видео, это может серьезно снизить производительность. Скорость получения пакетов из видеофайла ограничивается производительностью системы хранения данных (СХД), что, в свою очередь, влияет на производительность подсистемы визуализации. Задержки происходят, если в момент считывания пакетов из видеофайла СХД обрабатывала другие

команды либо файл являлся фрагментированным, что заставляет СХД искать фрагменты данного файла. Для преодоления проблем с производительностью была использована многопоточность.

Все потоки выполняются в адресном пространстве процесса. Выполняющийся процесс имеет как минимум один (главный) поток. Применение многопоточности наиболее эффективно при использовании многоядерных процессоров. На одноядерных процессорах многопоточное приложение также будет работать, но повышение производительности будет незначительным; возможно даже снижение производительности, если разные потоки одного процесса выполняют ресурсоемкие операции [6].

Хотя многопоточность способна существенно повысить производительность приложения, применять ее следует с осторожностью. Внедрение многопоточности существенно повышает сложность самого приложения и его отладки, а при некорректном использовании может привести к таким проблемам, как «состояние гонки» (когда работа системы зависит от того, в каком порядке выполняются части кода) и взаимная блокировка потоков (несколько потоков находятся в состоянии бесконечного ожидания ресурсов, занятых самими этими потоками), а также к трудностям при обработке ошибок (исключений) внутри приложения.

На данный момент каждая независимая часть декодера, кроме блока управления, работает в собственном потоке. Многопоточность позволяет значительно повысить производительность подсистемы декодирования при использовании многоядерного процессора. Однако данный подход имеет и свои недостатки. При воспроизведении значительного количества видеоматериалов одновременно будет создано большое количество потоков, превышающее количество доступных процессорных ядер (например, для пяти одновременно воспроизводимых видео будет создано 15 потоков). В то же время, чем больше потоков работают, тем чаще операционная система должна выполнять контекстное переключение, в течение которого текущий обрабатываемый поток приостанавливает свою работу, сохраняется вся информация о выполняемых операциях и данных, а затем управление передается другому потоку или процессу. Каждое переключение выполняется в течение определенного времени, которое можно было бы использовать для декодирования. Соответственно, в определенный момент добавление новых потоков не повысит, а снизит общую производительность приложения [7]. Так как каждый из потоков выполняет ресурсоемкие операции, может возникнуть ситуация, когда для одного из воспроизводимых видеоматериалов требуется отобразить следующий видеокадр, который еще не был декодирован. Выходом из данной ситуации может быть использование пула потоков.

Пул потоков – это структура, содержащая в себе определенное количество потоков обработки, обеспечивающая прием задач обработки от управляющей программы и их распределение по потокам. Зачастую количество потоков в пуле задается на основе информации о количестве доступных ядер процессора для использования всех доступных вычислительных мощностей процессора и снижения накладных расходов на контекстное переключение. Наиболее совершенные реализации пулов потоков позволяют перераспределять поставленные в обработку задачи между потоками для равномерной загрузки процессорных ядер [8].

Применительно к декодированию данных использование пула потоков позволит более равномерно распределять нагрузку при одновременной обработке нескольких видеоматериалов и таким образом готовить декодированные данные для большего числа одновременно воспроизводимых видеоматериалов, чем при использовании отдельных потоков для каждой части декодера.

#### **Обновление видеоматериалов в подсистеме визуализации**

Загрузка новых кадров видеоматериала (видеотекстуры) в подсистеме визуализации осуществляется в основном потоке программы (потоке рендеринга). Каждый кадр подсистемы визуализации проверяется на необходимость обновления видеотекстуры в трехмерной сцене. Если обновление требуется, то видеокادر берется из очереди декодированных кадров и помещается в память видеокарты в качестве текстуры, которая может быть наложена на поверхность любого трехмерного объекта [9].

При запуске видео создается пустая текстура с размерами видеокадра, которая затем применяется к материалу трехмерного объекта (материал объекта определяет, какие текстуры и шейдеры использует данный трехмерный объект). Перед применением новой текстуры к материалу происходит резервное копирование старого материала для возможности возвращения к исходным текстурам объекта после окончания воспроизведения видео. Также для повышения производительности используются два так называемых пиксельных буфера. Они применяются для хранения пиксельных данных (текстуры) в видеопамати и позволяют значительно снизить временные затраты на обновление видеотекстуры в подсистеме визуализации. Количество используемых пиксельных буферов было подобрано таким образом, чтобы получить лучшее соотношение объема используемой оперативной памяти и величины задержки при отображении декодированных видеокладов.

При использовании стандартной функциональности подсистемы визуализации для обновления текстур в видеопамати сначала из видеокадра в

промежуточный массив копируются данные, а затем из промежуточного массива помещаются в текстуру. При обновлении текстуры в видеопамати сначала происходит выделение нового места под данные и только потом копирование данных из промежуточного массива в видеопамать. Все описанные выше действия выполняются центральным процессором, что значительно снижает производительность подсистемы визуализации. Например, при воспроизведении FullHD-видео (1 920×1 080) время копирования данных из видеокадра в промежуточный массив составляет приблизительно 3–4 мс, а обновление текстуры из промежуточного массива в видеопамати – от 5 до 8 мс.

Преимущество пиксельных буферов состоит в том, что затраты процессорного времени необходимы только на помещение информации в буфер, а обновление текстуры из буфера происходит практически мгновенно (около 0.1 мс) за счет того, что обработка перемещения данных из пиксельного буфера в текстуру возлагается на видеокарту [10]. Так как операция обновления текстуры из пиксельного буфера является асинхронной, выполняется она очень быстро, в то время как операция обновления информации в пиксельном буфере может вызвать синхронизацию процессора и видеокарты, что приведет к задержкам. Также видеокarte после обновления пиксельного буфера может понадобиться некоторое время для обработки полученных данных. Поэтому при обновлении текстуры сразу после обновления пиксельного буфера может возникнуть задержка. Если же производить обновление видеотекстуры перед загрузкой новых данных, это даст видеокarte больше времени на обработку данных пиксельного буфера.

При воспроизведении одного видео с разрешением вплоть до 1 920×1 080 (FullHD) в подсистеме визуализации либо нескольких видео с разрешением не выше 720×576 (576р, SD) время генерации одного кадра не превышает 40 мс (в зависимости от аппаратной платформы). Однако при одновременном воспроизведении нескольких видео высокой четкости наблюдается значительное увеличение времени генерации кадра подсистемой визуализации.

Во время работы над решением проблемы производительности при одновременном воспроизведении видеоматериалов высокой четкости было разработано несколько методов. В первом разработанном методе новые декодированные кадры копировались в пиксельные буферы в отдельном потоке, в то время как их передача в память видеокарты осуществлялась в основном потоке визуализации. Однако данный метод себя не оправдал, так как узким местом было не время копирования информации в оперативной памяти, а передача новых данных в память видеокарты.

Второй метод основывался на использовании единого пиксельного буфера, который содержал

бы кадры всех воспроизводимых видео. Преимущество данного метода в однократном обновлении пиксельного буфера для всех воспроизводимых видеоматериалов во время *итерации цикла* (ИЦ) обработки подсистемы визуализации. Однако у него обнаружались и существенные недостатки: необходимость сохранять предыдущие видеокдры, чтобы избежать рассинхронизации видео и аудио; сложность реализации динамического расширения и уменьшения пиксельного буфера при добавлении или удалении видео в ходе работы приложения. Данный метод также не смог предоставить необходимый уровень производительности при обновлении нескольких видеоматериалов.

Метод, позволивший получить требуемую производительность при одновременном воспроизведении видеоматериалов высокой четкости, базируется на построении очереди воспроизводимых видео на основе весовых коэффициентов.

Каждому видеоматериалу при воспроизведении присваиваются два вида весовых коэффициентов: коэффициент разрешающей способности видео и коэффициент выполняемой операции. Коэффициент разрешающей способности определяется один раз при запуске видеоматериала и основан на разрешении видео: при более низком разрешении значение коэффициента увеличивается, а при более высоком уменьшается. Это связано с тем, что видеоматериал с более высоким разрешением обрабатывается дольше, чем с более низким. На данный момент коэффициент может принимать три различных значения: 8 (высокий приоритет обновления), 4 (средний приоритет обновления), 0 (низкий приоритет). Например, видеоматериалу с разрешающей способностью до 720×576 будет присвоен высокий приоритет обновления, с разрешением до 1 280×720 – средний, с разрешением выше – низкий.

Коэффициент выполняемой операции определяется заново при каждом обновлении видеоматериалов в системе. Коэффициенту присваивается значение, определяющее операцию, которая будет выполнена на следующей итерации цикла обработки подсистемы визуализации. Могут быть присвоены следующие значения: 3 (обновление видеотекстуры), 2 (обновление видеотекстуры и пиксельного буфера), 1 (обновление пиксельного буфера), 0 (исключение видео из обработки). Значение 0 коэффициенту присваивается в случае, когда на следующей ИЦ подсистемы визуализации видеотекстура не требует обновления, а все пиксельные буферы заполнены.

Значения коэффициентов подобраны таким образом, чтобы преобладающим был коэффициент разрешающей способности видео, так как разрешение видеоматериала в большей степени влияет на производительность, чем порядок выполнения операций. Список обновляемых видеоматериалов определяется на каждой ИЦ подсистемы визуализации.

На основе коэффициента разрешающей способности видео и коэффициента выполняемой операции определяется позиция видео в списке обновления. При первом обновлении видеоматериалов в системе порядок обновления определяется только по коэффициенту разрешающей способности видео, а коэффициенту выполняемой операции присвоено высшее значение 3 (обновление видеотекстуры).

Разработанный метод отображения видеоматериалов высокой четкости позволяет одновременно воспроизводить несколько видео в высоком разрешении. На тестовом оборудовании среднего класса одновременно воспроизводилось до трех видеоматериалов в разрешении 4K или до 6 в разрешении FullHD с сохранением управляемости ТОС и частоты кадров подсистемы визуализации на уровне 30 кадров в секунду. При улучшении аппаратного обеспечения увеличивается количество одновременно воспроизводимых видеоматериалов или появляется возможность использования видео с большим разрешением (5K, 8K).

### Заключение

Задача разработки ТОС является комплексной и требует решения ряда сопутствующих задач, в частности, задач обработки и визуализации видеоданных. Преимуществом использования в ТОС таких информационных ресурсов, как видеоматериалы, является возможность визуализации как реальных, так и моделируемых процессов. За счет одновременного воспроизведения могут быть отображены параллельно выполняемые процессы или же один процесс с различных ракурсов. Разработанный метод воспроизведения видеоматериалов в виртуальной трехмерной сцене на основе весовых коэффициентов для формирования порядка обновления позволяет одновременно отображать несколько видео в разрешении 4K с сохранением требуемой производительности подсистемы визуализации. Планируется дальнейшее развитие разработанного метода, в частности, применение пула потоков и отказ от цветового преобразования на процессоре в пользу преобразования с использованием мощностей видеокарт.

*Работа выполняется при поддержке РФФИ, грант № 17-07-00169.*

### Литература

1. Мамросенко К.А., Решетников В.Н. Моделирование подстилающей поверхности в имитационных системах // Программные продукты и системы. 2015. № 4. С. 70–74.
2. Решетников В.Н. Тригонометрические воспоминания. Сер.: Космические телекоммуникации. СПб: Изд-во РИП СПб, 2015. 138 с.
3. Гиацинтов А.М., Мамросенко К.А. Метод рип-проекции в подсистеме визуализации тренажерно-обучающей системы // Программные продукты и системы. 2014. № 4. С. 31–37.
4. MPEG-2 video compression. URL: <http://www.bbc.co.uk/>

rd/publications/rdreport\_1996\_02 (дата обращения: 11.03.2017).

5. Intel® Quick Sync Video. URL: <http://www.intel.com/content/www/ru/ru/architecture-and-technology/quick-sync-video/quick-sync-video-general.html> (дата обращения: 11.03.2017).

6. Miloš Ljumović. C++ Multithreading Cookbook. Packt Publ. Ltd, 2014, 422 p.

7. Уильямс Э. Параллельное программирование на C++ в действии. М.: ДМК Пресс, 2012. 672 с.

8. Thread Pools. URL: <https://msdn.microsoft.com/ru-ru/>

library/windows/desktop/ms686760(v=vs.85).aspx (дата обращения: 11.03.2017).

9. Гиацинтов А.М. Отображение разнородных видеоматериалов на гранях трехмерных объектов в подсистеме визуализации тренажерных обучающих систем // Программные продукты и системы. 2012. № 3. С. 80–86.

10. Pixel Buffer Object – OpenGL Wiki. URL: [https://www.khronos.org/opengl/wiki/Pixel\\_Buffer\\_Object](https://www.khronos.org/opengl/wiki/Pixel_Buffer_Object) (дата обращения: 11.03.2017).

Software & Systems

DOI: 10.15827/0236-235X.119.353-358

Received 21.03.17

2017, vol. 30, no. 3, pp. 353–358

### HIGH RESOLUTION VIDEO PLAYBACK IN VIRTUAL 3D ENVIRONMENT IN TRAINING SIMULATION SYSTEMS

A.M. Giatsintov<sup>1</sup>, Research Associate, [giatsintov@niisi.ras.ru](mailto:giatsintov@niisi.ras.ru)

K.A. Mamrosenko<sup>1</sup>, Ph.D. (Engineering), Head of Department, [mamrosenko\\_k@niisi.ras.ru](mailto:mamrosenko_k@niisi.ras.ru)

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

**Abstract.** The article describes the developed methods of high-resolution video playback in a visualization subsystem of training simulation systems (TSS). Usually TSS have a number of heterogeneous data resources. The following resource types are of interest for personnel training: dynamic process development charts; graphic materials of studied objects; 3D models of objects and their parts; simulation results in a video form; video records. Visualization subsystem provides rendering simulation results of environment and dynamic objects, and displaying the rendered image using display devices. Displaying videos in virtual 3D scenes is one of the requirements for TSS.

Displaying videos inside a virtual 3D scene is a complex task, as many factors should be taken in account, such as performance of a video card and a visualization subsystem (renderer). A renderer should visualize a 3D scene with acceptable frame rate (not less than 25 frames per second) and respond to commands, such as a 3D scene transformation or new objects loading.

The authors have developed and implemented a new decoder architecture in order to display several high definition videos in a virtual 3D scene. The architecture includes the following components: a decoder that decompresses audio and video packets; an audio playback system; a control module that allows starting, stopping playback, setting volume, etc.; an interface that interacts with a visualization subsystem, which is required for updating video images.

**Keywords:** training simulation systems, visualization system, video, video material, trainer, rendering, decoder.

**Acknowledgements.** The work has been done with the support of the RFBR, grant no. 17-07-00169.

### References

1. 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.).
2. Reshetnikov V.N. *Trigonometricheskie vospominaniya. Ser. Kosmicheskie telekommunikatsii* [Trigonometric Memories. Series "Space Telecommunications"]. St. Petersburg, RIP SPB Publ., 2015, 138 p.
3. Giatsintov A.M., Mamrosenko K.A. Rear-projection method in visualization subsystem of training simulation system. *Programmnye produkty i sistemy* [Software & Systems]. 2014, no. 4, pp. 31–37 (in Russ.).
4. *MPEG-2 Video Compression*. Available at: [http://www.bbc.co.uk/rd/publications/rdreport\\_1996\\_02](http://www.bbc.co.uk/rd/publications/rdreport_1996_02) (accessed March 11, 2017).
5. Intel® Quick Sync Video. Available at: <http://www.intel.com/content/www/ru/ru/architecture-and-technology/quick-sync-video/quick-sync-video-general.html> (accessed March 11, 2017).
6. Miloš Ljumović. *C++ Multithreading Cookbook*. Packt Publ. Ltd, 2014, 422 p.
7. Williams A. *C++ Concurrency in Action: Practical Multithreading*. Manning Publ., 2012, 528 p. (Russ.ed.: Moscow, ДМК Пресс, 2012, 672 p.).
8. *Thread Pools*. Available at: [https://msdn.microsoft.com/ru-ru/library/windows/desktop/ms686760\(v=vs.85\).aspx](https://msdn.microsoft.com/ru-ru/library/windows/desktop/ms686760(v=vs.85).aspx) (accessed March 11, 2017).
9. Giatsintov A.M. Playback of heterogeneous videos on 3D object's edges in visualization subsystem of training simulation system. *Programmnye produkty i sistemy* [Software & Systems]. 2012, no. 3, pp. 80–86 (in Russ.).
10. *Pixel Buffer Object – OpenGL Wiki*. Available at: [https://www.khronos.org/opengl/wiki/Pixel\\_Buffer\\_Object](https://www.khronos.org/opengl/wiki/Pixel_Buffer_Object) (accessed March 11, 2017).

УДК 621.382.323

DOI: 10.15827/0236-235X.119.359-366

Дата подачи статьи: 04.05.17

2017. Т. 30. № 3. С. 359–366

## **МОДЕЛИРОВАНИЕ РАСПРЕДЕЛЕНИЯ ПОТЕНЦИАЛА В ДВУХЗАТВОРНОМ ПОЛЕВОМ НАНОТРАНЗИСТОРЕ СО СТРУКТУРОЙ КРЕМНИЙ НА ИЗОЛЯТОРЕ С АСИММЕТРИЧНЫМ ЗАТВОРОМ**

*Н.В. Масальский, к.ф.-м.н., зав. сектором, volkov@niisi.ras.ru*

*(Федеральный научный центр Научно-исследовательский институт системных исследований  
РАН (ФНЦ НИИСИ РАН), Нахимовский просп., 36, корп. 1, г. Москва, 117218, Россия)*

Рассматриваются аналитическая 2D-модель распределения потенциала и вытекающая из нее модель порогового напряжения для тонкопленочного двухзатворного полевого нанотранзистора со структурой кремний на изоляторе с фронтальным затвором из двух последовательно соединенных материалов с разной работой выхода. Исследованы поведение потенциала в транзисторных структурах в области длин затворов менее 50 нм, его зависимости от смещений на стоке для различных конфигураций фронтального затвора.

Полученные результаты однозначно показывают, что применение двух материалов с разной работой выхода во фронтальном затворе приводит к эффективному подавлению короткоканальных эффектов из-за ступенчатой функции в профиле потенциала на границе раздела двух материалов. Сдвиг положения поверхностной позиции минимума поверхностного потенциала незначителен с увеличением смещений стока. Пиковое электрическое поле у стока значительно снижается по сравнению с аналогичной транзисторной структурой, но с однородным фронтальным затвором. Одновременно прогнозируются достижение более высокой активной межэлектродной проводимости и уменьшение подпороговой утечки по сравнению с классическими двухзатворными полевыми нанотранзисторами со структурой кремний на изоляторе в области длин затворов менее 50 нм.

Отличительными свойствами исследуемых структур являются существенное снижение пика электрического поля на границе рабочая область–сток, уменьшение паразитных эффектов порогового напряжения. Результаты моделирования хорошо согласуются с данными вычислительного эксперимента, полученными при помощи коммерчески доступного программного пакета ATLAS, предназначенного для моделирования сложных транзисторных структур.

Таким образом, применение двух материалов с разной работой выхода во фронтальном затворе двухзатворных полевых нанотранзисторов со структурой кремний на изоляторе улучшает их ключевые электрофизические характеристики по сравнению с двухзатворными полевыми транзисторами с однородным фронтальным затвором и с объемными аналогами.

**Ключевые слова:** *двухзатворный КНИ КМОП-нанотранзистор, 2D-уравнение Пуассона, асимметричный фронтальный затвор, распределение потенциала, пороговое напряжение.*

Двухзатворные полевые транзисторы со структурой *кремний на изоляторе* (КНИ) являются очень перспективной архитектурой для физически предельного масштабирования КНИ КМОП-нано технологий (*комплементарная логика на транзисторах металл-оксид-полупроводник*) в основном из-за их относительной простоты и технологичности [1]. Они также отличаются расширенной возможностью оптимизации электрофизических параметров, повышенной устойчивостью к деградационным эффектам, высокой активной межэлектродной проводимостью – это превосходные опции для продолжения углубленных теоретических и экспериментальных исследований устройств с такой архитектурой [2–5]. Одной из значимых конструкций являются асимметричные двухзатворные КНИ КМОП-нанотранзисторы. Они популярны, так как обеспечивают возможность управления пороговым напряжением в отличие от симметричных двухзатворных КНИ КМОП-транзисторов [5–10].

Повысить эффективность подавления *короткоканальных эффектов* (ККЭ) призвана новая транзисторная структура – двухзатворный КНИ КМОП-транзистор с асимметричным затвором [1, 5, 11, 12]. В данной структуре фронтальный затвор состоит из двух последовательно соединен-

ных материалов М1 и М2 с различными работами выхода. Такая конфигурация из-за скачка потенциала на границе соприкосновения двух материалов будет обеспечивать одновременное увеличение активной межэлектродной проводимости и подавление ККЭ по сравнению с двухзатворным КНИ КМОП-транзистором с затвором из одного материала. Предположительно, в рассматриваемой структуре пик электрического поля на краю рабочей области у стока будет уменьшаться, что влечет рост усредненного электрического поля под затвором. Это положительно сказывается на подвижности носителей [4], обуславливает минимизацию возможности влияния локализованных зарядов на проводимость транзистора [13], увеличивает эффективность управления при помощи затвора общей проводимостью транзистора. Ступенчатый профиль распределения поверхностного потенциала гарантирует экранирование области канала под материалом М1 со стороны истока от изменения потенциала на стоке [5, 12]. После достижения транзистором режима насыщения область под материалом М2 поглощает любое дополнительное напряжение сток–исток и, следовательно, область под М1 скрыта от возможных изменений потенциала на стоке [2, 3, 9].

Цель данной работы в том, чтобы создать аналитическую 2D-модель двухзатворного КНИ КМОП-нанотранзистора с фронтальным затвором, состоящим из двух последовательно соединенных материалов M1 и M2 с различными работами выхода, на основе аналитического решения двухмерного уравнения Пуассона. На основании разработанной аналитической модели для распределения поверхностного потенциала сформулировать аналитическую модель порогового напряжения. Для исследования специфики поведения потенциала в структурах в области длин затворов менее 50 нм численно исследованы его зависимости от смещения на стоке для различных конфигураций фронтального затвора. Результаты моделирования хорошо согласуются с данными вычислительного эксперимента, полученными при помощи коммерчески доступного программного пакета ATLAS, предназначенного для моделирования сложных транзисторных структур [14]. Одновременно предсказаны достижение более высокой активной межэлектродной проводимости и уменьшение подпороговой утечки по сравнению с классическими двухзатворными КНИ КМОП-нанотранзисторами в области длин затворов менее 50 нм. Отличительными свойствами исследуемых структур являются существенное снижение пика электрического поля на границе рабочая область–сток, уменьшение эффекта roll-off (резкого спада) порогового напряжения и DIBL-эффекта.

### Прототип устройства и постановка задачи

Функциональная схема двухзатворного КНИ КМОП-транзистора с фронтальным затвором, выполненный из двух разных материалов, приведена на рисунке 1. Фронтальный затвор состоит из двух

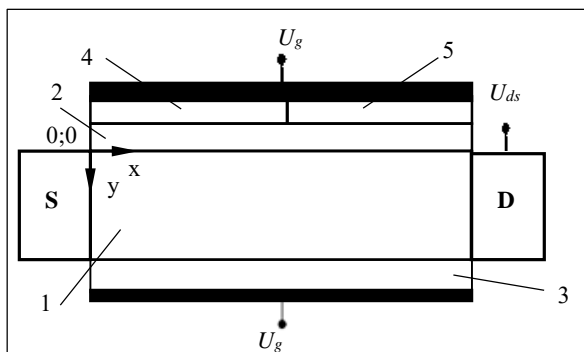


Рис. 1. Схема двухзатворного КНИ КМОП-транзистора с фронтальным затвором из двух материалов: 1 – кремниевая рабочая область, 2 – подзатворный окисел фронтального затвора, 3 – подзатворный окисел обратного затвора, 4 – часть фронтального затвора M1, 5 – часть фронтального затвора M2

Fig. 1. The scheme of dual-gate SOI CMOS transistor with double-material front gate

соединенных материалов M1 (p(+)-поликремния) и M2 (n(+)-поликремния) с длиной  $L_I$  и  $L_{II}$  соответственно. При этом выполняется условие  $L_I + L_{II} = L_g$ , где  $L_g$  – длина затвора (или длина рабочей области). Обратный (нижний) затвор выполнен из n(+)-поликремния.

Рассмотрим квазиклассическую задачу (в приближении зарядового разделения (ПЗР)) нахождения равновесного распределения потенциала в рассматриваемой КНИ-структуре [15]. Отметим, что возможность использования приближения ПЗР означает также и выбор начала отсчета потенциала  $\phi$ . В общем случае необходимо решать самосогласованную задачу, связанную с нахождением распределения потенциала  $\phi(x, y)$ , которое будет влиять и на распределение зарядов в рабочей области транзистора. Полагая, что концентрация легирования рабочей области транзистора постоянна по всему ее объему, и пренебрегая влиянием фиксированных окисных зарядов на электростатику канала, распределение потенциала в кремниевой тонкой рабочей области в режиме слабой инверсии может быть записано в виде 2D-уравнения Пуассона:

$$\frac{\partial^2 \phi(x, y)}{\partial x^2} + \frac{\partial^2 \phi(x, y)}{\partial y^2} = \frac{qN_A}{\epsilon_s}, \quad (1)$$

где  $q$  – заряд электрона;  $\phi(x, y)$  – потенциал в рабочей области транзистора;  $\epsilon_s$  – диэлектрическая проницаемость рабочей области;  $N_A$  – концентрация легирования рабочей области [7].

Для решения (1) используются следующие граничные условия:

$$\phi(x) \Big|_{y=0} = \phi_f(x),$$

$$\frac{\epsilon_{ox}}{t_{ox}} (U_g - U_{FB} - \phi(x)) = -\epsilon_s \frac{\partial \phi(x, y)}{\partial y} \Big|_{y=0},$$

$$\frac{\epsilon_{ox}}{t_{ox}} (U_g - U_{FB} - \phi(x, t_s)) = -\epsilon_s \frac{\partial \phi(x, y)}{\partial x} \Big|_{y=t_s}, \quad (2)$$

$$\phi(0, y) = U_{bi},$$

$$\phi(L_g, y) = U_{bi} + U_{ds},$$

где  $\phi(x)$  – поверхностный потенциал;  $\epsilon_{ox}$  – диэлектрическая проницаемость подзатворного окисла;  $t_{ox}$  – толщина подзатворного окисла;  $U_g$  – напряжение на затворах;  $U_{FB}$  – напряжение плоских зон;  $U_{bi}$  – встроенная разность потенциалов;  $t_s$  – толщина рабочей области транзистора;  $U_{ds}$  – напряжение сток–исток.

Распределение потенциала в вертикальном направлении, то есть его  $y$ -зависимость, может быть аппроксимировано следующей параболической функцией, обоснование которой для полностью обедненных КНИ КМОП-транзисторов можно представить так:

$$\phi(x, y) = \sum_{m=0}^N a_m(x) y^m, \quad (3)$$

где  $a_m(x)$  – произвольные коэффициенты, зависящие только от  $x$  [6].



### Метод решения

Поскольку в рассматриваемой транзисторной структуре фронтальный затвор состоит из двух различных материалов с разными работами выхода, фронтальные напряжения плоских зон для областей I и II (см. рис. 1) на фронтальном затворе будут различными, их можно представить в виде

$$U_{FB\_I} = \Phi_{MS\_I} - (\chi_{Si} + \frac{E_g}{2q} + \varphi_F)$$

$$\text{и } U_{FB\_II} = \Phi_{MS\_II} - (\chi_{Si} + \frac{E_g}{2q} + \varphi_F), \quad (4)$$

где  $\chi_{Si}$  – сродство электрона в кремнии;  $E_g$  – запрещенная зона кремния при 300 К;  $\varphi_F$  – уровень Ферми, где  $\varphi_F = \varphi_i \ln \frac{N_A}{n_i}$  ( $\varphi_i = \frac{kT}{q}$  – тепловой потенциал;  $T$  – температура;  $k$  – константа Больцмана;  $n_i$  – собственная концентрация носителей).

Так как на фронтальном затворе есть две области, распределение потенциала для каждой может быть записано аналогично (3) в виде

$$\varphi^{(I)}(x, y) = a_0^{(I)}(x) + a_1^{(I)}(x)y + a_2^{(I)}(x)y^2$$

при  $0 \leq x \leq L_i$  и  $0 \leq y \leq t_s$ , (5)

$$\varphi^{(II)}(x, y) = a_0^{(II)}(x) + a_1^{(II)}(x)y + a_2^{(II)}(x)y^2$$

при  $L_i \leq x \leq L_g$  и  $0 \leq y \leq t_s$ , (6)

где  $\varphi^{(I)}(x, y)$  и  $\varphi^{(II)}(x, y)$  – потенциал под областями I и II соответственно и коэффициенты  $a_i^{(j)}(x)$  – произвольные функции, где  $i = 0, 1, 2$  и  $j = I, II$ .

Уравнение Пуассона решается отдельно для двух областей фронтального затвора с использованием следующих граничных условий.

1. Электрическое поле на границе фронтальный затвор–окисел непрерывно для обеих областей:

$$\frac{\varepsilon_{ox}}{t_f}(U_g - U_{FB}^{(I)} - \varphi_f^{(I)}(x)) = -\varepsilon_s \frac{\partial \varphi^{(I)}(x, y)}{\partial x} \Big|_{y=0},$$

$$\frac{\varepsilon_{ox}}{t_f}(U_g - U_{FB}^{(II)} - \varphi_f^{(II)}(x)) = -\varepsilon_s \frac{\partial \varphi^{(II)}(x, y)}{\partial x} \Big|_{y=0}, \quad (7)$$

где  $\varphi_f^{(I)}(x)$  и  $\varphi_f^{(II)}(x)$  – поверхностный потенциал под областями I и II соответственно;  $t_f$  – толщина подзатворного окисла фронтального затвора.

2. Электрическое поле на границе обратный затвор–окисел и обратный канал непрерывно для обоих материалов на фронтальном затворе:

$$\frac{\varepsilon_{ox}}{t_b}(U_g - U_{FBb} - \varphi(x, t_s)) = -\varepsilon_s \frac{\partial \varphi^{(I)}(x, y)}{\partial x} \Big|_{y=t_s},$$

$$\frac{\varepsilon_{ox}}{t_b}(U_g - U_{FBb} - \varphi(x, t_s)) = -\varepsilon_s \frac{\partial \varphi^{(II)}(x, y)}{\partial x} \Big|_{y=t_s}, \quad (8)$$

где  $t_b$  – толщина подзатворного окисла обратного затвора;  $U_{FBb}$  – напряжение плоских зон на обратном затворе, такое же, что и на фронтальном для области II.

3. Поверхностный потенциал на поверхности двух разнородных материалов затвора на фронтальном затворе непрерывный:

$$\varphi^{(I)}(L_i, 0) = \varphi^{(II)}(L_i, 0). \quad (9)$$

4. Электрическое поле на поверхности двух материалов фронтального затвора непрерывно:

$$\frac{\partial \varphi^{(I)}(x, y)}{\partial x} \Big|_{x=L_i} = \frac{\partial \varphi^{(II)}(x, y)}{\partial x} \Big|_{x=L_i}. \quad (10)$$

5. Потенциал на краю рабочей области со стороны истока:

$$\varphi^{(I)}(0, y) = U_{bi}, \quad (11)$$

где  $U_{bi} = \varphi_i \ln(\frac{N_A N_{ds}}{n_i^2})$  – контактная разность потенциалов из-за разности концентраций легирования рабочей области и областей стока и истока, где  $N_{ds}$  – концентрация легирования области стока и истока.

6. Потенциал на краю рабочей области со стороны стока:

$$\varphi^{(II)}(L_g, y) = U_{bi} + U_{ds}, \quad (12)$$

где  $U_{ds}$  – приложенное напряжение между стоком и истоком (напряжение питания).

Выражения для коэффициентов в (5) и (6) определяются подстановкой решений (5)–(6) в (1); используя граничные условия (7)–(12), их можно записать в виде

$$a_0^{(j)}(x) = \varphi_f^{(j)}(x),$$

$$a_1^{(j)}(x) = \frac{C_f}{\varepsilon_s}(\varphi_f^{(j)}(x) - U_g + U_{FB}^{(j)}), \quad (13)$$

$$a_0^{(j)}(x) + a_1^{(j)}(x)t_s + a_2^{(j)}(x)t_s^2 = \varphi_b(x),$$

где  $C_f = \frac{\varepsilon_{ox}}{t_f}$  – емкость фронтального затвора,  $j = I, II$ .

Дифференцируя решение (3) по  $y$  с учетом (13), получаем выражение для  $a_2(x)$ :

$$a_2^{(j)}(x) = \frac{-(C_f + C_b + \frac{C_f C_b}{C_s})\varphi_f^{(j)}(x)}{(C_b + 2C_s)t_s^2} +$$

$$\frac{C_f(1 + \frac{C_b}{C_s})(U_g - U_{FB}^{(j)}) + C_b(U_g - U_{FB}^{(II)})}{(C_b + 2C_s)t_s^2}, \quad (14)$$

где  $C_f = \frac{\varepsilon_s}{t_s}$  – емкость рабочей области;  $C_b = \frac{\varepsilon_{ox}}{t_b}$  – емкость обратного затвора,  $j = I, II$ .

Для определения поверхностного потенциала необходимо решить уравнение Пуассона на границе Si–SiO<sub>2</sub>, которое можно представить, например, для области I в виде

$$\frac{d^2 \varphi_f^{(I)}(x)}{dx^2} - \alpha \varphi_f^{(I)}(x) = \varphi_0^{(I)}, \quad (15)$$

$$\text{где } \alpha = 2 \frac{(1 - \frac{C_f}{C_s} + \frac{C_f}{C_b})}{t_s^2 (1 + 2 \frac{C_s}{C_b})},$$

$$\varphi_0^{(I)} = \frac{qN_A}{\varepsilon_s} - 2 \frac{(U_g - U_{FB}^{(I)}) (\frac{C_f}{C_b} + \frac{C_f}{C_s})}{t_s (1 + 2 \frac{C_s}{C_b})} - 2 \frac{U_g - U_{FB}^{(II)}}{t_s^2 (1 + 2 \frac{C_s}{C_b})}.$$

Решением данного уравнения может быть следующее выражение:

$$\varphi_f^{(I)} = A_1^{(I)} \exp(y/l) + A_2^{(I)} \exp(-y/l) - l^2 \varphi_0^{(I)}, \quad (16)$$

где  $l$  – характеристическая длина  $l = \frac{1}{\sqrt{\alpha}}$ ,

$$A_1^{(I)} = ((U_{bi} + U_{ds} - l^2 \varphi_0^{(II)}) - (U_{bi} - l^2 \varphi_0^{(I)}) \exp(-\frac{L_g}{l}) -$$

$$-l^2 (\varphi_0^{(I)} - \varphi_0^{(II)}) \cosh(\frac{L_{II}}{l}) \frac{\exp(-\frac{L_g}{l})}{(1 - \exp(-2 \frac{L_g}{l}))},$$

$$A_2^{(I)} = ((U_{bi} - l^2 \varphi_0^{(I)}) - (U_{bi} + U_{ds} - l^2 \varphi_0^{(II)}) \exp(-\frac{L_g}{l}) +$$

$$+l^2 (\varphi_0^{(I)} - \varphi_0^{(II)}) \cosh(\frac{L_{II}}{l}) \frac{\exp(-\frac{L_g}{l})}{(1 - \exp(-2 \frac{L_g}{l}))}.$$

Для области II уравнение Пуассона можно представить в виде

$$\frac{d^2 \varphi_f^{(II)}(x)}{dx^2} - \alpha \varphi^{(II)}(x) = \varphi_0^{(II)}, \quad (17)$$

$$\text{где } \varphi_0^{(II)} = \frac{qN_A}{\varepsilon_s} - 2 \frac{(U_g - U_{FB}^{(II)}) (\frac{C_f}{C_b} + \frac{C_f}{C_s})}{t_s (1 + 2 \frac{C_s}{C_b})} -$$

$$-2 \frac{U_g - U_{FB}^{(II)}}{t_s^2 (1 + 2 \frac{C_s}{C_b})}.$$

Решение уравнения (17) по аналогии с (16) можно представить в виде

$$\varphi_f^{(II)} = A_1^{(II)} \exp(y/l) + A_2^{(II)} \exp(-y/l) - l^2 \varphi_0^{(II)}, \quad (18)$$

$$\text{где } A_1^{(II)} = A_1^{(I)} \exp(\frac{L_I}{l}) + \frac{l^2}{2} (\varphi_0^{(I)} - \varphi_0^{(II)}),$$

$$A_2^{(II)} = A_2^{(I)} \exp(-\frac{L_I}{l}) + \frac{l^2}{2} (\varphi_0^{(I)} - \varphi_0^{(II)}).$$

### Пороговое напряжение

В квазиклассическом приближении пороговое напряжение  $U_{th}$  двухзатворного КНИ КМОП-транзистора соответствует такому затворному напряжению, при котором сумма электронной плотности

на фронтальной и обратной поверхностях равняется пиковой плотности легирования рабочей области [7]. В общем случае аналитическое выражение для  $U_{th}$  вытекает из выражения для минимального значения фронтального поверхностного потенциала [9–11]. Для рассматриваемой нанотранзисторной структуры можно сделать некоторые упрощения с учетом симметричности структуры и одинакового напряжения  $U_g$ , приложенного к затворам. Тогда выражение  $U_{th}$  в длинноканальном случае для исследуемой структуры основывается на известной зависимости распределения потенциала от напряжения на затворах транзистора. В пороговом режиме первоначально инверсионный слой сформирован на внутренней поверхности обратного затвора p+ поликристаллического кремния. Распределение потенциала тогда изменяется линейно, в то время как фронтальный поверхностный потенциал фиксирован. После этого инверсионный слой на внутренней поверхности фронтального p+ поликристаллического кремния сформирован, распределение потенциала в рабочей области стабилизируется и поддерживается напряжением, приложенным к обоим затворам [9]. В рассматриваемой нанотранзисторной структуре в общем случае параметры  $t_f$  и  $t_b$  для фронтального и обратного затворов, которые являются толщинами подзатворного окисла, могут быть различными. Это приводит к тому, что данная структура характеризуется двумя различными пороговыми напряжениями, которые связаны с фронтальным и обратным затворами соответственно [5]. Опираясь на изложенное приближение, выражение для порогового напряжения на фронтальном и обратном затворах в длинноканальном случае имеет вид

$$U_{th\_b} = U_{FB}^{(I)} + 2\varphi_F + \frac{Q_s}{2} (1 + \varphi_t \frac{4C_s}{Q_s}) (\frac{1}{4C_s} + \frac{1}{C_f}) +$$

$$+ \varphi_t \ln(\varphi_t \frac{4C_s}{Q_s}) - \frac{\varepsilon_r t_f + t_s}{\varepsilon_r (t_f + t_b) + t_s} \Delta U_{FB},$$

$$U_{th\_f} = U_{FB}^{(I)} + 2\varphi_F +$$

$$+ \frac{Q_s}{2} \left( 1 + \varphi_t \frac{4C_s}{Q_s} \right) \left( \frac{1}{4C_s} + \frac{1}{C_f} \right) + \varphi_t \ln \left( \varphi_t \frac{4C_s}{Q_s} \right), \quad (19)$$

где  $U_{th\_b}$  – пороговое напряжение для обратного затвора;  $U_{th\_f}$  – пороговое напряжение для фронтального затвора;  $Q_s = qN_{Ats}$  – примесный (акцепторный) заряд рабочей области;  $\varepsilon_r = \frac{\varepsilon_s}{\varepsilon_{ox}}$  – от-

ношение диэлектрических проницаемостей;  $\Delta U_{FB} = U_{FB}^{(I)} - U_{FB}^{(II)}$  – разница между напряжениями плоских зон.

В длинноканальном случае примесный заряд  $Q_s$  играет заметную роль в поведении порогового напряжения [7]. Варьируя уровень легирования рабочей области, можно в некоторой степени управлять величиной порогового напряжения. Однако

для тонкопленочных короткоканальных устройств влиянием данного заряда на пороговые напряжения можно пренебречь [10]. Для КНИ КМОП-нотранзисторов с низколегированной рабочей областью это приближение приводит к следующему уравнению Пуассона:

$$\frac{\partial^2 \varphi(x, y)}{\partial x^2} + \frac{\partial \varphi(x, y)}{\partial y^2} \approx 0. \quad (20)$$

Для решения (20) используются параболическое приближение для распределения потенциала, аналогичное выражениям (5) и (6), и граничные условия (7)–(12). Сдвиг порогового напряжения из-за ККЭ для двухзатворного КНИ КМОП-нотранзистора с составным фронтальным затвором по аналогии с [9] можно представить так:

$$\Delta U_{th} = 2\sqrt{u_0 u_1} \exp\left(-\frac{L_1}{\sqrt{2\varepsilon_s t_s t_f}}\right), \quad (21)$$

$$\text{где } u_0 = U_{bi} - U_g - U_{FB}^{(I)} + \frac{\Delta U_{FB}}{2(1 + \frac{t_s}{2\varepsilon_s t_f})},$$

$$u_1 = \frac{1}{2} \frac{(U_{bi} + U_{ds} - U_g - U_{FB}^{(II)}) \sinh(\frac{L_1}{l}) + u_0 \sinh(\frac{L_2}{l})}{\cosh(\frac{L_1}{l}) \sinh(\frac{L_2}{l}) + \sinh(\frac{L_1}{l}) \cosh(\frac{L_2}{l})}.$$

Тогда в общем случае выражение для порогового напряжения двухзатворного КНИ КМОП-нотранзистора с составным фронтальным затвором имеет следующий вид:

$$U_{th} = U_{th\_long} - U_{th}, \quad (22)$$

где  $U_{th\_long}$  может быть или  $U_{th1}$ , или  $U_{th2}$ . Уравнение (22) применимо не во всех случаях. Простой вид поправки порогового напряжения обусловлен предположением, что  $L_1$  и  $L_{II}$  несильно различаются. При значительном отличии значений  $L_1$  и  $L_{II}$  искажается связь между фронтальным и обратными затворами, что приводит к нивелированию эффекта roll-off порогового напряжения. С учетом этого мы вводим поправочный коэффициент, эмпирическое выражение для которого является обобщением результатов большого числа расчетов, выполненных при помощи программы ATLAS:

$$\theta = 1 - \frac{|L_1 - L_{II}|}{\rho L_1}, \quad (23)$$

где  $\rho$  – подгоночный параметр, который зависит от отношения  $L_1$  и  $L_{II}$  и общей длины рабочей области. Нужно отметить, что, если  $L_1 = L_{II}$ , то  $\theta = 1$ . При уменьшении длины рабочей области и постоянном отношении  $L_1$  и  $L_{II}$  эффект roll-off порогового напряжения проявляется, хотя менее выражено, чем в случае классического двухзатворного КНИ КМОП-транзистора. Существенные отличия в пороговом напряжении начинают возникать примерно в субдиапазоне длин рабочей области 50 нм при отношении  $L_1$  и  $L_{II}$  3:1 (или 1:3). Очевидно, что

такие топологии вряд ли представляют практический интерес в силу незначительного влияния на характеристики нанотранзисторных структур. Поэтому заключительное выражение для  $U_{th}$  имеет вид

$$U_{th} = U_{th\_long} - \theta \Delta U_{th}. \quad (24)$$

### Результаты моделирования

Для модельных расчетов выбран прототип двухзатворного полностью обедненного КНИ КМОП-нотранзистора с фронтальным затвором из двух материалов с разной работой выхода (фронтальный затвор для формирования областей I и II с работой выхода 4,17 эВ и 5,25 эВ соответственно). Технологические параметры транзистора, отвечающие современным требованиям, таковы:  $L_g = 50$  нм,  $t_s = 10$  нм,  $t_f = 2,0$  нм,  $t_b = 2,0$  нм,  $N_A = 1 \times 10^{17}$  см<sup>-3</sup>,  $N_{ds} = 5 \times 10^{20}$  см<sup>-3</sup>.

На рисунке 2а приведены результаты моделирования распределения фронтального поверхностного потенциала вдоль рабочей области двухзатворного полностью обедненного КНИ КМОП-нотранзистора с фронтальным затвором из двух материалов с разной работой выхода. Длины областей с разной работой выхода равны и составляют половину общей длины рабочей области. Следует отметить, что поверхностный потенциал в центре на границе контакта двух областей представляет ступенчатую функцию на поверхности. Из-за этой ступенчатой функции область с большей работой выхода (область M1) фронтального затвора скрыта от изменений потенциала на стоке. Таким образом, ступенчатая функция подавляет влияние электрического поля, вызванного потенциалом между истоком и стоком (напряжением питания) в области M1, и, следовательно, нет никакого существенного изменения в поверхностном потенциале под M1 при увеличении смещения стока. Это означает, что напряжение на стоке имеет очень незначительное влияние на ток стока после насыщения [10], который, в свою очередь, значительно уменьшает проводимость стока и эффект DIBL для двухзатворного полностью обедненного КНИ КМОП-нотранзистора с фронтальным затвором из двух материалов с разной работой выхода.

На рисунке 2б приведены результаты расчетов распределения фронтального поверхностного потенциала вдоль рабочей области для различных комбинаций длин  $L_1$  и  $L_{II}$ . Из представленных результатов следует, что во втором случае нет никакого изменения в поверхностном потенциале, но в первом случае положение минимума поверхностного потенциала, лежащего под M1, смещается к истоку, так как параметр  $L_1$  уменьшается. Это вынуждает пик электрического поля в рабочей области смещаться больше к истоку. Однако характерный вид электрического поля остается неизменным. Кроме того, минимумы поверхностного

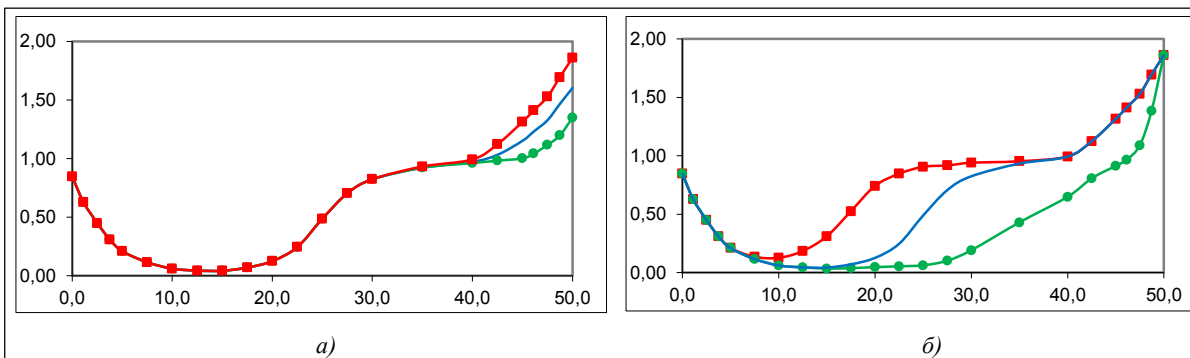


Рис. 2. Распределения фронтального поверхностного потенциала вдоль рабочей области:  
 а) при разных напряжениях питания, где 1 –  $U_{ds} = 1\text{ В}$ , 2 –  $U_{ds} = 0,75\text{ В}$ , 3 –  $U_{ds} = 0,5\text{ В}$ ;  
 б) при разных отношениях  $L_I$  и  $L_{II}$ , где 1 –  $L_I: L_{II} = 1:2$ , 2 –  $L_I: L_{II} = 1:1$ , 3 –  $L_I: L_{II} = 2:1$

Fig. 2. Front surface potential distribution along the workspace

потенциала для этих трех случаев различаются. Это происходит потому, что, поскольку  $L_I$  увеличивается, часть рабочей области, которой управляет фронтальный затвор с большей работой выхода, тоже увеличивается.

На рисунке 3 представлены результаты расчетов напряженности электрического поля вдоль рабочей области двухзатворных КНИ КМОП-нано-транзисторов с разным типом фронтального затвора: с однородным и асимметричным. Природа электрического поля для двух транзисторов одинаковая. Ближе к стоку напряженность поля резко возрастает. Так, на последних десяти процентах длины оно возрастает в два раза. Тем не менее, как видно из рисунка, пиковое электрическое поле у края рабочей области со стороны стока транзистора с асимметричным затвором значительно меньше (практически в 2,5 раза), чем для транзистора с однородным затвором, из-за использования материала затвора с более низкой работой выхода, чем материал затвора у истока. Из-за скачка поверхностного потенциала двухзатворного КНИ КМОП-нано-транзистора с фронтальным затвором,

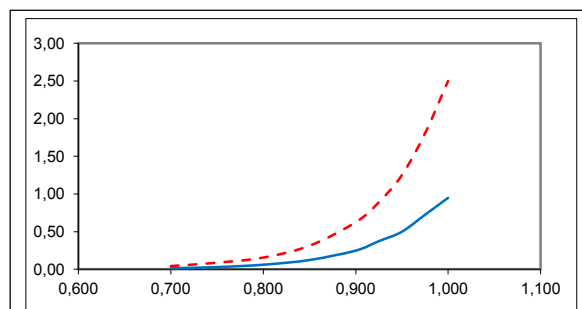


Рис. 3. Распределение электрического поля вдоль рабочей области у стока при  $U_g = 0,1\text{ В}$ ,  $U_{ds} = 0,5\text{ В}$ , где 1 – двухзатворный КНИ КМОП-нано-транзистор с фронтальным затвором с двумя материалами с разной работой выхода и  $L_I = L_{II}$ , 2 – обычный симметричный двухзатворный КНИ КМОП-нано-транзистор

Fig. 3. Electrical field distribution along the drain workspace when  $U_g = 0,1\text{ В}$ ,  $U_{ds} = 0,5\text{ В}$

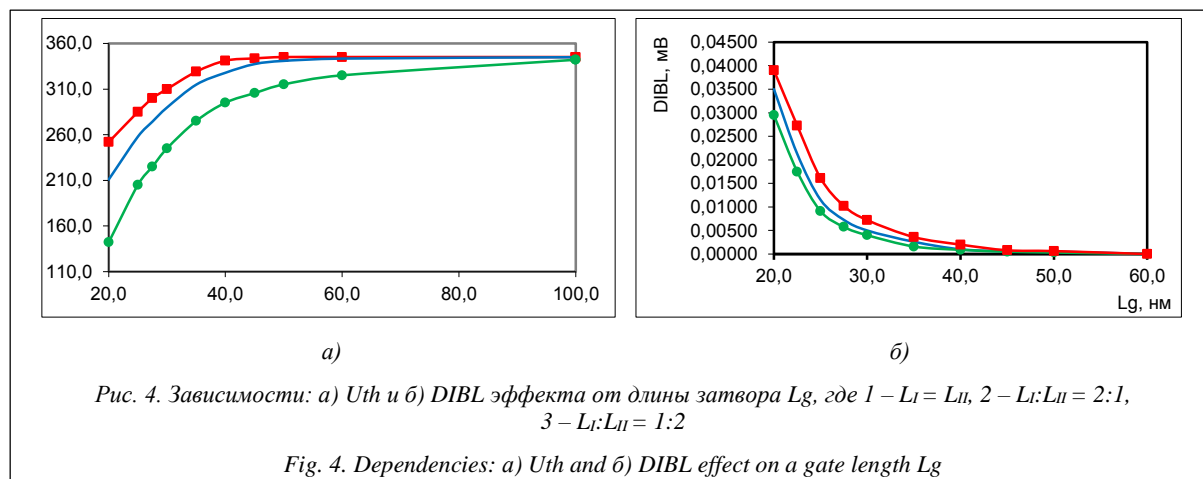
состоящим из двух материалов с разной работой выхода, пиковое электрическое поле стороны стока существенно снижается, что позволяет достигать одновременного подавления как короткоканальных эффектов, так и эффекта горячих носителей [12].

Зависимость порогового напряжения от длины затвора для двухзатворного КНИ КМОП-нано-транзистора с асимметричным фронтальным затвором для различных соотношений  $L_I/L_{II}$  показана на рисунке 4а. Следует отметить, что ККЭ во всех случаях являются серьезным препятствием при масштабировании длины рабочей области. Наиболее сильно это проявляется для случая  $L_I:L_{II} = 1:2$ , когда влияние области M1 на распределение потенциала незначительно, что отражено на рисунке 3 (нижняя кривая). С увеличением протяженности области M1 влияние эффекта roll-off уменьшается. Зависимость  $U_{th}(L_g)$  сглаживается. В предельном случае отклонение порогового напряжения составляет 82 мВ и 61 мВ для отношения  $L_I:L_{II} = 1:1$  и  $L_I:L_{II} = 2:1$  соответственно.

На рисунке 4б приведены результаты моделирования зависимости DIBL-эффекта от длины рабочей области исследуемого прототипа при разных отношениях  $L_I:L_{II}$ . Величина DIBL вычисляется как разность между пороговым напряжением при высоком напряжении сток–исток ( $U_{ds}^{(h)} = 0,5\text{ В}$ ) и пороговым напряжением при низком напряжении сток–исток ( $U_{ds}^{(l)} = 0,05\text{ В}$ ). Исходя из соотношений (21) и (24) выражение для DIBL можно записать в виде

$$DIBL = 2\theta(\sqrt{u_0(U_{ds}^{(h)})}u_1(U_{ds}^{(h)}) - \sqrt{u_0(U_{ds}^{(l)})}u_1(U_{ds}^{(l)})) \exp\left(-\frac{L_I}{\sqrt{2\varepsilon_r t_s t_f}}\right).$$

Из результатов моделирования следует, что величина DIBL резко возрастает при масштабировании длины рабочей области начиная примерно с 40 нм. При этом отношение длин областей  $L_I$  и  $L_{II}$  оказывает влияние на величину параметра DIBL. Уменьшение длины области II с меньшей работой



выхода приводит к снижению экранировки стока и, как следствие, к росту величины DIBL, а также сказывается на управляемости и активной межэлектродной проводимости транзисторной структуры. В то же время при больших длинах рабочей области эффект DIBL незначительный при любых отношениях  $L_I$  и  $L_{II}$ .

### Заключение

Разработаны 2D-аналитическая модель распределения потенциала и вытекающая из нее модель порогового напряжения для тонкопленочного двухзатворного КНИ КМОП-нотранзистора с фронтальным затвором из двух последовательно соединенных материалов с разной работой выхода. Исследованы поведение потенциала в транзисторных структурах в области длин затворов менее 50 нм, его зависимости от смещений на стоке для различных конфигураций фронтального затвора. Полученные результаты однозначно показывают, что применение двух материалов с разной работой выхода во фронтальном затворе приводит к эффективному подавлению короткоканальных эффектов из-за ступенчатой функции в профиле потенциала на границе раздела двух материалов. Сдвиг положения поверхностной позиции минимума поверхностного потенциала незначителен с увеличением смещений стока. Пиковое электрическое поле у стока значительно снижается по сравнению с аналогичной транзисторной структурой, но с однородным фронтальным затвором. Предсказаны достижение более высокой активной межэлектродной проводимости и уменьшение подпороговой утечки по сравнению с классическими двухзатворными КНИ КМОП-нотранзисторами в области длин затворов менее 50 нм. Отличительными свойствами исследуемых структур являются существенное снижение пика электрического поля на границе рабочей области и стока, уменьшение эффекта roll-off (резкого спада) порогового напряжения и DIBL-эффекта. Результаты моделирования хорошо согласуются с данными вычислительного эксперимента, которые получены при помощи ком-

мерчески доступного программного пакета ATLAS, предназначенного для моделирования сложных транзисторных структур. Таким образом, применение двух материалов с разной работой выхода во фронтальном затворе двухзатворных КНИ КМОП-нотранзисторов улучшает их ключевые электрофизические характеристики по сравнению с двухзатворными полевыми транзисторами с однородным фронтальным затвором и с объемными аналогами.

### Литература

1. International technology roadmap for semiconductor 2014 edition. URL: <http://public.itrs.net/> (дата обращения: 17.11.2016).
2. Djeflal F., Meguellati M., Benhaya A. A two-dimensional analytical analysis of subthreshold behavior to study the scaling capability of nanoscale graded channel gate stack DG MOSFETs. *Physica E*, 2009, vol. 41, no. 8, pp. 1872–1879.
3. Monga U., Fjeldly T.A. Compact subthreshold current modeling of short-channel nanoscale double-gate MOSFET. *IEEE Trans. Electron Devices*, 2009, vol. 56, no. 6, pp. 1533–1537.
4. Deb S., Singh N.B., Das D., De A.K., Sarkar S.K. Analytical model of threshold voltage and subthreshold slope of SOI and SON MOSFETs: A comparative study. *Jour. Electron Devices*, 2010, vol. 8, no. 1, pp. 300–309.
5. Li J., Liu H.X., Li B. Two-dimensional threshold voltage analytical model of DMG strained-silicon-on-insulator MOSFETs. *Jour. of Semiconductors*, 2010, vol. 31, no. 8, pp. 84008–84024.
6. Масальский Н.В. Характеристики двухзатворных КНИ КМОП-нотранзисторов для перспективных технологий с низким уровнем потребляемой мощности // *Микроэлектроника*. 2012. Т. 41. № 6. С. 436–444.
7. Colinge J.P. Multiple-gate SOI MOSFETs. *Solid-State Electronics*, 2004, vol. 48, no. 3, pp. 897–909.
8. Venkateshwar G.R., Kumar M.J. A new dual-material double-gate (DMDG) nanoscale SOI MOSFET – Two dimensional analytical modeling and simulation. *IEEE Trans. on Nanotechnology*, 2005, vol. 4, no. 3, pp. 260–268.
9. Goel K., Saxena M., Gupta M. Two-dimensional analytical threshold voltage model for DMG epi-MOSFET. *IEEE Trans Electron Devices*, 2005, vol. 52, no. 1, pp. 23–31.
10. Chiage T.K., Chen M.L. A new analytical threshold voltage model for symmetrical double-gate MOSFETs with high-k gate dielectrics. *Solid State Electron*, 2007, vol. 51, no. 3, pp. 387–395.
11. Venkataraman V., Nawal S., Kumar M.J. Compact analytical threshold-voltage model of nanoscale fully depleted strained-Si on silicon-germanium-on-insulator (SGOI) MOSFETs. *IEEE Trans Electron Devices*, 2007, vol. 54, no. 3, pp. 554–562.
12. Sharma R.K., Gupta M., Gupta R.S. Two-dimensional analytical subthreshold model of graded channel DG FD SOI n-MOSFET with gate misalignment effect. *Superlattices and Micro-*

structures, 2009, vol. 45, no. 1, pp. 91–103.

13. Goel K., Saxena M., Gupta M., Gupta R.S. Modeling and simulation of a nanoscale three-region tri-material gate stack (TRIMGAS) MOSFET for improved carrier transport efficiency and reduced hot electron effects. *IEEE Trans. Electron Devices*, 2006, vol. 53, no. 7, pp. 1623–1633.

14. Silvaco Int. 2004: ATLAS User's Manual A 2D numerical device simulator. URL: <http://www.silvaco.com/> (дата обращения: 22.10.2016).

15. Масальский Н.В. Проблемы схемотехнического моделирования нанотранзисторов со структурой «кремний на изоляторе» // Программные продукты и системы. 2013. № 4. С. 60–67.

Software & Systems

DOI: 10.15827/0236-235X.119.359-366

Received 04.05.17

2017, vol. 30, no. 3, pp. 359–366

## POTENTIAL DISTRIBUTION SIMULATION FOR A DUAL-GATE FIELD SILICON ON INSULATOR NANOTRANSISTOR WITH AN ASYMMETRIC GATE

N.V. Masalsky<sup>1</sup>, Ph.D. (Physics and Mathematics), Head of Sector, [volkov@niisi.ras.ru](mailto:volkov@niisi.ras.ru)

<sup>1</sup> Federal State Institution "Scientific Research Institute for System Analysis of the Russian Academy of Sciences" (SRISA RAS), Nakhimovsky Ave. 36/1, Moscow, 117218, Russian Federation

**Abstract.** The paper considers a 2D analytical model of potential distribution and a following threshold voltage model for a thin-film dual-gate field nanotransistor with "silicon-on-insulator" structure with a front gate from two sequentially connected materials with different work function. It also investigates potential behavior in transistor structures in the field of gate lengths less than 50 nanometers, its dependence on drain offsets for different configurations of a frontal gate.

The received results show that using two materials with different work function in a frontal gate leads to effective suppression of short-channel effects due to a step function in a potential profile on the boundary of two materials. The shift of a surface line of a surface potential minimum is insignificant with increasing drain biases. The peak electric field at a drain considerably decreases in comparison with similar transistor structure but with a uniform frontal gate. At the same time, it is predicted to be a higher active interelectrode conductance and reduction of subthreshold leak in comparison with classical two-gate field nanotransistors with structure "silicon-on-insulator" in the gate lengths less than 50 nanometers.

The distinctive properties of the researched structures are essential lowering of an electrical field peak on boundary of a working area and a drain, reduction of parasitic effects of threshold voltage. The simulation results are in good agreement with experiment data received by means of commercially available software package ATLAS intended for simulation of complex transistor structures.

Thus, using two materials with different work function in a front gate of dual-gate field nanotransistors with the structure "silicon-on-insulator" improves their key electrophysical characteristics in comparison with dual-gate field transistors with a uniform front gate and volume analogues.

**Keywords:** dual-gate SOI CMOS nanotransistor, 2D Poisson equation, asymmetric front gate, potential distribution, threshold voltage.

### References

1. International technology roadmap for semiconductor 2014 edition. Available at: <http://public.itrs.net/> (accessed November 17, 2014).
2. Djeflal F., Meguellati M., Benhaya A. A two-dimensional analytical analysis of subthreshold behavior to study the scaling capability of nanoscale graded channel gate stack DG MOSFETs. *Physica E*. 2009, vol. 41, no. 8, pp. 1872–1879.
3. Monga U., Fjeldly T.A. Compact subthreshold current modeling of short-channel nanoscale double-gate MOSFET. *IEEE Trans. Electron Devices*. 2009, vol. 56, no. 6, pp. 1533–1537.
4. Deb S., Singh N.B., Das D., De A.K., Sarkar S.K. Analytical model of threshold voltage and subthreshold slope of SOI and SON MOSFETs: A comparative study. *Jour. Electron Devices*. 2010, vol. 8, no. 1, pp. 300–309.
5. Li J., Liu H.X., Li B. Two-dimensional threshold voltage analytical model of DMG strained-silicon-on-insulator MOSFETs. *Jour. of Semiconductors*. 2010, vol. 31, no. 8, pp. 84008–84024.
6. Masalsky N.V. Characteristics of double gate SOI CMOS nanotransistors for perspective technologies with the low supply power. *Microelectronika* [Microelectronics]. 2012, vol. 41, no. 6, pp. 436–444 (in Russ.).
7. Colinge J.P. Multiple-gate SOI MOSFETs. *Solid-State Electronics*. 2004, vol. 48, no. 3, pp. 897–909.
8. Venkateshwar G.R., Kumar M.J. A new dual-material double-gate (DMDG) nanoscale SOI MOSFET – Two dimensional analytical modeling and simulation. *IEEE Trans. on Nanotechnology*. 2005, vol. 4, no. 3, pp. 260–268.
9. Goel K., Saxena M., Gupta M. Two-dimensional analytical threshold voltage model for DMG epi-MOSFET. *IEEE Trans. Electron Devices*. 2005, vol. 52, no. 1, pp. 23–31.
10. Chiage T.K., Chen M.L. A new analytical threshold voltage model for symmetrical double-gate MOSFETs with high-k gate dielectrics. *Solid State Electronics*. 2007, vol. 51, no. 3, pp. 387–395.
11. Venkataraman V., Nawal S., Kumar M.J. Compact analytical threshold-voltage model of nanoscale fully depleted strained-Si on silicon-germanium-on-insulator (SGOI) MOSFETs. *IEEE Trans. Electron Devices*. 2007, vol. 54, no. 3, pp. 554–562.
12. Sharma R.K., Gupta M., Gupta R.S. Two-dimensional analytical subthreshold model of graded channel DG FD SOI n-MOSFET with gate misalignment effect. *Superlattices and Microstructures*. 2009, vol. 45, no. 1, pp. 91–103.
13. Goel K., Saxena M., Gupta M., Gupta R.S. Modeling and simulation of a nanoscale three-region tri-material gate stack (TRIMGAS) MOSFET for improved carrier transport efficiency and reduced hot electron effects. *IEEE Trans. Electron Devices*. 2006, vol. 53, no. 7, pp. 1623–1633.
14. Silvaco Int. 2004: ATLAS User's Manual. A 2D numerical device simulator. Available at: <http://www.silvaco.com/> (accessed October 22, 2015).
15. Masalsky N.V. Nanotransistors circuitry simulation problems with silicon-on-insulator structure. *Programmye produkty i sistemy* [Software & Systems]. 2013, no. 4, pp. 60–67 (in Russ.).



УДК 519.684.6 + 519.685.1 + 519.688  
DOI: 10.15827/0236-235X.119.367-372

Дата подачи статьи: 03.04.17  
2017. Т. 30. № 3. С. 367–372

## **МНОГОФУНКЦИОНАЛЬНЫЙ ПРОГРАММНЫЙ КОМПЛЕКС ТЕПЛОВОГО ПРОЕКТИРОВАНИЯ ЭЛЕКТРОННЫХ СИСТЕМ: ТРЕБОВАНИЯ К АРХИТЕКТУРЕ И ФУНКЦИОНАЛЬНЫМ ВОЗМОЖНОСТЯМ МОДЕЛИРОВАНИЯ**

*А.Г. Мадера, д.т.н., профессор, зав. отделом, agmprof@mail.ru;*

*В.Н. Решетников, д.ф.-м.н., профессор, rvp\_@mail.ru*

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

Рассмотрены концепции математического и компьютерного моделирования, положенные в основу разработки и создания многофункционального программного комплекса для теплового проектирования сложных электронных систем. Показаны принципиальные недостатки зарубежных программных комплексов теплового проектирования, которые не позволяют использовать их в практике проектирования и создания конкурентоспособных электронных систем.

В статье обоснованы требования к многофункциональному программному комплексу для адекватного проектирования тепловых процессов и температурных распределений в электронных системах в условиях функционирования и эксплуатации, приближенных к реальным, а именно: программный комплекс должен обеспечивать возможность моделировать тепловые процессы в электронных системах, являющиеся нелинейными, нестационарными, трехмерными, интервально-стохастическими, а также учитывать влияние тепловой обратной связи, особенности конструкции и монтажа элементов в электронной системе, воздействие дестабилизирующих механических, климатических и радиационных факторов.

Архитектура многофункционального программного комплекса должна включать высокоэффективное математическое вычислительное ядро, развитую сервисную пользовательскую оболочку, отвечающую современным требованиям к сложным программным комплексам и системам. Сервисная оболочка, в свою очередь, должна обеспечивать визуальную, наглядную и удобную для восприятия форму задания исходных данных в виде цветных изображений температурных распределений и других тепловых характеристик на различных иерархических уровнях.

Программный комплекс реализуется в среде Microsoft Visual Studio, обеспечивающей объектно-ориентированный подход, на языке высокого уровня C#, под управлением семейства операционных систем Microsoft Windows. Модульность архитектуры позволяет эффективно расширять возможности и модернизацию программного комплекса.

**Ключевые слова:** программный комплекс, электронная система, тепловая модель, математическая модель, температурное распределение, стохастический, тепловая обратная связь, дестабилизирующие факторы.

Практика создания *электронных систем* (ЭлС), предназначенных для эксплуатации в условиях воздействия тепловых, механических, вибрационных, химических, радиационных и космических факторов и экстремальных параметров окружающей среды, предъявляет жесткие требования к системам автоматизированного проектирования ЭлС по всем видам дестабилизирующих воздействий. Тенденция к ужесточению и повышению требований к системам автоматизированного проектирования ЭлС и элементов ЭлС диктует, в свою очередь, повышение как уровня адекватности математического и компьютерного моделирования ЭлС, так и многофункциональности проектирования. Разработка *многофункционального программного комплекса* (МФПК) автоматизированного проектирования ЭлС является критически важной для создания новых конкурентоспособных ЭлС. При этом создание отечественных МФПК для проектирования ЭлС особенно актуально на современном этапе, поскольку обеспечивает решение важнейшей задачи по импортозамещению и импортобезопасности ПО.

Среди дестабилизирующих воздействий ЭлС одним из наиболее важных и значительных является тепловое воздействие, поскольку большая часть мощности (до 98 %), потребляемой в про-

цессе функционирования ЭлС, необратимо диссипируется в теплоту, приводящую к саморазогреву элементов и всей ЭлС в целом. В силу значительной зависимости электрических параметров элементов ЭлС от температуры одновременное воздействие как саморазогрева элементов ЭлС, так и повышенных температур окружающей среды приводит к высоким уровням температуры в элементах и к таким нежелательным последствиям, как выход характеристик ЭлС за пределы допустимых значений, потеря работоспособности, снижение быстродействия, надежности, помехозащищенности, невыполнение поставленной задачи. Возникающие в ЭлС температурные распределения обуславливают появление множества других эффектов различной физической природы, например, возникновение термомеханических напряжений в конструкции ЭлС, приводящих к короблениям, деформации, отслаиванию конструктивных элементов ЭлС, например, к отрыву шариковых выводов в корпусах BGA, короблению и отслаиванию слоев и токоведущих линий в слоях многослойной печатной платы, деформированию конструктивных элементов крепления и монтажа и пр. [1–3].

Таким образом, тепловые воздействия при функционировании ЭлС являются решающими ограничительными факторами на пути создания

конкурентоспособных ЭлС, а неадекватное тепловое проектирование ЭлС приводит к ошибкам при разработке ЭлС, обуславливая несоответствие параметров и характеристик ЭлС требуемым в ТЗ значениям. Поэтому важнейшей задачей при создании новых конкурентоспособных ЭлС, устойчивых к дестабилизирующим воздействиям, является разработка МФПК для *теплового проектирования ЭлС* (ТП ЭлС).

МФПК ТП ЭлС предназначен для обеспечения разработчиков элементов и ЭлС в целом высокоэффективным, адекватным реальности, с развитой пользовательской оболочкой инструментом, предназначенным для проведения всестороннего многовариантного математического и компьютерного моделирования тепловых процессов и температурных распределений в ЭлС.

Необходимо отметить, что существующие в настоящее время в зарубежных компаниях *программные комплексы* (ПК) для тепловых расчетов ЭлС (Beta Soft, TGM Thermal Analysis & Electronic System Cooling, SINDA, Ansys, FLOTHERM и другие, включая отдельные тепловые модули, встроенные в различные CAD/CAM/CAE) имеют следующие серьезные системные недостатки, не позволяющие использовать их в практике ТП ЭлС, в том числе и отечественной.

- Методы и модели, как и математические и компьютерные вычислительные алгоритмы, заложенные в зарубежных ПК, непользователю неизвестны и не раскрываются, что не позволяет судить об их адекватности, валидности и релевантности. Отсутствие информации об области и диапазоне применимости моделей, заложенных в конкретном ПК, сведений о математических моделях, методах решения уравнений математических моделей, используемых в данном ПК, не позволяет разработчикам новой техники доверять получаемым в ПК результатам и выводам и ориентироваться на них при создании новых ЭлС.

- Содержащиеся по умолчанию в зарубежных ПК физические и конструкционные параметры и характеристики материалов, элементной базы, конструкций ЭлС полностью привязаны к зарубежным технологиям, конкретным компаниям, конструкциям и принятым стандартам, которые неприменимы к отечественным материалам, технологиям, конструкциям и стандартам.

- Библиотеки элементов (микросхем, электро- и радиоэлементов), «вшитые» в структуру зарубежных ПК, недоступны для анализа специалистам по теплообмену и разработчикам элементной базы ЭлС, а их существенное отличие от отечественной номенклатуры элементов ЭлС делает их неприменимыми в отечественной практике теплового проектирования.

- Многофункциональность зарубежных ПК во многом ограничена вопреки декларируемому в них возможностям и зачастую не соответствует ре-

альной практике при попытках их применения к расчетам тепловых процессов в конкретных конструкциях ЭлС. Так, в зарубежных ПК отсутствует такая важная для адекватного теплового анализа ЭлС возможность, как моделирование температурных распределений в многослойной печатной плате ЭМ, в которой на сегодняшний день может насчитываться более нескольких десятков разнородных слоев. Например, в ПК Beta Soft количество слоев в печатной плате не превосходит трех, а в других зарубежных ПК имеется возможность задания только одного усредненного слоя, причем метод приведения множества слоев к одному не приводится.

- В зарубежных ПК отсутствуют (вопреки декларациям) принципиально важные возможности, которые необходимо учитывать при тепловом проектировании ЭлС, а именно моделирование

- температурных распределений в ЭлС при интервально-стохастической неопределенности тепловых, электрических и конструктивных параметров и характеристик ЭлС и стохастической неопределенности условий реальной окружающей среды;

- тепловой обратной связи, особенно сильно проявляющейся в элементах ЭлС с повышенной мощностью потребления, дестабилизирующих воздействиях и экстремальных окружающих условиях.

Приведенные системные недостатки зарубежных ПК не позволяют рекомендовать их к безусловному применению в практике теплового проектирования ЭлС и их элементов, что наряду с решением проблемы импортозамещения ПО еще больше повышает актуальность разработки и создания отечественных МФПК для теплового проектирования ЭлС.

#### **Требования к функциональным возможностям моделирования МФПК ТП ЭлС**

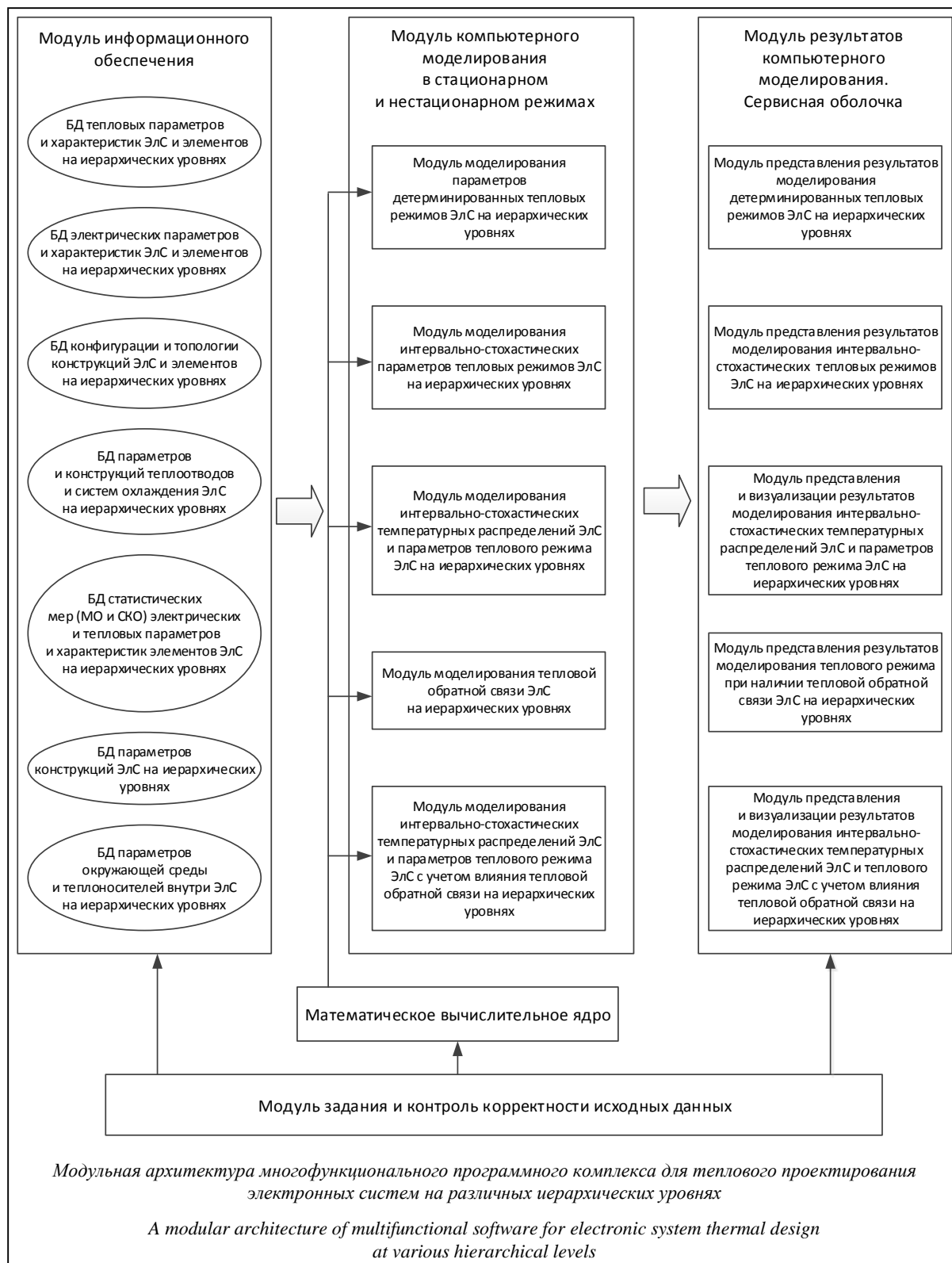
Разрабатываемый МФПК ТП ЭлС должен удовлетворять определенным требованиям и обладать следующими функциональными возможностями [4] (см. рисунок).

- Моделировать тепловые режимы и температурные распределения в ЭлС на различных иерархических уровнях, содержащих

- элементы ЭлС (микросхемы, электро- и радиоэлементы) – 1-й уровень;

- электронные модули, включающие многослойную печатную плату, установленные на ней микросхемы, электро- и радиоэлементы, теплоотводы, систему охлаждения электронных модулей, конструктивные элементы крепления и монтажа электронных модулей – 2-й уровень;

- панель (блок, субблок и т.п. по различной терминологии), объединяющая несколько электронных модулей вместе с конструктивными эле-



ментами крепления и монтажа, систему интерфейса, теплопроводы и систему охлаждения панели – 3-й уровень;

– стойку, в состав которой входят несколько панелей, элементы конструкции крепления и монтажа панелей в стойке, система интерфейса и охлаждения стойки – 4-й уровень.

При этом в МФПК ТП ЭлС должна быть обеспечена возможность моделирования всех конструктивных и физических особенностей элементов ЭлС и всей ЭлС в целом при их изготовлении, установке, монтаже и эксплуатации во всем диапазоне дестабилизирующих воздействующих факторов. Принципиально важной для адекватного моде-

лирования тепловых процессов в многослойных печатных платах является возможность учета всего множества разнородных в теплофизическом отношении слоев платы.

- Учитывать нелинейный, динамический и трехмерный характер тепловых процессов [3, 5, 6], развивающихся в ЭлС, сложный характер теплообмена между элементами ЭлС, а также разнородность конструктивных и теплофизических параметров элементов на различных иерархических уровнях ЭлС.

- Обладать возможностью моделирования тепловых режимов и процессов в ЭлС с учетом тепловой обратной связи, заключающейся во взаимовлиянии и взаимодействии электрических и тепловых процессов на различных иерархических уровнях ЭлС между собой, которая возникает в силу значительной зависимости электрических параметров и потребляемой элементами мощности [7] от температуры, а температуры элементов – от изменения мощностей потребления элементов [8–12]. Причем тепловая обратная связь в ЭлС может быть как положительной, так и отрицательной. В первом случае, если не принять специальные меры по охлаждению ЭлС, будет наблюдаться лавинообразный рост мощности потребления и температуры, ведущий к перегоранию элемента, во втором – снижение мощности потребления с ростом температуры при одновременном падении быстродействия.

- Учитывать интервально-стохастический характер тепловых процессов в ЭлС, который обусловливается статистическим технологическим разбросом тепловых, электрических и конструктивных параметров ЭлС, а также случайным характером параметров окружающей среды и/или теплоносителей (температуры, скорости и направления движения среды, влажности, плотности) в системе охлаждения ЭлС [5, 6, 13] на различных иерархических уровнях ЭлС.

- Моделировать тепловой режим ЭлС при наличии взаимно-обуславливающих эффектов, таких как тепловая обратная связь, интервально-стохастический разброс тепловых и электрических параметров ЭлС, воздействие дестабилизирующих механических, климатических и радиационных факторов.

- Учитывать влияние внешних тепловых воздействий на тепловые режимы ЭлС со стороны окружающих технических систем [2, 14].

- Архитектура МФПК ТП ЭлС должна содержать высокоэффективное математическое вычислительное ядро, а также развитую сервисную пользовательскую оболочку, отвечающую современным требованиям к сложным программным комплексам и системам [15–18]. Сервисная оболочка должна обеспечивать визуальную, наглядную и удобную для восприятия форму задания исходных данных, представления полученных результатов и визуализацию результатов в виде

цветных изображений изотерм температурных распределений и других тепловых характеристик ЭлС на различных иерархических уровнях.

Разрабатываемый в настоящее время в ФНЦ НИИСИ РАН программный комплекс МФПК ТП ЭлС реализуется в среде Microsoft Visual Studio, обеспечивающей объектно-ориентированный подход на языке высокого уровня C#, под управлением семейства операционных систем Microsoft Windows. Применение объектно-ориентированного подхода обеспечивает МФПК высокую степень модульности благодаря таким свойствам, как инкапсуляция, полиморфизм и позднее связывание. Модульность архитектуры МФПК позволяет эффективно расширять возможности и модернизировать программный комплекс, оказывая минимальное влияние на работоспособность системы в целом, что соответствует основным базовым принципам построения современных программных комплексов и обеспечивает создание надежного и эффективного комплекса моделирования.

#### Тепловые и математические модели тепловых процессов в ЭлС

Математические и тепловые модели, составляющие вычислительное ядро программного комплекса МФПК ТП ЭлС, должны в общем случае обладать достаточной общностью и сложностью, чтобы обеспечить адекватное численное моделирование тепловых процессов и тепловых режимов в сложных конструкциях ЭлС на всех иерархических уровнях с учетом всех сформулированных выше требований.

В силу чрезвычайной сложности поставленной задачи в качестве главного принципа моделирования тепловых процессов в МФПК ТП ЭлС авторами принят макроуровень моделирования [5, 18]. На макроуровне моделирования тепловые модели конструкции на каждом иерархическом уровне ЭлС представляются в виде направленных графов, содержащих вершины – температуры элементов ЭлС, и ветви – тепловые проводимости, моделирующие теплообмен кондукцией, конвекцией и излучением, и теплоемкости. Математические модели формируются на каждом иерархическом уровне в виде системы компонентных уравнений элементов ЭлС и системы матрично-топологических уравнений, описывающих структуру связей элементов в ЭлС.

Математическая модель на каждом иерархическом уровне, описывающая нелинейные, стохастические, нестационарные тепловые процессы в ЭлС и ее элементах, формируется для графовой тепловой модели в виде матрично-топологической системы нестационарных, нелинейных, интервально-стохастических уравнений в обыкновенных производных для каждого элементарного события  $\omega \in \Omega$  из пространства элементарных событий  $\Omega$  [13]:

$$C(\omega) \frac{dT(t, \omega)}{dt} + AG(T, \omega) A^T T(t, \omega) =$$

$$= J(T, t, \omega) + AG(T, t, \omega) T_a(t, \omega),$$

$$T(t, \omega) = T(0, \omega), (t, \omega) \in [0, \tau] \times \Omega,$$

где  $T = T(t, \omega)$  – вектор нестационарных интервально-стохастических температур в узлах графа тепловой модели ЭлС;  $C(\omega)$  – интервально-стохастическая матрица тепловых емкостей;  $G(T, \omega)$  – интервально-стохастическая матрица тепловых проводимостей ветвей графа тепловой модели, нелинейно зависящая от температур в узлах графа;  $J(T, t, \omega)$  – вектор нестационарных интервально-стохастических независимых мощностей тепловыделений в элементах ЭлС, нелинейно зависящих от температур в узлах графа;  $T_a(t, \omega)$  – вектор нестационарных интервально-стохастических независимых температур среды;  $A$  – матрица инцидентий графа,  $A^T$  – ее транспонированная матрица;  $T(0, \omega)$  – вектор начальных интервально-стохастических температур в узлах графа;  $\tau$  – рассматриваемый период времени;  $\omega \in \Omega$ ,  $(\Omega, U, P)$  – вероятностное пространство с  $\sigma$ -алгеброй подмножеств  $\Omega$  и вероятностью  $P$  на  $U$ .

Тепловые модели отдельных элементов конструкции ЭлС (многослойных печатных плат, теплоотводов и др.), для которых принципиальным является знание пространственных распределений температуры, также включаются в математическую и графовую тепловые модели ЭлС. В этом случае к обобщенной математической модели ЭлС (см. выше) добавляются математические модели на микроуровне [5], описывающие физические состояния и процессы в сплошных средах, аппаратом которых служат уравнения математической физики в частных производных – уравнения теплопроводности, гидро- и газовой динамики, полей электрического потенциала, диффузии носителей и пр. [19–21]. Тепловая обратная связь, моделирующая взаимодействие и взаимное влияние между собой теплового и электрического режимов ЭлС, учитывается в математической модели ЭлС в виде нелинейных зависимостей мощностей тепловыделений в элементах ЭлС, входящих в вектор  $J(T, t, \omega)$ , от температур элементов в векторе  $T = T(t, \omega)$  в узлах графа тепловой модели.

### Заключение

Разработанные концепции ТП ЭлС, математические и тепловые модели, положенные в основу МФПК ТП ЭлС – многофункционального программного комплекса для ТП ЭлС, позволяют создавать на практике методы математического и компьютерного моделирования сложных тепловых процессов в ЭлС и в их элементах, с помощью которых осуществляется полный цикл ТП сложных ЭлС различного назначения. Тепловая и математическая модели, описывающие тепловые процессы на различных иерархических уровнях ЭлС, учиты-

вают различные реальные аспекты тепловых процессов, с которыми сталкивается разработчик при создании и проектировании ЭлС: нелинейный, динамический, стохастический и нестационарный характер температурных распределений в элементах ЭлС и во всей ЭлС в целом; интервально-стохастический разброс электрических и тепловых параметров элементов ЭлС; воздействие тепловой обратной связи на тепловые и электрические режимы ЭлС; стохастический характер параметров окружающей среды. Сформулированные в статье основные требования к архитектуре и функциональным возможностям моделирования в МФПК ТП ЭлС позволяют создать высокоэффективный многофункциональный программный комплекс для ТП ЭлС и элементов ЭлС на всех иерархических уровнях.

### Литература

1. Конструкторско-технологическое проектирование электронной аппаратуры; [под общ. ред. В.А. Шахнова]. М.: Изд-во МГТУ им. Н.Э. Баумана, 2005. 568 с.
2. Чеканов А.Н. Расчеты и обеспечение надежности электронной аппаратуры. М.: КНОРУС, 2012. 440 с.
3. Ellison G.N. Thermal computations for electronics. Conductive, radiative, and convective air cooling. NY, CRC Press, 2011, 416 p.
4. Мадера А.Г. Концепция математического и компьютерного моделирования тепловых процессов в электронных системах // Программные продукты и системы. 2015. № 4. С. 79–86.
5. Мадера А.Г. Моделирование теплообмена в технических системах. М.: Науч. фонд «Первая исслед. лаборатория», 2005. 208 с.
6. Мадера А.Г., Кандалов П.И. Математическое моделирование интервально-стохастических тепловых процессов в технических системах при интервальной неопределенности определяющих параметров // Компьютерные исследования и моделирование. 2016. Т. 8. № 3. С. 501–520.
7. Бобков Г.Н., Мадера А.Г. Энергетические затраты, быстроедействие и проблема теплоотвода в микропроцессорах // Программные продукты и системы. 2013. № 4. С. 29–35.
8. Camarchia V., Capuleti F., Pirola M., Guettieri S.D.G., Ghione G. Self-consistent electrothermal modeling of class A, AB, and B power GaN HEMTs under modulated RF excitation. IEEE Transactions on microwave theory and techniques, 2007, vol. 55, no. 9, pp. 1824–1831.
9. Kuuse M., Loikkanen M., Bognar Gy. Theoretical investigation of thermal feedback effects in low power circuits. Therminec-2005, September, Belgirate, Italy, 2005.
10. Keller C.J., Antonetti V.W. Statistical thermal design for computer electronics. Electronic Packaging and Production, 1979, vol. 19, no. 3, pp. 55–62.
11. Rinaldi N., D'Alessandro V. Theory of electrothermal behavior of bipolar transistors: part II-two-finger devices. IEEE Transactions Electron Devices, 2005, vol. 52, no. 9, pp. 2022–2033.
12. Mueller O. Internal thermal feedback in four-poles especially in transistors. Proc. of the IEEE, 1964, vol. 52, pp. 924–930.
13. Madera A.G. Simulation of stochastic heat conduction processes. Intern. Jour. of Heat and Mass Transfer, 1994, vol. 37, no. 16, pp. 2571–2577.
14. Решетников В.Н. Космические телекоммуникации. Системы спутниковой связи и навигации. СПб: Ленинград. изд-во, 2010. 132 с.
15. Ли К. Основы САПР (CAD/CAM/CAE). СПб: Питер, 2004. 560 с.
16. Кандалов П.И. Программный комплекс моделирования стохастических температурных полей в технических системах STF-ElectronMod // Программные продукты и системы. 2016. № 4. С. 171–175.
17. Корячко В.П., Курейчик В.М., Норенков И.П. Теоретические основы САПР. М.: Энергоатомиздат, 1987. 400 с.

18. Норенков И.П. Основы автоматизированного проектирования. М.: Изд-во МГТУ им. Н. Э. Баумана, 2009. 430 с.
19. Тихонов А.Н., Самарский А.А. Уравнения математической физики. М.: Наука, 1999. 735 с.
20. Кляцкин В.И. Стохастические уравнения: теория и ее

приложения к акустике, гидродинамике и радиофизике. М.: Физматлит, 2008. Т. 1–2.

21. Алиев А.В., Мищенко О.В. Математическое моделирование в технике. Ижевск.: Изд-во Ин-та комп. исслед., 2012. 456 с.

Software & Systems

DOI: 10.15827/0236-235X.119.367-372

Received 03.04.17

2017, vol. 30, no. 3, pp. 367–372

## A SOFTWARE COMPLEX FOR ELECTRONIC SYSTEM THERMAL DESIGN: REQUIREMENTS FOR ARCHITECTURE AND FUNCTIONAL POSSIBILITIES OF MODELING

**A.G. Madera**<sup>1</sup>, *Dr.Sci (Engineering), Professor, agmprof@mail.ru*

**V.N. Reshetnikov**<sup>1</sup>, *Dr.Sc. (Physics and Mathematics), Professor, rvn\_@mail.ru*

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

**Abstract.** The paper considers fundamental concepts of mathematical and computer modeling, which are the basis of the development and creation of a multifunctional software package for thermal design of complex electronic systems. It shows the basic flaws of foreign thermal design software systems, which are not applicable in practice of designing and creating competitive electronic systems.

The article establishes the requirements for a multifunctional software package for adequate design of thermal processes and temperature distributions in real electronic systems under real conditions of operation and functioning. Namely, the software package should provide the ability to model thermal processes in electronic systems that are nonlinear, nonstationary, three-dimensional, interval stochastic, as well as take into account the influence of thermal feedback, design and installation of elements in electronic systems, effects of destabilizing mechanical, climatic and radiation factors.

The architecture of the multifunctional software package should contain a highly efficient mathematical computing core, a developed service user interface that meets modern requirements for complex software packages and systems. The service shell, in turn, should provide a visual, intuitive and easy-to-understand form of setting initial data in the form of color images of temperature distributions and other thermal characteristics at different hierarchical levels.

The software package is implemented in Microsoft Visual Studio that provides an object-oriented approach in the high-level language C# powered by Microsoft Windows operating system family. Modular architecture allows expanding the possibilities and upgrading the software complex effectively.

**Keywords:** software package, electronic system, thermal model, mathematical model, temperature distribution, stochastic, thermal feedback, destabilizing factors.

### References

1. *Konstruktorско-tekhnologicheskoe proektirovaniye elektronnoy apparatury* [Electronics Design and Engineering]. V.A. Shakhnov (Ed.). Moscow, Bauman MSTU Publ., 2005, 568 p.
2. Chekanov A.N. *Raschety i obespecheniye nadezhnosti elektronnoy apparatury* [Electronics Calculations and Reliability Control]. Moscow, KNORUS Publ., 2012, 440 p.
3. Ellison G.N. *Thermal computations for electronics. Conductive, radiative, and convective air cooling*. NY, CRC Press, 2011, 416 p.
4. Madera A.G. The concept of mathematical and computer simulation of thermal processes in electronic systems. *Programmnye produkty i sistemy* [Software & Systems]. 2015, no. 4 (112), pp. 79–86 (in Russ.).
5. Madera A.G. *Modelirovaniye teploobmena v tekhnicheskikh sistemakh* [Heat Exchange Simulation in Engineering Systems]. Moscow, Nauch. fond "Pervaya issled. lab.", 2005, 208 p.
6. Madera A.G., Kandalov P.I. Mathematical modeling of the interval stochastic thermal processes in technical systems at the interval indeterminacy of the determinative parameters. *Computer Research and Modeling*. 2016, vol. 42, no. 3, pp. 257–268.
7. Bobkov G.N., Madera A.G. Energy costs, speed and heat sink in microprocessors. *Programmnye produkty i sistemy* [Software & Systems]. 2013, no. 4, pp. 29–35 (in Russ.).
8. Camarchia V., Capuleti F., Pirola M., Guettieri S.D.G., Ghione G. Self-Consistent Electrothermal Modeling of Class A, AB, and B Power GaN HEMTs Under Modulated RF Excitation. *IEEE Trans. on Microwave Theory and Techniques*. 2007, vol. 55, no. 9, pp. 1824–1831.
9. Kuuse M., Loikkanen M., Bognar Gy. Theoretical investigation of thermal feedback effects in low power circuits. *Therminic-2005*. Belgirate, Italy, 2005.
10. Keller C.J., Antonetti V.W. Statistical thermal design for computer electronics. *Electronic Packaging and Production*. 1979, vol. 19, no. 3, pp. 55–62.
11. Rinaldi N., D'Alessandro V. Theory of electrothermal behavior of bipolar transistors: part II-two-finger devices. *IEEE Trans. Electron Devices*. 2005, vol. 52, no. 9, pp. 2022–2033.
12. Mueller O. Internal thermal feedback in four-poles especially in transistors. *Proc. of the IEEE*. 1964, vol. 52 (8), pp. 924–930.
13. Madera A.G. Simulation of stochastic heat conduction processes. *Int. Jour. of Heat and Mass Transfer*. 1994, vol. 37, no. 16, pp. 2571–2577.
14. Reshetnikov V.N. *Kosmicheskije telekommunikatsii (sistemy sputnikovoy svyazi i navigatsii)* [Space Telecommunications (Satellite Communication and Navigation Systems)]. St. Petersburg, 2010, 132 p.
15. Li K. *Osnovy SAPR (CAD/CAM/CAE)* [CAD Basics (CAD/CAM/CAE)]. St. Petersburg, Piter Publ., 2004.
16. Kandalov P.I. Software modeling of stochastic temperature fields in stf-electron mod technical systems. *Programmnye produkty i sistemy* [Software & Systems]. 2016, no. 4 (29), pp. 171–175 (in Russ.).
17. Koryachko V.P., Kureychik V.M., Norenkov I.P. *Teoreticheskie osnovy SAPR* [CAD Theoretical Basis]. Moscow, Energoatomizdat Publ., 1987.
18. Norenkov I.P. *Osnovy avtomatizirovannogo proektirovaniya* [The Basis of Automated Design Engineering]. 4th ed., Moscow, Bauman MSTU, 2009.
19. Tikhonov A.N., Samarsky A.A. *Uravneniya matematicheskoy fiziki* [Equations of Mathematical Physics]. Moscow Nauka Publ., 1999.
20. Klyatskin V.I. *Stokhasticheskie uravneniya. Teoriya i ee prilozheniya k akustike, gidrodinamike i radiofizike* [Stochastic Equations. Theory and its Applications to Acoustics, Hydrodynamics and Radiophysics]. Moscow Fizmatlit Publ., 2008.
21. Aliyev A.V., Mishchenko O.V. *Matematicheskoe modelirovaniye v tekhnike* [Mathematical Simulation in Engineering]. Izhevsk, Institute of Computer Science Publ., 2012.

УДК 004.8

DOI: 10.15827/0236-235X.119.373-377

Дата подачи статьи: 13.04.17

2017. Т. 30. № 3. С. 373–377

## **ИДЕНТИФИКАЦИЯ СОСТОЯНИЯ СЛОЖНОЙ ТЕХНИЧЕСКОЙ СИСТЕМЫ В УСЛОВИЯХ НЕОПРЕДЕЛЕННОСТИ ИЗМЕРИТЕЛЬНОЙ ИНФОРМАЦИИ**

*А.А. Генов, д.т.н., профессор, ведущий научный сотрудник, kt-mati@mail.ru  
(Центр визуализации и спутниковых информационных технологий ФНЦ НИИСИ РАН,  
Нахимовский просп., 36, корп. 1, г. Москва, 117218, Россия);*

*К.Д. Русаков, младший научный сотрудник, rusakov.msk@yandex.ru  
(Институт проблем управления им. В.А. Трапезникова РАН, ул. Профсоюзная, 65, г. Москва, 117997, Россия);*

*С.Ш. Хиль, к.т.н., доцент, skhill@mail.ru  
(Московский авиационный институт (национальный исследовательский университет),  
Волоколамское шоссе, 4, г. Москва, 125993, Россия)*

В статье обозначена актуальная задача идентификации функционального состояния сложной технической системы в условиях повышенной зашумленности с помощью перехода к другому пространству признаков наблюдаемого состояния, определено понятие «функциональное состояние».

Задача оценки функционального состояния показана как дуальная задача идентификации и распознавания образов. Уделено внимание возможному выбору параметров математических моделей, а также их структур в качестве новых признаков наблюдаемого состояния. Для решения задачи параметрической идентификации выбран метод наименьших квадратов, показывающий смену функционального состояния.

В задачах контроля и испытания сложными техническими системами немаловажную роль играет процесс анализа измерительной информации. Только пройдя все этапы обработки, можно принять решение о том или ином состоянии объекта испытаний. Однако сделать это сложно в случае поступления информации очень низкого качества. Устранение данной проблемы сводится в основном к поиску решений по формированию и совершенствованию соответствующих правил принятия управленческих решений. При этом в штатных условиях и ситуациях создание таких правил во многих случаях не вызывает особых проблем, однако в нештатных условиях сделать это проблематично. Анализ состояния сложных технических объектов или их подсистем по результатам измерений предполагает решение задач структурной и параметрической идентификации.

Управление сложными техническими объектами и их испытание характеризуются неопределенностью воздействия внешних факторов, аномальными и случайными погрешностями измерений. Для обеспечения требуемого качества анализа состояния их подсистем необходимо формирование принципов адаптивного управления процессом идентификации, методик оптимального выбора структуры моделей исходя из качества данных измерений. Особенно это актуально при автоматизации вторичной обработки на объектах с терминальным управлением.

**Ключевые слова:** идентификация состояния, функциональное состояние, сложная техническая система, повышенная зашумленность, распознавание образов, параметрическая идентификация, метод наименьших квадратов.

Повышение уровня автоматизации процессов управления состояниями сложных технических систем получает активное развитие. Решение данной проблемы сводится в основном к поиску способов формирования и совершенствования соответствующих правил принятия управленческих решений. При этом в штатных условиях и ситуациях создание таких правил во многих случаях не вызывает особых проблем, однако в нештатных условиях сделать это проблематично.

В задачах контроля и испытания сложных технических систем немаловажную роль играет анализ измерительной информации. Только пройдя все этапы обработки, можно принять решение о том или ином состоянии объекта испытаний. Однако это сложно сделать, если объем измерительной информации недостаточен и велика помеховая составляющая.

### **Дефиниция функционального состояния**

Одним из показателей технического состояния объекта является правильное/неправильное функционирование. Заметим, что при функционировании объекта испытаний система управления задей-

ствует в нем различные функции, узлы, агрегаты. Из этого следует, что под воздействием системы управления объект будет находиться в различных устойчивых и равновесных фазах своего функционирования, которые будем определять как функциональные состояния объекта испытаний. Понятие функционального состояния является математической интерпретацией режимов функционирования сложного технического объекта.

В [1] отмечено, что модель объекта должна отвечать целям ее использования. Применительно к задачам ситуационного управления это означает, что должно существовать такое соответствие между множеством состояний выбранной модели и множеством возможных целей управления объектом, которое позволяет однозначно идентифицировать каждую цель во множестве этих состояний. В математической форме это требование может быть записано в следующем виде:  $R \subset Q$ , где  $Q$  – отношение эквивалентности, определенное на множестве состояний модели таким образом, чтобы множество его классов на этом множестве взаимно-однозначно соответствовало множеству имен целей в управлении объектом. Все модели, удовлетворяющие данному условию, могут быть



использованы для решения задач ситуационного управления поведением объектов в пространстве их состояний. Это означает, что им присуще свойство *функциональности*, выражающееся в наборе возможностей (функций), которые предоставляет (выполняет) данный объект. На этом основании такие модели будем называть функциональными моделями объекта, а используемые в них состояния – функциональными состояниями. Таким образом, *функциональное состояние* – это атрибут детерминированной модели управляемых изменений в процессах функционирования технического объекта, который

а) характеризует в модели устойчивую и равновесную фазу этих процессов относительно заданного подмножества воздействий внешней среды и выбранного интервала времени существования объекта;

б) имеет уникальное имя, семантическую интерпретацию и идентифицируется в значениях параметров этих процессов, знание которых и оказываемых на объект управляющих воздействий в выбранный момент времени является необходимым и достаточным условием для прогнозирования будущих значений этих параметров через определенные в модели интервалы времени с точностью, позволяющей однозначно идентифицировать соответствующие им имена этого атрибута.

#### Постановка задачи оценки функционального состояния как задачи идентификации

Постановка задачи в общем виде будет следующей.

1. Система, относящаяся к классу динамических систем:  $\sum_1: \dot{g} = f_g(g, t, F, u)$ , где  $g$  – вектор состояния динамической системы  $g^T = [g_1 \ g_2 \ \dots \ g_n]$ ,  $g_i$  – параметры вектора состояния;  $u$  – вектор внешних воздействий динамической системы  $u^T = [u_1 \ u_2 \ \dots \ u_n]$ ;  $F$  – множество функциональных состояний динамической системы  $F_i \in F$ ,  $F_i = f(t, \Pi, \nu)$ , где  $\Pi$  – правило принятия решения об изменении функциональных состояний, которое выглядит следующим образом:

если  $g \in G_1$ , то  $F \rightarrow F_1$ ,  
 если  $g \in G_2$ , то  $F \rightarrow F_2$ ,  
 $\vdots$   
 если  $g \in G_n$ , то  $F \rightarrow F_n$

при следующих ограничениях:

при  $g \in G_1$   $T \geq T_1$   
 при  $g \in G_2$   $T \geq T_2$   
 $\vdots$   
 при  $g \in G_n$   $T \geq T_n$

2. Модель наблюдений:  $y = H \cdot g + \xi(t)$ , где  $y$  – вектор наблюдений;  $H$  – матрица связи наблюдаемых параметров и вектора состояния системы;  $\xi(t)$  – внешнее воздействие.

3. Модель измерений с аддитивной помехой:  $z = y + n(t)$ .

По результатам измерений  $z(t)$ ,  $t \in \Omega$ , определить функциональное состояние системы  $F_i = f(t, \Pi, \nu)$  на интервале времени  $t \in [t_k \dots t_n]$  при максимуме вероятности правильного принятия решения о функциональном состоянии системы:  $P_{\text{прФС}} \rightarrow \max$ .

Особенностью и новизной данной постановки является наличие зависимости изменений функциональных состояний от соответствующих правил принятия решений.

#### Постановка задачи оценки функционального состояния как задачи распознавания образов. Выбор признаков

Для решения данной задачи необходимо решить задачу классификации.

1. Известна модель функционирования объекта испытаний  $G(t) = \{g_k(t)\}$ ,  $k = 1, \dots, q$ , где  $g_k(t)$  – внутренние параметры состояния объекта.

2. На основе априорного анализа задано множество классов функционального состояния объекта испытаний:  $F = \{F_l\}$ ,  $l = 1, \dots, r$ .

3. Известно множество признаков  $\Pi = \{\pi_j\}$ ,  $j = 1, \dots, m$ , обеспечивающих наблюдаемость функциональных состояний при выбранном методе. При этом признаки являются функциями от внутренних параметров состояния объекта испытаний  $\pi_j = z[g_k(t)]$ .

4. Задано отношение  $H: F \rightarrow \Pi$  между множествами классов функциональных состояний  $F$  и признаков контроля в виде соответствующей матрицы.

5. Задано множество измеряемых параметров  $\Lambda = \{\lambda_i\}$ ,  $i = 1, \dots, n$ , позволяющее определить значения признаков. Известно, что в общем случае наблюдаемые параметры являются нелинейными функциями от параметров состояния  $\lambda(t) = h[g(t)] + \Delta\lambda$ , где  $\Delta\lambda$  – погрешность измерений, и признаков контроля  $\lambda_j = s(\pi)$ .

6. Измерительная система представлена в виде операторов измерений  $L: \{g_k(t)\} \rightarrow \lambda(t)$ , операторов первичной  $L_1: \lambda(t) \rightarrow \{\pi^*; t\}$  и вторичной  $L_2: \{\pi^*; t\} \rightarrow \{F_i^*\}$  обработки измерений, где каждый из операторов задается на множестве методов  $\{M\}$  и алгоритмов  $\{A\}$  обработки и анализа информации.

7. Задан показатель эффективности решения задачи определения функционального состояния  $P_{\text{прФС}} \rightarrow \max$ .

Стоит отметить, что вероятность правильного принятия решения уменьшается при увеличении шумовой составляющей в измерительной информации. Для ее повышения целесообразно перейти к другому пространству признаков наблюдаемого состояния. В качестве новых признаков будем рассматривать модели соответствующих функциональных состояний и параметры этих моделей.



## Численное моделирование

На примере гипотетической динамической системы, описывающей изменение заряда аккумуляторной батареи СА (рис. 1), покажем результат применения различных моделей функционального состояния в условиях высокого уровня помех (отношение сигнал/шум 20 дБ).

На рисунке 2 представлена часть информации, показывающей переход из одного функционального состояния в другое в условиях шума.

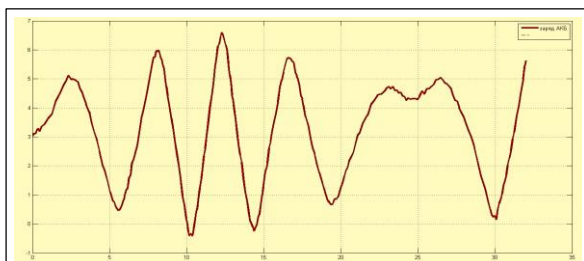


Рис. 1. Изменение заряда аккумуляторной батареи СА

Fig. 1. Changing CA battery charge

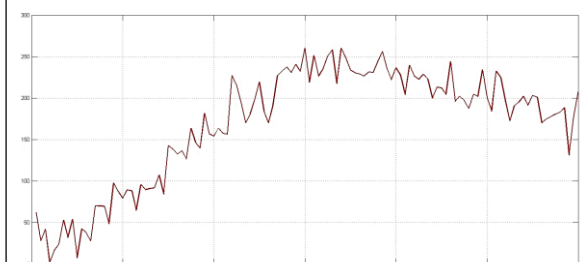


Рис. 2. Измерительная информация

Fig. 2. Measurement data

Интуитивно можно сделать вывод, что при высоком уровне шума более «грубая» модель точнее опишет процесс. Применяя модель более низкого порядка, например полином первой степени, отслеживаем изменение параметров этого полинома и, как только эти параметры изменились, можем сделать вывод о смене функционального состояния. Так как наиболее распространенным методом оценивания параметров, служащим базовым подходом к параметрической идентификации, является метод наименьших квадратов, который в предположении линейности и дискретности во времени объекта приводит к наиболее простым и универсальным решениям, воспользуемся им. Задача состоит в следующем: по имеющимся выборочным данным наблюдений за входным и выходным сигналами с интервалом дискретизации  $\Delta t$  требуется оценить значения параметров, обеспечивающих минимум величины функционала невязки между модельными и фактическими данными.

Таким образом, структура модели имеет вид  $y = a_1x + a_2$  при заданных коэффициентах  $a_1 = 4$ ,  $a_2 = 5$ .

Критерий минимума среднеквадратичной ошибки в этом случае определяется функционалом

$$\min_{a_i} J(a_i) = \min_{a_1, a_2, a_3} \sum_{j=1}^N [y_j - (a_2 + a_1x_j)]^2.$$

Система уравнений для нахождения коэффициентов  $a_i$  принимает вид

$$\begin{cases} \frac{\partial J}{\partial a_2} = -2 \sum_{j=1}^N (y_j - a_2 - a_1x_j) = 0, \\ \frac{\partial J}{\partial a_1} = -2 \sum_{j=1}^N (y_j - a_2 - a_1x_j) x_j = 0. \end{cases}$$

Преобразуя данное соотношение, получим следующую систему в матричном виде:

$$\begin{bmatrix} N & \sum_{j=1}^N x_j \\ \sum_{j=1}^N x_j & \sum_{j=1}^N x_j^2 \end{bmatrix} \begin{bmatrix} a_2 \\ a_1 \end{bmatrix} = \begin{bmatrix} \sum_{j=1}^N y_j \\ \sum_{j=1}^N y_j x_j \end{bmatrix}.$$

Для удобства примем следующие обозначения:

$$\begin{bmatrix} N & \sum_{j=1}^N x_j \\ \sum_{j=1}^N x_j & \sum_{j=1}^N x_j^2 \end{bmatrix} = X, \quad \begin{bmatrix} \sum_{j=1}^N y_j \\ \sum_{j=1}^N y_j x_j \end{bmatrix} = Y.$$

Таким образом, решением системы являются искомые выражения для коэффициентов уравнения

$$a_i: \begin{bmatrix} a_2 \\ a_1 \end{bmatrix} = X^{-1}Y.$$

Результаты представлены на рисунке 3.

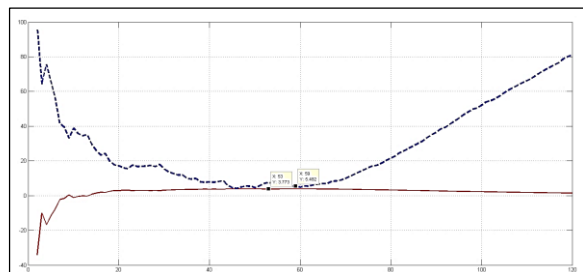


Рис. 3. Идентификация параметров линейной модели

Fig. 3. Identification of linear model parameters

## Заключение

Результаты эксперимента показывают, что в условиях шума в измерительной информации примерно на 60-й секунде происходит изменение параметров модели в виде полинома первой степени, что говорит о смене функционального состояния системы. Стоит отметить, что в зависимости от динамики процесса, уровня шумовых воздействий и режимов функционирования применяемые модели будут различными.

Таким образом, другие, более грубые модели, а также их параметры можно использовать в качестве признаков наблюдаемого состояния системы в

условиях различного качества измерительной информации, а следовательно, получить достоверную оценку функционального состояния системы.

### Литература

1. Кириллов Н.П. Функциональное состояние технического объекта. Дефиниция понятия // Авиакосмическое приборостроение. 2010. № 10. С. 31–40.
2. Васильев В.В. Математические методы анализа летательных аппаратов как объектов управления и испытания. М.: Изд-во МО РФ, 1992.
3. Дилигенская А.Н. Идентификация объектов управления. Самара: Изд-во СГТУ, 2009. 136 с.
4. Русаков К.Д. К оценке функционального состояния объектов РКТ в условиях различного качества измерительной информации // Нейрокомпьютеры и их применение: тез. докл. XIV Всерос. науч. конф. М.: Изд-во МГППУ, 2016. С. 89.
5. Русаков К.Д., Хиль С.Ш. О задаче выбора признаков наблюдаемого состояния сложного динамического объекта в условиях различного качества измерительной информации // Нейрокомпьютеры и их применение: тез. докл. XV Всерос. науч. конф. М.: Изд-во МГППУ, 2017. С. 246–248.
6. Рожнов А.В., Кривоножко В.Е., Лычев А.В. Построение гибридных интеллектуальных информационных сред и компо-

нентов экспертных систем на основе обобщенной модели анализа среды функционирования // Нейрокомпьютеры: разработка, применение. 2013. № 6. С. 3–12.

7. Кириллов Н.П. Структурно-функциональная модель причинно-следственных закономерностей управляемого поведения технических систем // Тр. СПИИРАН. СПб: Наука, 2008. Вып. 5. С. 285–299.

8. Васильев В.В., Галаев С.А., Лесниченко Р.И., Мезенцев А.В., Потюпкин А.Ю., Рудаев С.А. Методологические основы испытаний сложных технических систем. М.: Изд-во ВА ВВСН им. Петра Великого, 2013. 286 с.

9. Потюпкин А.Ю., Чечкин А.В. Интеллектуализация сложных технических систем: монография. М.: Изд-во МО РФ, 2013. 289 с.

10. Дмитриев А.К., Мальцев П.А. Основы теории построения и контроля сложных систем. Л.: Энергоатомиздат, 1988. 192 с.

11. Рожнов А.В. Творческие материалы «круглого стола». Часть II. Системная интеграция и моделирование новых эффектов в сфере интеллекта // Нейрокомпьютеры: разработка, применение. 2016. № 3. С. 3–12.

12. Лобанов И.А. Регенеративный анализ в задаче формирования адаптивного информационно-управляющего пространства // Нейрокомпьютеры и их применение: тез. докл. XIV Всерос. науч. конф. М.: Изд-во МГППУ, 2016. С. 26–28.

Software & Systems

DOI: 10.15827/0236-235X.119.373-377

Received 13.04.17

2017, vol. 30, no. 3, pp. 373–377

## IDENTIFICATION OF A COMPLEX TECHNICAL SYSTEM FUNCTIONAL STATE UNDER CONDITIONS OF MEASUREMENT DATA AMBIGUITY

**A.A. Genov**<sup>1</sup>, Dr.Sc. (Engineering), Professor, Leading Researcher, kt-mati@mail.ru

**K.D. Rusakov**<sup>2</sup>, Junior Researcher, rusakov.msk@yandex.ru

**S.Sh. Hill**<sup>3</sup>, Ph.D. (Engineering), Associate Professor, skhill@mail.ru

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

<sup>2</sup> V.A. Trapeznikov Institute of Control Sciences of RAS, Profsoyuznaya St. 65, 117997, Russian Federation

<sup>3</sup> Moscow Aviation Institute (National Research University), Volokolamskoe Highway, 4, Moscow, 125993, Russian Federation

**Abstract.** The article considers a topical problem of identification of a complex technical system functional state in conditions of increased noisiness using a transition to another attribute space of the observed state. The authors define the term “functional state”.

The paper shows the problem of estimating a functional state as a dual problem of identification and pattern recognition. It pays attention to a possible selection of parameters of mathematical models and their structures as new signs of the observed state. As an example, in order to solve the parametric identification problem, the authors have selected the method of least squares. It shows the changing of the functional state.

The process of analyzing measuring information plays an important role in the tasks of monitoring and testing by complex technical systems. Only after passing through all stages of processing, it is possible to decide on the condition of the test object. However, it is difficult to do in case of poor quality information. Elimination of this problem is mainly comes to searching solutions in order to form and improve relevant rules for management decision-making. At the same time, in normal conditions and situations, creation of such rules in many cases does not cause any special problems. Nevertheless, in abnormal conditions it is problematic. The state analysis of complex technical objects or their subsystems according to the measurement results assumes the solution of structural and parametric identification problems.

Management of complex technical objects and their testing is characterized by uncertainty of external influence, anomalous and accidental measurement errors. To ensure the required quality of their subsystems state analysis, it is necessary to form the principles of identification process adaptive control, the methods of an optimal choice of a model structure based on the quality of measurement data. This is particularly topical for automation of secondary processing at terminal control facilities.

**Keywords:** state identification, functional state, complex technical system, increased noisiness, pattern recognition, parametric identification, least squares method.

### References

1. Kirillov N.P. Functional condition of technical object concept definition. *Aviakosmicheskoe priborostroenie* [Aerospace Instrument-Making]. 2010, no. 10, pp. 31–40 (in Russ.).
2. Vasilev V.V. *Matematicheskie metody analiza letatelnykh apparatov kak obektov upravleniya i ispytaniya* [Mathematical Methods for Analysis of Aircrafts as Controlled and Test Objects]. Moscow, MO RF Publ., 1992.
3. Diligenskaya A.N. *Identifikatsiya obektov upravleniya* [Identification of Controlled Objects]. SSTU Publ., 2009.
4. Rusakov K.D. To the evaluation of the functional state of rocket and space equipment objects under conditions of different quality of measuring information. *XIV Vseros. nauch. konf. "Neyrokompyutery i ikh primeneniye"* [Proc. 14th All-Russian Science Conf. "Neurocomputers and Their Application"]. Moscow, MSUPE Publ., 2016, p. 89 (in Russ.).
5. Rusakov K.D., Khil S.Sh. On the problem of selecting the characteristics of the observed state of a complex dynamic object under conditions of different quality of the measurement information. *XV Vseros. nauch. konf. "Neyrokompyutery i ikh primeneniye"* [Proc. 15th All-Russian Science Conf. "Neurocomputers and Their Application"]. Moscow, MSUPE Publ., 2017, pp. 246–248 (in Russ.).
6. Rozhnov A.V., Krivonozhko V.E., Lychev A.V. Construction a hybrid intelligent information framework and components of expert systems using the generalized data envelopment analysis model. *Neyrokompyutery: razrabotka, primeneniye* [Jour. Neurocomputers]. 2013, no. 6, pp. 3–12 (in Russ.).
7. Kirillov N.P. The skeleton-functional model of the cause-effect regularities of the managed behavior of technical systems. *Trudy SPIIRAN* [SPIIRAS Proc.]. 2008, St. Petersburg, Nauka Publ., iss. 5, pp. 285–299 (in Russ.).
8. Vasilev V.V., Galaev S.A., Lesnichenko R.I., Mezentshev A.V., Potyupkin A.Yu., Rudaev S.A. *Metodologicheskie osnovy ispytaniy slozhnykh tekhnicheskikh sistem* [Methodological Basis for Testing Complex Technical Systems]. Moscow, Peter the Great Strategic Rocket Forces Academy Publ., 2013, 286 p.
9. Potyupkin A.Yu., Chechkin A.V. *Intellektualizatsiya slozhnykh tekhnicheskikh sistem* [Intellectualization of Complex Technical Systems]. Monograph, MO RF Publ., Moscow, 2013, 289 p.
10. Dmitriev A.K., Maltsev P.A. *Osnovy teorii postroeniya i kontrolya slozhnykh sistem* [Fundamentals of the Theory of Construction and Control of Complex Systems]. Leningrad, Energoatomizdat Publ., 1988, 192 p.
11. Rozhnov A.V. Round-table discussion. P. II. System integration and artificial intelligence new effects simulation. *Neyrokompyutery: razrabotka, primeneniye* [Jour. Neurocomputers]. 2016, no. 3, pp. 3–12 (in Russ.).
12. Lobanov I.A. Regenerative analysis in the task of forming an adaptive information-control space. *XIV Vseros. nauch. konf. "Neyrokompyutery i ikh primeneniye"* [Proc. 14th All-Russian Science Conf. "Neurocomputers and Their Application"]. Moscow, MSUPE Publ., 2016, pp. 26–28 (in Russ.).

### Примеры библиографического описания статьи

1. Генов А.А., Русаков К.Д., Хиль С.Ш. Идентификация состояния сложной технической системы в условиях неопределенности измерительной информации // Программные продукты и системы. 2017. Т. 30. № 3. С. 373–377. DOI: 10.15827/0236-235X.119.373-377.
2. Genov A.A., Rusakov K.D., Hill S.Sh. Identification of a complex technical system functional state under conditions of measurement data ambiguity. *Programmnye produkty i sistemy* [Software & Systems]. 2017, vol. 30, no. 3, pp. 373–377 (in Russ.). DOI: 10.15827/0236-235X.119.373-377.

УДК 004.4'233

DOI: 10.15827/0236-235X.119.378-383

Дата подачи статьи: 04.05.17

2017. Т. 30. № 3. С. 378–383

## ПРОБЛЕМЫ ОТЛАДКИ МНОГОПРОЦЕССНЫХ СИСТЕМ

*В.А. Галатенко, д.ф.-м.н., зав. сектором, galat@niisi.ras.ru;*

*К.А. Костюхин, к.ф.-м.н., старший научный сотрудник, kost@niisi.ras.ru*

*(Федеральный научный центр Научно-исследовательский институт системных исследований  
РАН (ФНЦ НИИСИ РАН), Нахимовский просп., 36, корп. 1, г. Москва, 117218, Россия)*

Статья посвящена проблеме отладки сложных многопроцессных систем. Разработка качественных сложных аппаратно-программных систем – длительный, трудоемкий процесс. Считается, что около половины времени уходит на отладку. Переход на многоядерные процессорные архитектуры сделал параллелизм нормой, равно как и специфичные для параллельных систем ошибки. Это делает отладку еще более важной и одновременно более сложной.

Основная особенность многопроцессных систем – использование сложных асинхронных взаимодействий между компонентами системы, эта особенность влияет на подходы к отладке систем, определяет выбор инструментов и методов отладки. Сложность отладки также обусловлена количеством и разнородностью компонентов многопроцессных систем, часть из которых могут быть аппаратными. Традиционный набор инструментов отладки в широком смысле (интерактивный отладчик, трассировщик, библиотеки самоконтроля, воспроизведения и отладки производительности) не теряют своей актуальности, но варианты их применения изменяются.

В работе сначала рассматриваются существующие средства и методы отладки, выделяются их достоинства и недостатки. Затем описываются существующие проблемы отладки многопроцессных систем и предлагается архитектура отладчика многопроцессных систем. В заключении представлены выводы авторов.

**Ключевые слова:** контролируемое выполнение, отладка, мониторинг, трассировка, многопроцессные системы.

Разработка качественных сложных аппаратно-программных систем – длительный и трудоемкий процесс. Считается, что около половины времени уходит на отладку (см., например, [1]). Переход на многоядерные процессорные архитектуры сделал параллелизм нормой, равно как и специфичные для параллельных систем ошибки. Это делает отладку еще более важной и одновременно более сложной.

Основная особенность многопроцессных систем – использование сложных, асинхронных взаимодействий между компонентами системы, эта особенность влияет на подходы к отладке систем, определяет выбор инструментов и методов отладки. Сложность отладки также обусловлена количеством и разнородностью компонентов многопроцессных систем, часть из которых могут быть аппаратными. Традиционный набор инструментов отладки (интерактивный отладчик, трассировщик, библиотеки самоконтроля, воспроизведения и отладки производительности) не теряет своей актуальности, но варианты применения инструментов отладки изменяются.

### Основные способы отладки, их возможные сочетания и особенности

**Интерактивная отладка.** Традиционно под отладкой понимается прежде всего интерактивная отладка. В то же время, чем сложнее отлаживаемая система, тем менее эффективна подобная отладка. Слишком много факторов приходится принимать во внимание и много данных необходимо проанализировать, чтобы ответить, например, на простые, но важные вопросы: случались ли уже в системе ошибки и каковы их первопричины.

Наиболее популярным инструментом интерактивной отладки является GDB [2]. На момент напи-

сания статьи он оставался однопроцессным; многопроцессные возможности были на начальной стадии реализации, и, как показал практический опыт, их нельзя было считать работоспособными. В то же время развитие GDB в интересующем авторов направлении активно обсуждается в литературе.

В [3] предложена реализация так называемой безостановочной многопроцессной отладки, ориентированной на слабосвязанные системы. Безостановочность заключается в возможности отладки отдельного потока в операционной среде Linux без остановки процесса целиком.

Такая функциональность достигается за счет расширения возможностей агента отладки, а также существенной переработки внутреннего устройства отладчика, создания механизма асинхронного обслуживания событий и контроля применимости отладочных действий. Этот шаг очень важен и для развития многопроцессной отладки.

Интерактивной отладке присуща также такая проблема, как внесение недопустимо серьезных возмущений в процесс функционирования отлаживаемых систем. Этот процесс и без отладчика, вообще говоря, является недетерминированным, поэтому заранее нельзя сказать, проявятся ли ошибки в данном конкретном запуске; отладочные действия усугубляют недетерминизм, что способно привести к исчезновению предмета отладки.

Проблеме минимизации возмущений, вносимых отладочными инструментами категории GDB, посвящена статья [4]. В ней описываются различные механизмы, необходимые для многопроцессной отладки: централизованный ввод-вывод отладочных сообщений, параллельный сбор данных во время выполнения, использование встроенных в приложение отладочных средств и перехват передачи сообщений между процессами.

**Командный язык отладки.** При отладке многопроцессных систем возрастает роль командного языка отладчика. Условные точки прерывания и реакция на их достижение могут и должны программироваться заранее. Весьма вероятно, что эти программы должны затрагивать не только непосредственно отлаживаемый процесс, но и процессы, возможно, функционирующие на других процессорах. Необязательно видеть состояние сразу всех процессов (для больших систем это не имеет смысла), но необходимо иметь возможность воздействовать на все процессы – поодиночке или группами. Подобное групповое воздействие – обязательная черта многопроцессного отладчика.

В настоящее время в качестве командного языка отладчика выбирается интерпретируемый полноценный язык программирования, такой как Python. Поддержка Python появилась в GDB версии 7.0. Наличие полноценного языка с большим количеством библиотек позволяет расширить возможности отображения данных, в том числе в графическом формате, организовать взаимодействие с базами данных и реализовать сложные механизмы управления приложением.

**Трассировка.** Вероятно, основным инструментом отладки многопроцессных систем можно считать трассировку. Она может иметь различную степень детализации – от машинной инструкции до программных транзакций. При аппаратной поддержке возмущения, вызываемые трассировкой, могут быть приемлемыми, а центр тяжести переносится на последующий анализ собранной информации. Еще одно применение этой информации – детерминированное воспроизведение запуска параллельной системы, при котором проявились ошибки. Детерминированное воспроизведение может сопровождаться интерактивной отладкой; вероятно, это можно считать наиболее уместным применением последней.

При трассировке с высокой степенью детализации объем собранной информации может исчисляться гигабайтами. Для его уменьшения существуют средства сжатия трасс и, что более существенно, условные точки трассировки, благодаря которым можно накапливать только информацию, вероятно, имеющую отношение к ошибкам и их проявлениям. В таких точках может также активизироваться интерактивный отладчик, который в данном случае остается средством ближнего боя.

Помещать события в трассу можно не только внешними средствами, но и из программы, используя механизмы самоконтроля и генерации событий. Подобное оснащение программ представляется наиболее целесообразным, так как оно учитывает семантику программ и предполагаемую первопричину разыскиваемых ошибок, минимизирует возмущения, вносимые в процесс функционирования систем, и объем собираемой информации.

И здесь возможна активизация интерактивного отладчика.

Практичный инструментарий трассировки – популярный предмет публикаций и реализаций [5, 6].

**Самоотладка.** Весьма удачным представляется подход, предложенный в статье [7] для отладки беспроводных сетей датчиков. Окна интерактивной отладки ассоциируются с некими представительными элементами аппаратно-программной конфигурации, но за счет использования групповых операций (глобальных команд) `gstop`, `gcontinue`, `gdetach`, `greset`, `gbreak` отладка сильно-связанных систем оказывается реальной.

Многопроцессорные конфигурации способны не только создавать новые проблемы, но и использоваться при отладке: на одних процессорах может функционировать отлаживаемая система, а на других – средства отладки. Если память подобной конфигурации является разделяемой, то возмущения, вносимые отладочными действиями, могут быть минимальными.

**Детерминированное воспроизведение выполнения.** Запуски, в которых проявляются ошибки, могут не только детерминированно воспроизводиться (примером каркаса для подобного воспроизведения может служить PinPlay [8]), но и синтезироваться как результат анализа программ (см. Execution Synthesis [9]). Вообще предварительный и «посмертный» анализ программ и данных, собранных в процессе функционирования, позволяет минимизировать вносимые отладкой возмущения (см. Analyzing Multicore Dumps [10], ConMem [11]).

Вариантом детерминированного воспроизведения может считаться механизм создания контрольных точек останова. Такие контрольные точки могут использоваться для более быстрого воспроизведения ошибки при ее обнаружении. При этом выполнение может быть неполным, но способствовать более быстрому воспроизведению аварийной ситуации.

**Отладка аппаратных конфигураций.** Современная постановка задачи отладки затрагивает такие аспекты, как отладка специфических аппаратных конфигураций, включающих в себя процессоры цифровой обработки сигналов или графические процессоры (см. Total Recall [12], NUDA [13]). Для отладки сложных сетей на кристалле предназначены инструменты, описанные в работе [14].

**Отладка производительности.** Еще один важный аспект – отладка производительности. Возможные решения в этой области представлены в работах [15, 16].

## Проблемы многопроцессной отладки

Все рассмотренные средства отладки полезны и для отладки последовательных приложений. Сложности при отладке многопроцессных приложений

связаны со значительным увеличением числа отслеживаемых сущностей, с усложнением их взаимодействий.

Рассмотрим, как увеличение количества сущностей отражается на отладочных действиях в системе.

**Взаимодействие с вычислительной средой.**

В многопроцессных взаимодействиях часто проявляется и особая сущность – подсистема взаимодействий. Она обеспечивает передачу данных между процессами и их синхронизацию. Это может быть система доставки сообщений, встроенная в ядро операционной системы, например, подсистема разделяемой памяти или программная библиотека, такая как MPICH [17]. Подобно тому, как реализация отладки требует поддержки операционной системы, в многопроцессных приложениях важно взаимодействие средств отладки и системы межпроцессного взаимодействия.

**Описание конфигурации отлаживаемой системы.** Первая проблема многопроцессной отладки – описание и отображение отлаживаемой конфигурации. При наличии большого количества вычислительных ресурсов возникает необходимость их распределения. Это распределение может быть статическим, когда конфигурация системы задается в файле или в специальной системе планирования. Примером такого распределения является конфигурация ARINC процессов в ОС3000.

Следует отметить, что проблема большого числа сущностей возникает не только при отладке многопроцессного приложения. Разработчику приложения приходится решать задачу разделения вычислительных ресурсов. Эта задача часто решается разделением множества модулей на функциональные компоненты, каждый из которых решает специфическую подзадачу. Например, некоторое множество процессов может отвечать за планирование, другое – за сбор информации, третье – за обработку, четвертое – за хранение. В такой системе естественным образом возникает необходимость работы с группами процессов в отладчике. Описание системы, созданное при ее проектировании, должно быть объединено с описанием системы в отладчике.

Распределение задачи по процессам может быть и динамическим. Например, сети автономных датчиков могут самоорганизовываться, определять имеющиеся ресурсы и планировать выполнение. Более того, часто в больших системах число компонент может со временем как расти, так и уменьшаться. Чем больше число процессов, тем больше вероятность отказа некоторых из них. Если приложение состоит из трех процессов, аварийное завершение одного из них равносильно завершению приложения. В системе из 1 000 процессов отказ – нормальное явление.

В любом случае отладочные средства должны иметь доступ к распределению процессов, возмож-

ность идентифицировать процессы, находить для каждого из них дополнительную информацию, такую как исходный код, расположение в памяти, аппаратную архитектуру.

**Сбор, отображение и анализ данных.** Ключевая проблема многопроцессной отладки – сбор и отображение большого количества данных. При этом нужно отметить, что в многопроцессных системах часто отсутствуют понятия времени и состояния или эти понятия являются размытыми. Соответственно, часто нельзя говорить о состоянии системы, смене состояний. В лучшем случае в системе присутствуют контрольные точки, в которых состояние системы определено. Многопроцессные системы порождают большое количество данных, слабо связанных между собой. Часто эти данные не позволяют говорить о том, корректно ли работает приложение.

Многопроцессный отладчик должен уметь работать со значительными потоками информации, а именно:

- собирать и группировать данные от различных процессов и групп процессов;
- анализировать данные, выявлять закономерности и аномалии;
- иметь возможность детализации представленной информации.

Наиболее популярное средство многопроцессной отладки – использование функций ввода-вывода для печати отладочной информации. Средства обработки вывода большого числа процессов описаны в статье [18]. Например, используется представление стеков вызова группы процессов, в которых функции объединены.

Кроме состояния системы, важно уметь отображать взаимодействия системы. Для сложных систем это может быть нетривиальной задачей.

**Управление выполнением.** Гибкое управление выполнением – традиционная возможность многопроцессных отладчиков. В стандарте HPDS описываются возможности управления группами процессов – группы процессов, групповые точки останова, барьеры и т.п. Группы процессов могут быть полезны и для других задач, например, для воспроизведения состояния. Воспроизведение работы подсистемы – более простая задача, если такое воспроизведение возможно.

Существует важная задача, решение которой в стандарте HPDS возложено на пользователя, – поддержание целостности системы в процессе отладки. Например, останов некоторой группы процессов для начала интерактивной отладки может не сказаться на работе системы. Связи компонентов необязательно должны быть жесткими, часто слабая связанность компонентов должна обеспечивать поддержание работы системы. Использование информации о таких связях в отладчике позволяет упростить отладку многопроцессных приложений. Например, система самоконтроля может остано-

ливать только группу процессов при наступлении аварийного состояния. Информация о связанности компонентов, их взаимодействиях может быть извлечена из конфигурации системы.

**Оптимизация отладочных действий.** Отладочные действия в многопроцессной конфигурации могут требовать значительных ресурсов. Многие задачи отладки, такие как воспроизведение, анализ стека вызовов, сжатие трассы, используют и процессор, и сетевую среду. При этом число данных растет даже быстрее, чем число процессов. Например, при трассировке взаимодействий многопроцессных систем объем информации может быть пропорционален квадрату числа процессов.

Экономия ресурсов при многопроцессной отладке достигается за счет следующих решений:

- обработка отладочных действий на каждом из вычислительных узлов;
- многоуровневая система агентов;
- использование широковебательных сообщений;
- интроспекция отладочных интерфейсов;
- возросшее использование эффективных отладочных средств, например трассировки;
- сжатие отладочной информации.

Масштабируемость отладочной системы – одна из главных проблем многопроцессной отладки. Сложность систем определяет необходимость использования специальных протоколов, а также архитектурных решений. При этом большинство протоколов, используемых в отладке, недостаточно расширяемые (например, простой протокол удаленной отладки, используемый в GDB).

В работе [19] описана реализация отладочных агентов с использованием протокола WSRF (Web Service Resource Framework). Это решение позволило создать отладчик для масштабной гетерогенной системы.

#### **Проблемы многопроцессной отладки на различных этапах поиска и устранения ошибок**

Функциональные свойства средств отладки – не единственное решение многопроцессных проблем. Другой класс вопросов, связанных с процессом отладки, показывает, что не только инструменты отладки должны быть другими, но и отлаживаемая система должна вести себя по-другому [20]. Рассмотрим, как процесс отладки определяет взаимодействия отладчика и многопроцессного приложения.

**Получение информации об ошибке.** Многопроцессная система состоит из большого набора компонент, возможно, слабосвязанных. Даже обнаружение ошибки является проблемой. Часто проявления ошибки настолько скрыты, что в потоке порождаемой информации их просто невозможно

увидеть. Например, некоторые компоненты системы могут перезапускаться по аварийному сигналу, но это никак не отразится на внешнем результате. О наличии ошибки можно судить только по падению производительности, или ошибка остается незамеченной.

Важно, чтобы некорректное поведение проявлялось как можно раньше. Возможные средства обнаружения ошибки включают средства самоконтроля, анализа поведения, обнаружения аномалий поведения [21].

**Изоляция ошибки.** Изоляция ошибки – важная задача даже при наличии инструментов воспроизведения. Последние не всегда позволяют работать с системой целиком, их применимость может ограничиваться некоторой подсистемой.

Этот аспект многопроцессной отладки более прост, чем отладка приложения в единой памяти. Средства изоляции компонент, слабая связанность, спецификации интерфейсов, средства контроля безопасности помогают найти проблемный компонент. Зачастую он более прост, чем сложная последовательная программа.

**Проверка гипотез.** После того как ошибка изолирована, необходимо найти ее причину. Поиск причины состоит из формулировки гипотез о ее нахождении и проверки этих гипотез. Эффективный способ проверки гипотез – использование встраиваемых в код агентов. Современные системы поддерживают способы модификации приложения, не требующие компиляции. Например, обработчики системных событий могут компилироваться на лету и эффективно выполняться. Примером такой системы может служить система [16], посвященная не только отладке производительности, но и традиционной многопроцессной отладке.

**Изучение последствий.** Заключительный этап исправления ошибки – изучение последствий ошибки. Как наличие ошибки отразилось на поведении системы в прошлом, к чему это привело? Возникали ли в прошлом ситуации, в которых ошибка могла привести к негативным последствиям? Привело ли к негативным последствиям исправление ошибки?

Система протоколирования должна быть разработана таким образом, чтобы протокол событий мог ответить на подобные вопросы. К сожалению, авторам неизвестны специальные инструменты, предназначенные для нахождения последствий ошибки.

#### **Архитектура многопроцессной отладки**

Увеличение числа сущностей отладки определяет возросшую важность архитектуры отладочных средств.

Архитектура позволяет решать проблемы минимизации последствий отладки, интеграции отладочных средств, адаптации отладочных средств к аппаратно-программной конфигурации. Большое число работ по многопроцессной отладке уделяет внимание архитектуре.

В качестве примера приведем архитектуру отладчика из [18]. Конечно, важным фактором, определяющим архитектуру отладчика, является программно-аппаратная платформа, но все же можно перечислить основные принципы построения многопроцессных отладчиков:

- древовидная структура;
- инкапсуляция;
- фильтрация данных на всех уровнях;
- агрегация данных;
- групповые операции;
- распределение полномочий отладки;
- использование хранилища данных для обеспечения доступа средств анализа;
- слабая связанность;
- асинхронные взаимодействия;
- интеграция с вычислительной платформой;
- интеграция средств отладки и приложений.

Перечисленные решения традиционны для архитектуры сложной многокомпонентной системы, например, системы управления. В большинстве случаев их применение оправдано, так как многопроцессная отладочная система действительно сложна. Подобная архитектура многопроцессных отладчиков обеспечивает их расширяемость, настраиваемость и эффективность.

### Заключение

Набор отладочных инструментов, применяемых в многопроцессной отладке, значительно шире набора инструментов традиционных последовательных отладчиков. Это усложнение – необходимость, определяемая сложностью отлаживаемой системы. Без средств визуализации трассировки, воспроизведения, самоконтроля найти ошибку зачастую невозможно. Эффективная отладка требует применения всего набора инструментов, причем каждый из них должен быть разработан с учетом сложности системы, с использованием вышеперечисленных подходов.

Важно, что подход к многопроцессной отладке требует не только применения различных инструментов, но и их интеграции в отлаживаемую систему, взаимодействия средств отладки с вычислительной средой. Эту новую по сравнению с традиционной отладкой особенность необходимо принимать во внимание не только при разработке отладчика, но и при проектировании приложений.

### Литература

1. Timmerman M., Gielen F., Lambrix P. High level tools for the debugging of real-time multiprocessor systems. ACM SIGPLAN Notices, 1993, vol. 28, iss. 12, pp. 151–157.
2. GNU Debugger. URL: <http://www.gnu.org/software/gdb/> (дата обращения: 03.05.2017).
3. Non-stop Multi-Threaded Debugging in GDB. URL: [http://www.codesourcery.com/publications/non\\_stop\\_multi\\_threaded\\_debugging\\_in\\_gdb.pdf](http://www.codesourcery.com/publications/non_stop_multi_threaded_debugging_in_gdb.pdf) (дата обращения: 03.05.2017).
4. Beynon M.D., Andrade H., Saltz J. Low-Cost Non-Intrusive Debugging Strategies for Distributed Parallel Programs, IEEE Intern. Conf. on Cluster Comp., 2002, pp. 439–442.
5. Yusen Li, Feng Wang, Gang Wang, Xiaoguang Liu, Jing Liu. MKtrace: an innovative debugging tool for multi-threaded programs on multiprocessor systems. Soft. Eng. Conf., APSEC 2007, pp. 510–517.
6. Moore L.J., Moya A.R. Non-intrusive debug technique for embedded programming, Soft. Reliability Eng., ISSRE 2003, pp. 375–380.
7. Yang Jing, Soffa Mary Lou, Selavo Leo, Whitehouse Kamin. Clairvoyant: a comprehensive source-level debugger for wireless sensor networks. Proc. of the 5th Intern. Conf. on Embedded Networked Sensor Systems, SenSys '07, 2007, pp. 189–203.
8. Patil Harish, Pereira Cristiano, Stallcup Mack, Lueck Gregory, Cowrie James. PinPlay: a framework for deterministic replay and reproducible analysis of parallel programs. Proc. 8th Annual IEEE/ACM Intern. Symp. Code Generation and Optimization, CGO'10, 2010, pp. 2–11.
9. Zamfir C., Candea G. Execution synthesis: a technique for automated software debugging. Proc. 5th Europ. Conf. on Comp. Systems, EuroSys '10, 2010, pp. 321–334.
10. Weeratunge Dasarath, Zhang Xiangyu, Jagannathan Suresh. Analyzing multicore dumps to facilitate concurrency bug reproduction, SIGPLAN Not., 2010, vol. 45, no. 3, pp. 155–166.
11. Zhang Wei, Sun Chong, Lu Shan. ConMem: detecting severe concurrency bugs through an effect-oriented approach. SIGARCH Comput. Archit. News, 2010, vol. 38, no. 1, pp. 179–192.
12. Sharif Ahmad, Lee Hsien-Hsin S. Total recall: a debugging framework for GPUs. Proc. 23rd ACM SIGGRAPH/Eurographics Sympos. on Graphics Hardware, GH '08, 2008, pp. 13–20.
13. Wen Chi-Neng, Chou Shu-Hsuan, Chen Tien-Fu, Su Alan Peisheng. NUDA: a non-uniform debugging architecture and non-intrusive race detection for many-core. Proc. 46th Annual Design Automation Conf., DAC '09, 2009, pp. 148–153.
14. Albrecht Mayer, Harry Siebert, Klaus D. McDonald-Maier. Boosting debugging support for complex systems on chip. Computer, 2007, vol. 40, no. 4, pp. 76–81.
15. West Paul E., Peress Yuval, Tyson Gary S., McKee, Sally A. Core monitors: monitoring performance in multicore processors. Proc. 6th ACM Conf. on Comp. Frontiers, CF '09, 2009, pp. 31–40.
16. Wisniewski R.W., Rosenberg B. Efficient, unified, and scalable performance monitoring for multiprocessor operating systems. Proc. 2003 ACM/IEEE Conf. on Supercomputing, SC '03, 2003, pp. 3–14.
17. MPICH. URL: [www.mcs.anl.gov/mpi/mpich/](http://www.mcs.anl.gov/mpi/mpich/) (дата обращения: 03.05.2017).
18. Susanne M. Balle, Bevin R. Brett, C.-P. Chih-Ping Chen, David LaFrance-Linden. Extending a traditional debugger to debug massively parallel applications. Jour. of Parallel and Distributed Computing, 2004, vol. 64, iss. 5, pp. 617–628.
19. Kurniawan D., Abramson D. A WSRFCompliant debugger for grid applications. Parallel and Distributed Proc. Sympos., 2007. IPDPS 2007. IEEE Intern., 2007, pp. 26–30.
20. Галатенко В.А., Костюхин К.А., Малиновский А.С., Шмырев Н.В. Моделирование и верификация программ как элемент контролируемого выполнения // Программные продукты и системы. 2008. № 4. С. 13–16.
21. Галатенко В.А., Костюхин К.А. Отладка и мониторинг распределенных разнородных систем // Программирование. 2002. № 1. С. 27–37.



## MULTIPROCESS SYSTEM DEBUGGING PROBLEMS

V.A. Galatenko<sup>1</sup>, Dr.Sc. (Physics and Mathematics), Head of Sector, galat@niisi.ras.ru

K.A. Kostyukhin<sup>1</sup>, Ph.D. (Physics and Mathematics), Senior Researcher, kost@niisi.ras.ru

<sup>1</sup> Federal State Institution "Scientific Research Institute for System Analysis of the Russian Academy of Sciences" (SRISA RAS), Nakhimovsky Ave. 36/1, Moscow, 117218, Russian Federation

**Abstract.** The article considers multiprocess complex systems debugging. The development of high-quality complex systems is a long and time-consuming process. It is believed that debugging takes about a half of this time. Switching to multi-core processor architectures has made parallelism a regular task, as well as errors specific for parallel systems. This makes debugging even more important and at the same time more complicated.

The main feature of multiprocessing systems is using complex, asynchronous interactions between system components. This feature influences the approaches to debugging, determines the choice of debugging tools and methods. The complexity of debugging is also determined by the quantity and heterogeneity of multi-process system components, some of them might be hardware. A traditional set of debugging tools is the following: an interactive debugger, a tracer, self-monitoring libraries, reverse execution. Their application changes.

The first chapter describes the existing debugging tools and techniques. The authors highlight their advantages and disadvantages. The next two chapters describe the existing problems of multiprocess system debugging. The fourth chapter provides the architecture of multiprocess system debugger. In conclusion authors present the results of their work.

**Keywords:** controlled execution, debugging, monitoring, tracing, multiprocess systems.

## References

1. Timmerman M., Gielen F., Lambrix P. High level tools for the debugging of real-time multiprocessor systems. *ACM SIGPLAN Notices*. 1993, vol. 28, iss. 12, pp. 151–157.
2. *GNU Debugger*. Available at: <http://www.gnu.org/software/gdb/> (accessed May 3, 2017).
3. *Non-stop Multi-Threaded Debugging in GDB*. Available at: [http://www.codesourcery.com/publications/non\\_stop\\_multi\\_threaded\\_debugging\\_in\\_gdb.pdf](http://www.codesourcery.com/publications/non_stop_multi_threaded_debugging_in_gdb.pdf) (accessed May 3, 2017).
4. Beynon M.D., Andrade H., Saltz J. Low-Cost Non-Intrusive Debugging Strategies for Distributed Parallel Programs. *IEEE Int. Conf. on Cluster Computing*. 2002, pp. 439–442.
5. Li Yu., Wang F., Wang G., Liu X., Liu J. MKtrace: An Innovative Debugging Tool for Multi-Threaded Programs on Multiprocessor Systems. *Software Engineering Conf. (APSEC 2007)*. 2007, pp. 510–517.
6. Moore L.J., Moya A.R. Non-intrusive debug technique for embedded programming. *Software Reliability Engineering (ISSRE 2003)*. 2003, pp. 375–380.
7. Yang Jing, Soffa M.L., Selavo L., Whitehouse Kamin. Clairvoyant: a comprehensive source-level debugger for wireless sensor networks. *Proc. of the 5th int. conf. on Embedded networked sensor systems (SenSys '07)*. 2007, pp. 189–203.
8. Patil H., Pereira C., Stallcup M., Lueck G., Cownie J. PinPlay: a framework for deterministic replay and reproducible analysis of parallel programs. *Proc. 8th Annual IEEE/ACM Int. Symp. on Code Generation and Optimization (CGO '10)*. 2010, pp. 2–11.
9. Zamfir C., Candea G. Execution synthesis: a technique for automated software debugging. *Proc. 5th European conf. on Computer systems (EuroSys '10)*. 2010, pp. 321–334.
10. Weeratunge Dasarath, Zhang Xiangyu, Jagannathan Suresh. Analyzing multicore dumps to facilitate concurrency bug reproduction. *ACM SIGPLAN Not.* 2010, vol. 45, no. 3, pp. 155–166.
11. Zhang Wei, Sun Chong, Lu Shan. ConMem: detecting severe concurrency bugs through an effect-oriented approach. *ACM SIGARCH Comput. Archit. News*. 2010, vol. 38, no. 1, pp. 179–192.
12. Sharif Ahmad, Lee Hsien-Hsin S. Total recall: a debugging framework for GPUs. *Proc. 23rd ACM SIGGRAPH/EUROGRAPHICS Symp. on Graphics Hardware (GH '08)*. 2008, pp. 13–20.
13. Wen Chi-Neng, Chou Shu-Hsuan, Chen Tien-Fu, Su Alan Peisheng NUDA: a non-uniform debugging architecture and non-intrusive race detection for many-core. *Proc. 46th Annual Design Automation Conf. (DAC '09)*. 2009, pp. 148–153.
14. Mayer A., Siebert H., McDonald-Maier K.D. Boosting Debugging Support for Complex Systems on Chip. *Computer*. 2007, vol. 40, no. 4, pp. 76–81.
15. West P.E., Peress Y., Tyson G.S., McKee S.A. Core monitors: monitoring performance in multicore processors, *Proc. 6th ACM conf. on Computing Frontiers (CF '09)*. 2009, pp. 31–40.
16. Wisniewski R.W., Rosenberg B. Efficient, Unified, and Scalable Performance Monitoring for Multiprocessor Operating Systems. *Proc. 2003 ACM/IEEE Conf. on Supercomputing (SC '03)*. 2003, pp. 3–14.
17. *MPICH*. Available at: [www.mcs.anl.gov/mpi/mpich/](http://www.mcs.anl.gov/mpi/mpich/) (accessed May 3, 2017).
18. Balle S.M., Brett B.R., Chih-Ping C.-P. Chen, LaFrance-Linden D. Extending a traditional debugger to debug massively parallel applications. *Jour. of Parallel and Distributed Computing*. 2004, vol. 64, iss. 5, pp. 617–628.
19. Kurniawan D., Abramson D. A WSRFCompliant Debugger for Grid Applications. *IEEE Int. Parallel and Distributed Processing Symp. IPDPS 2007*. 2007, pp. 26–30.
20. Galatenko V.A., Kostyukhin K.A., Malinovsky A.S., Shmyrev N.V. Program modelling and verification in controlled execution paradigm. *Programmye produkty i sistemy* [Software & Systems]. 2008, no. 4, pp. 13–16 (in Russ.).
21. Galatenko V.A., Kostyukhin K.A. Debugging and monitoring of the distributed heterogeneous systems. *Programming and Computer Software*. 2002, no. 1, pp. 27–37.

УДК 622.276.1/4

DOI: 10.15827/0236-235X.119.384-391

Дата подачи статьи: 30.03.17

2017. Т. 30. № 3. С. 384–391

## РЕШЕНИЕ ЗАДАЧ ОПТИМИЗАЦИИ ПРИ СУПЕРЭЛЕМЕНТНОМ МОДЕЛИРОВАНИИ РАЗРАБОТКИ НЕФТЯНЫХ МЕСТОРОЖДЕНИЙ

*И.В. Афанаскин, к.т.н., зав. группой, ivan@afanaskin.ru;*

*П.В. Ялов, инженер, petryalov@gmail.com;*

*А.М. Гуацингов, научный сотрудник, algts@inbox.ru;*

*А.В. Родителиев, ведущий программист, avrod\_94@mail.ru*

*(Федеральный научный центр Научно-исследовательский институт системных исследований РАН (ФНЦ НИИСИ РАН), Нахимовский просп., 36, корп. 1, г. Москва, 117218, Россия)*

Нефтяные месторождения России разрабатываются преимущественно с помощью заводнения. Большинство из них находятся на третьей или четвертой стадии разработки. Следовательно, обводненность продукции скважин составляет 80–90 % и более. В этих условиях с целью оптимизации разработки месторождений инженеры-нефтяники стараются уменьшить добычу и закачку воды при сохранении или увеличении добычи нефти. Для этого решаются задачи контроля и регулирования разработки месторождения с использованием различных математических моделей.

В данной работе рассматривается суперэлементная математическая модель заводнения, основанная на модели двухфазной фильтрации слабосжимаемых несмешивающихся жидкостей (нефти и воды) в упругом пласте по закону Дарси. Система дифференциальных уравнений для давления и насыщенности аппроксимирована на сетке Вороного полностью явным образом. Размеры суперэлементов сопоставимы с расстоянием между скважинами. Это позволяет проводить вычисления без использования специального ПО. Для решения обратных задач (определения коэффициентов модели или оптимизации параметров разработки) в работе используются метод Ньютона и метод сопряженных градиентов. В классической постановке обратной задачи методы теории оптимизации должны применяться непосредственно к математической модели исследуемого процесса. Однако в случае решения задач разработки нефтяных месторождений количество параметров для оптимизации может быть очень велико, а сложность математической модели довольно высока, поэтому применение методов теории оптимизации непосредственно к математической модели может быть очень трудоемким.

Для преодоления этого противоречия предлагается с помощью математической модели заводнения строить статистические зависимости показателей разработки от искомых параметров, а затем применять методы теории оптимизации уже не к математической модели, а к полученным статистическим зависимостям. Для иллюстрации такого подхода рассматривается решение задачи адаптации модели по абсолютной проницаемости. Установлено, что применение метода сопряженных градиентов непосредственно к модели заводнения дает ошибку в определении проницаемости 11,8 %. Применение того же метода к статистической зависимости ошибки адаптации модели (по накопленной добыче и закачке нефти и воды) от логарифма проницаемости дает ошибку в определении проницаемости лишь немногим больше – 15 %.

**Ключевые слова:** заводнение, экспресс-моделирование, оперативное моделирование, суперэлементная модель, суррогатная модель, сетка Вороного, обратные задачи.

Основным методом разработки нефтяных месторождений в России является заводнение. Большинство нефтяных месторождений находятся на третьей или четвертой стадии разработки. Следовательно, в продукции добывающих скважин присутствует большое количество воды. В этих условиях инженерам по разработке месторождений необходимо решать много задач по контролю и регулированию разработки. Для быстрого оценочного решения подобных задач может быть использована модель на базе концепции суперэлементов. Из-за большой неоднозначности исходных данных необходимо адаптировать модель к фактически замеренным показателям разработки (дебит нефти, газа и воды; забойное давление; пластовое давление и пр.). Задача адаптации модели сводится к решению задачи оптимизации. Кроме того, с помощью теории оптимизации и математического моделирования решают задачи оптимизации разработки нефтяных месторождений (оптимизации заводнения). Данная работа посвящена подходу, позволяющему с помощью простых методов (метод Нью-

тона, метод сопряженных градиентов) решать задачи оптимизации при суперэлементном моделировании разработки нефтяных месторождений. Используя только Excel Microsoft, описываемым методом можно решать задачи оптимизации с количеством параметров до 16. Используя специальные программы, количество параметров можно увеличить.

### Математическая модель двухфазной фильтрации нефти и воды

Система уравнений, описывающая упругую двухфазную фильтрацию, состоит из двух уравнений сохранения количества основных компонентов (объемов нефти и воды в стандартных условиях) и обобщенного закона Дарси, а капиллярными и гравитационными силами пренебрегается [1–6]:

– уравнения сохранения объемов нефти и воды

$$\frac{\partial}{\partial t} \left( \frac{mS_o}{B_o} \right) + \operatorname{div} \left( \frac{\vec{W}_o}{B_o} \right) = -q_o, \quad (1)$$

$$\frac{\partial}{\partial t} \left( \frac{m S_w}{B_w} \right) + \operatorname{div} \left( \frac{\vec{W}_w}{B_w} \right) = -\bar{q}_w + \bar{q}_a; \quad (2)$$

– обобщенный закон Дарси

$$\vec{W}_o = -\frac{k k_{ro}}{\mu_o} \operatorname{grad}(P), \quad (3)$$

$$\vec{W}_w = -\frac{k k_{rw}}{\mu_w} \operatorname{grad}(P), \quad (4)$$

где  $m$  – пористость;  $S_o$  и  $S_w$ ,  $B_o$  и  $B_w$ ,  $\vec{W}_o$  и  $\vec{W}_w$ ,  $\bar{q}_o$  и  $\bar{q}_w$ ,  $k_{ro}$  и  $k_{rw}$ ,  $\mu_o$  и  $\mu_w$  – насыщенность, объемный коэффициент, вектор скорости фильтрации, плотность источника (стока), моделирующего работу скважины, относительная фазовая проницаемость, вязкость нефти и воды соответственно;  $\bar{q}_a$  – плотность источника (стока), моделирующего работу законтурной области;  $k$  – абсолютная проницаемость;  $P$  – пластовое давление.

Систему уравнений (1)–(4) необходимо дополнить замыкающими соотношениями:

$$S_o + S_w = 1, \quad (5)$$

$$m = m_0[1 + C_r(P - P_0)], \quad (6)$$

$$B_o = B_{o0}[1 - C_o(P - P_0)], \quad (7)$$

$$B_w = B_{w0}[1 - C_w(P - P_0)], \quad (8)$$

где  $m_0$  – пористость при начальном пластовом давлении;  $B_{o0}$  и  $B_{w0}$  – объемный коэффициент нефти и воды при начальном пластовом давлении;  $C_r$ ,  $C_o$  и  $C_w$  – сжимаемость пласта, нефти и воды;  $P_0$  – начальное пластовое давление.

Систему уравнений (1)–(4) с учетом соотношений (5)–(8) можно преобразовать к следующей системе дифференциальных уравнений для водонасыщенности  $S \equiv S_w$  и давления  $P$  [7]:

$$m_0 \left[ C_r + (C_w - C_o)S + C_o \right] \frac{\partial P}{\partial t} = -B_{o0} \left[ \bar{q}_o + \operatorname{div} \left( \frac{\vec{W}_o}{B_o} \right) \right] - \quad (9)$$

$$-B_{w0} \left[ \bar{q}_w - \bar{q}_a + \operatorname{div} \left( \frac{\vec{W}_w}{B_w} \right) \right],$$

$$m_0 \left[ \frac{\partial S}{\partial t} + (C_r + C_w)S \frac{\partial P}{\partial t} \right] = -B_{w0} \left[ \bar{q}_w - \bar{q}_a + \operatorname{div} \left( \frac{\vec{W}_w}{B_w} \right) \right]. \quad (10)$$

Уравнения (9) и (10) дополняются начальными условиями  $P = P(x, y, z, t = 0)$ ,  $S = S(x, y, z, t = 0)$  и граничными условиями непротекания на внешних границах. Для моделирования законтурной водонасыщенной области используются источники/стоки  $\bar{q}_a$  в уравнениях (9) и (10) [1].

## Численная схема

Рассмотрим расчетную сетку, составленную из так называемых суперэлементов [8, 9] – ячеек с размерами, в плане сопоставимыми с расстоянием между скважинами (300–500 м). Тогда количество ячеек в модели будет примерно равно количеству скважин. Скважины используются как центры ячеек. В неразбуренной части объекта могут быть введены фиктивные скважины для построения сетки. Использование такой сетки позволяет сократить вычислительные затраты в тысячи раз [8].

Задача построения суперэлементной сетки в плане (2D-сетки) сводится к задаче построения для каждой скважины зоны дренирования, которая была введена А.П. Крыловым, а она, в свою очередь, к построению областей Вороного [7] (рис. 1):

- нанесение на карту скважин и границ;
- построение выпуклой оболочки по внешним скважинам;
- построение триангуляции Делоне по скважинам;
- построение областей Вороного по триангуляции.

Для преобразования суперэлементной сетки в 3D-сетку построенную в плане сетку копируют для разных слоев, изменяя при этом глубину залегания кровли ячеек и их толщину.

Большой размер ячеек позволяет использовать полностью явную схему [10]. Рассмотрим такую схему для уравнений (9) и (10) на сетке суперэлементов в плане (2D-плоская модель):

$$P_i^{n+1} = P_i^n - \Delta t^{n+1} \times$$

$$B_{o0} \left[ (q_o)_i^n + \sum_{j=1}^{N_i} (q_o)_{ij}^n \right] + B_{w0} \left[ (q_w)_i^n - (q_a)_i^n + \sum_{j=1}^{N_i} (q_w)_{ij}^n \right],$$

$$\times \frac{F_i h_i (m_0)_i \left[ C_r + (C_w - C_o)S_i^n + C_o \right]}{F_i h_i (m_0)_i \left[ C_r + (C_w - C_o)S_i^n + C_o \right]},$$

$$S_i^{n+1} = S_i^n -$$

$$-\Delta t^{n+1} \frac{B_{w0}}{F_i h_i (m_0)_i} \left[ (q_w)_i^n - (q_a)_i^n + \sum_{j=1}^{N_i} (q_w)_{ij}^n \right] -$$

$$-(C_r + C_w)S_i^n (P_i^{n+1} - P_i^n),$$

где  $n$  – номер шага по времени;  $i$  – номер ячейки;  $\Delta t^{n+1}$  – шаг по времени;  $F_i$ ,  $h_i$  и  $(m_0)_i$  – площадь, толщина и пористость при давлении  $P_0$  ячейки  $i$ ;  $(q_a)_i^n$  – дебит фазы  $\alpha = o, w$  скважины в ячейке  $i$  на шаге по времени  $n$ ;  $(q_a)_i^n$  – приток воды из законтурной области в ячейку  $i$ ;  $(q_a)_{ij}^n$  – переток фазы  $\alpha = o, w$  между ячейками  $i$  и  $j$  на шаге по времени  $n$ ;  $N_i$  – количество соседних ячеек для ячейки  $i$ .

Исходное состояние системы определяется заданными начальными условиями. Все шаги по времени рассчитываются таким образом, чтобы максимальное изменение давления и насыщенности по всей модели за один шаг по времени не превышало заданных значений.

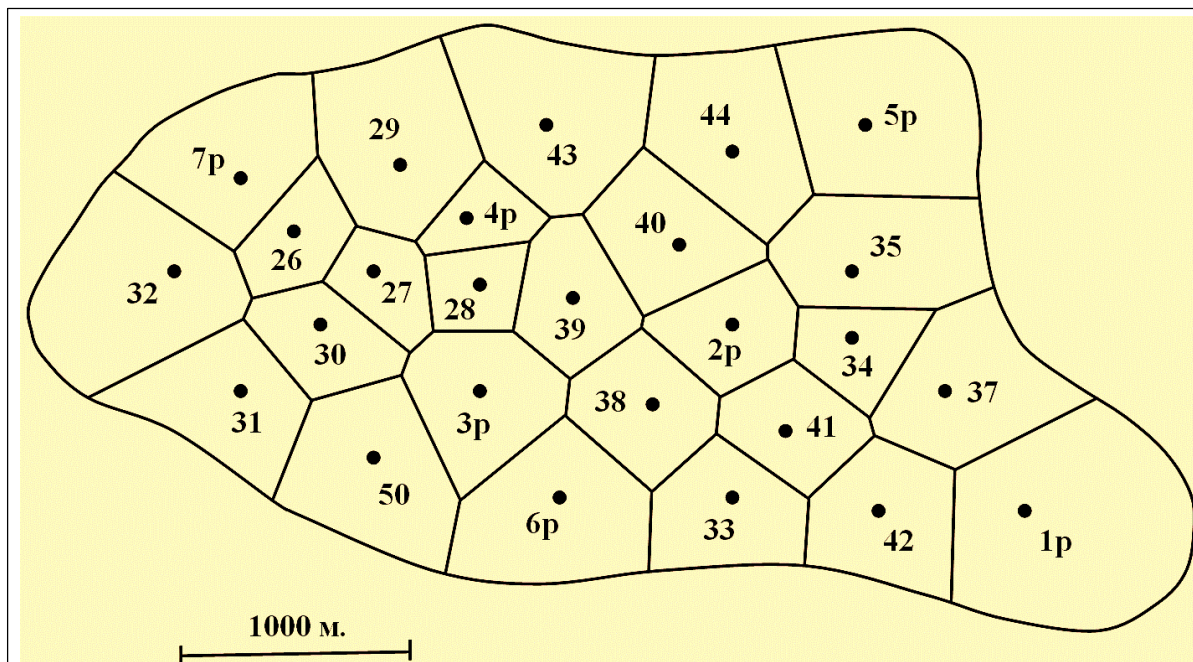


Рис. 1. Пример построения сетки суперэлементов в плане (точки – скважины, цифры – номера скважин)

Fig. 1. An example of super element grid constructing in a plan (points are wells, numbers are well numbers)

Переток между ячейками  $i$  и  $j$  (рис. 2) определяется как  $(q_{\alpha})_{ij}^n = \frac{k_{ij} (k_{ra})_{ij}^n}{\mu_{\alpha} (B_{\alpha})_{ij}^n} \cdot \frac{P_i^n - P_j^n}{L_{ij}} A_{ij} h_{ij}$ ,  $\alpha = o, w$ , где  $A_{ij}$  – длина совместного ребра ячеек  $i$  и  $j$ ;  $L_{ij}$  – расстояние между центрами ячеек (скважинами)  $i$  и  $j$ ;  $k_{ij} = \frac{2k_i k_j}{k_i + k_j}$ ;

$$(k_{ra})_{ij}^n = \begin{cases} (k_{ra})_i^n, & P_i^n > P_j^n, \\ (k_{ra})_j^n, & P_i^n < P_j^n; \end{cases}$$

$$(B_{\alpha})_{ij}^n = \frac{1}{2} [(B_{\alpha})_i^n + (B_{\alpha})_j^n]; \quad h_{ij} = \frac{1}{2} [h_i + h_j].$$

### Моделирование скважин

Дебиты скважин по жидкости  $q_l$ , нефти  $q_o$ , воде  $q_w$ , обводненность  $W$  и забойное давление  $P_w$  определяются из следующих соотношений:

$$(q_l)_i^n = PI_i^n (P_i^n - P_w^n),$$

$$(q_o)_i^n = (q_l)_i^n (1 - W_i^n),$$

$$(q_w)_i^n = (q_l)_i^n W_i^n,$$

$$W_i^n = \frac{(k_{rw})_i^n}{(k_{rw})_i^n + (k_{ro})_i^n \frac{\mu_w (B_w)_i^n}{\mu_o (B_o)_i^n}},$$

где  $PI_i^n$  – коэффициент продуктивности скважины  $i$  по жидкости, который определяется как

$$PI_i^n = \frac{2\pi k_i h_i}{\ln \left[ \frac{(R_c)_i}{(r_w)_i} \right] + Skin_i^n} \cdot \left[ \frac{(k_{ro})_i^n}{\mu_o (B_o)_i^n} + \frac{(k_{rw})_i^n}{\mu_w (B_w)_i^n} \right],$$

где  $(r_w)_i$  – радиус скважины по долоту;  $Skin_i^n$  – скин-фактор скважины;  $(R_c)_i$  – эквивалентный радиус блока, определяемый как  $(R_c)_i = 0,14 \sqrt{2F_i}$ .

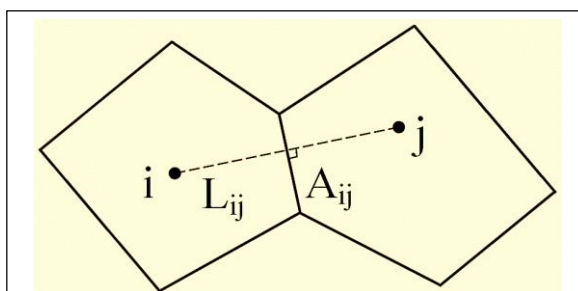


Рис. 2. Схема для расчета перетоков между ячейками

Fig. 2. Scheme for calculating crossflows between cells

Для добывающих скважин необходимо задать дебит жидкости или забойное давление, для нагнетательных – расход воды или забойное давление.

Нетрудно заметить, что в такой модели вода появится в продукции добывающей скважины сразу же после превышения текущей водонасыщенности ячейки значения насыщенности связанной водой. Это, конечно, неверно. Вода необходимо время для того, чтобы пройти расстояние от нагнетательной скважины до добывающей. Для воссоздания этого

эффекта используется ступенчатое задание функций относительной фазовой проницаемости:

$$(k_{rw})_i^n = \begin{cases} 0, & S_i^n < (S_c)_i, \\ A_i \left[ \frac{S_i^n - (S_{wcr})_i}{1 - (S_{wcr})_i} \right]^{\alpha_i}, & S_i^n \geq (S_c)_i, \end{cases}$$

$$(k_{ro})_i^n = \begin{cases} B_i, & S_i^n < (S_c)_i, \\ B_i \left[ \frac{1 - (S_{owcr})_i - S_i^n}{1 - (S_{owcr})_i - (S_{wcr})_i} \right]^{\beta_i}, & S_i^n \geq (S_c)_i, \end{cases}$$

где  $(S_c)_i$  – водонасыщенность на фронте вытеснения;  $(S_{wcr})_i$  – насыщенность связанной водой;  $(S_{owcr})_i$  – насыщенность остаточной нефтью; коэффициенты  $A_i, B_i, \alpha_i, \beta_i$  определяются по результатам интерпретации исследований ядра и могут зависеть от пористости или проницаемости.

Водонасыщенность на фронте вытеснения  $(S_c)_i$  определяется из соотношения

$$f'((S_c)_i) = \frac{f((S_c)_i) - f((S_{wcr})_i)}{(S_c)_i - (S_{wcr})_i},$$

где  $f(S_i)$  – функция Бакли–Лeverетта:

$$f(S_i) = \frac{(k_{rw})_i}{(k_{rw})_i + (k_{ro})_i \frac{\mu_w}{\mu_o}}.$$

Предложенная 2D-расчетная схема легко обобщается на случай 3D. При этом приток в скважину из каждого слоя определяется согласно его продуктивности при известном дебите скважины по жидкости или забойном давлении.

### Решение обратных задач методом Ньютона

Пусть необходимо найти минимум функции многих переменных  $f(X)$ , где  $X = (x_1, x_2, x_3, \dots, x_n)$ . Эта задача эквивалентна задаче нахождения значений  $X$ , при которых градиент функции  $f(X)$  равен нулю:

$$\text{grad}(f(X)) = 0. \quad (11)$$

Применим к (11) метод Ньютона:

$$\text{grad}(f(X^j)) + H(X^j)(X^{j+1} - X^j) = 0, \quad (12)$$

где  $j = 1, 2, 3, \dots, m$  – номер итерации;  $H(X)$  – гессиан функции  $f(X)$ .

Напомним, что гессиан функции – это симметричная квадратичная форма, описывающая поведение функции во втором порядке:

$$H(X) = \sum_{i=1}^n \sum_{j=1}^n a_{ij} x_i x_j,$$

где  $a_{ij} = \partial^2 f / \partial x_i \partial x_j$ , функция  $f(X)$  задана на  $n$ -мерном пространстве вещественных чисел.

В более удобном для вычислений виде формулу (12) можно представить в виде

$$X^{j+1} = X^j - H^{-1}(X^j) \text{grad}(f(X^j)).$$

Метод Ньютона прост в реализации, но нахождение матрицы Гессе сопряжено с большими вычислительными затратами из-за необходимости нахождения большого количества частных производных.

### Метод сопряженных градиентов

Метод сопряженных градиентов является методом нахождения локального минимума (максимума) функции с использованием информации о ее значениях и градиенте.

Метод сопряженных градиентов математически существенно сложнее метода Ньютона. Не углубляясь в особенности метода, рассмотрим итерационный алгоритм нахождения экстремума функции:  $X_k^{j+1} = X_k^j + \lambda S_k^j$ , где  $k$  – номер направления поиска;  $j$  – номер итерации, условие перпендикулярности направлений поиска  $k+1$  и  $k$ :  $\lambda = \arg \min_{\lambda} f(X_k^{j+1} + \lambda S_k^j)$ , направление поиска:

$$S_k^{j+1} = -\text{grad}(f(X_k^{j+1})) + \omega_k S_k^j,$$

$$\text{где } \omega_k = \frac{\|\text{grad}(f(X_k^{j+1}))\|^2}{\|\text{grad}(f(X_k^j))\|^2}.$$

Метод сопряженных градиентов математически сложнее метода Ньютона, но требует меньше вычислений.

Метод Ньютона и метод сопряженных элементов реализованы в надстройках Excel Microsoft.

### Программная реализация математической модели двухфазной фильтрации на основе концепции суперэлементов

Приведенная выше математическая модель, обобщенная на случай 3D, была описана в среде разработки Microsoft Visual Studio 15 на языке программирования C++. Для представления исходных данных, описывающих геолого-физические свойства пласта, был выбран формат в виде таблицы с использованием ключевых слов для обозначения блоков данных. Выбор представления обусловлен возможностью редактирования данных пользователем без использования специализированных программных средств (например в Microsoft Excel), а также экспорта данных из других симуляторов. При редактировании большого объема исходных данных пользователю необходимо хранить метаданные, относящиеся к ячейкам таблицы. Для реализации предложен алгоритм фильтрации, который удаляет метаданные, маркированные символами «//» из каждого блока при считывании файла. После успешного считывания исходных данных производятся проверки на целостность данных, размеры связанных блоков, размеры связанных интервалов перфораций и слоев, отсутствие пустых

ячеек внутри таблицы. При успешном считывании и прохождении всех проверок пользователю выводится список выполненных операций. В случае ошибки пользователь также получает сообщение с указанием позиции в блоке данных и способах устранения ошибки.

На следующем этапе выполняется алгоритм моделирования заводнения нефтяных месторождений с применением многопоточности. Создаются отдельные потоки для расчета статических данных, а затем очередь обработки функций, рассчитывающих динамические данные. На данном этапе производится расчет изменяемого шага по времени; в целях сокращения используемой оперативной памяти хранятся данные только по трем шагам. Далее происходит формирование выходного файла, в котором присутствуют ключевые слова, описание размерности данных, диапазоны адресов ячеек данных для блоков, а также нумерация скважин и слоев. Предлагаемый формат выходных данных позволяет произвести их анализ и обработку (например в Microsoft Excel), в том числе построить диаграммы без дополнительной подготовки.

Созданная программа получила название Oil production calculator, или сокращенно OPC. Она была проверена с помощью сравнительного тестирования с симуляторами Dz10 Каца Р.М. и Волгина Е.Р., Rubis Kappa Engineering, Eclipse Schlumberger. Получено хорошее совмещение результатов расчетов с помощью OPC и результатов, полученных с помощью вышеописанных программ.

### Новый подход к решению задач оптимизации при моделировании разработки нефтяных месторождений

Ввиду большого количества скважин при решении задач оптимизации количество искоемых параметров может быть очень велико. Тогда непосредственное использование метода Ньютона или метода сопряженных градиентов ведет к большим вычислительным затратам. На примере задачи адаптации по проницаемости суперэлементной модели заводнения нефтяного пласта рассмотрим новый метод решения задач оптимизации, позволяющий сократить вычислительные затраты без существенной потери точности вычислений.

Рассмотрим простейшую секторную модель с 4 скважинами – 1 нагнетательной и 3 добывающими. Параметры модели следующие:

- размеры модели в плане – 1 500×1 500 м;
- абсолютная проницаемость – переменная (см. табл. 1);
- пористость – переменная;
- начальное пластовое давление – 339 атм.;
- начальная водонасыщенность – 0,2 д.ед.;
- объемный коэффициент воды –  $1,02 \text{ м}^3/\text{м}^3$ ;
- вязкость воды – 0,36 сПз; сжимаемость воды –  $4,7 \cdot 10^{-5} \text{ 1/атм.}$ ;

- объемный коэффициент нефти –  $1,55 \text{ м}^3/\text{м}^3$ ;
- вязкость нефти – 0,397 сПз; сжимаемость нефти –  $1,01 \cdot 10^{-4} \text{ 1/атм.}$ ;
- растворимость газа в нефти – 203,5  $\text{м}^3/\text{м}^3$ ;
- давление насыщения нефти газом – 18,8 атм.;
- сжимаемость породы –  $4,7 \cdot 10^{-5} \text{ 1/атм.}$ ;
- капиллярным давлением пренебрегается;
- функции относительных фазовых проницаемостей (ОФП) приведены на рисунке 3;

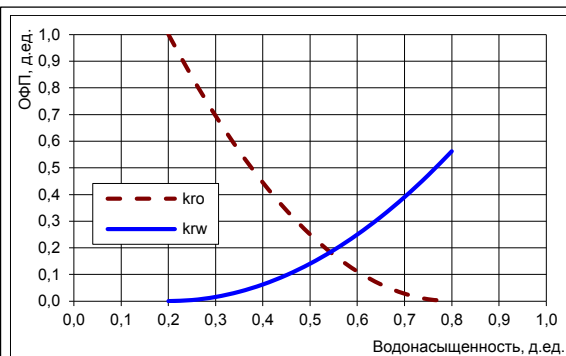


Рис. 3. Относительные фазовые проницаемости (ОФП) для нефти ( $k_{ro}$ ) и воды ( $k_{rw}$ )

Fig. 3. Relative phase permeabilities (ОФП) for oil ( $k_{ro}$ ) and water ( $k_{rw}$ )

- дебит жидкости добывающих скважин – данные в таблице 1; минимальное забойное давление в добывающих скважинах – 18,8 атм.;
- закачка воды в нагнетательную скважину без ограничений; забойное давление в нагнетательной скважине – данные в таблице 1;
- скин-фактор для всех скважин – 0 ед.;
- радиус скважин – 0,2 м;
- срок разработки – 35 лет;
- притока воды из-за контура нет.

Таблица 1

Параметры модели

Table 1

Model parameters

Параметр	Скважина (суперэлементная ячейка)			
	1	2	3	4
Площадь, тыс. м <sup>2</sup>	514	628	603	504
Толщина, м	25	22	19	15
Пористость, д. ед.	0,20	0,18	0,16	0,15
Абсолютная проницаемость, мД	70	53	44	38
Дебит, м <sup>3</sup> /сут.	-	236	168	114
Забойное давление, атм.	372	18,8	18,8	18,8

Примечание: 1 – нагнетательная скважина, 2, 3, 4 – добывающие.

Добывающие скважины управлялись дебитом, а нагнетательная – забойным давлением.

Расстановка скважин в плане и конфигурация суперэлементов показаны на рисунке 4.

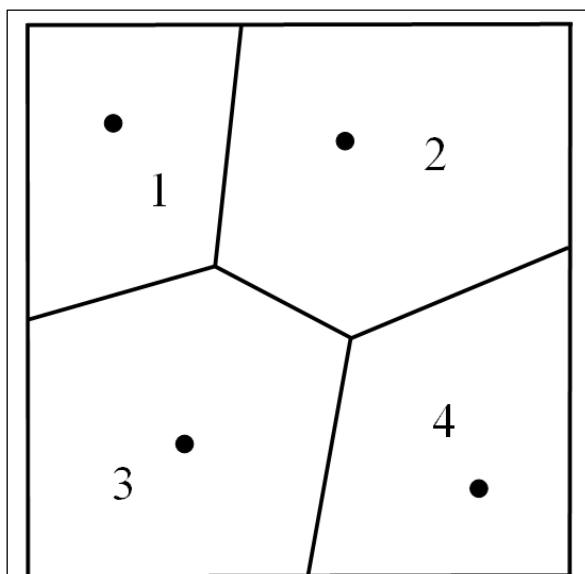


Рис. 4. Расстановка скважин в плане и конфигурация суперэлементов

Fig. 4. Arrangement of wells in a plan and configuration of superelements

Сведения о длине совместного ребра соседних ячеек для суперэлементной модели и расстоянии между центрами соседних ячеек (скважинами) приведены в таблице 2.

Таблица 2

**Геометрические параметры суперэлементной модели**

Table 2

**Geometric parameters of a superelement model**

Параметр	Пары ячеек				
	1-2	1-3	2-3	2-4	3-4
Длина совместного ребра ячеек, м	650	538	425	650	663
Расстояние между центрами соседних ячеек, м	638	889	925	1013	813

Итак, в данном случае задача оптимизации состоит в том, чтобы уточнить значения проницаемости, используемые в модели, зная историю разработки. Имея оценку проницаемости по данным геофизических и гидродинамических исследований, а также исследований керна, с помощью генератора случайных чисел рассчитаем  $n$  вариантов проницаемости по 4 штуки в каждой группе  $k_1, k_2, k_3, k_4$ , где цифры 1–4 соответствуют номерам ячеек. Примем  $n = 20$ . Рассчитаем значения показателей модели для каждой группы значений проницаемости. Запишем целевую функцию следующим образом:

$$F = \sum_{i=2}^4 \left( \frac{Q_{o,c} - Q_{o,f}}{Q_{o,f}} \right)_i^2 + \sum_{i=2}^4 \left( \frac{Q_{w,c} - Q_{w,f}}{Q_{w,f}} \right)_i^2 + \left( \frac{Q_{iw,c} - Q_{iw,f}}{Q_{iw,f}} \right)_{i=1}^2 \rightarrow \min,$$

где  $Q_o$  – накопленная добыча нефти по скважине;  $Q_w$  – накопленная добыча воды по скважине;  $Q_{iw}$  – накопленная закачка воды; индекс  $c$  – расчетные показатели; индекс  $f$  – фактические показатели; индекс  $i$  – номер ячейки. Для изучения эффективности предлагаемого метода в качестве фактических показателей примем показатели, полученные путем моделирования при значениях проницаемостей  $k_1 = 70, k_2 = 53, k_3 = 44, k_4 = 38$  мД.

Таким образом, для каждого набора проницаемостей ячеек 1–4 получено значение целевой функции  $F$ . Из опыта анализа разработки известно, что накопленная добыча и закачка прямо пропорциональны проницаемости либо ее логарифму. Составим следующие уравнения для целевой функции  $F$ :

$$F = a_1 k_1 + a_2 k_2 + a_3 k_3 + a_4 k_4 + a_5, \quad (13)$$

$$F = a_1 \lg(k_1) + a_2 \lg(k_2) + a_3 \lg(k_3) + a_4 \lg(k_4) + a_5, \quad (14)$$

$$F = a_1 k_1 + a_2 k_2 + a_3 k_3 + a_4 k_4 + a_5 \lg(k_1) + a_6 \lg(k_2) + a_7 \lg(k_3) + a_8 \lg(k_4) + a_9, \quad (15)$$

где  $a_1$ – $a_9$  – постоянные коэффициенты.

С помощью регрессионного анализа найдем коэффициенты соотношений (13)–(15), используя полученную таблицу  $F$  от  $k_1, k_2, k_3, k_4$ . Затем, применив метод сопряженных градиентов к (13)–(15), определим проницаемости  $k_1, k_2, k_3, k_4$ , приняв начальное приближение проницаемости 50 мД (для такого маленького числа параметров (4 штуки) не важно, какой метод использовать – Ньютона или сопряженных градиентов). Сравним результаты определения проницаемости, полученные с помощью применения методов Ньютона и сопряженных градиентов непосредственно к модели фильтрации, а также полученные путем применения метода сопряженных градиентов к регрессионным моделям (13)–(15) (см. табл. 3). Начальное приближение проницаемости – 50 мД. Для оценки достоверности определения проницаемости введем ошибку по проницаемости следующим образом:

$$\Delta = \frac{1}{4} \sum_{i=1}^4 \frac{|k_{i,c} - k_{i,f}|}{k_{i,f}} 100, \quad (16)$$

где индекс  $c$  – расчетная проницаемость; индекс  $f$  – фактическая проницаемость; индекс  $i$  – номер ячейки. Фактически соотношение (16) представляет среднюю относительную ошибку определения проницаемости.

Из таблицы 3 видно, что наиболее точные значения проницаемости можно получить, применяя метод сопряженных градиентов к модели фильтрации (ошибка – 11,8 %). Однако в случае, когда это невозможно (например, из-за большого числа искомых параметров), используя регрессионную модель и метод сопряженных градиентов (либо метод Ньютона), можно получить значения проницаемости с ошибкой не более 15,0 %. При этом важно правильно выбрать вид регрессионной модели.



Таблица 3

## Сравнение способов решения задачи оптимизации и определения проницаемости

Table 3

## Comparison of methods for solving the optimization problem and determining permeability

Способ определения проницаемости	Значение проницаемости				
	$k_1$ , мД	$k_2$ , мД	$k_3$ , мД	$k_4$ , мД	$\Delta$ , %
Факт	70	53	44	38	-
Метод Ньютона применительно к модели фильтрации	62,0	55,7	46,9	24,0	15,0
Метод сопряженных градиентов применительно к модели фильтрации	59,3	54,4	47,7	30,0	11,8
Метод сопряженных градиентов применительно к регрессионной модели (13)	61,9	58,4	47,8	50,0	15,5
Метод сопряженных градиентов применительно к регрессионной модели (14)	56,9	52,9	48,1	50,2	15,0
Метод сопряженных градиентов применительно к регрессионной модели (15)	39,8	54,1	53,3	49,8	23,5

## Заключение

В работе описана суррогатная модель заводнения нефтяной залежи при пластовом давлении выше давления насыщения. Модель представляет собой полностью явную численную схему на базе концепции суперэлементов. Это позволяет реализовать данный подход для любого месторождения без использования специального ПО.

Рассмотрены подходы к решению задач оптимизации при суперэлементном моделировании разработки нефтяных месторождений. При этом описано использование метода Ньютона и метода сопряженных градиентов. В качестве альтернативы при большом количестве искомых параметров (и/или больших размерах модели) предложено применять методы теории оптимизации не непосредственно к модели фильтрации, а к регрессионной модели, построенной на базе экспериментов, проведенных на модели фильтрации. Рассмотрен пример уточнения значений проницаемости при адаптации модели с четырьмя скважинами. На этом примере показана приемлемая точность такого подхода. Следует подчеркнуть, что важно правильно выбрать вид регрессионной модели.

*Работа выполнена при поддержке Программы фундаментальных исследований РАН № I.33П, НИР № 0065-2015-0111.*

## Литература

1. Азиз Х., Сеттари Э. Математическое моделирование пластовых систем. М.–Ижевск: Изд-во Ин-та комп. исследований, 2004. 416 с.
2. Каневская Р.Д. Математическое моделирование гидродинамических процессов разработки месторождений углеводородов. М.–Ижевск: Изд-во Ин-та комп. исследований, 2002. 140 с.
3. Кац Р.М., Волгин Е.Р., Афанаскин И.В. Численное моделирование двухфазной фильтрации нефти и воды // Тр. НИИСИ РАН, 2014. Т. 4. Вып. 2. С. 141–148.
4. Coats K.H., Thomas L.K., Pierson R.G. Compositional and black oil reservoir simulation. SPE 29111, 13<sup>th</sup> SPE Sympos. on Reservoir Simulation, 1995, pp. 149–162.
5. Cichlow G.B. Modern reservoir engineering – a simulation approach. New Jersey: Prentice Hall. Inc., 1979, p. 354.
6. Peaceman D.W. Fundamentals of numerical reservoir simulation. Amsterdam: Elsevier Scientific Publishing Co., 1977, p. 191.
7. Хисамутдинов Н.И., Хасанов М.М., Телин А.Г. [и др.]. Разработка нефтяных месторождений. В 4 т. Т. 1. Разработка нефтяных месторождений на поздней стадии. М.: Изд-во ВНИИОЭНГ, 1994. 240 с.
8. Мазо А.Б., Булыгин Д.В. Суперэлементы. Новый подход к моделированию разработки нефтяных месторождений // Нефть. Газ. Новации. 2011. Вып. 11. С. 6–8.
9. Булыгин Д.В., Мазо А.Б., Поташев К.А., Калинин Е.И. Геолого-технические аспекты суперэлементной фильтрационной модели нефтяных месторождений // Георесурсы, 2013. Вып. 3. С. 31–35.
10. Афанаскин И.В., Егоров А.А., Колеватов А.А. Экспресс-моделирование заводнения нефтяных месторождений с помощью концепции суперэлементов // Вестн. кибернетики. 2016. Вып. 2. С. 153–163.

# OPTIMIZATION PROBLEMS SOLUTION BASED ON SUPERELEMENT MODELING OF OIL-FIELD DEVELOPMENT

*I.V. Afanaskin*<sup>1</sup>, Ph.D. (Engineering), Head of Group, [ivan@afanaskin.ru](mailto:ivan@afanaskin.ru)  
*P.V. Yalov*<sup>1</sup>, Engineer, [petryalov@gmail.com](mailto:petryalov@gmail.com)  
*A.M. Giatsintov*<sup>1</sup>, Research Associate, [algts@inbox.ru](mailto:algts@inbox.ru)  
*A.V. Roditelev*<sup>1</sup>, Leading Programmer, [avrod\\_94@mail.ru](mailto:avrod_94@mail.ru)

<sup>1</sup> Federal State Institution "Scientific Research Institute for System Analysis of the Russian Academy of Sciences" (SRISA RAS), Nakhimovsky Ave. 36/1, Moscow, 117218, Russian Federation



**Abstract.** Oil fields of Russia are mainly developed by waterflooding. Most of them are in the 3rd or 4th stage of development. Consequently, the water cutting of well production is 80–90% or more. In these conditions, in order to optimize the development of deposits, oil engineers try to reduce water production and injection while maintaining or increasing oil production. For this purpose, there are the tasks of field development control and regulation. These problems are solved using various mathematical models.

This paper considers a superelement mathematical waterflooding model based on a two-phase filtration model for weakly compressible immiscible liquids (oil and water) in an elastic bed under the Darcy law. The system of differential equations for pressure and saturation is approximated on Voronoi diagram in an entirely explicit manner. The size of the superelements is comparable to the distance between the wells. This allows performing calculations without using special software. To solve inverse problems (determine model coefficients or optimize development parameters), the work uses Newton's method and the conjugate gradient method. In the classical setting of an inverse problem, the optimization theory methods should be applied directly to the mathematical model of the process under study. However, when solving oilfield development problems, the number of optimization parameters can be large, and the complexity of the mathematical model is quite high. Therefore, the application of optimization theory methods directly to a mathematical model can be very time-consuming. To overcome this contradiction, it is proposed to build statistical dependencies of the development indices on the required parameters using a mathematical model of waterflooding, and then to apply optimization theory methods no longer to the mathematical model, but to the statistical dependencies obtained.

To illustrate this approach, we consider the solution of the problem of model adaptation to absolute permeability. It is established that the application of the conjugate gradient method directly to the waterflooding model gives an error in determining the permeability of 11,8 %. Applying the same method to a statistical dependence of a model adaptation error (on accumulated production and injection of oil and water) on the logarithm of permeability gives an error in determining permeability of only a little more, it is 15 %.

**Keywords:** waterflooding, express modeling, prompt modeling, superelement model, Voronoi grid, inverse problems.

**Acknowledgements.** The article has been supported by the Program for Fundamental Research of the Russian Academy of Sciences no. 1.33P, Research no. 0065-2015-0111.

### References

1. Aziz Kh., Settari E. *Matematicheskoe modelirovanie plastovykh sistem* [Petroleum Reservoir Simulation]. Moscow-Izhevsk, ICS Publ., 2004, 416 p.
2. Kanevskaya R.D. *Matematicheskoe modelirovanie gidrodinamicheskikh protsessov razrabotki mestorozhdeniy uglevodorodov* [Mathematical Modeling of Hydrodynamic Processes of Development of Hydrocarbon Deposits]. Moscow-Izhevsk, ICS Publ., 2002, 140 p.
3. Kats R.M., Volgin E.R., Afanaskin I.V. Numerical Simulation of Two-Phase Oil and Water Filtration. *Trudy NIISI RAN* [Proc. Scientific-Research Institute for System Studies, Russian Academy of Sciences]. 2014, vol. 4, iss. 2, pp. 141–148 (in Russ.).
4. Coats K.H., Thomas L.K., Pierson R.G. Compositional and Black Oil Reservoir Simulation. SPE 29111. *13th SPE Symp. on Reservoir Simulation*. 1995, pp. 149–162.
5. Crichlow G.B. *Modern Reservoir Engineering – a Simulation Approach*. New Jersey, Prentice Hall Inc. Publ., 1979, 354 p.
6. Peaceman D.W. *Fundamentals of Numerical Reservoir Simulation*. Amsterdam, Elsevier Scientific Publ. Co., 1977, 191 p.
7. Khisamutdinov N.I., Khasanov M.M., Telin A.G. *Razrabotka neftyanykh mestorozhdeniy. Tom 1. Razrabotka neftyanykh mestorozhdeniy na pozdney stadii* [Development of Oil Fields. Vol. 1. Development of Oil Deposits at a Late Stage]. Moscow, VNIIOENG Publ., 1994, 240 p.
8. Mazo A.B., Bulygin D.V. Super Elements. A New Approach to Modeling the Development of Oil Fields. *Neft. Gaz. Novatsii* [Oil. Gas. Novations]. 2011, no. 11, pp. 6–8 (in Russ.).
9. Bulygin D.V., Mazo A.B., Potashev K.A., Kalinin E.I. Geological and Technical Aspects of the Superelement Filtration Model of Oil Fields. *Georesursy* [Georesources]. 2013, no. 3(53), pp. 31–35 (in Russ.).
10. Afanaskin I.V., Egorov A.A., Kolevator A.A. Instant Modelling of Oil Fields Waterflooding by Super Elements Concept Application. *Vestn. kibernetiki* [Proc. in Cybernetics]. 2016, no. 2 (22), pp. 153–163 (in Russ.).

### Примеры библиографического описания статьи

1. Афанаскин И.В., Ялов П.В., Гиацинтов А.М., Родителев А.В. Решение задач оптимизации при суперэлементном моделировании разработки нефтяных месторождений // Программные продукты и системы. 2017. Т. 30. № 3. С. 384–391. DOI: 10.15827/0236-235X.119.384-391.
2. Afanaskin I.V., Yalov P.V., Giatsintov A.M., Roditelev A.V. Optimization problems solution based on superelement modeling of oil-field development. *Programmnye produkty i sistemy* [Software & Systems]. 2017, vol. 30, no. 3, pp. 384–391 (in Russ.). DOI: 10.15827/0236-235X.119.384-391.

УДК 004.451.24

DOI: 10.15827/0236-235X.119.392-400

Дата подачи статьи: 30.03.17

2017. Т. 30. № 3. С. 392–400

## **РЕАЛИЗАЦИЯ КАНАЛОВ СПЕЦИФИКАЦИИ ARINC 653 В ОПЕРАЦИОННОЙ СИСТЕМЕ РЕАЛЬНОГО ВРЕМЕНИ БАГЕТ 3**

*А.Н. Годунов, к.ф.-м.н., зав. отделом, nkag@niisi.ras.ru;*

*В.А. Солдатов, к.т.н., старший научный сотрудник, nkvalera@niisi.ras.ru;*

*И.И. Хоменков, ведущий инженер, nkigor@niisi.ras.ru*

*(Федеральный научный центр Научно-исследовательский институт системных исследований  
РАН (ФНЦ НИИСИ РАН), Нахимовский просп., 36, корп. 1, г. Москва, 117218, Россия)*

В статье рассматриваются каналы спецификации ARINC 653, предназначенные для взаимодействия между процессами. Предлагаются основные принципы построения и методы взаимодействия драйверов каналов с отечественной операционной системой реального времени Багет 3, в которой интерфейс с пользователем базируется на спецификации ARINC 653 и стандарте POSIX. Доступ к каналам осуществляется через порты. У каждого канала имеются только один порт, передающий сообщения, а также один или несколько портов, принимающих сообщения. Данные в каналах могут передаваться только в одном направлении.

Прикладная программа может использовать каналы в процессах, соответствующих как спецификации ARINC, так и стандарту POSIX. Формулируются требования, предъявляемые к драйверам каналов, и описываются средства управления уведомлениями, которые используются при создании конкретного драйвера. Рассматриваются этапы инициализации, приема и передачи сообщений, а также алгоритмы их реализации.

Предложенные способы построения взаимодействия драйверов каналов с операционной системой реального времени позволяют избежать записи данных в память чужого процесса, что существенно повышает надежность функционирования системы. Этот же интерфейс используется и в распределенных вычислительных системах, когда отдельные узлы системы соединяются между собой посредством шин с общим доступом (VME, RapidIO, Fibre Channel) или с помощью сети Ethernet. Таблица связей по каналам определяется при конфигурировании системы.

Драйвер каналов, разработанный для коммуникационной среды RapidIO, отличающейся высоким быстродействием, используется в отечественной многопроцессорной системе цифровой обработки сигналов.

**Ключевые слова:** реальное время, операционная система, каналы, драйверы, многопроцессорные системы, RapidIO, POSIX, ARINC 653.

В 2008 году состоялся первый выпуск отечественной операционной системы реального времени (ОСРВ) Багет 3.0 [1, 2]. Разработка системы была вызвана прежде всего высокими требованиями к надежности ее выполнения на основе концепции изолированных разделов, которая нашла свое отражение в спецификации ARINC 653 [3]. Известные зарубежные производители ОСРВ, такие как vxWorks [4], LynxOS-178 [5], Integrity-178B [6], также выпустили версии ОС, удовлетворяющие этой спецификации.

Таким образом, ОСРВ Багет 3 базируется на следующих международных стандартах: спецификация ARINC 653, POSIX 1003.1, стандарт на мобильные операционные системы (программный интерфейс).

Стандарты POSIX и ARINC существенно отличаются друг от друга по составу функций, их именам, используемой терминологии. В ОСРВ Багет 3 в качестве основной была выбрана спецификация ARINC 653. Стандарт POSIX используется в той мере, в какой это допускается спецификацией ARINC 653.

Спецификация ARINC 653 определяет интерфейс (APEX – APplication EXecutive) между ОС и прикладными программами. APEX предписывает пространственное и временное разделение ресурсов. Терминология, используемая в ARINC 653, отличается от терминологии, принятой в стандарте POSIX. В ARINC 653 единицей планирования ре-

сурсов является раздел (partition). В стандарте POSIX этому термину соответствует понятие «процесс» (process). В рамках одного раздела возможно параллельное выполнение нескольких процессов, тогда как в POSIX в рамках одного процесса возможно параллельное выполнение нескольких потоков управления (thread). В настоящей статье использована терминология, принятая в стандарте POSIX.

Спецификация ARINC 653 предусматривает как пользовательские, так и системные процессы. Интерфейс пользовательских процессов должен соответствовать требованиям спецификации, а на интерфейс системных процессов в ARINC 653 не накладывается никаких ограничений. В ОСРВ Багет 3 для системных процессов используется интерфейс стандарта POSIX. Таким образом, под управлением ОСРВ Багет 3 могут одновременно выполняться как ARINC-процессы, так и POSIX-процессы. Все эти процессы, кроме главного системного процесса, будем называть пользовательскими. Все пользовательские процессы работают в пользовательском режиме процессора и используют виртуальную адресацию, что исключает доступ одних процессов к памяти других. Главный системный процесс, хотя и выполняется в привилегированном режиме работы процессора, в основном также использует виртуальную адресацию, доступ по физическим адресам применяется крайне редко. Организация виртуальной адресации позво-

ляет обеспечить контроль доступа к отдельным сегментам памяти, в частности, защиту пользовательских процессов друг от друга и защиту главного системного процесса от пользовательских процессов. Каждый процесс работает со своими виртуальными адресами, трансляцию которых в физические выполняет аппаратура компьютера. Таким образом, пользовательский процесс лишен возможности прямого обращения к страницам основной памяти, занятым информацией, относящейся к другим процессам.

### Взаимодействие между процессами

Согласно спецификации ARINC 653, процессы могут взаимодействовать между собой только путем передачи сообщений через каналы. Такое ограничение делает процессы слабо взаимосвязанными. Это в значительной мере уменьшает влияние ошибок в одних процессах на работу других процессов, а также упрощает восстановление работоспособности системы в случае обнаружении ошибок при работе пользовательских процессов. Если в результате обработки ошибки, возникшей при выполнении одного из пользовательских процессов, этот процесс будет повторно запущен (произведен рестарт), то для других процессов, которые взаимодействовали с перезапущенным процессом, это может привести лишь к потере нескольких сообщений.

Доступ к каналам производится через порты. Каждый канал имеет только один порт, передающий сообщения, и один или несколько портов, принимающих сообщения. Данные в канале могут передаваться только в одном направлении.

Каналы могут использоваться для передачи сообщений

- между процессами, которые выполняются на одном и том же процессорном модуле;
- между процессами, выполняемыми на разных процессорных модулях;
- между процессами и внешними устройствами.

В первом случае передача данных производится только средствами ОС, во втором и третьем случаях – драйверами и ОС.

Каждый канал может работать в одном из двух режимов: с очередью сообщений (queuing-канал) и без очереди сообщений (sampling-канал). Интерфейс взаимодействия прикладных программ с каналами не зависит от способа передачи данных, но зависит от режима работы канала. Для каждого режима работы используется свой набор функций.

Независимость интерфейса от способа передачи данных значительно повышает мобильность прикладных программ и позволяет выбирать способы передачи данных на этапе конфигурирования системы без внесения изменений в прикладные программы.

Каналы и порты должны быть описаны при конфигурировании системы. Указанные при конфигурировании параметры каналов (используемые порты, режим работы, направление передачи данных и др.) нельзя изменить во время работы системы.

Канал, работающий в режиме с очередью сообщений, имеет один порт приема сообщений (порт-получатель) и один порт отправки сообщений (порт-отправитель). Порт-отправитель содержит очередь отправляемых сообщений, а порт-получатель – очередь принимаемых сообщений. Размер очередей ограничен и указывается при создании порта. Каждое отправляемое сообщение доставляется получателю и может быть получено только один раз.

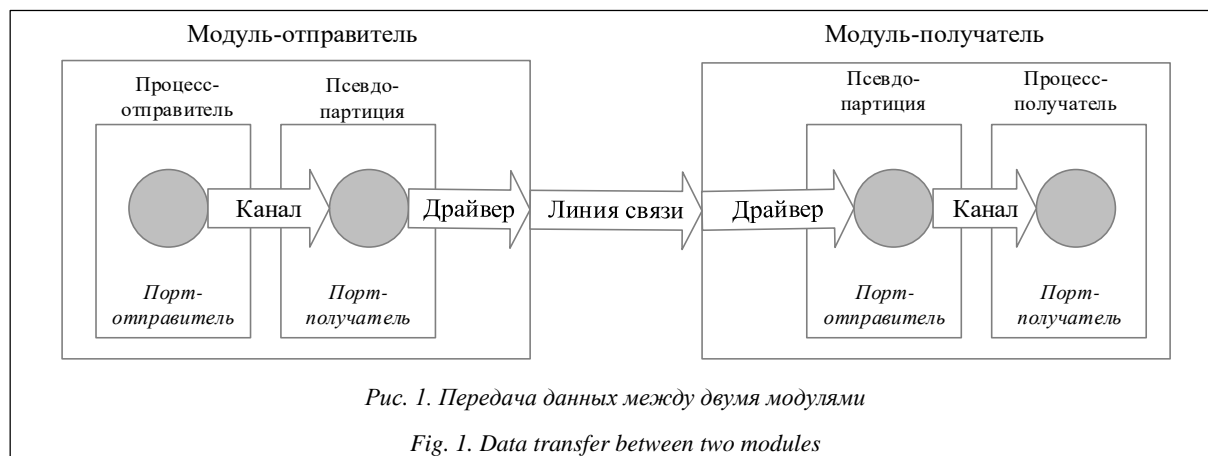
Канал, работающий в режиме без очереди сообщений, может иметь только один порт отправки сообщений и один или несколько портов приема сообщений. Каждое сообщение, получаемое портом приема, замещает ранее полученные сообщения (они просто теряются). Таким образом, из порта всегда считывается сообщение, полученное последним. Прочитанное сообщение считается годным, если с момента поступления его в порт приема до момента чтения из порта прошло времени не больше, чем на обновление порта.

Если при отправке сообщения оказалось, что по каким-то причинам предыдущее сообщение не было отправлено, новое сообщение замещает старое. Таким образом, отправка сообщения в порт всегда возможна и прикладная программа всегда может выводить сообщения в порт с нужной частотой (это не относится к процессу передачи данных от порта приема к порту передачи).

На рисунке 1 представлена модель интерфейса драйверов каналов с ОСРВ.

Как видно из рисунка, для передачи данных между двумя модулями создаются два канала: один на модуле, передающем данные, другой на модуле, принимающем данные. Канал, который находится на модуле, передающем данные, имеет порт-отправитель в пользовательском процессе ОСРВ, а порт-получатель – в псевдоразделе. Канал на модуле, принимающем данные, имеет порт-отправитель в псевдоразделе, а порт-получатель – в стандартном процессе ОСРВ.

Передача данных между портом-отправителем в процессе-отправителе и портом-получателем в процессе-получателе происходит в несколько этапов. На первом этапе сообщение из порта-отправителя в процессе-отправителе по каналу передается в порт-получатель псевдораздела модуля-отправителя. Передача данных по этому каналу производится операционной системой. На втором этапе полученное сообщение отправляется драйвером по линии связи модулю-получателю. На третьем этапе драйвер модуля-получателя принимает сообщение по линии связи и помещает его в порт-отправитель



псевдораздела модуля-получателя. На четвертом этапе это сообщение передается в порт-получатель в процессе-получателе.

В ОСПВ Багет 3 каналы могут использоваться не только в ARINC-процессах, но и в POSIX-процессах.

Драйвер канала для ОСПВ Багет 3 обычно состоит из обработчика прерываний и, возможно, одного или нескольких потоков главного (привилегированного) POSIX-процесса. Рассмотрим правила создания драйвера, разработанные для ОСПВ Багет 3.

#### Назначение и принципы работы канального драйвера

Как уже отмечалось, прием и передача сообщений между пользовательскими процессами происходит в несколько этапов. Драйвер каналов, который выполняется на 2-м и 3-м этапах, обеспечивает передачу и прием сообщений по физическим линиям связи между портом-получателем псевдораздела передающего модуля и портом-отправителем псевдораздела принимающего модуля. Управление устройством (прием и передача сообщений) производится драйвером в контексте специального потока привилегированного POSIX-процесса и/или в контексте функций обработки прерываний, которые также используют адресное пространство привилегированного POSIX-процесса.

Поток, в котором выполняется драйвер, может активизироваться либо периодически (периодический режим), либо по требованию (непериодический режим). В первом случае драйвер через равные интервалы времени выполняет требуемые действия по приему и/или передаче сообщений. Во втором случае драйвер активизируется с помощью функций уведомления о появлении нового сообщения или свободного буфера. Уведомления могут быть посланы из ARINC- или POSIX-процесса, а также из функции обработки прерываний с помощью соответствующих функций ОСПВ.

Накладные расходы при работе драйвера уменьшаются, если передавать и принимать не по

одному сообщению, а сразу группу сообщений. Периодический режим позволяет накопить несколько сообщений и потом передать сразу группу сообщений. В этом случае драйвер не извещается о появлении ни новых сообщений, ни новых свободных буферов (что также существенно снижает накладные расходы). Периодический драйвер не обрабатывает прерывания от аппаратуры и не заказывает их порождения. Он позволяет экономить время на обработке прерываний и уведомлений, но уступает непериодическому по скорости реакции. Интерфейс взаимодействия с драйвером, используемый в ОС РВ Багет, предполагает поддержку обоих режимов. Конкретный вариант может быть выбран на этапе конфигурирования.

Сообщения, отправляемые или получаемые драйвером, могут передаваться вместе с заголовком. Если в псевдоразделе находится порт-отправитель (порт-отправитель принадлежит драйверу), ОС игнорирует заголовок и передает прикладной программе сообщение без заголовка. Если в псевдоразделе находится порт-получатель, ОС формирует сообщение с заголовком или без него в зависимости от атрибутов, указанных при конфигурировании. Формируемый ОС (стандартный) заголовок содержит длину и номер сообщения.

Для размещения сообщений в памяти используются буфера. Все буфера одного канала имеют одинаковый размер, который указывается при конфигурировании. Для каждого буфера создается его описатель. Описатель буфера содержит указатель на буфер, длину сообщения и др. Буфера и их описатели создаются процессом-отправителем.

Функции драйвера должны обеспечивать выполнение следующих действий:

- инициализация драйвера;
- инициализация отдельного порта;
- чтение сообщений из порта и передача их по каналам связи;
- прием сообщений по каналам связи и отправка их через порт.

При инициализации ОСПВ вызывается специальная функция ОС инициализации каналов, при выполнении которой разбираются параметры кон-

фигурации каналов и инициализируются соответствующие каналные драйверы. В настоящее время в зависимости от оборудования и параметров конфигурации могут быть проинициализированы каналные драйверы, обеспечивающие межмодульную связь по сети Ethernet по протоколу UDP, по шине VME, по шине RapidIO.

### Инициализация драйвера

Каждый драйвер должен иметь свою функцию инициализации, которая вызывается при инициализации. В процессе инициализации драйвера он регистрирует в ОС функции драйвера, которые может использовать ОС для взаимодействия с драйвером. К ним относятся:

- функция `probe()`, которая позволяет определить, какие порты обслуживаются данным драйвером (вызывается в привилегированном *POSIX*-процессе);
- функция инициализации порта, которая вызывается при создании каждого порта, обслуживаемого данным драйвером.

Функция инициализации драйвера должна зарегистрировать драйвер в ОС с помощью функции `chanCreateDriver()`. Эта функция создает описатель драйвера в ОС и возвращает указатель на него. Доступ к описателю драйвера в ОС со стороны драйвера возможен только с помощью функций ОС (непосредственный доступ запрещен).

Далее драйвер должен поместить информацию о драйвере в описатель драйвера в ОС (с помощью функций ОС):

- указатель на функцию `probe()` драйвера;
- указатель на функцию инициализации порта драйвера;
- указатель на функцию запуска (старта) драйвера;
- имя драйвера.

Драйвер может создать свою управляющую таблицу (описатель) и зарегистрировать ее в управляющей таблице драйвера в ОС.

Функция проверки драйвера `probe()` входит в состав драйвера и используется ОС при поиске драйвера, который обслуживает данный (межмодульный) канал. Если драйвер обслуживает указанный канал, функция `probe()` возвращает 0, в противном случае не ноль.

### Инициализация порта

При создании порта, находящегося в псевдоразделе, ОС вызывает функцию инициализации порта драйвера и передает ей указатель на описатель порта ОС. Доступ к описателю порта в ОС со стороны драйвера возможен только с помощью функций ОС (непосредственный доступ запрещен).

Функция инициализации порта драйвера может получить указатель на управляющую таблицу

(описатель) драйвера и информацию об инициализируемом порте (канале). С помощью функций ОС драйвер может получить следующую информацию:

- идентификатор модуля-партнера;
- номер логической связи (подканала);
- имя канала;
- идентификатор канала;
- имя порта;
- размер буферов;
- число буферов;
- направление передачи данных (ввод или вывод);
- период обновления (для *sampling*-портов);
- тип канала (*queuing*, *sampling*, *SAP*).

Драйвер может создать свою управляющую таблицу (описатель) порта и зарегистрировать ее в управляющей таблице порта ОС (с помощью функций ОС). При инициализации канала (порта) драйвер также определяет функцию уведомления о появлении сообщения или свободного буфера в канале, а также наличие заголовков сообщений.

### Обслуживание буферов

Для приема и передачи сообщений через порт в ОСРВ Багет 3 был разработан нестандартный интерфейс взаимодействия с драйвером канала. Для работы с буферами и очередями буферов предназначены следующие функции:

- `chanGetBuffer()` – получить свободный буфер;
- `chanPutBuffer()` – освободить буфер;
- `chanGetMessage()` – получить сообщение;
- `chanPutMessage()` – отправить сообщение.

Эти функции можно использовать как для порта *queuing*, так и для *sampling*, но в зависимости от типа порта (*queuing* или *sampling*) функции используют разный алгоритм.

### Обслуживание буферов для *queuing*-каналов

Для каждого *queuing*-канала ОСРВ создает две очереди: очередь сообщений и очередь свободных буферов. Процесс-отправитель помещает отправляемые сообщения в очередь сообщений, а процесс-получатель извлекает сообщения из этой очереди. Очередь свободных буферов используется для передачи освободившихся буферов от процесса-получателя к процессу-отправителю. Процесс-получатель помещает свободные буфера в очередь, а процесс-отправитель извлекает их из очереди.

Все очереди организованы как циклические буфера, то есть состоят из массива идентификаторов буферов и двух указателей: указателя свободной области и указателя занятой области. Такая организация очередей упрощает синхронизацию при уста-

новке и извлечении элементов очереди. Длина каждого массива равна числу буферов плюс один. Так как в любой очереди буфер может содержаться не более одного раза, то его всегда можно поместить в очередь.

Массив очереди сообщений находится в памяти процесса-отправителя, там же находится и указатель свободной области (номер первого элемента в области свободных элементов) этой очереди. Указатель занятой области (номер первого элемента в области занятых элементов) находится в памяти процесса-получателя.

Массив очереди свободных буферов находится в памяти процесса-получателя, там же находится и указатель свободной области (номер первого элемента в области свободных элементов). Указатель занятой области (номер первого элемента в области занятых элементов) находится в памяти процесса-отправителя.

Если номер первого элемента в области занятых элементов меньше номера первого элемента в области свободных, то область занятых элементов находится между первым занятым (включительно) и первым свободным (не включая его). Все остальные элементы считаются свободными.

Если номер первого элемента в области занятых элементов больше номера первого элемента в области свободных, то область свободных элементов находится между первым свободным (включительно) и первым занятым (не включая его). Занятые элементы располагаются от первого занятого до конца массива и далее от начала массива до первого свободного (не включая его).

Если номер первого элемента в области занятых элементов равен номеру первого элемента в области свободных, считается, что все элементы свободны. Таким образом, всегда в массиве имеется хотя бы один свободный элемент (его нельзя использовать).

Занятый элемент массива содержит идентификатор буфера. В случае очереди отправленных сообщений это идентификатор буфера, содержащего сообщение, а в случае очереди свободных номеров – идентификатор свободного буфера.

Если нужно добавить буфер в очередь, выбирается первый элемент в области свободных (он лежит сразу после области занятых элементов). В этот элемент помещается идентификатор буфера. После этого указатель свободной области передвигается на 1 элемент. Если же он указывал на конец массива, то перемещается на начало массива.

Если нужно извлечь буфер из очереди, выбирается первый элемент в области занятых. Из этого элемента считывается идентификатор буфера. После этого указатель области занятых элементов передвигается на 1 элемент. Если же он указывал на конец массива, то перемещается на начало массива.

Функции `chanGetBuffer()` и `chanPutMessage()` используются при обслуживании портов-отправи-

телей, а функции `chanGetMessage()` и `chanPutBuffer()` при обслуживании портов-получателей.

Функция `chanGetBuffer()` извлекает буфер из очереди свободных буферов и возвращает его вызвавшей программе. Если очередь пуста, функция формирует запрос на уведомление о появлении свободного буфера и возвращает управление.

Функция `chanPutMessage()` помещает сообщение в очередь сообщений. Функция также посылает уведомление о появлении сообщения, если был неудовлетворенный запрос на уведомление.

Функция `chanGetMessage()` извлекает буфер из очереди сообщений и возвращает его вызвавшей программе. Если очередь пуста, функция формирует запрос на уведомление о появлении сообщения и возвращает управление.

Функция `chanPutBuffer()` помещает буфер в очередь свободных буферов. Функция также посылает уведомление о появлении свободного буфера, если был неудовлетворенный запрос на уведомление.

### Обслуживание буферов для *sampling*-каналов

В случае *sampling*-каналов буфера используются портами-отправителями для размещения в них отправляемых сообщений, а портами-получателями для доступа к полученным сообщениям. Один и тот же буфер может одновременно использоваться несколькими портами-получателями, но не может одновременно использоваться портом-отправителем и портом-получателем. Поэтому, если один из портов-получателей перестал использовать какой-либо буфер, это не означает, что буфер стал свободным и может использоваться портом-отправителем для нового сообщения.

Порт, принадлежащий псевдоразделу (драйверу), может одновременно использовать (закрепить за собой) несколько буферов. Максимальное число буферов, которое может быть закреплено за портом, обслуживаемым данным драйвером, должно быть определено при инициализации порта функцией `chanSetPortMaxBuffNum()`. Порты, принадлежащие пользовательскому процессу (ARINC или POSIX), не закрепляют за собой буфера.

Учет буферов, используемых в настоящее время портом (закрепленных за портом), производится ОС, управление буферами – функциями ОС. Функции `chanGetBuffer()` и `chanPutMessage()` используются при обслуживании портов-отправителей, а функции `chanGetMessage()` и `chanPutBuffer()` при обслуживании портов-получателей.

### Ввод сообщений

**Queueing-каналы (прием сообщений).** Для ввода сообщений через устройство создается канал, порт-получатель которого находится в обычном процессе, а порт-отправитель – в псевдораз-

деле. Сообщения, получаемые через обсуживаемое драйвером устройство, драйвер выводит через порт-отправитель канала.

Для получения сообщений из queing-порта прикладная программа (процесс-получатель) использует функцию OCPB Багет 3 RECEIVE\_QUEUEING\_MESSAGE. Эта функция запрашивает сообщение из канала с помощью функции ОС chanGetMessage(). Если в очереди есть (непрочитанные) сообщения, эта функция возвращает буфер с самым старым (непрочитанным) сообщением. Полученное сообщение копируется в буфер, указанный при вызове функции RECEIVE\_QUEUEING\_MESSAGE. Освободившийся буфер помещают в очередь с помощью функции chanPutBuffer(). Эта функция отправляет уведомление о появлении свободного буфера драйверу, если это требуется.

Если сообщение получить не удалось, функция формирует запрос на уведомление о получении сообщения и возвращает соответствующий код возврата. Если при вызове функции RECEIVE\_QUEUEING\_MESSAGE указано ненулевое время ожидания сообщения, поток объявляется неработоспособным.

Буфера для ввода сообщений с устройства канальный драйвер должен брать из очереди свободных буферов канала с помощью функции ОС chanGetBuffer(). Если свободных буферов нет, функция возвращает соответствующий код возврата и формирует запрос на уведомление о появлении свободных буферов в канале. Соответствующая функция уведомления драйвера должна быть зарегистрирована заранее (в противном случае уведомления не будут отправляться).

Для каждого выводимого в канал сообщения драйвер указывает его номер. Если сообщения нумерует (модуль) отправитель, используется его номер, в противном случае драйвер нумерует сообщения при получении. Если драйвер обнаружил потерю сообщений, он уведомляет об этом ОС (поле cb\_overflow в описателе буфера).

Буфера, содержащие полученные с устройства сообщения, драйвер помещает в очередь сообщений канала с помощью функции ОС chanPutMessage(). Перед вызовом функции chanPutMessage() драйвер должен поместить в описатель буфера следующую информацию: длина сообщения, порядковый номер сообщения, признак переполнения (если какие-то сообщения были утеряны).

Функция chanPutMessage() также отправит уведомление о появлении сообщения процессу-получателю, если это требуется. Функция обработки этого уведомления объявляет работоспособными потоки, которые ждут сообщений (столько потоков, сколько есть сообщений).

При периодической работе драйвера прерывания не порождаются аппаратурой и не обрабатываются драйвером.

**Sampling-каналы (чтение сообщений).** Если для ввода сообщений через устройство используется sampling-канал, то порты-получатели канала находятся в обычных процессах, а порт-отправитель – в псевдоразделе. Порт-отправитель обслуживается драйвером. Описатель порта находится в памяти привилегированного POSIX-процесса, там же находятся буфера и их описатели.

Поступающие с устройства сообщения драйвер выводит через порт-отправитель канала. Прикладная программа получает сообщение из порта-получателя канала с помощью функции ОС READ\_SAMPLING\_MESSAGE. Эта функция находит буфер, содержащий самое свежее сообщение, запоминает номер этого сообщения и копирует сообщение из буфера в память прикладной программы. После копирования функция проверяет, находится ли в буфере то же сообщение, что и до чтения. Если сообщение было затерто более свежим, снова запрашивается самое свежее сообщение и т.д.

Несколько потоков могут одновременно читать сообщения из одного и того же sampling-порта функцией READ\_SAMPLING\_MESSAGE. В силу этого sampling-порты обычных процессов не закрепляют буфера на время чтения сообщения. Функция READ\_SAMPLING\_MESSAGE не запрашивает и не посылает никаких уведомлений (так как это не требуется).

Для получения свободного буфера для ввода сообщений канальный драйвер может использовать функцию OCPB chanGetBuffer().

Перед использованием буфера устройством драйвер должен сбросить поле порядкового номера сообщения (записать -1) в описателе буфера. По завершении записи сообщения в буфер (ввода сообщения устройством) драйвер должен поместить в описатель буфера следующую информацию: длина сообщения, порядковый номер сообщения.

Далее драйвер должен вывести буфер с сообщением в порт канала с помощью функции chanPutMessage(). В результате вызова этой функции указанный буфер будет объявлен содержащим самое свежее сообщение.

В данном случае (когда все порты-получатели находятся в стандартных процессах) драйвер может взять управление буферами полностью на себя (например, циклически использовать все имеющиеся буфера) и не использовать функцию chanGetBuffer().

Драйвер не запрашивает никаких уведомлений и не порождает уведомлений (так как это не требуется).

### Вывод сообщений

**Queuing-каналы (отправка сообщений).** Для вывода сообщений через устройство создается канал, порт-отправитель которого находится в обычном процессе, а порт-получатель – в псевдоразделе.

Порт-получатель канала обслуживается драйвером. Сообщения, полученные через порт-получатель канала, драйвер выводит через устройство. Прикладная программа отправляет сообщения с помощью функции `OC SEND_QUEUEING_MESSAGE`. Эта функция запрашивает свободный буфер (из очереди свободных буферов канала) с помощью функции `OC chanGetBuffer()`. Если в очереди есть свободные буфера, функция возвращает свободный буфер. Отправляемое сообщение копируется в полученный буфер, и буфер с сообщением помещается в очередь сообщений канала функцией `OC chanPutMessage()`. Эта функция отправит уведомление об отправке сообщения драйверу, если это требуется.

Если свободных буферов нет, формируется запрос на уведомление о появлении свободного буфера в канале. Если после окончания указанного времени ожидания свободный буфер не появился, функция заканчивает свою работу с соответствующим кодом окончания.

Порт-получатель `queuing`-канала обслуживается драйвером канала. Драйвер извлекает буфера с сообщениями из очереди сообщений канала с помощью функции `OC chanGetMessage()` и выводит их через устройство. Если очередь сообщений пуста, функция возвращает соответствующий код возврата и формирует запрос на уведомление о появлении сообщений в канале. Соответствующая функция уведомления драйвера должна быть зарегистрирована заранее (в противном случае уведомления не будут отправляться).

Буфера, содержащие сообщения, уже отправленные через устройство, драйвер помещает в очередь свободных буферов канала с помощью функции `OC chanPutBuffer()`. Эта функция отправит уведомление о появлении свободного буфера процессу-получателю, если это требуется. Функция обработки этого уведомления объявляет работоспособными потоки, которые ждут свободных буферов (столько потоков, сколько свободных буферов).

**Sampling-каналы (запись сообщений).** При выводе сообщений через устройство посредством `sampling`-канала порт-отправитель канала находится в обычном процессе, а один или несколько портов-получателей – в псевдоразделе. Эти порты обслуживаются драйвером.

Поступающие через канал сообщения драйвер выводит через устройство; каждый раз выводятся самые свежие сообщения, так как более свежие сообщения замещают более старые (старые сообщения просто игнорируются).

Управление устройством производится драйвером в контексте обработчика прерываний или специального потока драйвера привилегированного `POSIX`-процесса.

Прикладная программа отправляет сообщения через порт-отправитель канала с помощью функ-

ции `OC WRITE_SAMPLING_MESSAGE`. Функция вначале запрашивает свободный буфер функцией `OC chanGetBuffer()`. Эта функция просматривает все порты и создает список всех используемых буферов (буфер, содержащий самое свежее сообщение, считается используемым), находит свободный буфер и возвращает его идентификатор.

В полученный таким образом буфер функция копирует сообщение и отправляет его в канал с помощью функции `OC chanPutMessage()`. В результате буфер перестает быть закрепленным за портом-отправителем и становится буфером, содержащим самое свежее сообщение.

Функция `chanPutMessage()` отправляет уведомление об отправке сообщения всем портам-получателям, которым это требуется. Функция уведомления может, например, объявить поток драйвера работоспособным. Для портов, находящихся в стандартных процессах, функция уведомления не используется.

Если один поток попытается выводить сообщение функцией `WRITE_SAMPLING_MESSAGE` в порт, в который сейчас выводит сообщение другой поток, его сообщение будет проигнорировано.

`Sampling`-каналы могут иметь несколько портов-получателей. Находящиеся в псевдоразделах порты-получатели обслуживаются драйверами устройств.

Драйвер получает наиболее свежее сообщение, выведенное в канал, с помощью функции `OC chanGetMessage()`. Если это сообщение не было выведено ранее, драйвер выводит его через обслуживаемое им устройство. После вывода сообщения драйвер освобождает содержащий его буфер с помощью функции `chanPutBuffer()` и он может быть повторно использован для вывода сообщения.

Функция `chanGetMessage()` всегда порождает запрос на уведомление о выводе сообщения. Драйвер должен заранее зарегистрировать функцию обработки уведомления, так как иначе уведомления не будут посылаться. При выводе очередного сообщения функция `WRITE_SAMPLING_MESSAGE` направляет извещение драйверу.

На рисунке 2 представлено графическое изображение описанного порядка ввода и вывода сообщений.

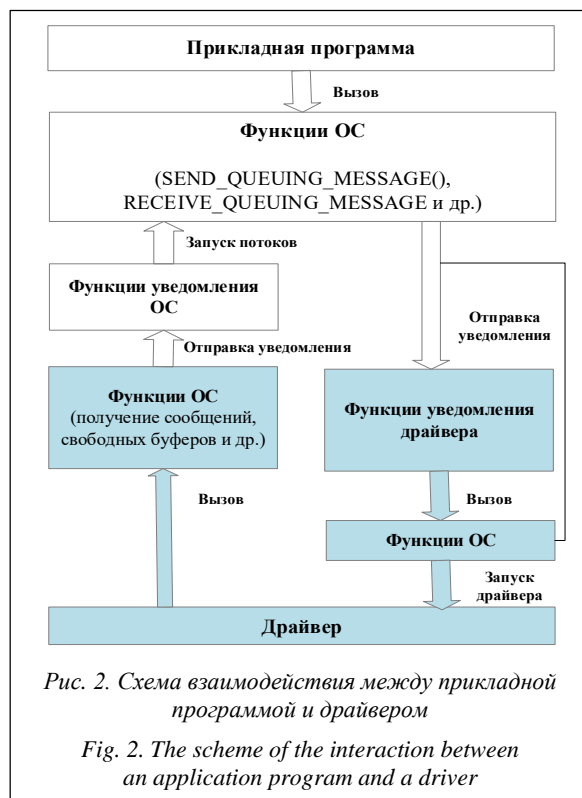
### Уведомления

Средства управления уведомлениями, входящие в состав `OSPB` Багет 3, играют важную роль при создании канальных драйверов. Рассмотрим типы существующих уведомлений, их назначение и пример использования в основном цикле головной функции драйвера канала.

Существуют следующие типы уведомлений (извещений):

- извещение о выводе сообщения в канал;
- извещение об освобождении буфера.





Уведомления о выводе сообщения порождаются владельцем порта-отправителя (обычным процессом или псевдоразделом) и направляются владельцу порта-получателя. Уведомления об освобождении буфера порождаются владельцем порта-получателя и направляются владельцу порта-отправителя.

Драйвер получает уведомление о выводе сообщения при обслуживании порта-получателя queueing-канала при выполнении следующих условий:

- драйвер заранее зарегистрировал функцию уведомления (о выводе сообщения);
- драйвер заранее породил запрос на уведомление о выводе сообщения.

Драйвер, обслуживающий порт-отправитель queueing-канала, получает уведомление об освобождении буфера при выполнении следующих условий:

- драйвер заранее зарегистрировал функцию уведомления (об освобождении буфера);
- драйвер заранее породил запрос на уведомление об освобождении буфера.

Запрос на уведомление о выводе сообщения в queueing-канал драйвер порождает неявно при вызове функции ОС `chanGetMessage()`, если очередь сообщений пуста. Запрос на уведомление об освобождении буфера драйвер также порождает неявно при вызове функции ОС `chanGetBuffer()`, если очередь свободных буферов пуста.

Драйвер, обслуживающий порт-получатель sampling-канала, получает уведомление о выводе сообщения при выполнении следующих условий:

- драйвер заранее зарегистрировал функцию уведомления (о выводе сообщения);
- драйвер заранее породил запрос на уведомление о выводе сообщения.

Запрос на уведомление о выводе сообщения в sampling-канал драйвер порождает неявно при вызове функции ОС `chanGetMessage()`.

Уведомления об освобождении буфера в sampling-канале никогда не порождаются, так как свободные буфера всегда есть.

Аппарат уведомлений является частью драйвера и создается разработчиком драйвера с помощью следующих функций OCPB:

- `chanNotificationInit()` – инициализация средств передачи уведомлений;
- `chanNotificationWait()` – ожидание уведомления или начала периода;
- `chanNotificationPut()` – постановка параметров уведомления в очередь;
- `chanNotificationGet()` – извлечение параметров уведомления из очереди.

Функция `chanNotificationInit()` должна быть вызвана потоком управления драйвера в начале его работы. Функция создает очередь уведомлений. Одному уведомлению соответствует один элемент очереди, который содержит два параметра. Функции уведомления драйвера помещают в очередь параметры уведомления с помощью функции `chanNotificationPut()`. Драйвер получает уведомления с помощью функции `chanNotificationGet()`. Если очередь уведомлений пуста, драйвер может использовать функцию `chanNotificationWait()` для ожидания появления уведомлений в очереди. Если драйвер работает в периодическом режиме, вызов функции `chanNotificationInit()` считается началом первого периода. Для ожидания начала следующего периода поток драйвера может использовать также функцию `chanNotificationWait()`.

## Заключение

В работе описана общая схема взаимодействия различных частей системы для обмена сообщениями в системах реального времени, сформулированы основные требования к драйверу, описаны этапы приема и передачи сообщений. Предложенные способы взаимодействия драйверов каналов с OCPB позволяют избежать записи данных в память чужого процесса, что существенно повышает надежность системы. Этот же интерфейс используется и в распределенных вычислительных системах, когда различные устройства системы соединяются между собой посредством шин с общим доступом.

Для OCPB Багет 3 разработаны и протестированы драйверы, обеспечивающие межмодульную связь по сети Ethernet по протоколу UDP, по шине VME, по шине RapidIO. Подготовлен прототип драйвера для Fibre Channel (FC).

Драйвер каналов, разработанный для коммуникационной среды RapidIO, которая отличается высоким быстродействием, используется в отечественной линейке процессоров [7] для решения задачи цифровой обработки сигналов [8] на многопроцессорной системе.

На описанную модель взаимодействия ОСРВ с драйвером канала получены патенты [9, 10].

### Литература

1. Годунов А.Н. Операционная система реального времени Baget 3.0 // Программные продукты и системы. 2010. № 4. С. 15–19.
2. Годунов А.Н., Солдатов В.А. Операционные системы семейства Багет (сходства, отличия и перспективы) // Программирование. 2014. № 5. С. 68–76.
3. Годунов А.Н., Солдатов В.А. Спецификация ARINC 653 и ее реализация в операционной системе реального времени Багет 3 // Программная инженерия. 2015. № 6. С. 3–17.
4. WIND RIVER VXWORKS 653 PLATFORM. URL: [http://www.windriver.com/products/product-overviews/PO\\_VE\\_6\\_9\\_Platform\\_0211.pdf](http://www.windriver.com/products/product-overviews/PO_VE_6_9_Platform_0211.pdf) (дата обращения: 29.03.2017).

6\_9\_Platform\_0211.pdf (дата обращения: 29.03.2017).

5. LynxOS-178 RTOS for DO-178B Software Certification. URL: <http://www.lynx.com/products/real-time-operating-systems/lynxos-178-rtos-for-do-178b-software-certification/> (дата обращения: 29.03.2017).

6. Safety Critical Products: INTEGRITY®-178B RTOS. URL: [http://www.ghs.com/products/safety\\_critical/integrity-do-178b.html](http://www.ghs.com/products/safety_critical/integrity-do-178b.html) (дата обращения: 29.03.2017).

7. Бобков С.Г. Импортозамещение элементной базы вычислительных систем // Вестн. РАН. 2014. Т. 84. № 11. С. 1010–1016.

8. Райко Г.О., Павловский Ю.А., Мельканович В.С. Технология программирования многопроцессорной обработки гидроакустических сигналов на вычислительных устройствах семейства «КОМДИВ» // Гидроакустика. 2014. № 20. С. 85–92.

9. Бетелин В.Б., Годунов А.Н., Грюнталь А.И., Солдатов В.А., Хоменков И.И. Способ передачи данных между процессами. Патент № 2592461, Российская Федерация; заявл. 12.05.2014, опубл. 20.07.2016.

10. Арышев С.И., Бетелин В.Б., Араkelов А.А., Солдатов В.А., Годунов А.Н., Малофеев Ю.Г., Хоменков И.И., Бобков С.Г. Способ передачи сообщений между вычислительными устройствами. Патент № 2611337, Российская Федерация; заявл. 13.01.2016, опубл. 21.02.2017.

Software & Systems

DOI: 10.15827/0236-235X.119.392-400

Received 30.03.17

2017, vol. 30, no. 3, pp. 392–400

### CHANNELS IMPLEMENTATION OF ARINC 653 SPECIFICATION IN RTOS BAGET 3

A.N. Godunov<sup>1</sup>, Ph.D. (Physics and Mathematics), Head of Department, [nkag@niisi.ras.ru](mailto:nkag@niisi.ras.ru)

V.A. Soldatov<sup>1</sup>, Ph.D. (Engineering), Senior Researcher, [nkvalera@niisi.ras.ru](mailto:nkvalera@niisi.ras.ru)

I.I. Homenkov<sup>1</sup>, Leading Engineer, [nkigor@niisi.ras.ru](mailto:nkigor@niisi.ras.ru)

<sup>1</sup> Federal State Institution "Scientific Research Institute for System Analysis of the Russian Academy of Sciences" (SRISA RAS), Nakhimovskiy Ave. 36/1, Moscow, 117218, Russian Federation

**Abstract.** The article considers ARINC specification 653 channels, which are intended for interprocess communication. The paper proposes the basic methods of designing channel driver, as well as their interaction with the Russian real-time operating system Baget 3 (RTOS Baget 3). RTOS Baget 3 user interface is based on ARINC 653 Specification and POSIX standard. The channel access is performed through ports. Every channel has the only port for sending messages and a single or several destination ports for receiving messages. Messages can be sent through a channel in only one direction. An application program can use the channels in the processes that correspond both ARINC and POSIX.

The paper specifies the requirements for channel drivers and describe notification control means used for driver development. It also considers the stages of initialization, sending/receiving messages and implemented algorithms. The proposed methods of implementing drivers interaction with RTOS allow avoiding data recording to the other process memory that significantly enhances the system reliability. Distributed computing systems use the same interface when separate subsystems are interconnected by means of shared access buses (VME, RapidIO, Fibre Channel) or Ethernet. The connection table for processor units is specified at the stage of system configuration. The channel driver, designed for RapidIO communication environment that offers fast operation speed, is used in a multiprocessor digital signal processing system.

**Keywords:** real-time, operating system, channels, drivers, multiprocessor systems, RapidIO. POSIX, ARINC 653.

### References

1. Godunov A.N. Real-time operating system baget 3.0. *Programmnye produkty i sistemy* [Software & Systems]. 2010, no. 4, pp. 15–19 (in Russ.).
2. Godunov A.N., Soldatov V.A. Operating Systems of the Baget Family (Likeness, Differences and Perspectives). *Programmirovaniye* [Programming and Computer Software]. 2014, no. 5, pp. 68–76 (in Russ.).
3. Godunov A.N., Soldatov V.A. ARINC Specification 653 and its Implementation in RTOS Baget 3. *Programmnaya inzheneriya* [Software Engineering]. 2015, no. 6, pp. 3–17 (in Russ.).
4. WIND RIVER VXWORKS 653 PLATFORM. Available at: [http://www.windriver.com/products/product-overviews/PO\\_VE\\_6\\_9\\_Platform\\_0211.pdf](http://www.windriver.com/products/product-overviews/PO_VE_6_9_Platform_0211.pdf) (accessed March 29, 2017).
5. LynxOS-178 RTOS for DO-178B Software Certification. Available at: <http://www.lynx.com/products/real-time-operating-systems/lynxos-178-rtos-for-do-178b-software-certification/> (accessed March 29, 2017).
6. Safety Critical Products: INTEGRITY®-178B RTOS. Available at: [http://www.ghs.com/products/safety\\_critical/integrity-do-178b.html](http://www.ghs.com/products/safety_critical/integrity-do-178b.html) (accessed March 29, 2017).
7. Bobkov S.G. Import substitution of the element base of computer systems. *Herald of the Russian Academy of Sciences*. 2014, vol. 84, iss. 11, pp. 1010–1016 (in Russ.).
8. Raiko G.O., Pavlovsky Yu.A., Melkanovich V.S. Technology of programming of multiprocessor processing of hydroacoustic signals on "KOMDIV" computer set. *Hydroacoustics*. 2014, no. 20 (2), pp. 85–92 (in Russ.).
9. Betelin V.B., Godunov A.N., Gryuntal A.I., Soldatov V.A., Homenkov I.I. *The Method for Transferring Data Between Processes*. Patent RF, no. 2692461, 2016 (in Russ.).
10. Aryashev S.I., Betelin V.B., Arakelov A.A., Soldatov V.A., Godunov A.N., Malofeev Yu.G., Homenkov I.I., Bobkov S.G. *A Method for Transmitting Messages Between Computing Devices*. Patent RF, no. 2611337, 2017 (in Russ.).

УДК 004.4

DOI: 10.15827/0236-235X.119.401-408

Дата подачи статьи: 17.04.17

2017. Т. 30. № 3. С. 401–408

## **АНАЛИЗ СОВРЕМЕННЫХ МЕТОДОВ ТЕСТИРОВАНИЯ И ВЕРИФИКАЦИИ ПРОЕКТОВ СВЕРХБОЛЬШИХ ИНТЕГРАЛЬНЫХ СХЕМ**

*Д.И. Слинкин, зав. группой, slin@nm.ru*

*(Федеральный научный центр Научно-исследовательский институт системных исследований  
РАН (ФНЦ НИИСИ РАН), Нахимовский просп., 36, корп. 1, г. Москва, 117218, Россия)*

Компании, разрабатывающие *сверхбольшие интегральные схемы* (СБИС), придерживаются определенных маршрутов проектирования и входящих в них маршрутов тестирования. Считается, что 60–80 % усилий команд-разработчиков аппаратных средств затрачивается на верификацию и отладку. Важным этапом является отладка моделей уровня регистровых передач (RTL). При этом не существует универсальной методики для решения этой задачи.

В статье анализируются зарубежные и отечественные публикации, посвященные отладке и верификации промышленных проектов СБИС. Разбираются четыре основные методологии: формальная верификация, имитационное тестирование, использование аппаратных ускорителей, создание прототипов на основе *программируемых логических интегральных схем* (ПЛИС). Для каждой из этих методологий приводится информация о способе выявления ошибок, существующих программных и аппаратных инструментах, которые могут быть применены в процессе отладки. Анализируются их особенности, такие как трудоемкость, требования к квалификации и численности команд верификаторов, стоимость необходимых инструментов, наличие метрик для оценки тестового покрытия. Приводятся названия конкретных промышленных проектов СБИС, в которых эти методологии были использованы: микропроцессоры, высокопроизводительные сетевые коммутаторы, графические процессоры. Называются примененные разработчиками инструменты.

Особое внимание уделено отладке проектов на ПЛИС. Разбираются следующие подходы: использование встроенного логического анализатора, внешнее контрольно-измерительное оборудование и их комбинирование.

На основе практического опыта показано, что четыре рассмотренные методологии отладки и верификации проектов СБИС имеют разграниченные области применения. Называются типы микросхем, для которых может быть использована конкретная методология. Кратко резюмируются их достоинства и недостатки.

**Ключевые слова:** *формальная верификация, имитационное тестирование, аппаратные ускорители, ПЛИС-прототипы, практическое применение.*

Ошибка в проекте *сверхбольших интегральных схем* (СБИС), обнаруженная на ранней стадии, удешевляет ее исправление. Наиболее дорогостоящим является изготовление новых фотошаблонов. Поэтому число итераций, связанных с изменением проекта на этапе тестирования готовых кристаллов, желательно минимизировать. Модель уровня RTL определяет основные характеристики будущей микросхемы. Статья посвящена анализу существующих методов отладки RTL-моделей с целью определения областей их применения. Рассмотрим основные методы отладки RTL-моделей.

**Формальная верификация (ФВ).** Применяются следующие методы ФВ: дедуктивный анализ, проверка эквивалентности, основанная на утверждениях верификация (ABV), проверка модели.

Дедуктивный анализ представляет собой математическое доказательство соответствия между верифицируемым проектом и спецификацией. Проект и спецификация должны быть представлены в терминах определенной формальной логики. Используемые средства автоматизированного доказательства теорем (theorem prover): HOL, Mizar, PVS, Coq, Isabelle, Alfa, ACL2. Метод применялся для отладки алгоритмов вещественной арифметики (умножение, деление, извлечение квадратного корня) процессора AMD-K7 (ACL2) [1] и сопроцессора вещественной арифметики процессора Intel Itanium (HOL-light) [2].

Основная идея ABV заключается в формализации знаний о работе проекта и применении их при моделировании в автоматическом режиме. Для контроля правильности работы проекта используются утверждения, написанные на языках PSL и SVA. Возможна автоматическая генерация утверждений при помощи DesignPSL. Существуют как свободно распространяемые библиотеки утверждений (OVL), так и коммерческие (QVL, CheckerWare). Библиотечные элементы содержат блоки для тестирования простых структур (счетчики, буферы, константы) и для сложных протоколов (шины AMBA, PCI, USB, интерфейс DDR) [3]. Библиотеки используются также при моделировании, эмуляции или для ПЛИС-прототипов. Для ПЛИС-прототипов возможно комбинирование проверки утверждений и задания режима работы программным тестом.

Проверка эквивалентности. Утверждение о работоспособности модели проекта основывается на доказательстве эквивалентности другой модели. Например, эквивалентности схемы на уровне транзисторов и RTL-модели. Инструментами являются Conformal, Formality, FormalPro.

Проверка модели связана с формальной проверкой выполнения на модели свойств поведения проектируемого объекта, специфицированных на языке формальной логики. Метод может быть полностью автоматизирован. Проблемой является

комбинаторный взрыв числа состояний системы при увеличении числа компонентов. Для ее решения используются символьная верификация на основе бинарных решающих диаграмм (BDDs) или решатели задач выполнимости булевых формул (SAT-solvers). Инструментами являются SMV, Magellan, Forte, Murphy. Метод использовался для верификации протоколов когерентности кэш-памяти в Intel (Murphy) [4] и для графических процессоров NVIDIA (Magellan) [5]. Для процессора Intel Core i7 проведена полная ФВ кластера EXE (Forte) [6]. При этом остались незамеченными пять проблем, три из которых были выявлены уже после изготовления кристалла. Проблемы связаны с некорректной спецификацией или с несвоевременным окончанием работ по ФВ [6].

В отличие от других подходов ФВ не обеспечивает равномерного улучшения качества результата тестирования от времени. Методология применяется в крупных компаниях с большими коллективами инженеров-верификаторов. Использование ФВ ограничено отладкой отдельных особо ответственных узлов. Главным недостатком является проверка абстрактной модели, а не реальной системы. Метод обеспечивает высокое качество верификации, но не гарантирует полное отсутствие ошибок.

**Имитационное тестирование.** В настоящее время имитационное тестирование, или моделирование, является самым распространенным подходом к верификации. Различают системную и модульную верификации. Модульная верификация выполняется для отдельного блока или для нескольких логически связанных блоков. Задачей системной верификации является проверка взаимодействия между компонентами, а следовательно, проверка устройства в целом [7].

Работа устройства моделируется при помощи программного RTL-симулятора. Существуют ПО с открытым исходным кодом (Icarus, GHDL, CVC) и бесплатные программы с ограниченной функциональностью (Active-HDL и ModelSim). Для рабочих мест на ПК применяются ModelSim, Active-HDL, Riviera-PRO, Silos, VeriLogger Extreme и др. Для предприятий предлагаются инструменты, обеспечивающие точное временное моделирование на вентильном уровне, что критически важно при передаче проектов СБИС на производство, например Incisive Enterprise Simulator, ModelSim/SE, Synopsys VCS. Проектирование на ПЛИС не требует дорогих симуляторов, используются инструменты ModelSim, Altera Quartus II и Xilinx ISE.

В процессе имитационного тестирования RTL-модель запускается в рамках тестовой системы (testbench), для создания которой используются или языки программирования C, C++, или языки описания аппаратуры SystemC, Verilog, VHDL, или специальные языки верификации OpenVera, SystemVerilog. Тестовая система выполняет три основ-

ные задачи: генерация тестовых воздействий, проверка правильности поведения модели и оценка полноты тестирования. Применяются следующие методы создания тестовых воздействий: ручная разработка, случайная генерация, генерация на основе шаблонов, генерация на основе конечных автоматов. Анализ ошибок и оценка полноты тестирования выполняются на основе отчета о тестировании. Для проверки правильности поведения модели используются ко-симуляция (сравнение результатов теста с эталонной моделью), тесты со встроенной проверкой и формальные спецификации [8]. Программы-симуляторы обычно имеют возможность оценки полноты покрытия RTL-модели: количество выполненных строк кода, статистика прохождения ветвлений, полнота комбинаций аргументов.

Для создания тестовых стендов используются специальные методологии верификации. В 2010 г. независимая организация Accellera, занимающаяся разработкой стандартов, предложила методологию UVM (Universal Verification Methodology), созданную на основе OVM и включающую в себя AVMM, URM, VMM и eRM. Методология UVM поддерживается тремя основными компаниями-разработчиками HDL-симуляторов: Cadence, Synopsys и Mentor.

В UVM применяются объектно-ориентированное программирование и моделирование на уровне транзакций (TLM). Для быстрого создания тестовых окружений и компонент верификации на языке System Verilog служит библиотека классов. Используются случайная генерация входных воздействий на основе ограничений и управляемая покрытие верификация. Предполагается повторное применение компонент верификации и тестовых окружений. UVM считается перспективной и широко распространенной методологией верификации.

В НИИСИ РАН методология UVM использовалась для отладки блока преобразования адресов TLB в составе микропроцессора [9]. Сообщается об использовании UVM для верификации контроллера прерываний APIC микропроцессора «Эльбрус 2S» в ЗАО «МЦСТ» и об использовании OVM для верификации модуля кэша второго уровня того же микропроцессора [7].

Также в НИИСИ РАН была разработана интегрируемая система INTEG, использующая случайную генерацию тестов по шаблонам. Система включает в себя графическую оболочку, графический редактор шаблонов, генератор случайных тестов, симулятор целевого процессора VMIPS (эталонная модель), симулятор RTL-модели (Verilog) и программу сравнения результатов. Система INTEG доказала свою полезность при верификации проекта микропроцессора с архитектурой MIPS64. Совершенствование разрабатываемых в НИИСИ РАН микропроцессоров, изменения в организации, спе-

нариях и инструментах тестирования обусловили разработку модернизированного программного комплекса INTEG2 [10].

В ИСП РАН для верификации микропроцессов и других программируемых устройств (сопроцессоров, контроллеров) на системном уровне разработан инструмент MicroTESK [11]. Генератор создает набор тестовых программ небольшого размера. Имеется поддержка автоматической генерации тестовых шаблонов на основе комбинаторных техник. MicroTESK использовался для отладки MIPS-64 совместимого микропроцессора, DSP-сопроцессора и арифметических сопроцессоров вещественных чисел. Для верификации на модульном уровне используются технология UniTESK и инструмент CTESK, осуществляющий автоматизированную разработку тестов на основе формальных спецификаций. CTESK был применен для отладки Verilog-моделей кэш-памяти второго уровня и буфера трансляции адресов MIPS64-совместимого микропроцессора. Инструмент C++TESK предполагает создание спецификации и всех компонент тестовой системы на языке Си++. Тестовые системы создаются на основе эталонных программных моделей различных уровней абстракции. Сообщается о применении рассматриваемого метода в виде библиотеки классов для девяти проектов верификации модулей промышленных микропроцессоров [12].

Бутылочным горлышком имитационного тестирования является низкая частота моделирования, десятки герц. Скорость моделирования отдельного блока на порядок выше скорости моделирования устройства в целом. Комбинаторная сложность тестов для современных систем на кристалле (СнК) высока, поэтому в литературе зачастую сообщается о тестировании только отдельных, наиболее ответственных блоков микропроцессоров. Ограничивающим фактором становится не только машинное время для прохождения тестов, но и время разработки полного набора тестов. Необходимы высокий уровень знаний UVM-инженеров в области разработки аппаратных средств, в области разработки ПО, а также знание методов верификации. Неполная документация способна снизить качество верификации. Недостатком случайной генерации тестов является малая вероятность нахождения сложных ошибок, для которых требуется привести модель устройства в особое состояние. Генерация тестов при обходе конечного автомата позволяет находить сложные ошибки, но требует адекватности используемой эталонной модели [12].

Достоинствами имитационного моделирования являются наличие метрик, позволяющих количественно оценивать тестовое покрытие, и возможность обнаружения непредсказуемых ошибок при выполнении случайных тестов. Автоматическая генерация входных воздействий и автоматическая проверка результатов тестирования снижают тру-

доемкость тестирования и обеспечивают высокий процент тестового покрытия.

**Аппаратные ускорители.** Для увеличения скорости моделирования на несколько порядков применяются аппаратные ускорители, или платформы эмуляции. Представленные на рынке ускорители реализованы на специальных процессорах, на заказных или на коммерческих ПЛИС (табл. 1). Все три архитектуры являются масштабируемыми, они позволяют размещать проекты любого размера, от IP-блоков до полных систем. Важным является тестирование взаимодействия на системном уровне, поддерживаемое аппаратными ускорителями. Частота зависит от многих факторов и составляет от единиц МГц до 100 МГц (для HAPS-80). Обычно поддерживаются методология UVM, верификация на основе утверждений, многопользовательская работа, модели уровня транзакций (TLM).

Современные СнК могут включать в себя несколько процессорных ядер и высокопроизводительные интерфейсы. Для управления используются операционная система и драйверы.

Таблица 1

Аппаратные ускорители и платформы для эмуляции  
Table 1  
Hardware accelerators and platforms for emulation

Платформа	Компания	Микросхемы	Количество вентилей в проекте
Palladium	Cadence	Специализированные процессоры	До 2 млрд
Protium	Cadence	ПЛИС Xilinx Virtex-7	До 100 млн
Veloce II	Mentor	Специализированные ПЛИС Crystal2	От 256 млн до 2 млрд
ZeBu Server-3	Synopsys	ПЛИС Xilinx Virtex-7	300 млн, масштабируется до 3 млрд
HAPS-80	Synopsys	ПЛИС Xilinx Virtex UltraScale	До 1,6 млрд
Prodigy Cloud Cube 32	S2C	ПЛИС Xilinx Virtex-7 (UltraScale, Kintex-7) или Altera Stratix IV	До 1,4 млрд

Эмуляция на основе процессоров реализована в ускорителях семейства Palladium. Аппаратная часть эмулятора состоит из массива специализированных процессоров, выполняющих булевы операции тестируемого проекта. Эмулятор включает в себя от 103 до 104 арифметико-логических устройств. Задача ПО состоит в разделении проекта между процессорами и в оптимальном планировании булевых операций в корректной временной последовательности. Поддержка стандартных интерфейсов реализована за счет SpeedBridges – карт адаптации частот эмулятора и периферийных устройств.

Основными преимуществами Palladium являются быстрое время компиляции и полная видимость проекта на полной скорости эмуляции. Palladium эффективен в режиме ICE, поддерживаемом набором карт SpeedBridges. Имеется возможность оценки питания проекта с целью уменьшения энергопотребления.

Главные недостатки Palladium – большие физические размеры и высокое энергопотребление по сравнению с ПЛИС-эмулятором эквивалентной вычислительной мощности [13]. Ограничена скорость в режиме Transaction-based acceleration. Также следует отметить высокую стоимость аппаратных эмуляторов.

Ускорители на основе специализированных ПЛИС обеспечивают стопроцентный доступ без компиляции пробников и быструю трассировку временных диаграмм. Недостатком является использование фермы рабочих станций для быстрой компиляции, скорость которых ниже, а физические размеры больше, чем у эквивалентного эмулятора на основе коммерческих ПЛИС.

Ускорители на основе коммерческих ПЛИС обладают наименьшими физическими размерами и потребляемой мощностью. Они достигают более высокой скорости выполнения. Недостатки заключаются в меньшей скорости компиляции по сравнению с другими двумя архитектурами, по крайней мере, для проектов в 10 млн логических элементов или меньше. Полная видимость проекта достигается в обмен на высокую скорость эмуляции.

Cadence сообщает об использовании Palladium XP II компанией Realtek для разработки и верификации проекта СнК. Отмечаются ускорение в 250 раз по сравнению с предыдущей методологией, улучшение качества верификации, экономия времени за счет повторного использования более чем 90 % предыдущих настроек среды моделирования, а также оценка потребляемой мощности в пределах 5 % от фактически измеренной мощности СнК.

Palladium XP использовался компанией Altair Semiconductor (выкуплена Sony в 2016 г.) для верификации и валидации проекта Internet of Thing. Сообщается об уменьшении времени цикла разработки на 20 % [14]. Выполнены разработка ПО и его проверка в аппаратном контексте.

Компания NVIDIA применяла установку, состоящую из 16 шасси, под названием Tigris (Cadence) для эмуляции архитектуры GPU Fermi. Самая крупная установка Palladium XP под названием Indus использовалась для эмуляции архитектуры следующего поколения Kepler [15].

В НИИСИ РАН Palladium был использован для отладки микропроцессора 1890BM8Я. В настоящее время ведется тестирование проектируемого микропроцессора 1890BM108Я с использованием Protium.

Компания AMD использовала ZeBu для эмуляции графического процессора (GPU), имеющего

240 млн вентиляей. Было задействовано от 50 до 70 ПЛИС. Применен подход, основанный на транзакциях. Проверялось функционирование GPU в среде IBM PC-совместимой ЭВМ. Различные материнские платы IBM-PC моделировались при помощи виртуальной машины VirtualBox. Производилась загрузка ОС Linux, драйвера, выполнялись операции чтения/записи памяти эмулируемого GPU. Также были выполнены 20 базовых аппаратных тестов для GPU [16].

Платформа эмуляции Veloce была использована Barefoot Networks для проверки высокопроизводительного сетевого коммутатора Tofino 6.5 ТБ/сек. Важным было использование инструмента Veloce Virtual LAB Ethernet, включающего в себя генератор и монитор пакетов Ethernet (EPGM) [17].

НАPS-70 использовался для верификации двухъядерного микропроцессора Baikal-T1 компании «Байкал электроникс». Микропроцессор работает на частоте 1,2 ГГц, использует 32-битные ядра MIPS Warrior P5600, имеет в своем составе ряд высокоскоростных периферийных устройств, таких как SATA, PCI Express, и Ethernet 10 Гбит/с. Тестирование проводилось в несколько этапов. Сообщается о загрузке ОС Linux и запуске одного ядра на частоте 25 МГц [18].

Достоинствами аппаратных ускорителей являются короткое время старта работ по верификации и хорошие возможности для отладки.

Недостаток состоит в высокой стоимости при избыточных возможностях для некоторых задач, таких, например, как тестирование контроллера периферийного интерфейса. Частота моделирования существенно ниже по сравнению со специальными платами для отладки ПЛИС-прототипов.

ПЛИС-прототипы. Высокая скорость эмуляции, в сотни МГц, достигается при помощи модулей на основе ПЛИС. Можно использовать как готовые, так и специально разработанные отладочные платы. Возможно непосредственное встраивание ПЛИС-прототипа в реальную систему, например, путем установки в разъем PCI-e IBM-PC совместимой ЭВМ. Отладочная плата содержит различные устройства: генераторы тактовых сигналов, приемопередатчики Ethernet, высокопроизводительное ОЗУ, видеоинтерфейсы.

В НИИСИ РАН отладочные платы разрабатываются и изготавливаются специально в количестве нескольких экземпляров. Такой подход позволяет максимально приблизить их характеристики к будущим модулям за счет установки соответствующих компонент. При этом начало тестирования задерживается до момента получения плат. Стоимость таких плат ниже по сравнению с аппаратными ускорителями. Возможность параллельной разработки системного ПО (драйверов) и микросхемы снижает время выхода на рынок. Разнообразие отладочных плат на рынке свидетельствует об их активном применении.

Существуют три подхода к отладке проектов на ПЛИС: использование встроенного логического анализатора, использование внешнего контрольно-измерительного оборудования и их комбинирование.

Встроенный логический анализатор предполагает размещение на ПЛИС аппаратных средств отладки, предназначенных для отслеживания внутренних состояний в моменты срабатывания триггеров, устанавливаемых разработчиками. Обычно анализатор встраивается в ПЛИС за счет добавления одного или нескольких IP-ядер. Для размещения логики встроенного анализатора и буферов памяти требуется от 5 % до 10 % ресурсов ПЛИС. Через порт JTAG ядро анализатора можно сконфигурировать и динамически управлять им. Этот порт используется и для вывода данных. Встроенный логический анализатор позволяет отслеживать любые внутренние сигналы в реальном времени. Для получения доступа к сигналам работа системы останавливается, цепочки сканирования (scan chains) вместе с захваченными данными передаются из ПЛИС. Затем полученная информация анализируется на IBM-PC с использованием программ с графическим интерфейсом или текстовой консоли. В настоящее время на рынке предлагаются различные встроенные логические анализаторы (табл. 2).

Отладка проектов на ПЛИС может производиться с использованием внешнего контрольно-измерительного оборудования, такого как цифровые осциллографы смешанных сигналов или логические анализаторы. Ведущие изготовители – компании Tektronix и Agilent. Примерами являются осциллографы смешанных сигналов серии MSO4000 и логические анализаторы серии TLA компании Tektronix.

Технологические платы с ПЛИС должны иметь специальные разъемы для подключения внешнего оборудования. Сложность отладки обусловлена ограниченным доступом к внутренним сигналам ПЛИС, взаимным влиянием сигналов на печатной плате. Многие устройства имеют ограниченное число выводов, меньшее, чем количество сигналов, требуемых для отладки. Поскольку возможности встроенных логических анализаторов ограничены, существуют программы, такие как Altera Signal-

Probe, которые позволяют выбирать нужные внутренние сигналы и перенаправлять их на свободные выводы ПЛИС.

Возможно комбинирование встроенных ядер отладки и внешнего контрольно-измерительного оборудования. Например, ядро Agilent Trace Core 2 (ATC-2) в составе программы Xilinx Chipscope Pro Analyzer специально разработано для подключения внешнего логического анализатора Agilent и обеспечивает доступ анализатора к внутренним сетям проекта на ПЛИС. Логический анализатор обеспечивает задание сложных условий захвата данных. Для сохранения данных используется большой объем ОЗУ, позволяющий отслеживать разнесенные во времени события. Возможно сопоставление сигналов проекта на ПЛИС с другими сигналами системы.

Программа FPGAView от компании First Silicon Solutions (FS2) используется с логическими анализаторами серии TLA или осциллографом смешанных сигналов от компании Tektronix. Предоставляется возможность измерения различных внутренних сигналов ПЛИС Altera или Xilinx без необходимости перекомпиляции проекта. Автоматическое изменение имен каналов на имена выбранных внутренних сигналов облегчает интерпретацию результатов.

Встроенный логический анализатор не требует дополнительных выводов и использует для реализации своих функций внутренние ресурсы ПЛИС. Объем ОЗУ внутри ПЛИС ограничивает суммарное количество отладочной информации, являющееся компромиссом между количеством захватываемых сигналов и глубиной их отслеживания.

Внешнее контрольно-измерительное оборудование дороже встроенного логического анализатора, но обладает определенными преимуществами. Цифровой осциллограф позволяет запускаться от широкого спектра аналоговых и цифровых сигналов, а также от сигналов последовательных шин, захватывать их с различным разрешением по времени. Внешний логический анализатор открывает доступ к различным состояниям запуска и может захватывать очень длинные последовательности данных с высоким разрешением по времени. Внешний осциллограф позволяет захватывать до 10 млн точек, а логический анализатор до 256 млн. Име-

Таблица 2

## Встроенные логические анализаторы

Table 2

## Built-in Logic Analyzers

Анализатор	ПЛИС	Интеграция с САПР	Особенность	Количество каналов данных	Объем буфера
Altera SignalTap II	Altera	Quartus II	Частота выше 300 МГц	2 048	128 К
FS2 Logic Navigator	Actel	-	Интегрируется в проект на языке Verilog или VHDL	256	64 К
Xilinx Chipscope Pro	Xilinx	Quartus II	Интеграция с внешним логическим анализатором Agilent	4 096	128 К



ется преимущество в динамичности: не требуется останавливать работу системы для анализа захватываемых данных. Внешнее оборудование позволяет изучать временные диаграммы сигналов с разрешением меньше наносекунды, тогда как встроенный логический анализатор может захватывать данные только синхронно с внутренней тактовой частотой ПЛИС. Использование внешнего контрольно-измерительного оборудования имеет некоторые ограничения: получение доступа к новому набору сигналов требует перекомпиляции проекта, что отнимает время, может привести к изменению временных характеристик и скрыть исходные проблемы. Обычно число отладочных выводов мало, что ограничивает обзор и глубину анализа.

Эти ограничения можно преодолеть путем комбинирования отладочных ядер и внешнего контрольно-измерительного оборудования. Выбор конкретной методики зависит от особенностей проекта. Если отладка ограничивается функциональными проблемами внутри ПЛИС, то достаточно возможностей встроенного логического анализатора. Внешнее оборудование предпочтительнее при определении предельных значений временных характеристик, сопоставлении внутренней активности ПЛИС с работой других устройств на печатной плате.

Недостаток отладки проектов на ПЛИС состоит в отсутствии метрик покрытия RTL-кода и метрик покрытия функциональности. Тесты в основном находят ошибки, поиск которых ведется целенаправленно. Поэтому некоторые ошибки могут остаться незамеченными. Выявить их помогают случайные тесты. Считается, что утверждения System Verilog Assertions (SVA) сокращают время тестирования проекта на 50 %. При их использовании улучшается наблюдаемость проекта, упрощаются поиск и исправление недостатков.

Примером отладки ПЛИС-прототипа является эмуляция ядра x86 микропроцессора Intel Atom (см. [19]). Авторами предложена методология использования RTL-проекта и создания его ПЛИС-синтезируемой версии с применением стандартного инструментария. Использовалась специально разработанная отладочная плата на основе ПЛИС Xilinx Virtex-5. Наиболее сложным аспектом являлась механическая адаптация разъема Socket7 материнской платы и ПЛИС.

Отладка сводилась к регрессионному тестированию и сравнению проекта с существующим процессором Intel Pentium. Чтобы избежать ошибок, вносимых в RTL-код непреднамеренно, использовались три метода: RTL-моделирование, сравнение исполнения программ и онлайн-отладка. Для проверки функциональной полноты выполнялась загрузка стандартных ОС: MS-DOS, Linux, Windows XP и различных приложений. Также оценивалась производительность [19].

В НИИСИ РАН были разработаны ПЛИС-прототипы графического ускорителя и универсального микропроцессора с различными внешними интерфейсами. Графический 2D-ускоритель реализует аппаратную поддержку системы команд X Windows, принятую для ОС семейства UNIX. Ускоритель 2D является встроенным узлом микросхем 1890ВГ10Т и 1890ВГ16Т. Метод встречной оптимизации программ под аппаратуру служит повышению реальной (в отличие от пиковой) производительности, и наоборот, создаваемая аппаратура учитывает назначение и особенности программ пользователей. Метод встречного тестирования предполагает привлечение контрольных тестовых задач от потенциальных пользователей [20]. Предложен метод тестирования производительности и корректности микропроцессоров нацеленными тестовыми программами [21]. Данные методы предполагают использование ПЛИС-прототипов. Они применялись для микропроцессоров 1890ВМ5Ф, 1890ВМ6Я, 1890ВМ8Я и 1890ВМ9Я.

### Заключение

Рассмотренные методы отладки и верификации проектов СБИС имеют достаточно разграниченные области применения.

Формальная верификация используется крупными компаниями для ответственных узлов, например, арифметических алгоритмов процессоров, сопроцессоров, кэш-памяти, графических процессоров. Требуется работа коллектива специалистов.

Имитационное моделирование широко распространено и обычно применяется для отладки отдельных блоков микросхем. Частота моделирования составляет десятки герц, что ограничивает область применения. Сильными сторонами являются наличие метрик покрытия, автоматическая генерация тестов, методологии OVM/UVM.

Аппаратные ускорители обеспечивают высокую скорость, обычно несколько МГц. Они позволяют производить тестирование проектов СнК на системном уровне. При этом обеспечивается полная видимость проекта. Недостаток состоит в больших размерах и высокой стоимости аппаратного ускорителя. Основная сфера применения – микропроцессоры, графические процессоры, сетевые приложения. Платы для создания ПЛИС-прототипов обладают наибольшей скоростью в сотни МГц, но меньшими отладочными возможностями. Размеры позволяют встраивать их в реальную систему, а стоимость сравнительно невелика.

Возможен комбинированный подход, включающий, например, имитационное моделирование отдельных блоков и использование ПЛИС-прототипов для проверки проекта в целом.

### Литература

1. Russinoff D. A mechanically checked proof of IEEE compliance of the floating-point multiplication, division, and square root



- algorithms of the AMD-K7\* processor. *Jour. of Computation and Mathematics*, 1998, pp. 148–200. URL: <http://citeseerx.ist.psu.edu/viewdoc/download?rep=rep1&type=pdf&doi=10.1.1.212.6294> (дата обращения: 07.04.2017).
2. Harrison J. Floating-point verification using theorem proving. URL: <http://www.cl.cam.ac.uk/~jrh13/papers/sfm.pdf> (дата обращения: 07.04.2017).
3. Лохов А.Л. Современные методы функциональной верификации цифровых HDL-проектов: методология ABV, библиотеки OVL и QVL // *Современная электроника*. 2010. № 1. С. 56–59.
4. Chou C.-T., Manna P.K., Park S. A simple method for parameterized verification of cache coherence protocols. 2004. URL: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.101.5271&rep=rep1&type=pdf> (дата обращения: 07.04.2017).
5. Synopsys' Magellan Deployed by NVIDIA to maximize verification productivity on next-generation graphics processing units. 2004. URL: <http://news.synopsys.com/index.php?s=20295&item=122694> (дата обращения: 07.04.2017).
6. Kaivola R., Ghughal R., Narasimhan N., Telfer A., Whittemore J., Pandav S., Slobodova A., Taylor C., Frolov V., Reeber E., Naik A. Replacing testing with formal verification in intel coretm7 processor execution engine validation. 2010. URL: <http://is.muni.cz/el/1433/jaro2010/IA159/um/intel.pdf> (дата обращения: 07.04.2017).
7. Ровнягин М.М., Лебедев М.С., Чудновский А.Л. Современные методы верификации и особенности их применения // *Современные информационные технологии и ИТ-образование* // VI Междунар. науч.-практич. конф.: сб. избр. тр.; [под ред. В.А. Сухомлина]. М.: Изд-во ИНТУИТ.РУ, 2011. С. 1009–1020.
8. Камкин А.С. Верификация микропроцессоров: борьба с ошибками и управление качеством // *Электроника: НТБ*. 2010. № 3. С. 98–104. URL: <http://www.electronics.ru/journal/article/57> (дата обращения: 07.04.2017).
9. Барских М.Е., Аряшев С.И., Рогаткин Б.Ю. Современные методы функциональной верификации RTL-моделей блоков СБИС микропроцессора // *Проблемы разработки перспективных микро- и наноэлектронных систем: сб. тр.; [под общ. ред. А.Л. Стемповского]*. М.: Изд-во ИПИМ РАН, 2014. Ч II. С. 119–122. URL: <http://www.mes-conference.ru/data/year2014/pdf/D066.pdf> (дата обращения: 07.04.2017).
10. Захаров А.В., Хисамбеков И.Ш., Котович Н.В., Кравченко А.А., Осипов А.С., Кольцов П.П., Коганов М.А., Грибков И.В., Куцаев А.С. Развитие системы стохастического тестирования микропроцессоров INTEG // *Программные продукты и системы*. 2010. № 2. С. 14–23.
11. Камкин А.С., Коцыняк А.М., Смолов С.А., Сортон А.А., Татарников А.Д., Чупилко М.М. Средства функциональной верификации микропроцессоров // *Тр. ИСП РАН*. Т. 26 (1). С. 149–200. URL: <http://docplayer.ru/271393-Sredstva-funkcionalnoy-verifikacii-mikroprocessorov.html> (дата обращения: 07.04.2017).
12. Чупилко М.М. Динамическая верификация цифровой аппаратуры на основе формальных спецификаций: автореф. дис. канд. физ.-мат. наук. М.: ИСП РАН, 2012. 24 с.
13. Rizzatti L. Hardware emulation: three decades of evolution. Part III. Verification Horizons. 2015. URL: <https://verification-academy.com/verification-horizons/november-2015-volume-11-issue-3/hardware-emulation-three-decades-of-evolution-part-iii> (дата обращения: 07.04.2017).
14. Altair semiconductor adopts cadence palladium XP platform for advanced IoT SoC development. 2015. URL: [https://www.cadence.com/content/cadence-www/global/en\\_US/home/company/newsroom/press-releases/pr/2015/altair-semiconductor-adopts-cadence-palladium-xp-platform-for-advanced-iot-soc-development.html](https://www.cadence.com/content/cadence-www/global/en_US/home/company/newsroom/press-releases/pr/2015/altair-semiconductor-adopts-cadence-palladium-xp-platform-for-advanced-iot-soc-development.html) (дата обращения: 07.04.2017).
15. Mackey H. Sneak peek: inside NVIDIA'S emulation lab. 2011. URL: <https://blogs.nvidia.com/blog/2011/05/16/sneak-peek-inside-nvidia-emulation-lab> (дата обращения: 07.04.2017).
16. Ross A., Star A. Enabling greater reliability, scalability and flexibility of GPU emulation at AMD using a hybrid virtual-machine based approach. 2015. URL: <http://www.techdesignforums.com/practice/technique/reliable-scalable-flexible-gpu-emulation-using-hybrid-virtual-machine-approach/> (дата обращения: 07.04.2017).
17. Mentor Graphics Veloce emulation platform helps barefoot networks verify the world's first fully programmable switch. *Jour. Electronic Engineering*. 2016. URL: [http://www.eejournal.com/archives/news/20160719\\_06/](http://www.eejournal.com/archives/news/20160719_06/) (дата обращения: 07.04.2017).
18. Osipenko P. Synopsys and Baikal Electronics SoC team applies Synopsys HAPS and AMBA transactors to accelerate the availability of prototypes and improve product quality. 2015. URL: [https://www.synopsys.com/content/dam/synopsys/verification/prototyping/success-stories/baikal\\_ss.pdf](https://www.synopsys.com/content/dam/synopsys/verification/prototyping/success-stories/baikal_ss.pdf) (дата обращения: 07.04.2017).
19. Wang P.H., Collins J.D., Weaver C.T., Kuttanna B., Salamian S., China G.N., Schuchman E., Schilling O., Doil T., Steibl S., Wang H. Intel atom processor core made FPGA-synthesizable. URL: <http://www.cse.wustl.edu/~roger/565M.fl2/p209-wang.pdf> (дата обращения: 01.04.17).
20. Чибисов П.А. Встречное тестирование высокопроизводительных микропроцессоров: автореф. дис. канд. техн. наук. М.: ИСП РАН, 2013. 23 с.
21. Зубковская Н.В. Метод тестирования производительности и корректности микропроцессоров при помощи нацеленных тестовых программ: автореф. дис. канд. техн. наук М.: ИИИСИ РАН, 2013. 24 с.

## ANALYSIS OF MODERN METHODS FOR VLSI PROJECT TESTING AND VERIFICATION

*D.I. Slinkin*<sup>1</sup>, Head of Group, *slin@nm.ru*<sup>1</sup> *Federal State Institution "Scientific Research Institute for System Analysis of the Russian Academy of Sciences" (SRISA RAS), Nakhimovskiy Ave. 36/1, Moscow, 117218, Russian Federation*

**Abstract.** VLSI development companies are keeping selected design flows and a testing process is a part of them. It is considered that from 60 % to 80 % of development teams' efforts are spent on VLSI project verification and debugging. Debugging of register-transfer level (RTL) models is an important phase. There is no universal way to solve this problem.

The article is devoted to the analysis of foreign and domestic publications on industrial VLSI project debugging. There are four main methodologies that are being considered: formal verification, simulation testing, using of hardware accelerators, prototyping based on programmable logic integrated circuits (FPGAs). For each of these methodologies there is information on an error detection method, existing software and hardware debugging tools. The paper analyzes some of their features, such as labor consumption, requirements for qualification and size of verification group, the cost of necessary tools and the availability of metrics to evaluate the test coverage. There are names of some industrial VLSI projects that used these methodologies, such as microprocessors, high-performance network switches, graphics processors. The paper mentions the tools used by the developers.

Special attention is paid to debugging FPGA projects. The following approaches are considered: using the built-in logic analyzer, external control and measuring equipment and their combination.

Finally, based on the practical experience the paper shows that four VLSI project debugging and verifying methodologies have different applications. It mentions the types of VLSI, which use one of these methodologies. Their advantages and disadvantages are briefly summarized.

**Keywords:** formal verification, simulation testing, hardware accelerators, FPGA prototypes, practical application.

### References

1. Russinoff D. A mechanically checked proof of IEEE compliance of the floating-point multiplication, division, and square root algorithms of the AMD-K7\* processor. *London Mathematical Society Journ. of Computation and Mathematics*. 1998, pp. 148–200. Available at: <http://citeseerx.ist.psu.edu/viewdoc/download?rep=rep1&type=pdf&doi=10.1.1.212.6294> (accessed April 7, 2017).
2. Harrison J. *Floating-Point Verification using Theorem Proving*. Available at: <http://www.cl.cam.ac.uk/~jrh13/papers/sfm.pdf> (accessed April 7, 2017).
3. Lokhov A. Advanced Methods for Functional Verification of HDL Projects: ABV Methods, OVL and QVL Libraries. *Sovremennaya elektronika* [Modern Electronics]. Moscow, 2010, no. 1, pp. 56–59 (in Russ.).
4. Chou C.-T., Manna P.K., Park S. *A Simple Method for Parameterized Verification of Cache Coherence Protocols*. Intel Corp. 2004. Available at: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.101.5271&rep=rep1&type=pdf> (accessed April 7, 2017).
5. Synopsys' Magellan Deployed by NVIDIA to Maximize Verification Productivity on Next-Generation Graphics Processing Units. 2004. Available at: <http://news.synopsys.com/index.php?s=20295&item=122694> (accessed April 7, 2017).
6. Kaivola R., Ghughal R., Narasimhan N., Telfer A., Whittemore J., Pandav S., Slobodova A., Taylor C., Frolov V., Reeber E., Naik A. *Replacing Testing with Formal Verification in Intel CoreTMi7 Processor Execution Engine Validation*. 2010. Available at: <http://is.muni.cz/el/1433/jaro2010/IA159/um/intel.pdf> (accessed April 7, 2017).
7. Rovnyagin M.M., Lebedev M.S., Chudnovsky A.L. Modern methods of verification and their applications features. *Sb. izbran. tr. VI Mezhdunar. nauch.-praktich. konf. "Sovremennye informatsionnye tekhnologii i IT-obrazovanie"* [Proc. 6th Int. Science and Practice Conf. "Modern IT and IT-education"]. V.A. Sukhomlin (Ed.). Moscow, 2011, pp. 1009–1020 (in Russ.).
8. Kamkin A.S. Microprocessor Verification. Combating Errors and Quality Control. *Elektronika: NTB* [Electronics: STB]. Moscow, 2010, no. 3, pp. 98–104 (in Russ.).
9. Barskikh M.E., Aryashev S.I., Rogatkin B.Yu. Modern methods of functional verification RTL-models blocks for VLSI microprocessor. *Problemy razrabotki perspektivnykh mikro- i nanoelektronnykh sistem: sb. tr.* [Proc. "Development Issues of Advanced Micro and Nanoelectronic Systems"]. A.L. Stempkovsky (Ed.). Moscow, 2014, pp. 119–122 (in Russ.).
10. Zakharov A.V., Khisambiev I.Sh., Kotovich N.V., Kravchenko A.A., Osipov A.S., Koltsov P.P., Koganov M.A., Gribkov I.V., Kutsaev A.S. The development of the system for microprocessor random testing INTEG. *Programmnye produkty i sistemy* [Software & Systems]. Tver, 2010, no. 2, pp. 14–23 (in Russ.).
11. Kamkin A.S., Kotsynyak A.M., Smolov S.A., Sortov A.A., Tatarnikov A.D., Chupilko M.M. Tools for Functional Verification of Microprocessors. *Tr. ISP RAN* [Proc. of ISP RAS]. Moscow, 2014, vol. 26, iss. 1, pp. 149–200 (in Russ.).
12. Chupilko M.M. *Dinamicheskaya verifikatsiya tsifrovoy apparatury na osnove formalnykh spetsifikatsiy* [Dynamic Verification of Digital Equipment Based on Formal Specifications]. PhD Author's Thesis. Moscow, 2012, 24 p.
13. Rizzatti L. Hardware Emulation: Three Decades of Evolution - Part III. *Verification Horizons*. Mentor Graphics Verification Academy. 2015, vol. 11, iss. 3. Available at: <https://verificationacademy.com/verification-horizons/november-2015-volume-11-issue-3/hardware-emulation-three-decades-of-evolution-part-iii> (accessed April 7, 2017).
14. *Altair Semiconductor Adopts Cadence Palladium XP Platform for Advanced IoT SoC Development*. San Jose, Calif., 2015. Available at: [https://www.cadence.com/content/cadence-www/global/en\\_US/home/company/newsroom/press-releases/pr/2015/altair-semiconductor-adopts-cadence-palladium-xp-platform-for-advanced-iot-soc-development.html](https://www.cadence.com/content/cadence-www/global/en_US/home/company/newsroom/press-releases/pr/2015/altair-semiconductor-adopts-cadence-palladium-xp-platform-for-advanced-iot-soc-development.html) (accessed April 7, 2017).
15. Mackey H. *Sneak Peek: Inside NVIDIA'S Emulation Lab*. 2011. Available at: <https://blogs.nvidia.com/blog/2011/05/16/sneak-peek-inside-nvidia-emulation-lab> (accessed April 7, 2017).
16. Ross A., Star A. *Enabling greater reliability, scalability and flexibility of GPU emulation at AMD using a hybrid virtual-machine based approach*. 2015. Available at: <http://www.techdesignforums.com/practice/technique/reliable-scalable-flexible-gpu-emulation-using-hybrid-virtual-machine-approach/> (accessed April 7, 2017).
17. Mentor Graphics Veloce Emulation Platform Helps Barefoot Networks Verify the World's First Fully Programmable Switch. *Electronic Engineering*. 2016. Available at: [http://www.eejournal.com/archives/news/20160719\\_06/](http://www.eejournal.com/archives/news/20160719_06/) (accessed April 7, 2017).
18. Osipenko P. Synopsys and Baikal Electronics SoC Team Applies Synopsys HAPS and AMBA Transactors to Accelerate the Availability of Prototypes and Improve Product Quality. 2015. Available at: [https://www.synopsys.com/content/dam/synopsys/verification/prototyping/success-stories/baikal\\_ss.pdf](https://www.synopsys.com/content/dam/synopsys/verification/prototyping/success-stories/baikal_ss.pdf) (accessed April 7, 2017).
19. Wang P.H., Collins J.D., Weaver C.T., Kuttanna B., Salamian S., China G.N., Schuchman E., Schilling O., Doil T., Steibl S., Wang H. *Intel Atom Processor Core Made FPGA-Synthesizable*. Available at: <http://www.cse.wustl.edu/~roger/565M.fl2/p209-wang.pdf> (accessed April 7, 2017).
20. Chibisov P.A. *Vstrechnoe testirovanie vysokoproizvoditelnykh mikroprotssessorov* [Counter Testing of High-Performance Microprocessors]. PhD Author's Thesis. Moscow, 2013, 23 p.
21. Zubkovskaya N.V. *Metod testirovaniya proizvoditelnosti i korrektnosti mikroprotssessorov pri pomoshchi natselennykh testovykh program* [Method for Testing the Performance and Correctness of Microprocessors Using Targeted Test Programs]. PhD Author's Thesis. Moscow, 2013, 23 p.

УДК 004.051

DOI: 10.15827/0236-235X.119.409-419

Дата подачи статьи: 17.04.17

2017. Т. 30. № 3. С. 409–419

## СРАВНЕНИЕ ПРОИЗВОДИТЕЛЬНОСТИ ОТЕЧЕСТВЕННЫХ И ИМПОРТНЫХ МИКРОПРОЦЕССОРОВ

Н.Д. Байков, младший научный сотрудник, [nknikita@niisi.ras.ru](mailto:nknikita@niisi.ras.ru);

А.Н. Годунов, к.ф.-м.н., зав. отделом, [nkag@niisi.ras.ru](mailto:nkag@niisi.ras.ru)

(Федеральный научный центр Научно-исследовательский институт системных исследований  
РАН (ФНЦ НИИСИ РАН), Нахимовский просп., 36, корп. 1, г. Москва, 117218, Россия)

В статье сравнивается производительность трех различных процессоров архитектуры MIPS – RM7000, XLP316 и 1890BM8Я и описываются их архитектурные особенности.

Для оценки производительности процессоров при помощи языков программирования C и Assembler строится и применяется методика тестирования, включающая в себя три последовательных этапа.

На первом этапе тестирования измеряется время выполнения отдельных процессорных инструкций при условии, что эти инструкции и необходимые для их работы данные лежат в кэш-памяти первого уровня. Инструкции делятся на несколько групп, для представителей каждой из которых в статье приводятся результаты. На втором этапе измеряется эффективность работы со структурой кэш-памяти первого и второго уровней. В данной статье приводятся только результаты измерений времени обращения в кэш-память второго уровня и в основную память. На третьем этапе сравниваются полученные измерения с теоретическими оценками, построенными на основе результатов первых двух этапов тестирования. Для этого используются синтетические тесты производительности.

Достоинством методики является независимость результатов измерений от используемого компилятора и операционной системы. Все измерения производятся в тактах процессора средствами специальных регистров сопроцессора.

**Ключевые слова:** тестирование, MIPS, производительность, микропроцессор.

В статье сравнивается производительность процессоров архитектуры MIPS [1] – RM7000 [2], XLP316 (далее – XLP) [3] и экспериментальной версии процессора 1890BM8Я (далее – BM8) [4]. Основной целью тестирования является выявление особенностей архитектуры, оказывающих влияние на производительность процессоров, а также классов задач, на которых процессоры показывают наилучшую производительность. Результаты тестирования содержат информацию о времени выполнения как отдельных процессорных инструкций, так и некоторых модельных задач. Предлагаемая методика тестирования имеет следующие особенности:

- все измерения проводились в тактах процессора с целью получить характеристику производительности, которая не зависит от тактовой частоты;
- было изучено влияние кэш-памяти первого и второго уровней на производительность;
- все тесты задействовали только одно ядро процессора;
- при разработке использовались языки программирования Assembler и C; для их трансляции в машинный код – один и тот же компилятор на всех трех процессорах.

Многие тесты производительности (Whetstone [5], Dhrystone [6] и др.) реализованы при помощи языков программирования высокого уровня. Как следствие, на результаты измерений влияют и оптимизирующие возможности компилятора. В данной статье производительность процессоров и влияние компилятора на нее изучаются отдельно друг от друга.

Каждый из тестируемых процессоров имеет уникальную и достаточно сложную архитектуру, и,

как следствие, состояния этих процессоров в каждый момент времени определяются большим количеством различных параметров, осуществить полный учет которых достаточно трудно. В данной статье рассматривается упрощенная модель, согласно которой состояние процессора определяется содержимым кэш-памяти процессора и последовательностью исполняемых им инструкций.

### Технические характеристики

Основные параметры изучаемых процессоров приведены в таблице 1. Для всех трех процессоров функциональность ALU зависит от конвейера, на котором этот блок находится. Например, операции целочисленного умножения и деления выполняются только на F-конвейере для процессоров RM7000 и BM8 и только на ALU2 для процессора XLP. На процессоре RM7000 инструкции вещественной арифметики не вынесены на отдельный конвейер и обрабатываются F- и M-конвейерами.

### Методика тестирования

Тестирование процессоров проводилось в три этапа. На первом этапе измерялось время выполнения процессорных инструкций при условии, что сами инструкции и все необходимые для работы данные уже находятся в соответствующих кэшах процессоров. Тестирование охватывало четыре группы инструкций: целочисленная арифметика, вещественная арифметика, передачи управления, чтение/запись в память. В статье приводятся результаты измерений для нескольких представителей каждой из указанных групп.

Таблица 1

## Технические характеристики

Table 1

## Technical specifications

Характеристика	Процессор		
	RM7000	XLP	BM8
Тактовая частота процессора	400,0 МГц	1,0 ГГц	700,0 МГц
Частота памяти	66,6 МГц	800,0 МГц	400,0 МГц
Количество ядер	1	4	2
L1 – кэш инструкций (iL1)	16 Кб (4-канальный)	64 Кб (2-канальный)	32 Кб (8-канальный)
L1 – кэш данных (dL1)	16 Кб (4-канальный)	32 Кб (2-канальный)	16 Кб (4-канальный)
L2 – кэш	256 Кб (4-канальный)	512 Кб (8-канальный)	512 Кб (4-канальный)
L3 – кэш	Отсутствует	4 Мб (16-канальный)	Отсутствует
Размер строки кэша в L1	32 байта	32 байта	32 байта
Размер строки кэша в L2	32 байта	64 байта	32 байта
Конвейеры выполнения инструкций	2 конвейера: ALU1/BR/FPU, ALU2/LSU/COP0/FPU	5 конвейеров: ALU0/LSU0, ALU1/LSU1, ALU2, ALU3/BR/COP0, FPU	3 конвейера: ALU1/BR, ALU2/LSU/COP0, FPU

Примечание. В таблице приняты следующие обозначения: FPU – обработка инструкций вещественной и комплексной арифметики, ALU – целочисленной арифметики, BR – передачи управления, LSU – чтения/записи в память, COP0 – работы с управляющим сопроцессором. Для первого и второго конвейеров процессоров RM7000 и BM8 также используются обозначения F и M соответственно.

На втором этапе тестирования изучалось влияние на производительность процессоров кэшей первого и второго уровней.

На третьем этапе сравнивалась производительность процессоров на следующих модельных задачах: функция копирования данных, метод Рунге–Кутты, метод Гаусса.

Для автоматизации процесса тестирования с помощью языков программирования Assembler и C была построена тестовая система. В ее функции входили подготовка начального состояния процессора для каждого теста, а также непосредственное измерение характеристик производительности процессоров.

Перед началом тестирования содержимое всех кэшей очищалось, в том числе и содержимое кэша третьего уровня на процессоре XLP. Инструкции тестовой системы размещались в кэше инструкций первого уровня и оставались там до окончания тестирования всех функций. Для инструкций каждой тестируемой функции и данных, с которыми она работает, независимо моделировались следующие ситуации: присутствие инструкций/данных в соответствующем кэше первого уровня, присутствие только в кэше второго уровня, отсутствие в кэшах всех уровней. После окончания тестирования каждой функции ее инструкции и данные удалялись из кэшей. Таким образом, в кэшах процессора одновременно могли находиться только инструкции тестовой системы и одной из тестируемых функций, а также данные, необходимые для ее работы. Кэши изучаемых процессоров организованы так, что в них можно одновременно разместить указанные инструкции и данные вне зависимости от адресов, которые они имеют в памяти, не вызывая при этом вытеснения строк (количества кэш-каналов достаточно, чтобы при возникновении конфликтов

между адресами функций или данных можно было разместить их в разных секциях соответствующего кэша). Загрузка инструкций тестовой системы и тестируемых функций в кэши происходила при помощи вызовов этих функций, а загрузка данных – при помощи операций чтения. Для исключения данных из кэшей использовалась инструкция cache. Проведенные операции позволили минимизировать влияние тестовой системы на результаты измерений и предотвратить вытеснение строк из кэшей в связи с их переполнением.

В качестве средств измерения использовались регистры процессора Performance Counter Control и Performance Counter, которые позволяют регистрировать различные события во время работы программы, такие как такты процессора, промахи в кэш-память процессора и т.д. При каждом выполнении тестируемой функции тестовая система опрашивала счетчик до и после ее выполнения и сохраняла разницу этих значений. Анализ полученных результатов производился в зависимости от этапа тестирования. Для проведения первых двух этапов конструировались семейства однотипных тестовых функций. Функции одного семейства последовательно повторяли заданную инструкцию или набор инструкций заданное количество раз и отличались друг от друга только количеством повторений. Кроме указанных инструкций, функции также могли содержать какие-либо дополнительные инструкции, которые требуются для корректной работы теста, например, первоначальной инициализации используемых регистров. В частности, для каждого такого семейства конструировалась пустая тестовая функция, которая состояла только из дополнительных инструкций и характеризовала величину накладных расходов. Оценка интересующих авторов факторов проводилась на основе се-

рии измерений для функций, принадлежащих одному семейству. На третьем этапе тестирования такие семейства уже не требовались и измерения проводились непосредственно для программных реализаций модельных задач.

Важной технической деталью является то, что процессор XLP поддерживает внеочередное исполнение инструкций. Каждый из его конвейеров оснащен очередью, вмещающей в себя до 16 инструкций. Если в данный момент времени нет свободного конвейера, который может выполнить текущую инструкцию, она помещается в очередь к одному из них, а процессор переходит к обработке следующей инструкции. Если при этом окажется, что есть свободный конвейер для выполнения этой инструкции и она не зависит по данным от уже находящихся в очередях инструкций, процессор немедленно приступит к ее обработке, тогда как предыдущая инструкция еще будет находиться в очереди. Эта архитектурная особенность XLP позволяет избежать простоя всех конвейеров из-за высокой нагрузки на один из них. Так как указанная особенность распространяется и на конвейер, ответственный за работу с управляющим сопроцессором, повторный опрос счетчика после возврата из тестируемой функции может произойти раньше, чем будет завершена обработка всех инструкций теста, которые на данный момент находятся в очереди другого конвейера, то есть результат измерений будет некорректным. Эту проблему удастся решить. Заметим, что только один из конвейеров процессора XLP способен обрабатывать инструкции работы с управляющим сопроцессором, то есть эти инструкции сохраняют порядок относительно друг друга. Добавляя загрузку результата работы тестируемой функции на один из временных регистров управляющего сопроцессора перед повторным опросом счетчика, формируем зависимость по данным между инструкциями и гарантируем, что опрос произойдет после вычисления этого результата. Наконец, для корректности итоговых результатов прерывания во время работы каждого теста были запрещены.

### Результаты измерений

Рассмотрим процессорные инструкции. Прежде чем перейти к результатам измерений, необходимо уточнить, что будем понимать под длительностью выполнения инструкций. Тестируемые процессоры используют конвейерный способ обработки инструкций. Полное время прохождения инструкцией всех стадий конвейера не дает точного представления о производительности процессора, поскольку не учитывает параллелизм в выполнении инструкций. Более точно оценить вклад каждой отдельной инструкции в общее время выполнения программы позволяют следующие величины [7]:

- время освобождения блока выполнения конвейера (Repeat Rate), то есть количество тактов от

момента начала стадии выполнения инструкции до момента, когда конвейер сможет начать эту стадию для следующей инструкции;

- задержка результата выполнения инструкции (Latency), то есть количество тактов от момента начала стадии выполнения инструкции до момента, когда для следующих инструкций будет доступен результат выполнения.

Также существует понятие пропускной способности конвейера (Pipeline Throughput). На наборе из  $n$  инструкций это отношение количества инструкций  $n$  к полному времени их прохождения по конвейеру.

Стадия выполнения для инструкции может быть начата на подходящем свободном конвейере только в том случае, если доступны результаты всех инструкций, от которых она зависит. В противном случае возникает простой конвейера до получения необходимых данных. Для многих инструкций задержка результата оказывается больше времени освобождения блока выполнения. Поскольку при выполнении реальных задач нет никаких гарантий того, что результат работы таких инструкций не понадобится уже в следующей инструкции, для оценки производительности процессоров на первом этапе тестирования больше внимания уделяется измерению задержки результата. Для этого достаточно сделать инструкции тестируемых функций зависимыми.

Описание результатов начнем с инструкций целочисленной арифметики. Приведем результаты для трех инструкций из этой группы: сложение с постоянным значением `addi`, умножение `mul` и целочисленное деление при помощи пары инструкций `div` и `mflo`. Инструкция `mflo` считывает результат целочисленного деления `div` из специального LO-регистра, создавая зависимость по данным между соседними инструкциями `div`. Для других операций, например, выполняемых при помощи инструкций `andi`, `add`, `sub` и т.д., тестирование проводится аналогично.

Структура всех тестов одинакова и состоит из этапа инициализации, последовательности повторяющихся операций и возврата управления. Приведем пример исходного кода:

```
li v0, C1; // Инициализация v0
addi v0, v0, C2;
... // Повторяем вышестоящую инструкцию нужное
... // количество раз вручную, не используя
... // передачи управления
jr ra;
nop;
```

В таблице 2 указана продолжительность тестов в тактах процессора в зависимости от количества повторений выбранной операции целочисленной арифметики (где 0 соответствует пустой функции). Все указанные результаты воспроизводятся при повторном тестировании. Отметим, что время выполнения операции деления при помощи пары инструкций `div` и `mflo` не на всех процессорах является фиксированным. Здесь приведены результаты

Таблица 2

## Инструкции целочисленной арифметики (такты)

Table 2

## Integer arithmetic instructions (clock cycles)

Инструкция	Процессор	Количество повторений																
		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
addi	RM7000	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
	XLP	29	33	33	33	33	34	35	36	37	38	39	40	41	42	43	44	45
	BM8	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
mul	RM7000	6	8	11	14	17	20	23	26	29	32	35	38	41	44	47	50	53
	XLP	34	33	35	38	41	44	47	50	53	56	59	62	65	68	70	73	76
	BM8	13	14	16	18	20	22	24	26	28	30	32	34	36	38	40	42	44
div + mflo	RM7000	6	44	84	124	164	204	244	284	324	364	404	444	484	524	564	604	644
	XLP	29	44	59	73	87	100	113	125	136	148	161	173	185	197	209	221	233
	BM8	13	24	34	44	53	62	70	78	101	124	147	170	193	216	239	262	285

частного случая начальных значений регистров v0 и t0.

На процессоре RM7000 зависимость количества тактов от количества операций линейна для инструкции addi, а для инструкции mul и пары инструкций div, mflo почти линейна, за исключением переходов от пустых функций. Действительно, с увеличением на единицу количества повторений addi длительность теста возрастает ровно на 1 такт во всех случаях. Для инструкции mul и пары инструкций div, mflo при переходе от пустых функций длительность теста возрастает на 2 и 38 тактов процессора соответственно, а при дальнейшем увеличении количества операций – на 3 и 40 тактов за каждое повторение соответствующей операции. Отличие первого приращения от остальных указывает на то, что освобождение блока выполнения от mul и mflo происходит быстрее, чем будет доступен их результат (для возврата управления их результат не требуется). Также было обнаружено, что на процессоре RM7000 задержка получения результата инструкции mul зависит от размера аргументов. Для 16-битных аргументов она равнялась 3 тактам, а для 32-битных – 4 тактам. На других процессорах подобной зависимости не наблюдалось.

Результаты процессоров BM8 и XLP аналогичны результатам процессора RM7000. Укажем лишь на некоторые особенности. На процессоре BM8 полученные значения задержки результата для инструкций addi и mul составили 1 и 2 такта соответственно, а задержка результата выполнения пары инструкций div и mflo варьировалась от 8 до 23 тактов.

На процессоре XLP полученные результаты проявляют линейную зависимость лишь после нескольких повторений операции. Для инструкций addi и mul наблюдаемые значения задержки составили 1 и 3 такта. Задержка на паре div и mflo, как и в случае процессора BM8, не была фиксированной и требовала от 11 до 15 тактов. Большая величина накладных расходов (то есть время выполнения пустой функции) на процессоре XLP по сравнению с остальными процессорами частично объясняется

влиянием на результат дополнительных операций с управляющим сопроцессором для предотвращения преждевременного опроса счетчика событий.

Таким образом, на операциях целочисленного сложения и деления лучшую производительность в тактах процессора показал процессор XLP, а на умножении – BM8.

Аналогичные тесты были проведены для инструкций сопроцессора вещественной арифметики. Тесты отличались только способом инициализации исходных регистров. Рассмотрим результаты для четырех видов инструкций, работающих с числами формата double: сложение add.d, умножение mul.d, деление div.d и смешанная операция умножения и сложения madd.d.

На процессоре XLP результаты измерений не воспроизводились. Уже в случае пустой функции результат измерений мог меняться от запуска к запуску. Например, для 5 повторений инструкции add.d в 50 запусках теста результат измерений изменялся от 57 до 116 тактов. На процессорах RM7000 и BM8 колебаний при измерениях не наблюдалось. Результаты этой группы тестов приведены в таблице 3 (для процессора XLP выбраны результаты одного из запусков).

Оценивать увеличение длительности теста при увеличении количества повторений инструкций на единицу на процессоре XLP будем в среднем. Получаем приближенные значения 5,9, 6,1, 36,2 и 10,2 такта для инструкций add.d, mul.d, div.d и madd.d, соответственно. На процессорах RM7000 и BM8 искомые величины могут быть вычислены точно. Не считая пустых функций, длительности тестов на указанных процессорах линейно зависят от количества инструкций. На процессоре RM7000 длительность теста увеличивается на 4, 5, 36 и 5 тактов за каждую инструкцию add.d, mul.d, div.d и madd.d соответственно, а на процессоре BM8 – на 3, 3, 19 и 5 тактов. Для тестируемых инструкций вещественной арифметики освобождение блока выполнения конвейера происходит не позже, чем будет доступен результат работы инструкций, поэтому полученные значения совпадают с временем задержки

Таблица 3

## Инструкции вещественной арифметики (такты)

Table 3

## Floating point arithmetic instructions (clock cycles)

Инструкция	Процессор	Количество повторений																
		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
add.d	RM7000	9	14	18	22	26	30	34	38	42	46	50	54	58	62	66	70	74
	XLP	53	85	47	83	70	100	85	91	104	127	98	122	114	131	133	136	148
	BM8	17	21	24	27	30	33	36	39	42	45	48	51	54	57	60	63	66
mul.d	RM7000	9	15	20	25	30	35	40	45	50	55	60	65	70	75	80	85	90
	XLP	71	45	92	56	70	64	110	80	84	103	93	118	133	135	126	135	168
	BM8	17	21	24	27	30	33	36	39	42	45	48	51	54	57	60	63	66
div.d	RM7000	9	46	82	118	154	190	226	262	298	334	370	406	442	478	514	550	586
	XLP	83	75	157	156	205	226	281	304	340	397	437	455	516	530	571	614	663
	BM8	17	37	56	75	94	113	132	151	170	189	208	227	246	265	284	303	322
madd.d	RM7000	10	16	21	26	31	36	41	46	51	56	61	66	71	76	81	86	91
	XLP	52	93	72	68	91	96	110	127	139	147	166	169	194	191	197	205	215
	BM8	17	24	29	34	39	44	49	54	59	64	69	74	79	84	89	94	99

результата этих инструкций. Таким образом, наименьшие значения задержки результата для группы инструкций вещественной арифметики наблюдались на процессоре BM8.

В отличие от тестов целочисленной арифметики переход от пустых функций длится дольше всех последующих переходов. Причиной оказываются инструкции dmtcl, для которых задержка результата выполнения больше времени освобождения блока выполнения конвейера. В ходе отдельных тестов было получено, что для освобождения блока выполнения конвейера от инструкции dmtcl на процессорах RM7000 и BM8 требуется 1 такт, а задержка результата равна 2 и 3 тактам соответственно.

Перейдем к тестированию инструкций передачи управления. Эффективность выполнения этих инструкций существенно зависит от используемых алгоритмов предсказания переходов и может меняться в зависимости от ситуации. Оценка эффективности проводилась в четырех тестах: простой цикл, вложенные передачи управления, перекрест-

ные передачи управления и безусловные передачи управления.

Результаты тестов представлены в таблице 4. Лучшие результаты в этих тестах показал процессор RM7000, которому требуется ровно 2 такта на выполнение одной передачи управления, тогда как процессорам XLP и BM8 требуется до 5 тактов процессора. Отдельно можно отметить только результаты процессора BM8 в первом тесте, которому после нескольких начальных итераций требуется только 1 такт на выполнение передачи управления.

Наконец, перейдем к изучению инструкций чтения/записи. Тест инструкций чтения требовал дополнительной подготовки. Буфер, из которого считывались данные, заполнялся двойными словами, каждое из которых содержало адрес следующего. Это позволило сформировать зависимость по данным между инструкциями, которая вынуждает процессор XLP выполнять инструкции последовательно.

Результаты обоих тестов воспроизводятся точно на процессорах RM7000 и BM8, а на процес-

Таблица 4

## Инструкции передачи управления (такты)

Table 4

## Branch instructions (clock cycles)

Тест	Процессор	Количество повторений																
		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Тест 1	RM7000	8	9	11	13	15	17	19	21	23	25	27	29	31	33	35	37	39
	XLP	14	27	29	31	34	35	37	39	41	43	45	47	49	51	53	55	57
	BM8	19	19	28	31	32	33	34	35	36	37	38	39	40	41	42	43	44
Тест 2	RM7000	9	13	17	21	25	29	33	37	41	45	49	53	57	61	65	69	73
	XLP	17	32	42	50	60	66	74	82	90	98	106	114	122	130	138	146	154
	BM8	25	35	45	55	65	75	85	95	105	115	125	135	145	155	165	175	185
Тест 3	RM7000	8	9	11	13	15	17	19	21	23	25	27	29	31	33	35	37	39
	XLP	15	18	45	49	33	62	37	39	67	43	45	47	49	51	53	55	57
	BM8	19	25	28	34	37	43	46	52	55	61	64	70	73	79	82	88	91
Тест 4	RM7000	5	7	9	11	13	15	17	19	21	23	25	27	29	31	33	35	37
	XLP	14	16	22	24	30	32	38	40	46	48	54	58	62	64	70	74	78
	BM8	12	15	20	23	28	31	36	39	44	47	52	55	60	63	68	71	76

соре XLP наблюдаются редкие и незначительные по величине (в пределах 1-2 тактов) колебания. Результаты измерений приведены в таблице 5.

На процессоре XLP была получена наибольшая величина накладных расходов. На инструкциях записи происходит скачок длительности теста при переходе от пустой функции к функции с одной инструкцией записи и далее в среднем длительность теста возрастает приблизительно на 1 такт процессора за каждую дополнительную инструкцию записи. На инструкциях чтения прирост длительности теста составил 4 такта за каждую инструкцию.

Перейдем к результатам процессоров RM7000 и BM8. На инструкциях чтения процессоры показали одинаковый результат: длительность теста возрастает на 2 такта процессора за каждую инструкцию чтения. При тестировании инструкций записи было обнаружено, что результаты RM7000 и BM8 уже не совпадают. На каждую инструкцию записи процессору RM7000 требуется ровно один такт, тогда как на процессоре BM8 зависимость количества тактов от количества инструкций становится нелинейной и в среднем требует более одного такта процессора.

Часто при выполнении инструкции чтения соседние инструкции не зависят от ее результата. Производительность в этом случае определяется временем освобождения блока выполнения конвейера. Измерения показали, что на это требуется 1 такт на всех трех процессорах. Также отметим, что на процессоре XLP имеются сразу два конвейера, способных обрабатывать инструкции чтения/записи, что также позволяет ускорить работу процессора.

### Время обращения к кэш-памяти

Хотя влияние кэш-памяти на производительность складывается из многих факторов, ограничимся лишь измерением времени обращения к ней. Влияние других факторов (например, политики кэширования) изучаться не будет. Также отметим, что измеряемая величина не является только характеристикой производительности процессора, так как зависит также от используемой системы памяти.

В процессе тестирования существенных различий в длительности получения данных или инструкций из памяти обнаружено не было. Поэтому рассмотрим только результаты тестов для случая работы с инструкциями. Для измерений использовалось семейство тестовых функций из первого этапа тестирования для инструкции `addi`. При их работе чтение данных из памяти не происходит. Были рассмотрены два случая.

1. Инструкций нет в кэше инструкций первого уровня iL1, но они есть в кэше второго уровня L2.

2. Инструкций нет в кэшах iL1 и L2. В случае процессора XLP инструкции также отсутствуют в кэше L3.

Поскольку инструкции и данные попадают в кэши строками, количество обращений из кэша iL1 в кэш L2 определяется количеством строк кэша iL1, требуемых для хранения функции. Тестовые функции задавались так, чтобы начало каждой из них соответствовало началу строки в кэшах первого и второго уровней. В таком случае количество требуемых для хранения функции строк кэша однозначно определяется по количеству инструкций в ней. Размер строки кэша первого уровня у всех трех процессоров одинаков – 32 байта (см. табл. 1). Поскольку на хранение одной инструкции выделяются 4 байта, в одной строке кэша помещаются ровно 8 инструкций, следовательно, для получения количества строк кэша необходимо округлить вверх результат деления на 8 количества инструкций, из которого состоит тестируемая функция.

Результаты первого случая приведены в таблице 6. На процессорах RM7000 и BM8 при переходе от 5 повторений инструкции `addi` к 6 повторениям происходят скачки в длительности тестов. Далее подобные скачки повторяются с периодом в 8 инструкций `addi`. Учитывая то, что, помимо инструкций `addi`, в теле тестовых функций присутствуют ровно 3 дополнительные инструкции (`li`, `jr` и `por`), получаем, что скачки происходят одновременно с увеличением количества строк кэша iL1, требуемых для хранения тестовой функции. На процессоре RM7000 зависимость монотонна, а величина скачка равна 9 тактам процессора. На процессоре BM8 монотонность нарушается. Тем не менее, скачок при увеличении количества строк кэша

Инструкции чтения/записи (такты)

Load/store instructions (clock cycles)

Инструкция	Процессор	Количество повторений																
		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
sd	RM7000	7	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
	XLP	54	61	61	64	63	66	66	69	68	71	71	73	73	76	76	77	78
	BM8	18	19	20	22	23	25	26	28	29	31	32	34	35	37	38	40	41
ld	RM7000	7	9	11	13	15	17	19	21	23	25	27	29	31	33	35	37	39
	XLP	49	53	57	60	64	68	72	76	80	84	88	92	96	100	104	108	112
	BM8	18	19	21	23	25	27	29	31	33	35	37	39	41	43	45	47	49

Таблица 5

Table 5



на процессоре VM8 оценим величиной в 4-5 тактов. На процессоре XLP наблюдаются пары скачков при переходе от 13 к 14 повторениям и от 14 к 15 повторениям и далее, как показало тестирование, с периодом в 16 инструкций. Размер скачков оценивается величиной в 5-6 тактов процессора.

Перейдем ко второму случаю. Его результаты приведены в таблице 7. Для процессора VM8 результаты отсутствуют, так как оценить длительность обращения в память в рамках используемой модели на данном процессоре не удалось. Анализ результатов проводится аналогично первому случаю. Отметим только, что на процессоре XLP размер строки кэша L2 равен 64 байтам. Как видно из таблицы 7, скачки в длительностях тестов на процессорах RM7000 и XLP происходят одновременно с увеличением количества строк кэша L2, требуемых для хранения функции. Размер этих скачков приблизительно равен 110–120 тактам для обоих процессоров.

В процессе тестирования для процессора XLP также были выявлены две важные особенности, которые позволяют существенно ускорить его работу при промахах в кэш-память процессора. Было обнаружено, что процессор XLP способен обрабатывать параллельно промахи в кэш-память для различных инструкций из очереди выполнения при условии, что известны адреса данных в памяти. Если требуется, например, последовательно прочитать данные из буфера, указанная особенность позволяет добиться существенного прироста к производительности. Также было обнаружено, что при выполнении инструкций запись обращений в память не происходит. Вместо этого процессор XLP помещает записываемые данные в специальную очередь на запись, из которой они впоследствии попадают в основную память. При этом запись данных в очередь осуществляется почти так же быстро, как если бы данные находились в кэше данных первого уровня.

## Модельные задачи

Представим результаты измерений для трех модельных задач: функция копирования данных, метод Рунге–Кутты и метод Гаусса [8].

Начнем с задачи копирования данных из одного буфера в другой (назовем их Src и Dst) при помощи функции memcpy, поддерживаемой стандартами POSIX и C99. Адреса буферов в памяти были выровнены на границу строк кэшей первого и второго уровней, а кэши данных очищены перед началом тестирования. Существует несколько реализаций функции копирования (memcpy); в данном случае использовалась реализация, которая перемещает 4 двойных слова из буфера Src в буфер Dst за одну итерацию основного цикла. Инструкции этой функции были предварительно размещены в кэше инструкций первого уровня. Рассматривались не перекрывающиеся в памяти буферы размером от 1 Кб до 8 Мб в следующих случаях:

- данные обоих буферов отсутствуют в кэшах;
- буфер Src предварительно прочитан (то есть, если размер кэша больше размера буфера, весь буфер находится в этом кэше после прочтения);
- буфер Dst предварительно прочитан;
- оба буфера были прочитаны перед началом теста.

Результаты измерений представлены в таблице 8 (вместо общей длительности теста указывается время копирования 1 Кб). На RM7000 полученные данные хорошо согласуются с результатами первых этапов тестирования. Действительно, при отсутствии буферов в кэшах время копирования, главным образом, определяется количеством обращений в основную память. Количество таких обращений можно вычислить явно. Поскольку размер строки кэша второго уровня на процессоре RM7000 равен 32 байтам, а адрес используемых буферов выровнен на границу строки кэша, для хранения 1 Кб в кэш-памяти второго уровня на процес-

Таблица 6

Длительность обращения в кэш L2 (такты)

Table 6

The L2 cache access duration (clock cycles)

Процессор	Количество повторений																
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
RM7000	13	14	14	15	16	17	26	25	26	27	27	28	29	30	39	38	39
XLP	44	45	46	47	48	49	50	51	52	53	54	55	56	57	62	68	68
VM8	17	18	19	20	21	22	32	27	28	29	30	31	32	33	43	38	39

Таблица 7

Длительность обращения в память (такты)

Table 7

The memory access duration (clock cycles)

Процессор	Количество повторений																
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
RM7000	115	116	116	117	118	119	229	228	229	230	230	231	286	233	343	342	343
XLP	161	151	163	164	165	166	167	168	158	159	171	161	174	163	281	275	275

Таблица 8

## Копирование памяти (такты/Кб)

Table 8

## Memory copying (clock cycles/Kb)

Бу- фер	Процес- сор	Размер буфера											
		4Кб	8Кб	16Кб	32Кб	64Кб	128Кб	256Кб	512Кб	1Мб	2Мб	4Мб	8Мб
Src(-) Dst(-)	RM7000	7 773	7 776	7 776	7 776	7 776	7 780	8 654	9 092	9 312	9 422	9 477	9 504
	XLP	809	791	797	806	804	803	804	808	811	812	850	875
	BM8	1 666	1 702	1 689	1 676	1 676	1 668	1 671	1 911	2 032	2 091	2 240	2 263
Src(+) Dst(-)	RM7000	4 099	4 101	4 229	4 352	4 417	4 448	7 007	8 270	8 901	9 216	9 374	9 453
	XLP	233	224	222	223	276	302	359	431	478	503	663	806
	BM8	1 195	1 189	1 093	1 040	1 009	1 003	990	1 630	1 861	2 120	2 200	2 189
Src(-) Dst(+)	RM7000	4 093	4 098	4 280	4 378	4 430	4 454	7 011	8 271	8 901	9 216	9 374	9 453
	XLP	838	800	807	800	796	793	799	804	807	808	844	875
	BM8	983	961	988	1 003	1 005	1 011	1 010	1 579	1 956	2 122	2 202	2 243
Src(+) Dst(+)	RM7000	289	300	643	849	953	1 011	5 269	7 400	8 466	8 999	9 265	9 398
	XLP	230	225	224	276	298	311	332	449	484	504	708	798
	BM8	451	451	561	616	644	658	666	1 457	1 781	2 078	2 181	2 231

соре RM7000 требуются 32 строки. Поскольку работа идет сразу с двумя буферами, количество строк необходимо удвоить. Длительность чтения 1 строки из основной памяти на процессоре RM7000 оценивалась величиной приблизительно в 110 тактов процессора, отсюда получаем, что только для 64 обращений к основной памяти без учета времени выполнения инструкций потребовалось бы приблизительно 7 040 тактов. Полученная теоретическая оценка близка к данным таблицы 8 для буферов малых размеров. При увеличении размера копируемых буферов до 256 Кб начинается рост результатов измерений, связанный с переполнением кэша второго уровня и необходимостью вытеснения строк из кэшей.

Изучим также результаты процессора RM7000 для случая, когда оба буфера находятся в кэше данных первого уровня. В таком случае инструкции записи и независимые инструкции чтения, как показало тестирование на первом этапе, должны выполняться за 1 такт процессора. Для передачи управления требуются 2 такта. Учитывая, что передачи управления и инструкции целочисленной арифметики могут быть выданы на выполнение одновременно с инструкциями чтения/записи, получаем, что копирование 4 двойных слов можно осуществить за 9 тактов. Следовательно, на копирование 1 Кб требуется 288 тактов. Соответствующий результат измерений из таблицы 8 равен 289 тактам, то есть практически совпадает с предложенной оценкой.

Скачки в измерениях для процессоров XLP и BM8, как и в случае процессора RM7000, объясняются переполнением какого-либо из кэшей. Из данных таблицы 8 следует, что процессор BM8 в большинстве случаев показал большую производительность, чем процессор RM7000, но лучшие результаты в этом тесте были получены при тестировании процессора XLP, что особенно заметно при копировании буферов большого размера. Сравним результаты процессора XLP и процессора RM7000

в случае копирования 4 Кб при полном отсутствии данных в кэш-памяти. Хотя на втором этапе тестирования было получено, что количество тактов для обращения в основную память на рассматриваемых процессорах приблизительно совпадает, процессор XLP обладает рядом преимуществ, в числе которых вдвое больший размер строки кэша второго уровня, отсутствие обращений к основной памяти на инструкциях записи и параллельная обработка промахов в кэш-память. Поскольку процессор XLP обладает двумя LSU-блоками, в очередях на выполнение может находиться до 32 инструкций чтения/записи. В рассматриваемом случае этого достаточно для одновременного копирования 16 двойных слов, то есть двух строк кэша второго уровня на процессоре XLP. Таким образом, скорость копирования данных на процессоре XLP должна быть как минимум в 8 раз выше скорости копирования на процессоре RM7000, что подтверждают результаты измерений.

В случае работы с некешируемой памятью скорость копирования данных вне зависимости от общего размера буферов составила приблизительно 19 000, 36 250 и 5 800 тактов/Кб на процессорах RM7000, XLP и BM8 соответственно.

В качестве другой модельной задачи был рассмотрен метод Рунге–Кутты 4-го порядка точности. Количество обращений к памяти в этой задаче мало. Промежуточные результаты вычислений не сохранялись, что также снизило количество обращений к памяти. Метод Рунге–Кутты применялся к осциллятору Ван-дер-Поля, который задается следующей системой уравнений:

$$\begin{cases} \frac{dx}{dt} = y, \\ \frac{dy}{dt} = \mu(1 - x^2)y - x, \end{cases}$$

где  $x$  – координата точки;  $y = dx/dt$  – скорость точки;  $\mu$  – некий коэффициент. Уравнение рассматривалось на фиксированном по времени отрезке при

разном количестве шагов по времени. Для программной реализации поставленной задачи использовался язык программирования C. В данном случае изучалась работа функций, построенных с использованием ключей компилятора -O0 и -O3, то есть при отсутствии оптимизации и с высоким уровнем оптимизации. На всех трех процессорах использовался один и тот же компилятор и, как следствие, один и тот же объектный код. Все инструкции и данные, необходимые для работы программы, предварительно размещались в соответствующих кэшах процессоров.

Из таблицы 9 видно, что при отсутствии оптимизации полученные результаты на всех трех процессорах достаточно близки. Лучшую производительность в тактах процессора в этом случае показал процессор XLP. Использование оптимизации приводит к сокращению времени выполнения на всех тестируемых процессорах. Лучшую производительность в тактах процессора в этом случае показал уже процессор BM8.

Сравним объектный код программ, полученных при разных уровнях оптимизации. Так как каждый шаг метода Рунге–Кутты требует одних и тех же операций, в обоих случаях программа имеет структуру цикла. В зависимости от уровня оптимизации в цикле используются различные наборы инструкций. Укажем количество повторов каждой из них.

С ключом -O0

dmtcl – 156, ld – 94, dmfc1 – 68, mul.d – 34,  
add.d – 24, sdc1 – 24, lui – 18, addiu – 17,  
nop – 17, move – 16, sd – 10, jr – 8,  
jal – 8, sub.d – 8, ldc1 – 4, lw – 3,  
div.d – 2, sw – 2, bnez – 1, slti – 1.

С ключом -O3:

mul.d – 12, add.d – 10, madd.d – 8,  
msub.d – 4, sub.d – 4, bnez – 1, addiu – 1.

За счет оптимизации количество используемых инструкций сократилось на порядок (с 515 до 40 инструкций). Компилятору удалось избавиться от инструкций перемещения данных между регистрами, обращений к памяти и большей части передач управления. Также сократилось количество инструкций вещественной арифметики.

Из таблицы 9 получаем, что при отсутствии оптимизации на выполнение одной итерации цикла требуется приблизительно 854 такта на процессоре RM7000, 706 тактов на XLP и 792 такта на BM8.

Так как все необходимые инструкции и данные уже находятся в соответствующих кэшах первого уровня, длительность теста определяется эффективностью выполнения отдельных процессорных инструкций, а также выигрышем от одновременного выполнения независимых инструкций на нескольких конвейерах. Последний в данном случае можно оценить. Сформулируем несколько утверждений относительно структуры программы:

- каждая инструкция вещественной арифметики зависит от пары инструкций dmtcl, которые загружают ее аргументы на регистры сопроцессора вещественной арифметики;
- результат каждой инструкции вещественной арифметики выгружается на регистр общего назначения при помощи инструкции dmfc1, то есть выполнение dmfc1 не начнется, пока не станет доступным результат вычислений;
- в слотах задержки всех инструкций передачи управления расположены инструкции por;
- для большей части остальных инструкций время освобождения блока выполнения конвейера происходит через 1 такт процессора, а дополнительных простоев из-за задержки результата не возникает.

Предположим, что все инструкции программы выполняются последовательно, если только это не инструкции передачи управления, параллельно с которыми могут выполняться только их слоты задержки. Тогда, учитывая результаты первого этапа тестирования и сделанные утверждения о структуре программы, можно формально оценить длительность такого выполнения одной итерации цикла. На процессоре RM7000 получаем:

Передачи управления:	17×2
Инструкции mul.d:	+34×5
Инструкции add.d:	+24×4
Инструкции sub.d:	+8×4
Инструкции div.d:	+2×36
Простой на dmtcl:	+(34+24+8+2)×(2-1)
Прочие инструкции:	+(515-17×2-34-24-8-2)×1
	= 885 тактов.

Разница между полученной оценкой и результатом измерений составила всего 31 такт процессора, следовательно, использование нескольких конвейеров на процессоре RM7000 практически не дает прироста к производительности из-за большого количества зависимостей по данным между соседними инструкциями. Аналогичные результаты по-

Метод Рунге–Кутты (такты)

Таблица 9

The Runge–Kutta method (clock cycles)

Table 9

Количество шагов	Ключ -O0			Ключ -O3		
	RM7000	XLP	BM8	RM7000	XLP	BM8
256	218 780	181 281	202 995	24 455	33 524	19 091
512	437 404	362 686	405 747	48 779	67 068	38 063
1,024	874 652	722 888	811 227	97 415	134 128	75 923

Таблица 10

## Метод Гаусса (такты)

Table 10

## The Gauss method (clock cycles)

Размер матрицы	Ключ -O0			Ключ -O3		
	RM7000	XLP	BM8	RM7000	XLP	BM8
32	1 120 314	856 320	1 218 764	330 119	161 709	233 262
64	8 502 498	6 221 187	9 330 940	2 361 472	1 043 660	1 652 070
128	66 270 288	48 281 943	72 851 443	17 711 872	7 999 142	12 369 516
256	610 119 486	377 854 617	575 599 138	267 932 792	60 719 264	73 889 891

Таблица 11

## Результаты тестирования (такты)

Table 11

## Test results (clock cycles)

Инструкция	RM7000	XLP	BM8
addi	1	1	1
mul	3–4	3	2
div + mflo	40	11–15	8–23
add.d	4	5,9	3
mul.d	5	6,1	3
div.d	36	36,2	19
madd.d	5	10,2	5
dmtcl	2	–	3
Передачи управления	2	2–4	1–5
sd	1	1	1–2
ld	2	4	2
Обращение в L2	9	5–6	4–5
Обращение в память	110–120	110–120	–
Копирование 1 Кб (данные находятся в кэше dL1)	289	229	451
Копирование 1 Кб (данные отсутствуют в кэш-памяти)	7 773–9 504	809–891	1 666–2 263
Итерация метода Рунге–Кутты (-O3)	95	130	74
Метод Гаусса (n=128, -O3)	17 711 872	7 999 142	12 369 516

лучаются на процессоре BM8. Для процессора XLP разница между такой оценкой и реальным результатом оказывается уже значительной, что, вероятно, является следствием особенностей его архитектуры, которые позволяют изменять порядок выполнения инструкций и за счет этого добиваться лучших результатов, несмотря на большее по сравнению с остальными процессорами время выполнения отдельных типов инструкций.

При высоком уровне оптимизации, как следует из таблицы 9, для выполнения одной итерации цикла на процессорах RM7000, XLP и BM8 требуется 95, 130 и 74 такта соответственно. Отметим, что процессор XLP в этом случае оказался медленнее RM7000 и BM8, а это хорошо согласуется с результатами первого этапа тестирования для инструкций вещественной арифметики.

Перейдем к результатам третьей задачи. В ней изучалась производительность процессоров при решении системы линейных уравнений методом Гаусса с выбором главного элемента по столбцу. Он уже содержит не только большое количество вычислений, но и большое количество обращений к памяти. Результаты измерений для этого теста приведены в таблице 10. Аналогично методу Рунге–Кутты рассматриваются результаты для двух уровней оптимизации.

Наибольшую производительность в этом тесте показал процессор XLP в случаях как отсутствия оптимизации, так и ее высокого уровня. Результаты процессора BM8 при высоком уровне оптимизации оказались близкими к результатам процессора XLP.

### Заключение

Подведем итоги проведенного тестирования. Разработанная тестовая система показала свою эффективность на практике. За исключением нескольких экспериментов, была измерена длительность выполнения многих низкоуровневых операций. Благодаря этой информации обнаружено несколько технических особенностей архитектуры, влияющих на производительность при выполнении модельных задач. Основные результаты измерений (в тактах процессора) отражены в таблице 11.

Проведенные измерения также показали, что среди процессоров невозможно однозначно выбрать лидера по производительности. При определенных условиях каждый из трех процессоров может показать лучшую производительность, чем два других. Например, вычислительные задачи с небольшой долей обращений к памяти быстрее всего выполняются на процессоре BM8, даже если при-

нять во внимание тактовые частоты процессоров. Для копирования больших объемов данных, напротив, лучше всего подходит процессор XLP, на котором большая длительность загрузки строк в кэш-память компенсируется увеличением их размера и их одновременной загрузкой.

### Литература

1. MIPS Architecture For Programmers, vol. III: The MIPS64 and microMIPS64 Privileged Resource Architecture, Revision 5.04, 2014.
2. PMC-Sierra. URL: <https://www.pmc.com> (дата обращения: 23.03.2017).

3. NetLogic Microsystems. URL: <https://www.netlogicmicro.com> (дата обращения: 23.03.2017).
4. Бобков С.Г. Импортзамещение элементной базы вычислительных систем // Вестн. РАН. 2014. Т. 84. № 11. С. 1010–1016.
5. Curnow H.J. and Wichman B.A. a synthetic benchmark. *Computer Jour.*, 1976, vol. 19, iss. 1, pp. 43–49.
6. Reinhold P. Weicker. Dhrystone: a synthetic systems programming benchmark. *Communications of the ACM*, 1984, vol. 27, iss. 10, pp. 1013–1030, DOI: 10.1145/358274.358283.
7. Zargham M.R. *Computer architecture: single and parallel systems*. Upper Saddle River, NJ, Prentice Hall, 1996, 472 p.
8. Бахвалов Н.С., Жидков Н.П., Кобельков Г.М. Численные методы. М.: Бином. Лаборатория знаний, 2003. 632 с.

Software & Systems  
DOI: 10.15827/0236-235X.119.409-419

Received 17.04.17  
2017, vol. 30, no. 3, pp. 409–419

## COMPARISON OF RUSSIAN AND FOREIGN MICROPROCESSOR PERFORMANCE

**N.D. Baykov**<sup>1</sup>, Junior Researcher, [nknikita@niisi.ras.ru](mailto:nknikita@niisi.ras.ru)

**A.N. Godunov**<sup>1</sup>, Ph.D. (Physics and Mathematics), Head of Department, [nkag@niisi.ras.ru](mailto:nkag@niisi.ras.ru)

<sup>1</sup> Federal State Institution "Scientific Research Institute for System Analysis of the Russian Academy of Sciences" (SRISA RAS), Nakhimovsky Ave. 36/1, Moscow, 117218, Russian Federation

**Abstract.** This article provides performance comparison for three different MIPS processors (RM7000, XLP316 and 1890VM8YA) and describes their architecture details.

To compare processor performance, the testing technique is developed and implemented using C and Assembler. The technique consists of three consecutive stages.

At the first stage the authors measure processor instructions execution time providing that both instructions and data required are already stored in a primary cache. The instructions are divided into several groups. There are the results for members of each group. Primary and secondary cache efficiency is benchmarked on the second stage. The article provides the results for secondary cache and RAM access time only. The third stage uses synthetic performance tests. The obtained results are compared with theoretical estimations based on the results of the first two stages.

The advantage of the proposed technique is in its independence from compiler and operating system specified. All measurements are carried out in clock cycles using special purpose coprocessor registers.

**Keywords:** testing, MIPS, performance, microprocessor.

### References

1. MIPS Architecture For Programmers. Vol. III: The MIPS64 and microMIPS64. Privileged Resource Architecture. Revision 5.04, 2014.
2. PMC-Sierra. Available at: <https://www.pmc.com> (accessed March 23, 2017).
3. NetLogic Microsystems. Available at: <https://www.netlogicmicro.com> (accessed March 23, 2017).
4. Bobkov S.G. Import substitution of the element base of computer systems. *Vestnik Rossiyskoy akademii nauk* [Herald of the Russian Academy of Sciences]. 2014, vol. 84, iss. 11, pp. 1010–1016 (in Russ.).
5. Curnow H.J., Wichman B.A. A Synthetic Benchmark. *Computer Jour.* 1976, vol. 19, iss. 1, pp. 43–49.
6. Weicker R.P. Dhrystone: A Synthetic Systems Programming Benchmark. *Communications of the ACM*. 1984, vol. 27, iss. 10, pp. 1013–1030.
7. Zargham M.R. *Computer Architecture*. Prentice Hall Publ., 1996.
8. Bakhvalov N.S., Zhidkov N.P., Kobelkov G.M. *Chislennyye metody* [Numerical Methods]. Moscow, Binom. Laboratoriya znaniy Publ., 2003, 632 p.

УДК 512.6, 517.9, 519.6

DOI: 10.15827/0236-235X.119.420-424

Дата подачи статьи: 29.05.17

2017. Т. 30. № 3. С. 420–424

## **ГИБРИДНЫЕ КОГНИТИВНЫЕ НЕЧЕТКИЕ СИСТЕМЫ УПРАВЛЕНИЯ АВТОНОМНЫМ РОБОТОМ НА ОСНОВЕ НЕЙРОИНТЕРФЕЙСА И ТЕХНОЛОГИИ МЯГКИХ ВЫЧИСЛЕНИЙ**

*С.В. Ульянов, д.ф.-м.н., профессор, ulyanovsv@mail.ru;*

*А.Г. Решетников, к.т.н., доцент, agreshetnikov@gmail.com;*

*А.А. Мамаева, аспирант, allabard@yandex.ru*

*(Университет «Дубна», Институт системного анализа и управления,  
Университетская ул., 19, г. Дубна, 141980, Россия)*

В статье обсуждаются возможности применения нейроинтерфейса совместно с различными типами регуляторов на типовом примере управления автономным транспортным средством. Проведена оценка возможностей применения методов и средств технологии интеллектуальных вычислений для повышения надежности функционирования системы управления.

Цель данной работы – на основе эксперимента показать возможности эффективного применения когнитивного интерфейса («мозг–компьютер–исполнительное устройство») на примере управления транспортным средством (мобильным роботом), раскрыть возможность применения современных технологий управления и показать роль и необходимость интеллектуальных вычислений в работе интерфейса «мозг–компьютер» для повышения надежности и робастности системы управления.

В частности, рассмотрена возможность управления типовым движением объекта (вперед, назад, влево, вправо, обход препятствий) посредством когнитивного шлема с помощью стандартного блока распознавания команд и различных типов систем управления, в том числе на основе оптимизатора баз знаний на мягких вычислениях.

**Ключевые слова:** когнитивная система управления, когнитивный регулятор, нейроинтерфейс, мягкие вычисления, интеллектуальная система управления.

До настоящего времени в теории и процессах проектирования *интеллектуальных систем управления* (ИСУ) как систем, основанных на знаниях (в виде соответствующих *баз знаний* (БЗ)), проектирование самих БЗ осуществлялось экспертом на основе или интеллектуальных вычислений типа мягких вычислений с применением генетических алгоритмов, или нечетких нейронных сетей. Роль человека-оператора в контуре управления не учитывалась в явном виде либо описывалась передаточными функциями упрощенного типа. Довольно давно установлено [1], что в контуре управления больших многоконтурных связанных систем управления часто бывает до 75 % избыточной информации, которая не используется или мешает принятию решения. Поэтому одной из центральных проблем разработки ИСУ являлся поиск конструктивного решения задачи проектирования БЗ в заданной проблемно-ориентированной области применения при наличии неопределенности и риска.

Включение в контур управления человека-оператора часто ассоциировалось с появлением и повышением риска в принятии решения из-за человеческого фактора, несмотря на возможность учета интеллектуального ресурса самого оператора. В свою очередь, одно из направлений технологического роста интерфейсов вычислительных систем, неразрывно связанных с ростом вычислительной мощности, направлено именно на интеграцию психофизиологических особенностей поведения пользователя (оператора) с виртуальным и информационным полем эффективной деятельности человека.

Такое развитие порождает новые классы аппаратной и программной реализации интерфейсов и сталкивается с задачами обработки огромного количества индифферентных данных пользователя. Очевидной становится необходимость наличия интеллектуальных свойств адаптации и обучения ПО такого рода интерфейсов.

Большое внимание уделяется, например, вопросу применения когнитивного интерфейса совместно с различными типами структур управления. Работы зарубежных исследователей показали возможности использования стимулирующих сигналов для обучения и при генерации управляющих команд оператором [2, 3]. Однако извлечение знаний и разработка БЗ непосредственно из мыслительного процесса (электроэнцефалограммы, ЭЭГ) не рассматривались. Напротив, результаты работы [4] показали возможность восстановления образов мыслительного процесса непосредственно из сигналов ЭЭГ. Отметим, что и в подобных системах не рассматривается возможность извлечения знаний из сигнала ЭЭГ оператора для формирования адаптивной обучающейся системы.

Обычно для тренировки когнитивных способностей оператора широко применяются специализированные тренажеры – игры, например нейроинтерфейс ЕРОС [5]. Эти тренажеры не обладают специализированным программным модулем для обучения и адаптации системы управления (самой программы) к физиологическим особенностям оператора [6, 7]. В свою очередь, зарекомендовавшие себя в широком классе областей мягкие вычисления (генетические алгоритмы, нечеткая логика и

нечеткие нейронные сети) позволяют спроектировать интеллектуальную когнитивную систему управления, обладающую такими качествами.

Соответственно, технологический рост данного направления неразрывно связан с наличием и развитием специализированных программных инструментов проектирования ИСУ. В качестве такого инструментария в данной работе выступает *оптимизатор баз знаний* (ОБЗ) на мягких вычислениях, являющийся (с точки зрения теории нечетких систем) универсальным аппроксиматором для формирования БЗ *нечеткого регулятора* (НР). Структурно оптимизатор включает в себя три генетических алгоритма, на каждом этапе работы которого формируется оптимальная структура нечеткой нейронной сети с оптимальным количеством продукционных правил и значений весовых коэффициентов узлов. В свою очередь, это гарантирует робастность системы управления в условиях помех и смешанных сигналов [8].

Обобщенно технологию проектирования можно представить в виде схемы, изображенной на рисунке 1.

Для снятия сигнала о мозговой активности [9, 10] использовался когнитивный шлем Emotiv EPOC+ совместно с поставляемым ПО (EPOC Control Panel).

Структурно Emotiv EPOC+ состоит из 14 каналов для снятия сигнала. Для метода отбора проб используется последовательная выборка. Частота ее – 128 выборок в секунду (внутренний 2048 Гц) с разрешением оцифровки 14 бит и пропускной способностью 0,2–45 Гц.

Поставляемое ПО позволяет получать, распознавать и регистрировать ЭЭГ-сигнал со шлема, ментальные команды и назначать на них определенные действия (например, посылать управляющий сигнал на Bluetooth-устройство для движения автономного робота *вперед, назад, влево и вправо*).

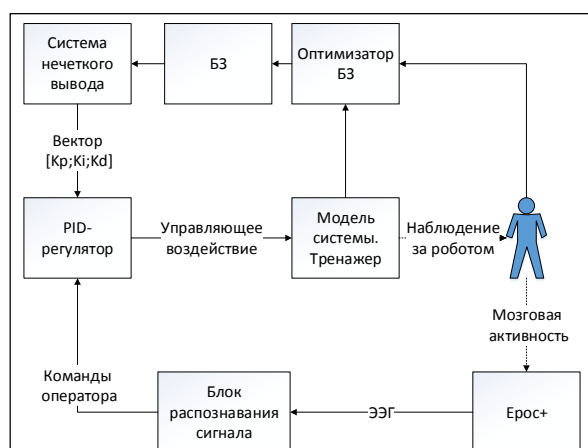


Рис. 1. Структура технологии проектирования когнитивной системы управления

Fig. 1. The structure of a cognitive control system design technology

Обучение ментальной команде заключается в записи состояния покоя и состояний, когда субъект представляет образ какого-либо действия, концентрируя мозговую активность в определенном участке мозга. Запись таких состояний занимает 8 секунд.

Качество распознавания команд разработчики такого оборудования обычно связывают с уровнем подготовки оператора в генерировании различных ментальных команд. В комплекте ПО поставляются специализированные игры – тренажеры для обучения и тренировок.

### Проектирование когнитивного регулятора для управления автономным роботом

Обычно в качестве системы управления на объекте устанавливают регулятор, который в зависимости от ментальных команд оператора вырабатывает управляющее воздействие для исполнительных механизмов. В качестве такого регулятора может выступать, например, простой релейный регулятор, где для конечного множества выходных команд (вперед, назад, влево, вправо) вырабатываются одни и те же управляющие воздействия.

В данной работе испытывались пропорциональный регулятор, *пропорционально-интегральный* (ПИ) регулятор с блоком нечеткого вывода, *пропорционально-интегральный дифференцирующий* (ПИД) регулятор с различными коэффициентами усиления.

Для проведения эксперимента был выбран объект управления – мобильный робот в виде трехколесной машинки с Bluetooth-управлением (рис. 2).

В устройстве управления в качестве процессора для управления используется Arduino Uno совместно с драйвером двигателя – Pololu Dual MC33926. Два Мотора – DC 9V Motor. Bluetooth-

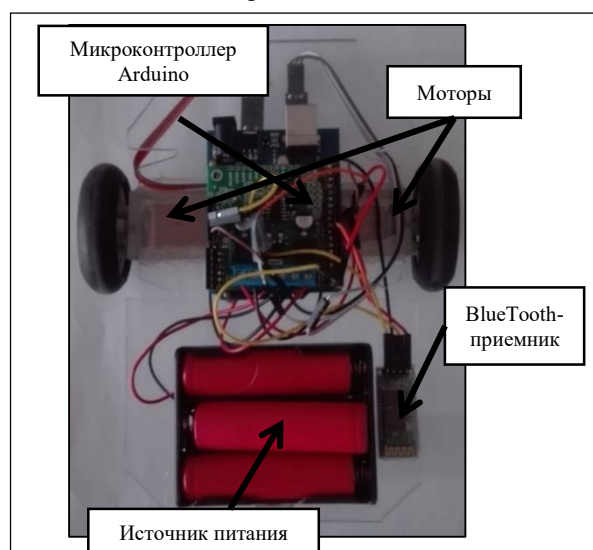


Рис. 2. Устройство управления

Fig. 2. A control device

модуль – НС-05. Источником питания служат три 3V Li-On-батареи.

Руководствуясь принципом неразрушения нижнего исполнительного уровня управления (рис. 1), применение интеллектуальной надстройки в виде ОБЗ основано на извлечении сигналов обучения из сигналов траектории управления оператором. Так, в процессе управления роботом на основе заданного типа регулятора регистрируются траектории движения (рис. 3). Из выходного сигнала блока распознавания путем сложения формируется интегральная составляющая, далее эксперт выбирает соответствующее управляющее воздействие на основе предыдущего опыта использования системы.

Для аппроксимации обучающего сигнала с целью извлечения знаний применялся разработанный ОБЗ с выбранной моделью нечеткого вывода (Сугено 1-го порядка). В частности, использовались сигнал с блока распознавания сигнала ЕРОС, а также интегральная величина сигнала и идентификационный номер требуемой команды.

На следующем этапе проектирования БЗ для нечеткого вывода в автоматическом режиме формируется полная БЗ и осуществляется дальнейшая оптимизация правых частей правил (рис. 4 и 5).

На этапе создания БЗ полученный обучающий сигнал используется для формирования лингвистических переменных для входных значений. В данной работе количество нечетких множеств в каждой лингвистической переменной было ограничено до трех.

Другими словами, на вход в нейронную сеть поступают команды от программного модуля распознавания сигнала ЕРОС (вперед, назад, влево и вправо), на выход поступают значения для команды движения устройства, установленные экспертом (рис. 4).

Применение БЗ совместно с ПИ-регулятором обусловлено необходимостью формирования более мягкой структуры для управления. С этой целью создавались *лингвистические переменные* (ЛП) для каждой из записанных команд в системе распознавания, формировалась полная БЗ с использованием нечетких множеств ЛП (рис. 6). Правая часть регулятора содержала соответствующие значения для управляющего воздействия с применением ПИ-регулятора. Таким образом, уровень активации правила в базе правил соответствовал уровню активации управляющего воздействия.

На рисунке 7 приведен результат когнитивного управления движением мобильного робота в лабиринте на основе ПИ-регулятора.

В работе рассматривались следующие регуляторы для управления роботом: П-, ПИ-, ПИД-регуляторы с постоянными и переменными коэффициентами.

На рисунке 8 представлены выходные команды систем управления для управления роботом. На графике зеленым цветом показан целевой сигнал,

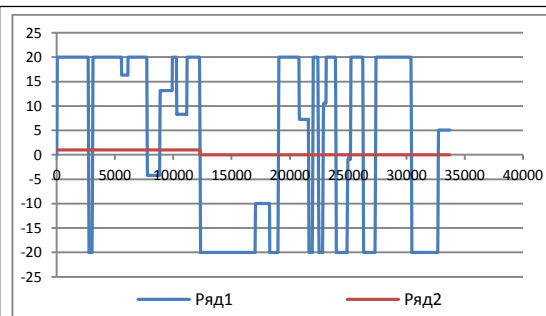


Рис. 3. Обучающий сигнал

Fig. 3. A training signal

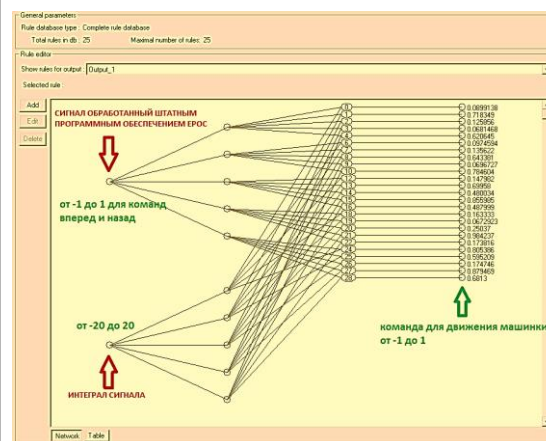


Рис. 4. Нейронная сеть нечеткого вывода

Fig. 4. A fuzzy inference neural network



Рис. 5. Структурная схема формирования правила в БЗ

Fig. 5. A block diagram for knowledge base rule formation

который соответствует движению вперед и назад, другими цветами выделены уровни активации П- (на рисунке Ерос) и ПИ-регулятора с БЗ.

Как видно из графика на рисунке 8, в момент, когда задача оператора – движение машинки назад, и происходит концентрация мыслительного процесса на этой команде, стандартный блок не всегда правильно определяет управляющее воздействие и робот движется рывками или даже в другую сто-



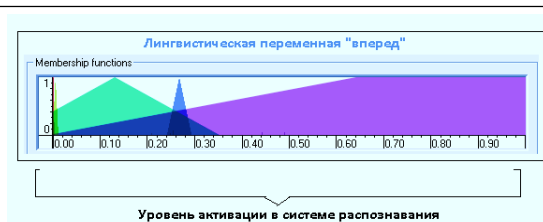


Рис. 6. Пример лингвистической переменной для команды «вперед»

Fig. 6. An example of the linguistic variable for the command "forward"

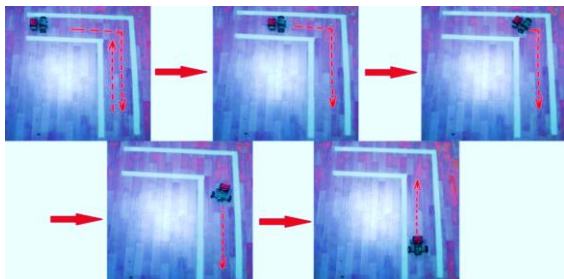


Рис. 7. Траектория движения при управлении мобильным роботом на основе ПИ-регулятора

Fig. 7. A motion trajectory when controlling a mobile robot based on a PI-regulator

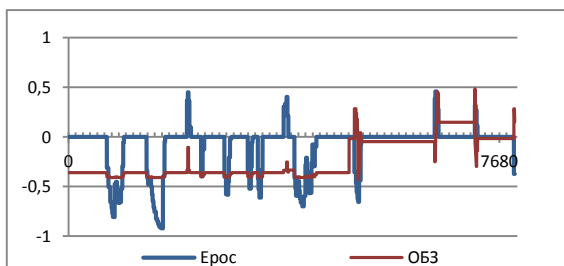


Рис. 8. Управляющие воздействия, вырабатываемые релейным и ПИ-регулятором при движении вперед и назад

Fig. 8. Controlling actions produced by a relay regulator and a PI-regulator when moving forward and backward

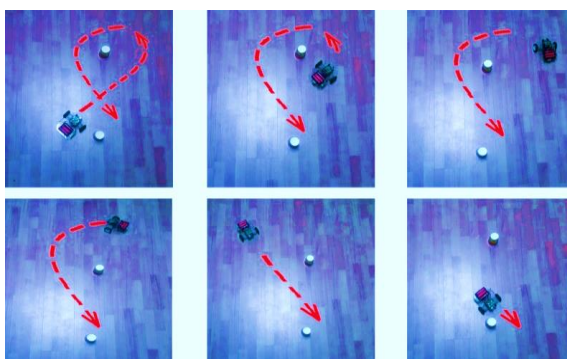


Рис. 9. Объезд препятствия системой управления с ПИД-регулятором

Fig. 9. Obstacle avoidance by a PID-regulator control system

рону (ложное срабатывание). Однако дополнительная надстройка в виде интегральной составляющей в БЗ позволяет сгладить последовательность команд и уменьшить количество ошибок при достижении цели.

При работе системы распознавания когнитивных сигналов часто возникают неверные (ложные) срабатывания, вызванные артефактами мышечной и мыслительной активности, или команды недостаточно хорошо распознаются в связи с уровнем подготовки оператора. На пропорциональный регулятор это оказывает серьезное воздействие, и работа системы вызывает трудности при эксплуатации.

Дополнительно была рассмотрена задача объезда препятствия с использованием системы управления с ПИД-регулятором и интегрированной БЗ (рис. 9).

Как можно заметить, повышение интеллектуальности ПО влияет на выработку эффективного управляющего воздействия регулятором и на работу системы в целом. Однако установление оптимальных значений коэффициентов для каждого момента времени является актуальной и очень важной задачей. При неверной (ложной) установке значений также наблюдается неверное срабатывание, объект управления движется рывками.

### Заключение

Результаты показали, что качество управления объектом значительно повышается при применении разработанной схемы управления. Проведенная работа демонстрирует необходимость создания унифицированной технологии проектирования систем управления для нейроинтерфейсов и недостаточность робастности функционирования поставляемого в настоящее время ПО.

Стоит отметить, что дальнейшее развитие когнитивных технологий управления неразрывно связано с методами регистрации и интерпретации ЭЭГ мозга оператора с применением специализированных фильтров, сглаживающих сигналы ЭЭГ и удаляющих помехи и шумы (артефактов), вызванные психофизиологическим состоянием и внешними факторами, и с применением интеллектуальных технологий в ПО исполнительных механизмов для обучения и адаптации систем управления с оператором в контуре управления.

### Литература

1. Петров Б.Н., Уланов Г.М., Ульянов С.В., Хазен Э.М. Информационно-семантические проблемы в процессах управления и организации. М.: Наука, 1977. 452 с.
2. Po-Lei Lee, Hsiang-Chih Chang, Tsung-Yu Hsieh, Hua-Ting Deng, and Chia-Wei Sun. A brain-wave-actuated small robot car using ensemble empirical mode decomposition-based approach. IEEE Transactions on Systems, Man, and Cybernetics. Part a: Systems and Humans, 2012, vol. 42, no. 5, pp. 1053–1064.
3. Becedas J. Brain-machine interfaces: basis and advances. IEEE Transactions on Systems, Man, and Cybernetics. Part a: Systems and Humans, 2012, vol. 42, no. 6, pp. 825–836.
4. Miyawaki Y., Uchida H., Yamashita O., Sato M., Mori-

to Y., Tanabe H., Sadato N., and Y. Kamitani. Visual image reconstruction from human brain activity using a combination of multiscale local image decoders. *Neuron*, 2008, vol. 5, no. 60, pp. 915–929.

5. Jaime Gomez-Gil, San-Jose-Gonzalez, Luis Fernando Nicolas-Alonso, Sergio Alonso-Garcia. Steering a tractor by means of an emg-based human-machine interface. *Sensors*, 2011, no. 11, pp. 7110–7126. DOI: 10.3390/s110707110.

6. Kane M.J., Conway A.R.A., Miur T.K. & Colflesh G.J.H. Working memory, attention control, and the N-back task: a question of construct validity. *Jour. of Experimental Psychology: Learning, Memory, and Cognition*, 2007, no. 33, pp. 615–622.

7. Anguera J.A. et al. Video game training enhances cognitive control in older adults. *Nature*, 2013, vol. 501, pp. 97–101.

8. Ульянов С.В., Литвинцева Л.В., Добрынин В.Н., Мишин А.А. Интеллектуальное робастное управление: технологии

мягких вычислений. М.: Изд-во ВНИИгеосистем, 2011. 406 с.

9. Barker A.T., Jalinous R., Freeston I.L. Non-invasive magnetic stimulation of human motor cortex. *Lancet*, 1985, vol. 11, pp. 1106–1107.

10. Бархатова И.А., Соколов И.А., Шмыков Г.Ю., Ульянов С.В. Интеллектуальные самоорганизующиеся когнитивные регуляторы. Ч. 2: Модели когнитивных интерфейсов «мозг–устройство» // Системный анализ в науке и образовании. 2015. № 1. URL: <http://www.sanse.ru/archive/32> (дата обращения: 23.05.2017).

11. Бархатова И.А., Ульянов С.В. Интеллектуальные самоорганизующиеся когнитивные регуляторы. Ч. 3: Информационно-термодинамический закон распределения качества и обобщенная структура интеллектуального когнитивного управления // Системный анализ в науке и образовании. 2015. № 1. URL: <http://www.sanse.ru/archive/32> (дата обращения: 23.05.2017).

Software & Systems

DOI: 10.15827/0236-235X.119.420-424

Received 29.05.17

2017, vol. 30, no. 3, pp. 420–424

### HYBRID COGNITIVE FUZZY CONTROL SYSTEMS FOR AN AUTONOMOUS ROBOT BASED ON NEUROINTERFACE AND SOFT COMPUTING

S.V. Ulyanov<sup>1</sup>, Dr.Sc. (Physics and Mathematics), Professor, [ulyanovsv@mail.ru](mailto:ulyanovsv@mail.ru)

A.G. Reshetnikov<sup>1</sup>, Ph.D. (Engineering), Associate Professor, [agreshetnikov@gmail.com](mailto:agreshetnikov@gmail.com)

A.A. Mamaeva<sup>1</sup>, Postgraduate Student, [allabard@yandex.ru](mailto:allabard@yandex.ru)

<sup>1</sup> Dubna State University, Institute of the System Analysis and Control, Universitetskaya St. 19, Dubna, 141980, Russian Federation

**Abstract.** The article discusses the possibility of applying neurointerface together with different types of regulators via the typical example of controlling an autonomous vehicle. There is an assessment of application possibilities of intelligent computing methods and means to improve the control system performance reliability.

The aim of this work is to show experimentally the possibilities of cognitive interface effective application (“brain-computer-actuating device”) on the example of motor vehicle driving (a mobile robot). The paper also reveals modern management technologies application and shows the role and the necessity of intelligent computing in the operating “brain-computer” interface in order to improve the reliability and robustness of the control system.

In particular, the paper considers the possibility of controlling the movement of the object (forward, backward, left, right, bypass obstacles) with the help of a cognitive helmet using a standard command recognition block and different types of control systems, including knowledge base optimizer based on soft computing.

**Keywords:** cognitive control system, cognitive control, neurointerface, intelligent computing, intelligent control system.

### References

1. Petrov B.N., Ulanov G.M., Ulyanov S.V., Khazen E.M. *Informatsionno-semanticheskie problemy v protsessakh upravleniya i organizatsii* [Information and semantic problems in management and organization processes]. Moscow, Nauka Publ., 1977, 452 p.

2. Lee Po-Lei, Chang Hsiang-Chih, Hsieh Tsung-Yu, Deng Hua-Ting, Sun Chia-Wei. A Brain-Wave-Actuated Small Robot Car Using Ensemble Empirical Mode Decomposition-Based Approach. *IEEE Trans. on Systems, Man, and Cybernetics. Part a: Systems and Humans*. 2012, vol. 42, no. 5, pp. 1053–1064.

3. Bectedas J. Brain-Machine Interfaces: Basis and Advances. *IEEE Trans. on Systems, Man, and Cybernetics. Part a: Systems and Humans*. 2012, vol. 42, no. 6, pp. 825–836.

4. Miyawaki Y., Uchida H., Yamashita O., Sato M., Morito Y., Tanabe H., Sadato N., Kamitani Y. Visual Image Reconstruction from Human Brain Activity using a Combination of Multiscale Local Image Decoders. *Neuron*. 2008, vol. 5, no. 60, pp. 915–929.

5. Jaime Gomez-Gil, San-Jose-Gonzalez, Nicolas-Alonso L.F., Alonso-Garcia S. Steering a Tractor by Means of an EMG-Based Human-Machine Interface. *Sensors*. 2011, no. 11, pp. 7110–7126.

6. Kane M.J. Working Memory, Attention Control, and the N-Back Task: A Question of Construct Validity. *Jour. of Experimental Psychology: Learning, Memory, and Cognition*. 2007, no. 33, pp. 615–622.

7. Anguera J.A., Boccanfuso J., Rintoul J.L., Al-Hashimi O., Faraji F., Janowich J., Kong E., Larraburo Y., Rolle C., Johnston E., Gazzaley A. Video game training enhances cognitive control in older adults. *Nature*. 2013, vol. 501, pp. 97–101.

8. Ulyanov S.V., Litvinseva L.V., Dobrynin V.N., Mishin A.A. *Intellektualnoe robastnoe upravlenie: tekhnologii myagkikh vychisleniy* [Intelligent robust control: soft computing technology]. Moscow, Vniigeosystem Publ., 2011.

9. Barker A.T., Jalinous R., Freeston I.L. Non-invasive magnetic stimulation of human motor cortex. *Lancet*. 1985, vol. 11, pp. 1106–1107.

10. Barkhatova I.A., Sokolov I. A. Shmykov G.U., Ulyanov S.V. Intelligent Self-Organizing Cognitive Controllers. Pt. 2: Cognitive Brain Machine Interface Models. *Sistemny analiz v nauke i obrazovanii* [System Analysis in Science and Education]. 2015, no. 1. Available at: <http://www.sanse.ru/archive/32> (accessed May, 23, 2017).

11. Barchatova I.A., Ulyanov S.V. Intelligent Self-Organized Cognitive Controllers. Pt. 3: Information-Thermodynamics Quality Distribution Law And Generalized Structure Of Intelligent Cognitive Control. *Sistemny analiz v nauke i obrazovanii* [System Analysis in Science and Education]. 2015, no. 1. Available at: <http://www.sanse.ru/archive/32> (accessed May, 23, 2017).

УДК 519.876.5

DOI: 10.15827/0236-235X.119.425-434

Дата подачи статьи: 30.01.17

2017. Т. 30. № 3. С. 425–434

## **ОБЗОР СРЕДСТВ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ КОЛЛЕКТИВОВ РОБОТОВ С ЭЛЕМЕНТАМИ СОЦИАЛЬНОЙ ОРГАНИЗАЦИИ**

*М.А. Ровбо, инженер-исследователь, rovbooma@gmail.com;*

*Е.Е. Овсянникова, инженер-исследователь, eeovsyana@yandex.ru  
(Национальный исследовательский центр «Курчатовский институт»,  
пл. Академика Курчатова, 1, г. Москва, 123182, Россия);*

*А.А. Чумаченко, магистрант, chumachenkoa93@mail.ru  
(Национальный исследовательский университет «Высшая школа экономики»,  
ул. Мясницкая, 20, г. Москва, 101000, Россия)*

В работе рассматриваются механизмы, важные для адекватного моделирования группы искусственных агентов, особенно мобильных роботов, организованных в систему с элементами социальной структуры. Сформулированы основные механизмы для моделирования искусственных коллективов с социальной структурой.

Проведен обзор инструментов моделирования, основанный на этих требованиях, определяющий преимущества и недостатки существующих систем для моделирования, симуляции и исследования подобных систем. Обзор сосредоточен вокруг средств моделирования биологических систем (в частности, муравьев, так как они являются ярким примером социальной группы), агентного моделирования и специализированных инструментов симуляции роботов.

Сделан акцент на уровень поддержки различных социальных механизмов в системах моделирования, а также на их возможности моделирования, особенно большого количества агентов (в частности, поддержка вычислений на кластерах), на умение симулировать пользовательские модели и реализацию различных аспектов социальной группы. Выяснилось, что, хотя есть много общих средств для создания и симуляции моделей, существует очень мало поддержки и готовых реализаций даже широко используемых механизмов, таких как феромоны, предоставленных системами для создания и исследования группы робототехнических агентов с социальной структурой. С другой стороны, средства, направленные на моделирование систем с элементами социального поведения, таких как системы моделирования поведения муравьев, не очень масштабируемы или ограничены в наборе аспектов, которые они могут моделировать, и нет удобных механизмов их расширения для углубленных исследований.

На основе обзора была выявлена потребность в комплексной системе моделирования, специально направленной на исследования в области искусственных групп агентов с элементами социальной структуры, и сформулированы требования к такой системе.

**Ключевые слова:** агентное моделирование, моделирование социального поведения, групповая робототехника, имитационное моделирование, инструменты разработки, обзор, кластерные вычисления, биоподобные системы, зусоциальность, робот.

Основной задачей групповой робототехники является построение такой системы, которая давала бы новые качества и улучшенные характеристики за счет взаимодействия многих отдельных агентов, обладающих относительно простыми правилами. Основным методом изучения данных систем является имитационное и агентное моделирование, для чего исследователи пользуются различными средами и библиотеками.

Одним из направлений исследований в групповой робототехнике является изучение биологических систем с целью применения принципов их организации для построения коллектива роботов, поскольку эти системы демонстрируют многие желаемые качества, такие как устойчивость к внешним воздействиям, возможность работы в неизвестной среде, эффективность выполнения задач. Новым и весьма перспективным направлением в этой области является подход, основанный на социальном поведении и различных механизмах, связанных с ним [1]. Искусственные коллективы, реализующие эти механизмы, в дальнейшем будем называть коллективами с социальной структурой. В частности, большое распространение получило

построение коллективов, подобных муравейнику, который является классическим примером зусоциального сообщества, что считается высшей степенью социальной организации. Несмотря на весьма сложное устройство, возможности и поведение отдельного муравья, наличие интересных явлений в структуре муравейника, таких как полиэтизм, мобилизация, трофобиоз, накопление и передача знаний, иерархия и других [2], активно изучаются лишь некоторые отдельные механизмы. Например, нахождение пути по феромонам, которое имеет ряд различных описанных в литературе моделей, программных реализаций и используется как для управления движением физических агентов и их моделей, так и для поиска пути на графах.

Однако, помимо развития идеи организации агентов с помощью феромонов, несомненно важными являются и другие механизмы коллективов с социальной структурой. Для полноценного всестороннего изучения искусственных коллективов, строящихся по подобным принципам, важно наличие такой системы имитационного моделирования, которая предоставляет средства реализации в ней основных элементов коллектива с социальной

структурой, обеспечивает масштабируемость системы, возможность ее удобного расширения, поддержку сбора данных для последующего анализа. Существует немало обзоров систем агентного моделирования (например [3–6]), однако в основном они касаются систем общего, универсального назначения. Цель данного обзора – анализ имеющихся специализированных систем моделирования, пригодных для решения задач коллективной робототехники, ориентированных на создание социальных структур.

### Модели социального поведения

Существует ряд форм, видов и проявлений элементов социального поведения, таких как когезия (стремление держаться группой), контагиозное и агонистическое поведение, кастовость, образование коалиций и т.д. Данные механизмы наблюдались у животных и описаны исследователями, в частности, этология сложного поведения тех же муравьев описана в классических работах [2, 7]. Некоторые механизмы свойственны высокоуровневым формам жизни, например, агрессивные и антагонистические формы поведения шимпанзе рассматривались в [8], а контагиозное поведение – в работе [9]. В работе [1] показано, что для реализации феноменов социального поведения необходим целый ряд ключевых механизмов:

- индивидуальные морфологические и психические различия индивидуумов (например, за счет реализации механизма темперамента и эмоций);
- локальное взаимодействие особей, в том числе коммуникация;
- средства языкового общения;
- формирование коалиций;
- дифференциация функций (распределение ролей и задач);
- выделение иерархической структуры (лидера, групп и т.п.) и другие.

Не все из перечисленных механизмов должны быть ярко выражены в коллективе как социальной структуре, однако можно говорить об их наличии в том или ином виде и, соответственно, система моделирования должна их поддерживать хотя бы на уровне соответствующих базовых модулей и интерфейсов. Для более сложных агентов необходима поддержка создания моделей внутреннего и внешнего мира. Такие модели важны для локализации агента [10], навигации и коммуникации [11], кооперации между агентами [12], планирования действий [13], адаптации к повреждениям [14] и многих других задач, хотя последний пример больше свойственен для более сложно устроенных роботов, чем для тех, на которых зачастую ориентируются исследования в групповой робототехнике.

Система моделирования необязательно должна предоставлять средства реализации всех этих меха-

низмов, поскольку для некоторых из них можно воспользоваться специализированными библиотеками, но хотя бы возможность их интеграции должна быть. Чем более явно наличествует поддержка всех элементов коллектива агентов с социальной структурой, тем больше подходит программное средство для создания и изучения этих моделей.

Конечно, поскольку речь идет о моделировании робототехнических распределенных систем, система моделирования должна поддерживать и такие понятия, как отдельный агент, пространство, время, задавать в некотором виде среду, в которой взаимодействуют агенты, предоставлять возможность модульной реализации отдельных аспектов агента. Стоит отметить, что одним из типов коммуникации между агентами является стигмергия, то есть опосредованная передача информации через изменение внешней среды, например, оставление феромонов [15]. Также система должна уметь поддерживать модель из нескольких взаимодействующих коллективов, поскольку среди задач групповой робототехники есть такие, которые связаны с делением и слиянием групп роботов [16].

Несомненно, для изучения отдельных элементов системы также полезно иметь возможность использовать уже готовые алгоритмы для реализации других аспектов системы без необходимости написания программы для всех механизмов сразу. Например, при изучении различных моделей эмоций роботов использовать модули, реализующие один из способов моделирования локальной коммуникации или группового движения. Использование таких стандартных поведенческих модулей позволит, как минимум, сосредоточиться на содержательной части изучаемой проблемы.

В данной работе была сделана попытка сформулировать требования и найти систему моделирования, наиболее подходящую для построения модели множества взаимодействующих искусственных агентов, организованных в коллектив с социальной структурой, и для изучения его с использованием вычислительного эксперимента.

Хотя существуют различные методы моделирования групповых систем [17] и разные типы имитационного моделирования [18], внимание в данной статье сосредоточено на системах агентного имитационного моделирования, ориентированных на дискретные имитационные модели, поскольку метод моделирования в этом случае поддерживает понятие агентов и предполагает возможность реализации модели на ЭВМ.

Таким образом, можно выделить некий предварительный ряд требований к такой системе моделирования:

- поддержка больших групп (тысячи и десятки тысяч особей);
- возможность моделирования сложной внутренней структуры;

- поддержка локальных взаимодействий;
- моделирование пространственных отношений;
- поддержка моделирования коммуникаций и языкового общения.

### Системы моделирования поведения эусоциальных животных

Поскольку рассматриваются построение и моделирование искусственной системы из мобильных агентов, организованных в коллектив с социальной структурой, сначала был произведен поиск систем моделирования, созданных для изучения биологических эусоциальных коллективов. При возможности модификации правил и изменения ряда параметров агентов и окружающей среды такая система моделирования могла бы быть применена и для изучения свойств искусственного коллектива.

Яркие представители животного мира с эусоциальной организацией – муравьи, пчелы и некоторые другие насекомые, поэтому в обзор попали в основном программы и библиотеки, относящиеся к моделированию поведения муравьев, такие как Aints [19], Myrmedrome [20], AntMe [21] и т.д. Стоит также упомянуть о наличии платформ для соревнований программ, которые могут быть выполнены в форме муравейников [22], хотя это, скорее, платформы для тестирования, нежели для разработки, и в качестве библиотеки моделирования для исследований не подходят.

**Myrmedrome** – это агентный симулятор колоний муравьев, построенный на принципе реакции муравьев исключительно на локальные события [23]. Муравьи взаимодействуют друг с другом с помощью химических сигналов. Графический интерфейс программы представлен на рисунке 1 (белые линии – пути следования из феромонов). Алгоритмы управления агентами недетерминированные: на каждом шаге, когда все параметры муравьев обновляются, существует вероятность, что выбранное действие не будет выполнено. Это делает систему адаптивной к непредвиденным ситуациям.

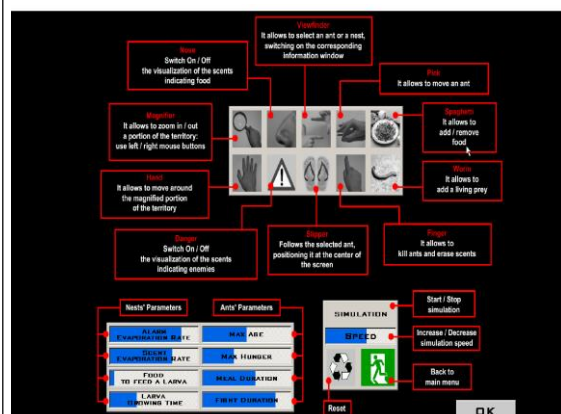
Myrmedrome имитирует жизнь колонии муравьев на ограниченном участке среды. Социальная организация основана на кастовой системе, в которой есть рабочие и солдаты. Первые выполняют основные функции по поиску еды, которую они хранят в совместных желудках и которой делятся с другими муравьями. Рабочие, исследуя территорию, при встрече с добычей пытаются ее убить, а также ищут муравьев из других колоний, рассматривая их как источник опасности и оповещая об их присутствии при помощи феромонов. По этим феромонам солдаты, защищающие гнездо, находят источник опасности и стремятся его уничтожить. Программа предоставляет пользователю возмож-

ность манипулировать средой и муравьями: добавлять еду, двигать муравьев, менять параметры феромона, количество муравьев и некоторые другие. Она является демонстрацией работы муравейника, но не библиотекой для моделирования, и исходный код этой программы для его модификации не предоставляется.

**AntMe** позиционируется как игра [21], обучающая объектно-ориентированному программированию на примере управления муравейником путем написания соответствующего кода на языке C# или же Visual Basic .NET. В ней программируемым муравьям нужно собирать еду (сахар и яблоки) в гнездо и бороться с жуками, организуясь с помощью феромонов. Взаимодействия локальны: муравей «видит» и взаимодействует лишь с ближайшим окружением. Для изменения алгоритма муравья используются обработчики событий и набор методов-команд, таких как «идти вперед», «вернуться в муравейник». Доступ к внутренним параметрам (состоянию) осуществляется через параметры объекта. Муравьев можно разбить на касты,



а)



б)

Рис. 1. Графический интерфейс программы:  
а) окно симуляции, б) настраиваемые параметры и опции программы

Fig. 1. A graphic program interface:  
а) a simulation window, б) configurable parameters and program options



каждая из которых будет обладать своим набором параметров, таких как скорость передвижения, величина переносимого груза и т.п. Возможность расширения функционала программы путем написания плагинов заявлена, однако описание этого в документации на сайте проекта отсутствует.

Несмотря на наличие визуализации, а также реализацию некоторых важных механизмов (локальность взаимодействия, коммуникация через феромоны, распределение ролей), данная программа является скорее площадкой для знакомства с объектно-ориентированным программированием в увлекательной интерактивной форме, нежели средством моделирования для изучения коллективного управления, что и заявлено авторами.

**NetLogo: Ants.** NetLogo – агентно-ориентированный язык программирования и интегрированная среда разработки [24]. Язык реализован на виртуальной машине Java, что делает программы, написанные на нем, кроссплатформенными. Среда NetLogo содержит встроенные библиотеки моделей, в частности, имитирующие биологические системы. Среди них есть библиотека под названием Ants (муравьи) [25]. В этом проекте колония муравьев занимается фуражированием. Каждый отдельный муравей обладает определенным набором простых правил поведения: ищет еду, несет ее в гнездо, когда обнаруживает, оставляя феромоны по мере движения. Если другие муравьи чувствуют этот запах, они следуют по нему к еде. Феромон со временем распространяется в пространстве путем эмуляции диффузии и исчезает. Настраиваемые функции: скорость испарения запаха, скорость распространения запаха, количество муравьев. Среда NetLogo имеет встроенную систему сбора различной информации и ее графического вывода, а также позволяет менять параметры системы во время ее работы. Характерные возможности программы изображены на примере ее работы (рис. 2). Воз-

можно расширение модели путем расположения источников питания в различных местах, внедрения различных стратегий нахождения обратного пути к гнезду и изменения силы химического запаха.

Несмотря на то, что модель имеет возможности расширения, они представлены в рамках механизма поиска пищи. В то же время для построения социального коллектива необходима реализация других различных механизмов, таких как локальные коммуникации между агентами, построение иерархической структуры и т.д.

К языку NetLogo предоставляются открытый исходный код [26], а также позволяющие его модификацию ряд различных библиотек (например Ants) и обширная документация. Однако модификация исходного кода NetLogo является гораздо более сложной задачей, чем изменение или добавление моделей, написанных на нем, и большинству пользователей целесообразнее работать с предоставляемыми этим языком инструментами в рамках связанных с ними ограничений и особенностей.

Таким образом, NetLogo, являясь гибкой и расширяемой средой агентного моделирования, не реализует многие механизмы, необходимые для построения коллектива с социальной структурой (к примеру, эмоции, языковое общение), а ее язык налагает некоторые ограничения [27].

### Системы имитационного моделирования

Поскольку достаточно гибкой системы моделирования, удовлетворяющей поставленным требованиям, среди сред моделирования биологических систем не найдено, был осуществлен поиск по программам и библиотекам, позволяющим строить модели различных систем, не ограниченных выбранным типом. Подобные системы могут предоставлять поддержку моделирования среды различного уровня, в том числе симуляцию законов физики, а также пространственное представление различного уровня.

**AnyLogic**– это ПО для имитационного моделирования. Оно позволяет симулировать модели, построенные с использованием трех различных подходов: системная динамика, дискретно-событийное моделирование, агентное моделирование и их комбинации [28]. Среда обладает современным графическим интерфейсом, который позволяет создавать интерактивную анимацию для наглядности модели и интеграцию с геоинформационными системами, базами данных и пользовательскими библиотеками. Разработанные модели можно экспортировать в приложения на языке Java. AnyLogic использует модульную объектно-ориентированную парадигму. Помимо задания различных последовательностей взаимодействий элементов модели, в среде также можно использовать стохастические процессы для моделирования случайных событий.

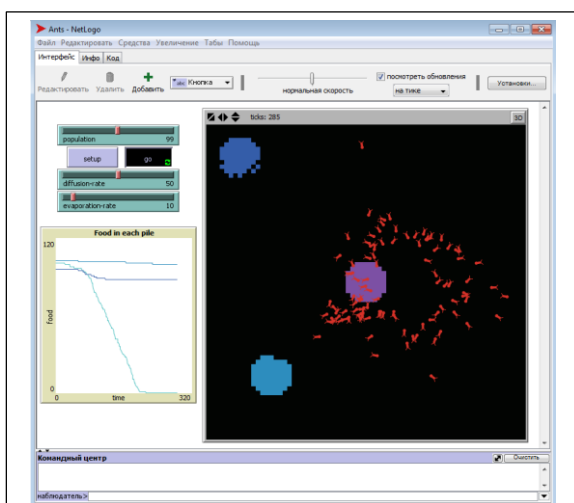


Рис. 2. Особенности программы NetLogo: Ants

Fig. 2. Features of the program NetLogo: Ants

Тем не менее, система предназначена в основном для моделирования бизнес-процессов, логики и производства, включая стратегические модели развития рынков и компаний, физические законы в этой среде напрямую не поддерживаются, и их необходимо самостоятельно реализовывать. Также стоит отметить, что ПО является платным, хотя есть и бесплатные версии, доступные при соблюдении определенных ограничений.

**Simplex3.** Система имитационного моделирования Simplex3 [29] позволяет строить дискретные модели и обладает собственным встроенным языком описания эксперимента Simplex-EDL. Simplex3 – среда разработки имитационного моделирования с собственным языком описания модели, разработанная в университетах Нюрнберг-Эрланген и Пассау (Германия), которая развивалась в течение нескольких итераций, включая предшественника Simplex2 [30]. Она позволяет строить системно-динамические и дискретно-событийные модели и расширять их с помощью языка C, но полноценной поддержки агентного моделирования в системе нет. Код системы закрытый, с намерением разработчика коммерциализировать проект, однако пока что, судя по сайту, проект требует лицензии и не предоставляет очевидного способа (кроме прямого запроса автору) получить (скачать) систему или приобрести лицензию. Поддержка продукта в виде документации и примеров весьма ограниченная, особенно на английском языке (основные работы на немецком языке [31]).

**Repast** – набор инструментов: библиотек и модулей интегрированной среды разработки Eclipse, позволяющий симулировать агентные модели [32].

Поддерживает языки программирования Java, Logo, C++, имеет встроенный расширяемый графический интерфейс, экспорт данных, реализует интерфейсы, позволяющие работать с рядом внешних программ, обрабатывающих и визуализирующих данные (RStudio, обработчики таблиц наподобие Excel, WEKA Data Mining Software и некоторые другие), генерирует переносимый stand-alone-код на Java, во многих случаях скрывает проблемы параллельности и синхронизации и решает их за пользователя, имеет версию для использования на вычислительных кластерах (Repast for High Performance Computing [33]). Графический интерфейс среды может настраиваться пользователем. Иллюстративный пример, показывающий одновременно визуализацию модели и вывод собираемых данных, приведен на рисунке 3.

Однако, несмотря на возможность удобным образом редактировать определенные элементы интерфейса, выводить графики и выбирать экспортируемые данные, в нем непросто менять интерфейс произвольным образом, добавлять нестандартное взаимодействие с элементами симуляции и дополнительные пользовательские окна и элементы интерфейса.

В Repast нет стандартизованного и поддерживаемого библиотекой способа организации взаимодействия между агентами: для общения или иных действий агентов, объектом которых выступает другой агент, необходимо вызывать методы объекта взаимодействия или редактировать его параметры напрямую. Это вызывает необходимость следить за его состоянием, учитывая возможность параллельного действия нескольких агентов на

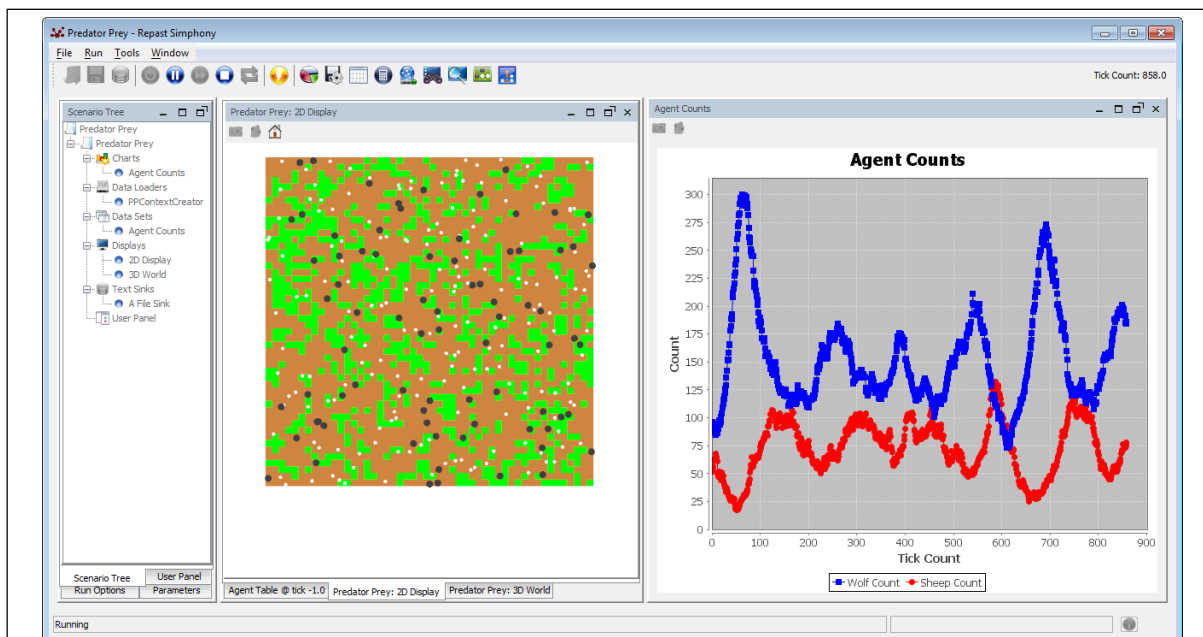


Рис. 3. Графический интерфейс Repast на примере модели «Охотник–Жертва»

Fig. 3. Repast graphical interface on the example of "Hunter–Victim" model

него (то есть заниматься синхронизацией, что обычно выполняется библиотекой Repast в других случаях), а также затрудняет написание двустороннего диалога (взаимодействия) между агентами, происходящего в течение одного шага симуляции. Создание модулей или надстройки над Repast возможно, однако, хотя документация и есть, поддержка этого ПО несколько ограниченная, разработка и расширение системы могут быть весьма проблематичными.

### Системы симуляции роботов

Отдельно стоит упомянуть системы, предназначенные для моделирования групп роботов как наиболее приближенные к задаче построения искусственного коллектива.

**Webots.** Система моделирования Webots (см. [34, 35]) является кроссплатформенным (Windows, Mac и Linux) симулятором роботов с множеством возможных интеграций: MATLAB, ROS (Robotic Operating System [36]) и API для языков программирования C++, Java и Python. Webots предоставляет обширную библиотеку моделей сенсоров, интерактивную 3D-визуализацию, поддержку CAD-систем, набор роботов и окружений, библиотеку для коммуникаций между роботами и вывод данных в специальные окна. Среди имеющихся библиотек есть наиболее используемые сенсоры и исполнительные устройства: датчики света, акселерометры, камеры и некоторые другие. Основой Webots является библиотека физической симуляции ODE (Open Dynamics Engine [37]), которая позволяет моделировать динамику твердых тел. Это дает возможность определения столкновений тел, расчета сил, моментов, учета трения и многое другое. Модель робота задает необходимые параметры, позволяющие протестировать ее в симуляции среды с учетом физических эффектов, и включает в себя графические свойства, такие как форма, размер, положение и ориентация, цвет и текстура объекта. Для некоторых моделей есть возможность переноса алгоритма управления из среды разработки в реального робота. Программа является коммерческой.

**Gazebo** – это программа и набор библиотек, позволяющие моделировать поведение роботов с учетом физических эффектов [38]. Модель робота описывается на специальном языке SDF либо URDF, которые являются надстройкой над XML. Набор роботов и модель мира подгружаются в систему моделирования, после чего можно начинать симуляцию. Работа всегда происходит в 3D-мире. Поддерживается режим как с графическим отображением объектов, так и без него для ускорения обработки.

Основой Gazebo является физический движок, который можно при необходимости переключить перед началом симуляции. По умолчанию исполь-

зуется ODE. Стандартные средства Gazebo предоставляют возможность моделировать силу тяжести, некоторые виды трения, динамику движения твердого тела и некоторые другие. Возможности среды можно расширить, используя готовые или написав собственные плагины. С их помощью можно как изменить или добавить физический эффект, так и ввести специфические правила поведения объектов: добавить внешне контролируемые актуаторы робота или датчик. Примерами возможностей этих расширений являются инфракрасные дальнометры, видеокамера, сервомоторы, управляемые внешними программами, встроенный в систему моделирования контроллер и даже физические эффекты передвижения в жидкости.

Несмотря на эти многочисленные достоинства, Gazebo не очень подходит для симуляции систем с большим количеством агентов из-за достаточно большой ресурсоемкости производимой обработки, поэтому моделировать работу систем с более, чем 1 000 агентов затруднительно.

**ARGoS** – многопоточный симулятор, ориентированный на поддержку большого количества роботов в различных физических средах в рамках одной симуляции. Он поддерживает как 2D-, так и 3D-моделирование [39]. Большое внимание в нем уделено модульности, в результате чего предоставляется возможность использовать разработанный код как в симуляции, так и на реальных роботах, выбирать между различными визуализациями и физическими движками, пользоваться ускорением вычислений за счет параллельного выполнения расчетов с несколькими типами синхронизации и даже совмещать несколько различных библиотек физической симуляции в одной модели мира. Помимо этого, ARGoS предоставляет симуляцию нескольких разных типов коммуникации между роботами: RFID-метки, WiFi и т.д.

ARGoS доступен в виде исходных кодов (под лицензией MIT [40]), а также готовых пакетов под MacOSX и нескольких дистрибутивов GNU/Linux. Имеется некоторая поддержка Windows. Система поддерживает пользовательские расширения. Скорость симуляции может достигать 40 % ускорения по сравнению с реальным временем для 10 000 роботов [41]. Однако, как и многие другие симуляторы, ARGoS не предоставляет особой поддержки механизмам социального взаимодействия в коллективах роботов, а его масштабируемость, судя по всему, ограничивается использованием локальных ресурсов.

Основные характеристики рассмотренных систем отображены в таблице. Эти критерии расшифровываются следующим образом: имитация физических свойств объекта – моделирование физических явлений, механических ограничений (трение, вязкость, связи между частями объекта и т.п.); реализация библиотек элементов робота – набор готовых реализаций сенсоров и актуаторов; наличие



### Характеристики систем System characteristics

Система моделирования	Имитация физических свойств объекта	Реализация библиотек элементов робота	Наличие поддержки коммуникации между агентами	Поддержка большого количества агентов	Уровень поддержки	Поддержка вычислений на кластерах	Наличие социально ориентированных библиотек
Myrmedrome	-	-	+	+	-	-	+
AntMe	-	-	-	+	+-	-	+
NetLogo: Ants	-	-	-	+	+	-	+
AnyLogic	-	-	+	+	+	-	-
Simplex3	-	-	-	+	-	-	-
Repast	-	-	-	+	+-	+	+
Webots	+	+	+	-	+	-	-
Gazebo	+	+	-	-	+-	-	-
ARGoS	+	+	+	+	-+	-	-

поддержки коммуникации между агентами – моделирование общения (связи) между агентами и предоставление стандартного интерфейса определения протокола и сообщений; поддержка большого количества агентов – приспособленность системы для работы с большими (порядка 1 000, 10 000) группами роботов; уровень поддержки – суммарная оценка качества пользовательской документации, уровня поддержки со стороны сообщества и разработчиков, интуитивность работы с системой, документированность кода (в случае открытого кода); поддержка вычислений на кластерах – встроенная возможность работы системы на кластере; наличие социально ориентированных библиотек – предоставление встроенных или пользовательских библиотек, реализующих хотя бы некоторые из социальных механизмов.

Еще раз отметим, что в обзор попали лишь некоторые системы, характерные для своего класса, интересные с точки зрения критериев, определенных спецификой задач социальной робототехники. Отдельно можно сказать о таких известных и часто упоминаемых системах моделирования, как Microsoft Robotics Developer Studio (MRDS) [42] и Ant Farm Simulator, которые наряду с некоторыми другими не попали в обзор, так как или относятся к одному из рассмотренных классов, или предназначены для решения других категорий задач агентного моделирования, например транспортных, логистических и т.д.

### Заключение

Сравнительный анализ приведенных программных продуктов показывает, что они лишь частично удовлетворяют специфике моделирования многоагентных робототехнических систем и искусственных коллективов с социальной структурой, и позволяет составить более точные требования к такой среде моделирования.

ПО моделирования биологических коллективов обладает весьма ограниченным набором возможностей и расширяемостью, а также зачастую не задумывалось как библиотека моделирования для

изучения коллективного управления. Исключением является модель Ants для NetLogo, но она тоже не предоставляет особых средств для реализации многих важных механизмов коллективов с социальной структурой.

Несмотря на универсальность и мощность, в средах разработки и библиотеках имитационного моделирования не хватает хотя бы некоторых из следующих элементов: индивидуальные психические различия агентов, поддержка локальной коммуникации, языковое общение, алгоритмы формирования коалиций, дифференциация функций, выделение иерархической структуры, поддержка пространственных и временных отношений.

ПО моделирования роботов больше подходит для отдельных роботов и физических эффектов, а не для систем с большим количеством агентов либо имеет те же проблемы, что и системы имитационного моделирования: специальные средства для моделирования коллективов с социальной структурой не предоставляются, что означает необходимость исследователю самостоятельно реализовывать многочисленные механизмы.

На основе проведенного обзора можно сформулировать следующие функциональные требования к библиотеке (или среде) моделирования для коллективов с социальной структурой в области робототехники:

- масштабируемость, возможность моделирования с применением вычислительных кластеров и соответствующая поддержка со стороны библиотеки (пользователь, насколько это возможно, не должен следить за распараллеливанием кода и синхронизацией);

- поддержка определенного функционала, свойственного агентным системам, и его предоставление в виде библиотек: агент, взаимодействие между агентами, среда, законы взаимодействия со средой, пространство, время;

- реализация некоторых базовых механизмов социального поведения и структур и поддержка дополнения и расширения таких механизмов, как индивидуальные психические различия, дифференциация функций и т.п.;

- поддержка создания моделей внутреннего и внешнего мира агента;
- поддержка модели из нескольких взаимодействующих коллективов.

Кроме того, выделим дополнительные технические требования, которые представляются важными для использования системы различными группами исследователей:

- предоставление инструментов или интерфейсов для интеграции кода в сторонний пользовательский графический интерфейс и визуализации данных, а также базового варианта интерфейса;
- экспорт данных в общепринятые, открытые форматы (CSV, GraphML и т.п.), позволяющие интеграцию с другими инструментами;
- открытость кода для чтения (Open source [43]), соблюдение стилей и наличие хорошей документации: хорошо описанное API, ее общее устройство (архитектура, концепции);
- открытость кода для модификации, поддержка плагинов или иных стандартных способов расширения функциональности для исправления ошибок пользователями и возможности расширения до большего класса поддерживаемых систем.

Таким образом, проведенный обзор существующих систем моделирования для изучения робототехнических коллективов с социальной структурой позволил сформулировать требования к желаемой системе и выявил неудовлетворенную потребность в наличии специализированной системы агентного моделирования, учитывающей их особенности.

*Работа выполнена при частичной финансовой поддержке грантов РНФ № 16-11-00018 и РФФИ № 16-29-04412 офи\_м.*

#### Литература

1. Карпов В.Э. Модели социального поведения в групповой робототехнике // Управление большими системами. 2016. № 59. С. 165–232.
2. Захаров А.А. Муравей, семья, колония. М.: Наука, 1978. 144 с.
3. Adamatzky A., Komosinski M. (eds.). Artificial life models in software. Springer, London, 2009, no. 2, 441 p.
4. Allan R. Survey of agent based modelling and simulation tools. 2009. URL: <http://www.grids.ac.uk/Complex/ABMS/> (дата обращения: 08.01.2017).
5. Kravari K., Bassiliades N. A survey of agent platforms. J. Artif. Soc. Soc. Simul., 2015, vol. 18, no. 1, p. 11.
6. Nikolai C., Madey G. Tools of the trade: a survey of various agent based modeling platforms. J. Artif. Soc. Soc. Simul., 2009, vol. 12, no. 2, p. 2.
7. Длусский Г.М. Муравьи рода формика. М.: Наука, 1967. 236 с.
8. Тинберген Н. Социальное поведение животных; [под ред. П.В. Симонова]. М.: Мир, 1993. 152 с.
9. Гудолл Д. Шимпанзе в природе: поведение. М.: Мир, 1992. 670 с.
10. Karpov V., et al. Multi-robot exploration and mapping based on the subdefinite models. Lecture Notes in Comp. Sc.: Ronzhin A., Rigol G., Meshcheryakov R. (eds.), Springer, 2016, vol. 9812, pp. 143–152.
11. Карпова И.П. Псевдоаналоговая коммуникация в группе роботов // Мехатроника, автоматизация, управление. 2016. Т. 17. № 2. С. 94–101.
12. Kulnich A.A. A model of agents (robots) command behavior: the cognitive approach. Autom. Remote Control. 2016, vol. 77, no. 3, pp. 510–522.
13. Кулинич А.А. Модель командной работы агентов с BDI архитектурой // КИИ-2016: сб. тр. XV Национальн. конф. 2016. С. 344–352.
14. Bongard J., Zykov V., Lipson H. Resilient machines through continuous self-modeling. Science, 2006, vol. 314, no. 5802, pp. 1118–1121.
15. Ровбо М.А. Распределение ролей в гетерогенном муравьино-подобном коллективе // КИИ-2016: сб. тр. XV Национальн. конф. 2016. Т. 2. С. 363–371.
16. Воробьев В.В. Алгоритм кластеризации коллектива роботов // Беспилотные транспортные средства с элементами искусственного интеллекта: сб. тр. III Всерос. науч.-практич. сем. Иннополис, 2015. С. 50–59.
17. Purushotham Muniganti, Albert Oller Pujol. A survey on mathematical models of swarm robotics. Proc. Conf. Workshop on Physical Agents. 2010. URL: [https://www.researchgate.net/publication/230793772\\_A\\_Survey\\_on\\_Mathematical\\_models\\_of\\_Swarm\\_Robotics](https://www.researchgate.net/publication/230793772_A_Survey_on_Mathematical_models_of_Swarm_Robotics) (дата обращения: 08.01.2017).
18. Бондаревский А.С., Лебедев А.В. Имитационное моделирование: определение, применяемость и техническая реализация // Фундаментальные исследования. 2011. № 12–3. С. 535–541.
19. Économou M., et al. Aints. 2010. URL: <http://a-i-nts.blogspot.ru/2010/03/presentation-of-aints.html> (дата обращения: 08.01.2017).
20. Cristiani E. MYRMEDROME. 2016. URL: [http://www.emilianocristiani.it/myrmedrome/main\\_en.html](http://www.emilianocristiani.it/myrmedrome/main_en.html) (дата обращения: 08.01.2017).
21. Wendel T. AntMe. 2016. URL: <https://service.antme.net/> (дата обращения: 08.01.2017).
22. AI Challenge Ants. 2011. URL: <http://ants.aichallenge.org/> (дата обращения: 08.01.2017).
23. Cacace S., Cristiani E., D'Eustacchio D. Myrmedrome: simulating the life of an ant colony. Imagine Math 2: Between Culture and Mathematics, Springer Milan, 2013, pp. 201–210.
24. Wilensky U. NetLogo. 1999. URL: <http://ccl.northwestern.edu/netlogo/> (дата обращения: 08.01.2017).
25. Wilensky U. NetLogo Ants model. 1997. URL: <http://ccl.northwestern.edu/netlogo/models/Ants> (дата обращения: 08.01.2017).
26. NetLogo Source Code. 2016. URL: <https://github.com/NetLogo/NetLogo> (дата обращения: 08.01.2017).
27. Lytinen S.L., Railsback S.F. The evolution of agent-based simulation platforms: a review of NetLogo 5.0 and ReLogo. Eur. Meet. Cybern. Syst. Res., 2010, pp. 1–11.
28. Borshchev A. The big book of simulation modeling: multi-method modeling with AnyLogic 6. Amazon Digital Services LLC, 2015, 614 p.
29. SOI-Software. Simplex3. 2016. URL: <http://www.simplex3.net/Body/Introduction/English/indexAbstract.html> (дата обращения: 08.01.2017).
30. Dörmhöfer K. Die Benutzerumgebung von SIMPLEX II – Eine offene graphische Bedienoberfläche zur Erstellung und Auswertung von Simulationsmodellen. Univ. Erlangen-Nürnberg, 1991, pp. 1–336.
31. Eschenbacher P. Entwurf und Implementierung einer formalen Sprache zur Beschreibung dynamischer Modelle. Institut für Mathematische Maschinen und Datenverarbeitung, 1989, 328 s.
32. Argonne National Laboratory. Repast Symphony. 2016. URL: [http://repast.sourceforge.net/repast\\_symphony.php#](http://repast.sourceforge.net/repast_symphony.php#) (дата обращения: 08.01.2017).
33. Argonne National Laboratory. Repast for High Performance Computing. 2016. URL: [http://repast.sourceforge.net/repast\\_hpc.php#](http://repast.sourceforge.net/repast_hpc.php#) (дата обращения: 08.01.2017).
34. Michel O. Webots TM: Professional Mobile Robot Simulation. Int. J. Adv. Robot. Syst., 2004, vol. 1, no. 1, pp. 39–42.
35. Cyberbotics Ltd. Webots. 2016. URL: <https://www.cyberbotics.com/webots.php> (дата обращения: 08.01.2017).
36. ROS Indigo Igloo. 2014. URL: <http://wiki.ros.org/indigo> (дата обращения: 08.01.2017).
37. Smith R. Open Dynamics Engine. 2016. URL: <http://www.ode.org/> (дата обращения: 08.01.2017).

38. Koenig N., Howard A. Design and use paradigms for gazebo, an open-source multi-robot simulator. 2004 IEEE/RSJ Int. Conf. Intell. Robot. Syst., 2004, vol. 3, pp. 2149–2154.

39. Pinciroli C. et al. ARGoS: A modular, parallel, multi-engine simulator for multi-robot systems // Swarm Intell. 2012, vol. 6, no. 4, pp. 271–295.

40. The MIT License (MIT). 2016. URL: <https://opensource.org/licenses/MIT> (дата обращения: 08.01.2017).

41. Pinciroli C. et al. ARGoS: A modular, multi-engine simulator for heterogeneous swarm robotics // IEEE Int. Conf. Intell. Robot. Syst., 2011, pp. 5027–5034.

42. Microsoft. Microsoft Robotics Developer Studio 4. 2012. URL: <https://www.microsoft.com/en-us/download/details.aspx?id=29081> (дата обращения: 08.01.2017).

43. Open Source Initiative. The Open Source Definition. 2007. URL: <https://opensource.org/osd> (дата обращения: 08.01.2017).

Software & Systems

DOI: 10.15827/0236-235X.119.425-434

Received 30.01.17

2017, vol. 30, no. 3, pp. 425–434

## REVIEW OF SIMULATION MODELING TOOLS FOR ROBOT GROUPS WITH SOCIAL ORGANIZATION ELEMENTS

**M.A. Rovbo**<sup>1</sup>, Research Engineer, [rovboma@gmail.com](mailto:rovboma@gmail.com)

**E.E. Ovsyannikova**<sup>1</sup>, Research Engineer, [eeovsyan@yandex.ru](mailto:eeovsyan@yandex.ru)

**A.A. Chumachenko**<sup>2</sup>, Graduate Student, [chumachenkoa93@mail.ru](mailto:chumachenkoa93@mail.ru)

<sup>1</sup> National Research Centre “Kurchatov Institute”, Akademika Kurchatova Sq. 1, Moscow, 123182, Russian Federation

<sup>2</sup> National Research University Higher School of Economics, Myasnitskaya St. 20, Moscow, 101000, Russian Federation

**Abstract.** The paper discusses mechanisms that are important for a proper simulation of artificial agents' group, especially mobile robots, which are organized into a system with the elements of a social structure.

The authors provide an overview of simulation tools based on these requirements to determine advantages and disadvantages of existing systems for modeling, simulating and researching such systems. They identify key instruments for a simulation of artificial groups with social structures. The review focuses on simulation tools for biological systems (in particular, ants since they are a good example of a social group), agent-based modeling and robot-specialized simulation tools.

The paper emphasizes the level of support of various social mechanisms in simulation tools, as well as their capability to model a large number of agents (in particular, cluster computing support) and the ability to simulate user-provided models and implementations of different aspects of a social group. It was found out that, although there are many general modeling and simulation tools, there is very little support and ready-to-use implementations of even widely used mechanisms, such as pheromones, which are provided by systems to build and research a group of robotic agents with a social structure. On the other hand, the tools that aim at simulating systems with social behavior elements, such as ant modeling systems, are not very scalable or limited in aspects they can simulate. There are no mechanisms to expand them for further research.

Based on the review, the paper has detected a lack of a comprehensive simulation system that specifically aims at the research of artificial agent groups with social structure elements. It has also formulated a set of requirements for such a system.

**Keywords:** agent-based simulation, social behavior modeling, swarm robotics, simulation modeling, development tools, review, cluster computing, bioinspired systems, eusociality, robot.

**Acknowledgements.** The work has been partially supported by the Russian Science Foundation, project no. 16-11-00018, RFBR 16-29-04412 ofi\_m.

## References

1. Karpov V.E. Models of social behaviour in the group robotics. *Upravlenie bolshimi sistemami* [Large-scale Systems Control]. 2016, no. 59, pp. 165–232 (in Russ.).
2. Zakharov A.A. *Muravey, semya, koloniya* [Ant, family, colony]. Moscow, Nauka Publ., 1978, 144 p.
3. *Artificial Life Models in Software*. Komosinski M., Adamatzky A. (Eds.). London, Springer Publ., 2009, no. 2, 441 p.
4. Allan R. *Survey of Agent Based Modelling and Simulation Tools*. 2009. Available at: <http://www.grids.ac.uk/Complex/ABMS/> (accessed January 8, 2017).
5. Kravari K., Bassiliades N. A Survey of Agent Platforms. *J. Artif. Soc. Soc. Simul.* 2015, vol. 18, no. 1, p. 11.
6. Nikolai C., Madey G. Tools of the Trade: A Survey of Various Agent Based Modeling Platforms. *J. Artif. Soc. Soc. Simul.* 2009, vol. 12, no. 2, p. 2.
7. Dlussky G.M. *Muravyi roda formika* [Formica Genus Ants]. Moscow, Nauka Publ., 1967, 236 p.
8. Tinbergen N. *Sotsialnoe povedenie zhivotnykh* [Social Behaviour of Animals]. Simonov P.V. (Ed.) Moscow, Mir Publ., 1993, 152 p.
9. Gudoll D. *Shimpanze v prirode: povedenie* [Chimpanzee in the Wild: Behaviour]. Moscow, Mir Publ., 1992, 670 p.
10. Karpov V., Migalev A., Moscovsky A., Rovbo M., Vorobiev V. Multi-robot Exploration and Mapping Based on the Subdefinite Models. *Interactive Collaborative Robotics. ICR 2016. Lecture Notes in Computer Science*. Ronzhin A., Rigoll G., Meshcheryakov R. (Eds.). Springer Publ., Cham, 2016, vol. 9812, pp. 143–152.
11. Karpova I.P. Pseudo-Analog Communication in Robot Groups. *Mekhatronika, avtomatizatsiya, upravlenie* [Mechatronics, Automation, Control]. 2016, vol. 17, no. 2, pp. 94–101.
12. Kulinich A.A. A model of agents (robots) command behavior: The cognitive approach. *Autom. Remote Control*. 2016, vol. 77, no 3, pp. 510–522.

13. Kulinich A.A. BDI Architecture Model of Agents' Team Work. *Proc. 15th National Conf. on Artificial Intelligence with Int. Participation (CAI 2016)*. Smolensk, 2016, pp. 344–352 (in Russ.).
14. Bongard J., Zykov V., Lipson H. Resilient Machines Through Continuous Self-Modeling. *Science*. 2006, vol. 314, no. 5802, pp. 1118–1121.
15. Rovbo M.A. Role Assignment in a Heterogeneous Ant-Like Collective. *Proc. 15th National Conf. on Artificial Intelligence with Int. Participation (CAI 2016)*. Smolensk, 2016, vol. 2, pp. 363–371 (in Russ.).
16. Vorobev V.V. Robot collective clusterization algorithm. *III Vseros. nauch.-praktich. seminar "Bespilotnye transportnye sredstva s elementami iskusstvennogo intellekta"*. Innopolis [Proc. 3rd All-Russian Science and practice Workshop on AI-Powered Unmanned Vehicles. Innopolis]. 2015, pp. 50–59 (in Russ.).
17. Purushotham Muniganti, Albert Oller Pujol. A survey on mathematical models of swarm robotics. *Proc. Conf. Workshop on Physical Agents*. 2010. Available at: [https://www.researchgate.net/publication/230793772\\_A\\_Survey\\_on\\_Mathematical\\_models\\_of\\_Swarm\\_Robotics](https://www.researchgate.net/publication/230793772_A_Survey_on_Mathematical_models_of_Swarm_Robotics) (accessed January 8, 2017).
18. Bondarevsky A.S., Lebedev A.V. Simulation modeling: definition, applicability and technical implementation. *Fundamentalnye issledovaniya* [Fundamental research]. 2011, iss. 12 (pt. 3), pp. 535–541 (in Russ.).
19. Économou M., Kostenko S., Hauchecorne É., Vandon R. *Aints*. 2010. Available at: <http://a-i-nts.blogspot.ru/2010/03/presentation-of-aints.html> (accessed January 8, 2017).
20. Cristiani E. *MYRMEDROME*. 2016. Available at: [http://www.emilianocristiani.it/myrmedrome/main\\_en.html](http://www.emilianocristiani.it/myrmedrome/main_en.html) (accessed January 8, 2017).
21. Wendel T. *AntMe*. 2016. Available at: <https://service.antme.net/> (accessed January 8, 2017).
22. *AI Challenge Ants*. 2011. Available at: <http://ants.aichallenge.org/> (accessed January 8, 2017).
23. Cacace S., Cristiani E., D'Eustacchio D. Myrmedrome: Simulating the Life of an Ant Colony. *Imagine Math 2: Between Culture and Mathematics*. Milano, Springer Publ., 2013, pp. 201–210.
24. Wilensky U. *NetLogo*. Center for Connected Learning and Computer-Based Modeling. 1999. Available at: <http://ccl.northwestern.edu/netlogo/> (accessed January 8, 2017).
25. Wilensky U. *NetLogo Ants model*. Center for Connected Learning and Computer-Based Modeling. 1997. Available at: <http://ccl.northwestern.edu/netlogo/models/Ants> (accessed January 8, 2017).
26. *NetLogo Source Code*. 2016. Available at: <https://github.com/NetLogo/NetLogo> (accessed January 8, 2017).
27. Lytinen S.L., Railsback S.F. The evolution of agent-based simulation platforms: a review of NetLogo 5.0 and ReLogo. *Eur. Meet. Cybern. Syst. Res.* 2010, pp. 1–11.
28. Borshchev A. *The Big Book of Simulation Modeling: Multimethod Modeling with AnyLogic 6*. Amazon Digital Services LLC, 2015, 614 p.
29. *SOI-Software. Simplex*. 2016. Available at: <http://www.simplex3.net/Body/Introduction/English/indexAbstract.html> (accessed January 8, 2017).
30. Dörnhöfer K. *Die Benutzerumgebung von SIMPLEX II – Eine offene graphische Bedienoberfläche zur Erstellung und Auswertung von Simulationsmodellen* [The User Environment of SIMPLEX II – An Open Graphical User Interface for Creating and Evaluating Simulation Models]. Univ. Erlangen-Nürnberg Publ., 1991 (in German).
31. Eschenbacher P. *Entwurf und Implementierung einer formalen Sprache zur Beschreibung dynamischer Modelle* [Design and Implementation of a Formal Language for Describing Dynamic Models]. Institute of Mathematical Machines and Data Processing Publ., 1989, 328 p. (in German).
32. Argonne National Laboratory. *Repast Symphony*. 2016. Available at: [http://repast.sourceforge.net/repast\\_symphony.php#](http://repast.sourceforge.net/repast_symphony.php#) (accessed January 8, 2017).
33. Argonne National Laboratory. *Repast for High Performance Computing*. 2016. Available at: [http://repast.sourceforge.net/repast\\_hpc.php#](http://repast.sourceforge.net/repast_hpc.php#) (accessed January 8, 2017).
34. Michel O. Webots TM : Professional Mobile Robot Simulation. *Int. J. Adv. Robot. Syst.* 2004, vol. 1, no. 1, pp. 39–42.
35. *Cyberbotics Ltd. Webots*. 2016. Available at: <https://www.cyberbotics.com/webots.php> (accessed January 8, 2017).
36. *ROS Indigo Igloo*. 2014. Available at: <http://wiki.ros.org/indigo> (accessed January 8, 2017).
37. Smith R. *Open Dynamics Engine*. 2016. Available at: <http://www.ode.org/> (accessed January 8, 2017).
38. Koenig N., Howard A. Design and use paradigms for gazebo, an open-source multi-robot simulator. *2004 IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IEEE Cat. No.04CH37566)*. 2004, vol. 3, pp. 2149–2154.
39. Pinciroli C., Trianni V., O'Grady R., Pini G., Brutschy A., Brambilla M., Mathews N., Ferrante E., Di Caro G., Ducatelle F., Birattari M., Gambardella L.M., Dorigo M. ARGoS: A modular, parallel, multi-engine simulator for multi-robot systems. *Swarm Intell.* 2012, vol. 6, no. 4, pp. 271–295.
40. *The MIT License (MIT)*. 2016. Available at: <https://opensource.org/licenses/MIT> (accessed January 8, 2017).
41. Pinciroli C., Trianni V., O'Grady R., Pini G., Brutschy A., Brambilla M., Mathews N., Ferrante E., Di Caro G., Ducatelle F., Stirling T., Gutiérrez Á., Gambardella L.M., Dorigo M. ARGoS: A modular, multi-engine simulator for heterogeneous swarm robotics. *IEEE Int. Conf. Intell. Robot. Syst.* 2011, pp. 5027–5034.
42. *Microsoft. Microsoft Robotics Developer Studio 4*. 2012. Available at: <https://www.microsoft.com/en-us/download/details.aspx?id=29081> (accessed January 8, 2017).
43. *Open Source Initiative. The Open Source Definition*. 2007. Available at: <https://opensource.org/osd> (accessed January 8, 2017).

УДК 004.932

DOI: 10.15827/0236-235X.119.435-438

Дата подачи статьи: 20.04.17

2017. Т. 30. № 3. С. 435–438

**РАЗРАБОТКА СИСТЕМЫ СТЕРЕОЗРЕНИЯ ДЛЯ МОБИЛЬНОГО РОБОТА***С.В. Ульянов, д.ф.-м.н., профессор, ulyanovsv@mail.ru;**А.Г. Решетников, к.т.н., доцент, agreshetnikov@gmail.com;**К.В. Кошелев, студент, kirill\_koshelev18@rambler.ru**(Университет «Дубна», Институт системного анализа и управления,  
ул. Университетская, 19, г. Дубна, 141980, Россия)*

Основным источником развития робототехнических систем и систем искусственного интеллекта является усовершенствование вычислительной техники, устройств и сенсоров, позволяющих получать информацию об окружающей среде. Этот процесс неразрывно связан с развитием ПО бортовых процессоров, которые используются при управлении автономным роботом. В данной статье приводится краткое описание модуля разрабатываемой системы распознавания образов автономного робота, базирующегося на технологии стереозрения.

В настоящее время все чаще упоминаются системы, состоящие из комбинаций нескольких видеокамер, лазерных радаров, тепловизоров и других источников получения информации об окружающей среде. Широкое применение обобщение такого класса получило при создании автопилотов для транспортных средств, систем контроля безопасности, беспилотной авиаразведки, различных бытовых роботов. Применение стереозрения позволяет получать данные о глубине изображения и расстоянии до объектов, строить трехмерную картину окружающего мира.

В работе также дано описание настройки используемого оборудования (пара камер, закрепленных на общем основании): описан процесс калибровки стереопары (поиск калибровочного шаблона на изображении производился с помощью фреймворка ROS). Изображения, полученные с камер, подвергаются ректификации – преобразованию, при котором оба изображения проецируются на плоскость, параллельную линии, соединяющей оптические центры камер.

В статье рассмотрены основы проективных преобразований и эпиполярной геометрии, а также идея, лежащая в основе построения карт глубины и смещений. В качестве результата представлен программный модуль распознавания, разработанный с помощью языка программирования Python.

**Ключевые слова:** распознавание образов, стереозрение, Robot Operating System (ROS), системы управления, робототехника, калибровка камер.

В большинстве случаев для машинного зрения и распознавания достаточно использовать одну камеру [1–3]. Стереозрение, в определенной степени повторяя особенности развития природного зрения, позволяет бортовой системе получать информацию не только о цвете и яркости объекта, но и о расстоянии до него, о его геометрической форме, о препятствиях на пути к объекту. В технологии стереозрения используются, как правило, две камеры, работающие синхронно, что позволяет восстанавливать форму и расположение наблюдаемых объектов в трехмерном пространстве. При этом создается трехмерная полигональная модель какого-либо объекта, формируемая на основе анализа изображений этого объекта, полученных разнесенными в пространстве камерами.

Для операционной системы робота *Robot Operating System* (далее ROS) был разработан программный модуль распознавания объектов [4]. Фреймворк ROS представляет собой набор библиотек, утилит и соглашений. Система ROS предоставляет разработчику большое количество вспомогательного кода, позволяя ему сконцентрироваться на важнейших частях создаваемого ПО. Создание программного модуля распознавания объектов состоит из следующих этапов: настройка стереопары из двух камер, построение карты смещений, использование карты смещений для получения облака точек с помощью 3D-визуализатора Rviz, создание программного модуля распознавания объектов на языке программирования Python.

**Настройка стереопары  
из двух камер**

Данная настройка является первым этапом при проектировании системы стереозрения в ROS. Для выполнения поставленной задачи была использована пара камер, соединенных общим основанием. Для обеспечения возможности обмена данными между ROS и камерами необходимо создать в рабочем каталоге, который связан с ROS, пакет с файлом инициализации. В файле содержатся параметры для работы с изображением, а именно разрешение, имена видеоустройств. Каждая камера по сути становится узлом с определенным именем, а узлы, в свою очередь, взаимодействуют с ROS [4].

Калибровка камер позволяет исправлять дисторсию (абберацию оптических систем), при которой коэффициент линейного увеличения изменяется по полю зрения объектива. При этом нарушается геометрическое подобие между объектом и его изображением. Калибровка стереопары проводится с помощью фреймворка ROS методом шахматной доски. Калибровка камер обычно выполняется за счет многократной съемки калибровочного шаблона, на изображении можно легко выделить ключевые точки (рис. 1), для которых известны их относительные положения в пространстве.

Затем находятся коэффициенты, связывающие координаты проекций, матрицы камер и положения точек шаблона в пространстве [4, 5].

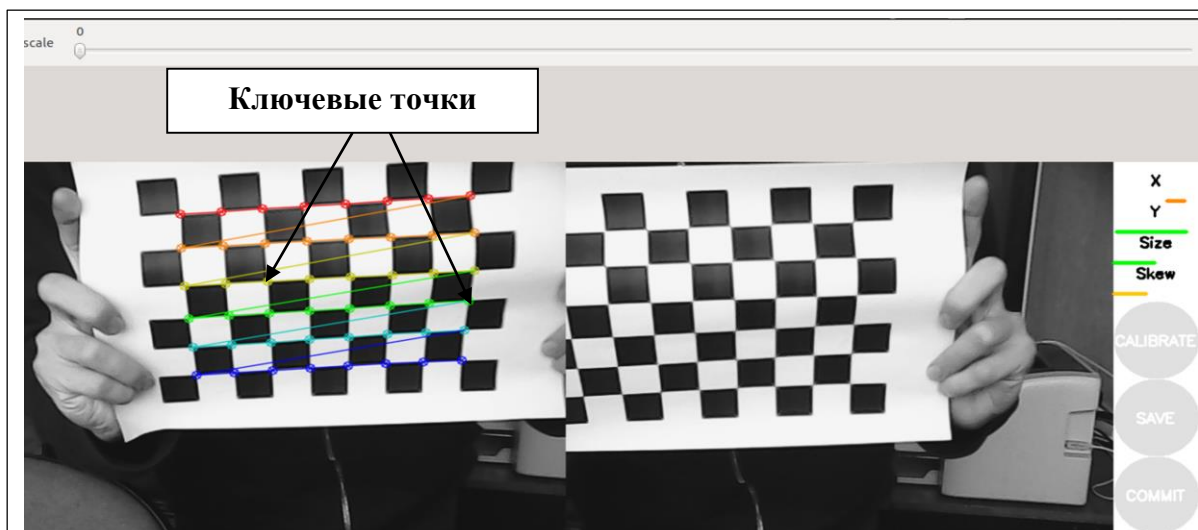


Рис. 1. Процесс калибровки (выделение ключевых точек)

Fig. 1. The calibration process (selection of key points)

### Построение карты смещений

Для построения трехмерного изображения с помощью 3D-визуализатора Rviz используются карты глубины и карты смещений. Рассмотрим такое понятие, как карта глубины [6]. Карта глубины – это способ представления объемных 3D-изображений в виде изображения, в котором каждому пикселю присваивается дополнительный параметр – глубина. Этот параметр показывает, на каком расстоянии от плоскости изображения расположен данный пиксель. Карта глубины может быть получена с помощью специальной камеры глубины (например, сенсор *Kinect* является своего рода такой камерой), а также может быть построена по изображениям, полученным со стереопары.

При построении карты смещений для каждой точки на одном изображении выполняется поиск парной ей точки на другом изображении. По паре соответствующих точек можно выполнить триангуляцию и определить координаты их прообраза в трехмерном пространстве. При известных трехмерных координатах прообраза глубина вычисляется как расстояние до плоскости камеры (рис. 2). Для каждой точки на изображении с первой камеры соответствующую ей парную точку нужно искать в той же строчке на изображении со второй камеры. Для каждого пикселя левого изображения с координатами  $(x_0, y_0)$  выполняется поиск соответствующего пикселя на правом изображении. При этом предполагается, что пиксель на правом изображении должен иметь координаты  $(x_0 - d, y_0)$ , где  $d$  – величина, называемая смещением (рис. 2) [7, 8]. Поиск соответствующего пикселя выполняется путем вычисления максимума функции отклика, в качестве которой может выступать, например, корреляция окрестностей пикселей. В результате получается карта смещений (рис. 3).

### Получение облака точек с помощью 3D-визуализатора Rviz

Визуализатор Rviz является модулем фреймворка ROS, предназначенным для отображения данных с различных датчиков (включая камеры, сонары, дальномеры). С его помощью, например, можно построить трехмерную карту помещения, а

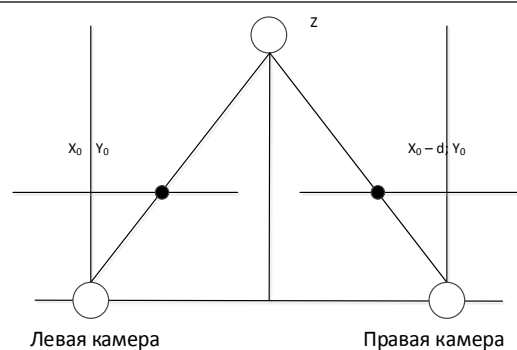


Рис. 2. Поиск соответствующих пар точек

Fig. 2. Search for matching pairs of points



Рис. 3. Полученная карта смещений

Fig. 3. A displacement map

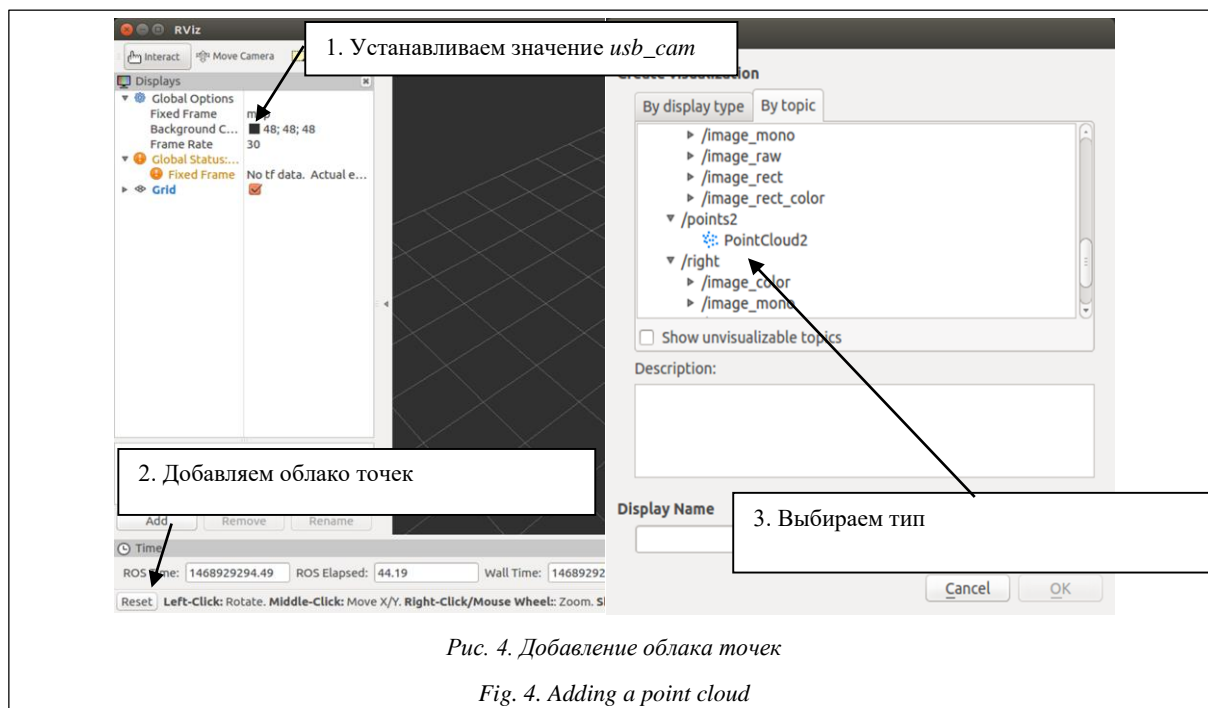


Рис. 4. Добавление облака точек

Fig. 4. Adding a point cloud

робот под управлением ROS, используя полученные данные об объектах и препятствиях, сможет автономно перемещаться в этом помещении. Первоначально в области глобальных настроек Global Options напротив поля Fixed Frame указываем имя пакета, посредством которого ROS взаимодействует с камерами. Затем нажимаем кнопку Add и выбираем из списка PointCloud2 (рис. 4).

Сообщение sensor\_msgs/PointCloud2 описывает облако точек в трехмерном пространстве [4]. На рисунке 5 показано трехмерное изображение, полученное с помощью стереопары из двух камер.

### Программный модуль системы распознавания

При выполнении задачи распознавания объектов программный код создавался на языке программирования Python. В качестве объекта был выбран шар желтого цвета. Разработанный модуль производит распознавание объекта (рис. 6).

Следует также отметить, что данный модуль позволяет осуществлять слежение за объектом. Однако программа распознает лишь те объекты, форма и цвет которых указаны непосредственно в коде, что несколько снижает ее эффективность. Решить данную проблему можно путем введения самоорганизации в процесс распознавания. Для повышения точности и качества распознавания при разработке системы будут использоваться интеллектуальные вычисления (применение нейросетового подхода с элементами нечеткой логики и эволюционных алгоритмов обучения) [9, 10].

### Заключение

В работе приводится описание процесса калибровки стереопары из двух камер, необходимого для практического применения технологии стереозрения. На основе изображений, полученных со стереопары, построена карта смещений, используемая для создания трехмерных изображений окружаю-

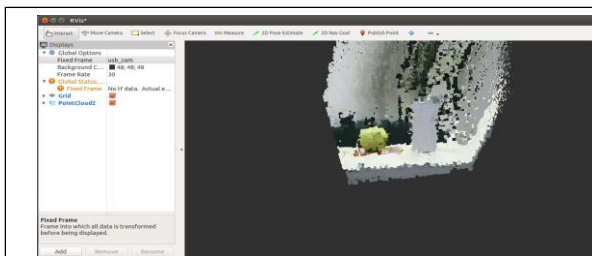


Рис. 5. 3D-изображение, полученное с помощью визуализатора Rviz

Fig. 5. 3D image obtained with the Rviz visualizer



Рис. 6. Распознавание шара

Fig. 6. Ball recognition



щих объектов. Результатом является программный модуль, с помощью которого осуществляется распознавание объектов и слежение за ними. Использование двух камер (стереопары) позволяет значительно повысить качество распознавания объектов. Изобилие систем машинного зрения не устраняет главные недостатки систем распознавания (погрешность распознавания при изменении ракурса объекта, изменение освещения, чувствительность ПО и т.д.) [10]. Использование интеллектуальных вычислений в совокупности с технологией стереозрения позволяет создать эффективную систему распознавания.

### Литература

1. Kuznetsova A., Sigal L. Expanding object detector's Horizon: Incremental Learning Framework for object detection in videos. IEEE Conf. CVPR 2015. URL: <https://www.disneyresearch.com/publication/expanding-object-detectors-horizon/> (дата обращения: 19.03.2017).

2. Распознавание образов для программистов: примеры работ. URL: <http://recog.ru/> (дата обращения: 9.03.2017).

3. Яне Б. Цифровая обработка изображений. М.: Техносфера, 2007. 584 с.

4. Robot operating system (ROS): documentation, support, tutorials. URL: <http://wiki.ros.org/> (дата обращения: 22.02.2017).

5. Szpytko J., Hyla P. Stereovision 3D type workspace mapping system architecture for transport devices. Jour. of KONES Powertrain and Transport, 2010, vol. 17, pp. 496–502.

6. Pentland A.P. Depth of Scene from Depth of Field. Proc. Image Understanding Workshop, 1982, pp. 253–259.

7. Bobick A.F., Intille S.S. Large occlusion stereo. Int. Jour. of Computer Vision, 1999, vol. 33, pp. 181–200.

8. Форсайт Д.А., Понс Ж. Компьютерное зрение: современный подход. М.: Вильямс, 2004. 928 с.

9. Ульянов С.В., Добрынин В.Н., Неведов Н.Ю., Петров С.П., Полунин А.С., Решетников А.Г. Генетические и квантовые алгоритмы. Ч. 1. Инновационные модели в обучении // Системный анализ в науке и образовании. 2010. № 3. URL: <http://www.sanse.ru/archive/17> (дата обращения: 11.03.2017).

10. Ульянов С.В., Петров С.П. Квантовое распознавание лиц и квантовая визуальная криптография: модели и алгоритмы // Системный анализ в науке и образовании. 2012. № 1. URL: <http://www.sanse.ru/archive/23> (дата обращения: 11.03.2017).

Software & Systems

DOI: 10.15827/0236-235X.119.435-438

Received 20.04.17

2017, vol. 30, no. 3, pp. 435–438

### DEVELOPMENT OF A PATTERN RECOGNITION SYSTEM FOR A MOBILE ROBOT

S.V. Ulyanov, Dr.Sc. (Physics and Mathematics), Professor, [ulyanovsv@mail.ru](mailto:ulyanovsv@mail.ru)

A.G. Reshetnikov, Ph.D, Senior Lecturer, [agreshetnikov@gmail.com](mailto:agreshetnikov@gmail.com)

K.V. Koshelev, Student, [kirill\\_koshelev18@rambler.ru](mailto:kirill_koshelev18@rambler.ru)

<sup>1</sup> Dubna State University, Institute of the system analysis and management, Universitetskaya St. 19, Dubna, 141980, Russian Federation

**Abstract.** The main development source of robotic systems and artificial intelligence systems is improvement of computer technology, devices and sensors that allow obtaining information about the environment. This process is inextricably linked with the development of software on-board processors, which are used in autonomous robot managing. The article gives a brief description of a module of a developed image recognition system for an autonomous robot, which is based on a stereo vision technology.

Nowadays, more and more articles mention the systems consisting of combinations of several video cameras, laser radars, thermal imagers and other sources of information about the environment. Equipment of this class became widespread when creating autopilots for vehicles, security control systems, unmanned aerial reconnaissance, various household robots. Stereo vision allows obtaining data on image depth, the distance to objects. It also provides an opportunity to build a three-dimensional picture of the surrounding world.

The paper describes the configuration of the equipment used (a pair of cameras fixed on a common base): a stereopair calibrating process, the calibration pattern on the image was searched using the ROS framework. Images obtained from cameras are subjected to rectification (a transformation when both images are projected onto a plane parallel to the line that connects the optical centers of the cameras).

The article considers the fundamentals of projective transformations and epipolar geometry. The paper presents the idea that underlies the construction of depth and displacement maps. As a result, it presents a recognition software module developed using the Python programming language.

**Keywords:** stereovision, pattern recognition, Robot Operating System (ROS), control systems, robotics, camera calibration.

### References

1. Kuznetsova A., Sigal L. Expanding object detector's horizon: incremental learning framework for object detection in videos. *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2015*. Available at: <https://www.disneyresearch.com/publication/expanding-object-detectors-horizon/> (accessed March 19, 2017).

2. *Raspoznavanie obrazov dlya programmistov: primery robot* [Image Recognition for Programmers: Examples]. Available at: <http://recog.ru/> (accessed March 9, 2017).

3. Яне Б. *Tsifrovaya obrabotka izobrazheniy* [Digital Image Processing]. Moscow, Tekhnosfera, 2007, 584 p.

4. *Ros.org. Robot Operating System (ROS): documentation, support, tutorials*. Available at: <http://wiki.ros.org/> (accessed February 22, 2017).

5. Szpytko J., Hyla P. Stereovision 3D type workspace mapping system architecture for transport devices. *Jour. of KONES Powertrain and Transport*. 2010, vol. 17, pp. 496–502.

6. Pentland A. P. Depth of scene from depth of field. *Proc. Image Understanding Workshop*. 1982, pp. 253–259.

7. Bobick A.F., Intille S.S. Large occlusion stereo. *Int. Jour. of Computer Vision*. 1999, vol. 33, pp. 181–200.

8. Forsyth D.A., Ponce J. *Computer Vision: a Modern Approach*. Prentice Hall Publ., 2003, 693 p. (Russ.ed.: Moscow, 2004, 928 p.).

9. Ulyanov S.V., Dobrynin V.N., Nefedov N.Yu., Petrov S.P., Polunin A.S., Reshetnikov A.G. Genetic and quantum algorithms. Pt. 1. Innovative models in training. *Sistemny analiz v nauke i obrazovanii* [Jour. of Systems Analysis in Science and Education]. 2010, no. 3. Available at: <http://www.sanse.ru/archive/17> (accessed March 11, 2017).

10. Ulyanov S.V., Petrov S.P. Quantum Face Recognition and Quantum Visual Cryptography: Models and Algorithms. *Sistemny analiz v nauke i obrazovanii* [Jour. of Systems Analysis in Science and Education]. 2012, no. 1. Available at: <http://www.sanse.ru/archive/23> (accessed March 11, 2017).



УДК 004.[62+91]

DOI: 10.15827/0236-235X.119.439-446

Дата подачи статьи: 20.01.17

2017. Т. 30. № 3. С. 439–446

## **ФРЕЙМВОРК ДЛЯ АНАЛИЗА И ПРОГНОЗИРОВАНИЯ ВРЕМЕННЫХ РЯДОВ ПРИ РАЗРАБОТКЕ КОМПОНЕНТ ПРОАКТИВНЫХ СИСТЕМ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ**

*М.В. Шербаков, д.т.н., главный научный сотрудник, maxim.shcherbakov@vstu.ru;*

*К.С. Задиран, студент, konstantin.zadiran@gmail.com;*

*А.В. Голубев, аспирант, ax.golubev@gmail.com;*

*Аль-Гунаид Моххаммед Амин, преподаватель, mohammadalgunaid@gmail.com*

*(Волгоградский государственный технический университет,  
просп. Ленина, 28, г. Волгоград, 400005, Россия)*

В статье описан разработанный фреймворк для автоматизации исследований в области проектирования математического и ПО проактивных систем поддержки принятия решений. В частности, рассматривается проблема анализа и прогнозирования временных рядов для формирования компонент автоматизации прогнозирования различных процессов.

На практике для прототипирования компонент используются различные библиотеки анализа данных (на языках R, Python). Основная проблема – отсутствие или недостатки реализации методики анализа, то есть последовательности действий при анализе. Кроме этого, следует выделить ряд сдерживающих факторов, влияющих на эффективность проектирования компонент анализа и прогнозирования: рутинные операции при ручном анализе временных рядов, отнимающие много времени разработчика; недостаточная квалификация разработчика как сдерживающий фактор реализации компонент проактивных систем; зачастую необходимость анализа большого числа однотипных данных.

Предлагаемое программное решение, представляющее собой ПО на языке Python, позволяет автоматизировать процесс анализа временных рядов и формировать отчет в формате LaTeX. Формирование осуществляется в автоматическом режиме в соответствии с методологией CRISP-DM. Отчет содержит результат комплексного анализа временных рядов в соответствии с опубликованными и признанными методиками.

Все функции фреймворка можно отнести к одной из следующих групп: функции загрузки данных и формирование внутреннего фрейма данных; функции дескриптивного анализа временных рядов с визуализацией; функции прогнозирования и функции формирования отчета. Показан пример использования фреймворка для решения задачи поддержки принятия проактивных управленческих решений в системах энергетического менеджмента (прогнозирование потребления электроэнергии).

**Ключевые слова:** проактивные системы, программный фреймворк, анализ временных рядов, прогнозирование временных рядов, визуализация данных.

Низкая степень автоматизации бизнес-процессов все еще остается серьезной проблемой для производства и бизнеса, а следовательно, и для разработчиков автоматизированных систем и системных интеграторов [1]. Развитие современных подходов и методов анализа данных позволяет расширить область объектов автоматизации, автоматизировать операции, связанные с принятием решений (поддержка принятия решений), повышая уровень зрелости бизнес-процессов предприятий.

Одним из направлений автоматизации является совершенствование механизмов прогнозирования развития ситуации и оценка риска возникновения неблагоприятных исходов. В этом случае рассматривается новый класс систем поддержки принятия решений – проактивные системы принятия решений. Эти системы позволяют выявлять проактивные ситуации, требующие внимания или вмешательства заинтересованных лиц. В результате вырабатываются и принимаются превентивные меры для минимизации риска возникновения неблагоприятной ситуации [2].

В основе подобных систем лежат модели и методы прогнозирования временных рядов. Разработку ПО, выполняющего функции прогнозирования, целесообразно выносить в отдельные неболь-

шие проекты, в которых осуществляется быстрое прототипирование решений с использованием современных средств и инструментов анализа данных. Несмотря на большое количество инструментов для проведения анализа временных рядов, например, язык программирования R или Python с его набором библиотек для работы с данными, остается проблема, связанная с реализацией методики построения модели прогнозирования. В зависимости от квалификации разработчика сроки реализации проекта могут существенно варьироваться. Выделим ряд факторов, имеющих большое влияние при решении задачи анализа и прогнозирования: ручной анализ временных рядов, отнимающий время разработчика; недостаточная квалификация разработчика как сдерживающий фактор реализации систем; зачастую необходимость анализа большого числа однотипных данных.

В качестве решения может быть предложен фреймворк (<https://github.com/zadiran/Dataworks>) для проведения исследований при анализе и прогнозировании временных рядов согласно методологии CRISP-DM [3]. В статье предлагается программное решение, позволяющее автоматизировать процессы анализа временных рядов и формирования отчетов в формате LaTeX.

Авторы исследовали работы, в которых описаны методики для анализа и прогнозирования временных рядов [4–7]. Это позволило выделить две основные проблемы: исходный код для представленных методов или отсутствует, или не всегда понятен разработчику.

При работе с временными рядами для анализа и прогнозирования можно использовать разные пакеты, на приобретение которых требуются средства. Для работы с ними необходима соответствующая квалификация [8–16]. Немаловажным фактором использования ПО является возможность реализации своих методов. Интеграция существующего ПО в собственные решения ограничена лицензией.

Многие разработчики и аналитики обычно реализуют свои собственные наработки. На это затрачивается время, которое можно было бы использовать более эффективно.

В последнее время все большее распространение получает язык программирования R, который был создан для статистической обработки данных и работы с графикой. Для него имеется большое количество пакетов [17], расширяющих функциональные возможности по обработке и прогнозированию временных рядов [18, 19]. Одним из них является *forecast* [20], в котором реализованы основные модели, используемые для прогнозирования.

Помимо R, большой популярностью пользуется Python. Это высокоуровневый язык программирования общего назначения, ориентированный на повышение производительности разработчика и читаемости кода, а также имеющий минималистичный синтаксис. Репозиторий насчитывает порядка 90 000 пакетов [21], среди которых можно выделить такие, как *pandas* [22, 23], *matplotlib* [24] и *numpy* [25], являющиеся базовыми для любого исследователя.

### Описание фреймворка

Основная цель фреймворка – уменьшение времени на первичный анализ временных рядов для дальнейшего построения модели прогнозирования. От пользователя (аналитика данных) требуется задать минимум значений параметров. Структуру фреймворка определяют следующие сущности.

- ***data\_provider*** – абстрактный класс, предоставляющий доступ к источнику данных:

***get\_data(string[] = None) : DataFrame*** – метод, используемый для получения всего набора данных; в качестве параметра принимает список столбцов, для которых требуется получить данные (если параметр не указан, метод возвращает данные для всех столбцов);

***get\_training\_set(string[] = None) : DataFrame*** – метод, используемый для получения тренировочного набора данных; в качестве параметра принимает список столбцов, для которых требуется полу-

чить данные (если параметр не указан, возвращает данные для всех столбцов);

***get\_testing\_set(string[] = None) : DataFrame*** – метод, используемый для получения тестового набора данных; в качестве параметра принимает список столбцов, для которых требуется получить данные (если параметр не указан, возвращает данные для всех столбцов);

***load\_data()*** – метод, используемый для загрузки данных из источника в оперативную память (данные во фреймворке загружаются только по требованию, а не при создании экземпляра *data\_provider*);

***divide\_data()*** – метод, используемый для разделения набора данных на тренировочный и тестовый наборы;

***try\_separate\_timestamp()* : dict** – метод для автоматического выделения колонок с датами и формирования статистики по выделенным столбцам.

- ***csv\_data\_provider*** – реализация класса *data\_provider* для CSV-файлов.

- ***visualizer*** – класс, предоставляющий API для построения графиков.

- ***forecast\_handler*** – класс, предоставляющий API для прогнозирования:

***get\_benchmark\_result()* : dict** – метод получения результатов прогнозирования для бенчмарк-модели;

***get\_error\_measurements(dict) : dict*** – метод расчета значений показателей ошибок для всех столбцов в словаре, переданном в качестве параметра;

***get\_error\_measurements\_for(dict, str) : dict*** – метод расчета значения;

***grid\_search()* : dict** – метод, используемый для осуществления *grid search*.

- ***report\_generator*** – абстрактный класс, который предоставляет API для генерации отчетов:

***data\_loading(bool = None)*** – метод, генерирующий отчет для этапа загрузки данных; параметр отвечает за добавление результата в файл с отчетом или его перезапись;

***charts(bool = None)*** – метод, генерирующий графики по данным: временной ряд, гистограмма, агрегация, графики по времени дня; параметр отвечает за добавление результата в файл с отчетом или его перезапись;

***grid\_search(bool = None)*** – метод, генерирующий отчет по результату прогнозирования; параметр отвечает за добавление результата в файл с отчетом или его перезапись.

- ***latex\_report\_generator*** – реализация класса *report\_generator* для генерации отчетов в формате LaTeX.

Данная структура графически представлена на рисунке 1.

Все функции фреймворка можно отнести к одной из следующих групп:

- загрузка данных и формирование внутреннего фрейма данных;

- дескриптивный анализ временных рядов с визуализацией;
- прогнозирование;
- формирование отчета.

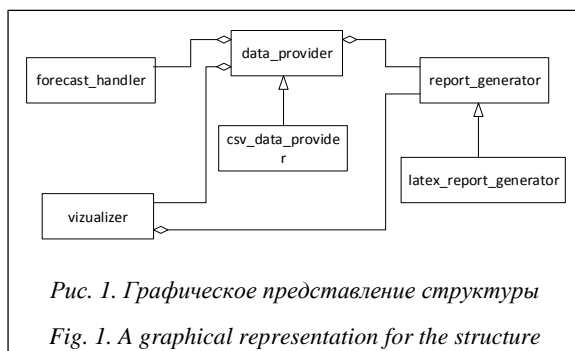


Рис. 1. Графическое представление структуры

Fig. 1. A graphical representation for the structure

**Загрузка данных и формирование внутреннего фрейма данных.** Для загрузки данных пользователь задает значения параметров `<data_provider_parameters, forecast_handler_params, vis_params>`.

Параметр `data_provider_parameters` включает:  
`'filepath'` – имя файла с путем;  
`'testing_set_size'` – размер выборки для тестирования.

Параметр `forecast_handler_params` включает:  
`forecasting_horizon` – опциональный параметр, горизонт прогнозирования (задание параметров сезонности (частоты));  
`forecasting_columns` – список столбцов, для которых выполняется прогнозирование.

Параметр `vis_params` включает:  
`'output_directory'` – путь к директории с отчетом и картинками;

`'timestamp_column'` – имя столбца, содержащего отметки дата-время;

`'timestamp_parameters'` – параметры для обработки колонок, при автоматическом определении колонок с отметками даты-времени; включает логические параметры `'dayfirst'` и `'yearfirst'`, указывающие положение дня и года в строке с датой (в случае, если этот параметр не указан, значения параметров подбираются исходя из оптимального варианта – наименьшего количества различных интервалов между соседними отметками даты-времени функции прогнозирования).

Пример кода по заданию параметров прогнозирования:

```
from framework.data import csv_data_provider as dp
from framework.forecasting import forecast_handler
```

```
# create data provider
data_provider_parameters = {
    'filepath': 'data/demo1.csv',
    'testing_set_size': 96
}
data_provider = dp(data_provider_parameters)
```

```
# create forecast handler
forecast_handler_params = {
```

```
'forecasting_columns':
['Open','High','Low','Close','Volume'],
'forecast_horizon': 96
}
```

```
fh = forecast_handler(data_provider, forecast_handler_params, benchmark_model, models, error_measurements)
```

Далее пользователь задает имя временной колонки, по которой фреймворк может произвести загрузку в `dataframe`, выделить столбцы для прогнозирования, выявить пропуски, произвести индексацию по временным меткам.

**Дескриптивный анализ временных рядов с визуализацией.** Фреймворк загружает данные и формирует отчет о загрузке, который включает следующую информацию:

- ошибки загрузки (если имеются);
- список колонок в файле с загрузкой;
- число записей;
- детальная информация по каждой колонке, имеющей значения, представленные в числовом виде: минимальное значение, максимальное значение, среднее, медиана;
- информация о столбце с временной меткой (если можно выделить автоматически): дата начала наблюдений, дата окончания наблюдений, интервалы наблюдений.

За визуализацию данных во фреймворке отвечает модуль `visualizer.py`.

Пример кода для визуализации данных:  
**from** framework.data **import** csv\_data\_provider **as** dp  
**from** framework.visualization **import** visualizer **as** vs

```
# create data provider
data_provider_parameters = {
    'filepath': 'data/demo1.csv', # path to data
    'testing_set_size': 96, # size of testing set
    'missing_values_processing_policy': None # missing values processing method
}
data_provider = dp(data_provider_parameters)

# plot charts
vis_params = {
    'output_directory': 'report/images/', # directory, where charts will be placed
    'timestamp_column': 'DateTime', # timestamp column
    'timestamp_parameters': {'dayfirst': True} # parameters for parsing timestamp
}
v = vs(vis_params, data_provider)
v.plot_everything()
```

Модуль поддерживает построение временных рядов, построение гистограмм, построение агрегированных данных по дням, неделям, месяцам (группировка данных по часам).

**Прогнозирование.** Задание списка моделей кандидатов с параметрами осуществляется в списке `models` словарями с параметрами. Представим задание областей изменения параметров моделей кандидатов:

```
models = [
```

```

{ 'model' : naive_model() },
{ 'model' : random_model() },
{ 'model' : moving_average_model(),
  'parameters' : [
    model_param('lookback', 7, (4, 10), 1),
    model_param('lag', 30, None, None)
  ]
},
{ 'model' : simple_average_model(),
  'parameters' : [
    model_param('frequency', 365, None, None),
    model_param('lag', 2, None, None)
  ]
},
{ 'model' : arma_model(),
  'parameters' : [
    model_param('p', 2, None, None),
    model_param('q', 0, None, None)
  ]
}
]

```

Меры точности задаются списком функций, который передается в *forecast\_handler* вместе с моделями и данными. На данный момент реализованы следующие меры точности: MAE, RMSE, MSE, MdAE, MAPE, MdAPE, RMSPE, RMdSPE, MASE, MdASE, RMSSE, SMAPE, SMdAPE [26].

Использование grid search для выбранных моделей с мерами точности:

```

from framework.data import csv_data_provider as dp
from framework.visualization import visualizer as vs

from framework.forecasting import naive_model, random_model

from framework.measurement.absolute import mean_absolute_error as mae
from framework.measurement.absolute import root_mean_square_error as rmse
from framework.measurement.absolute import mean_square_error as mse

from framework.measurement.absolute import median_absolute_error as mdac

from framework.forecasting import forecast_handler

# create data provider
data_provider_parameters = { ... }
data_provider = dp(data_provider_parameters)

# create charts builder
vis_params = { ... }
v = vs(vis_params, data_provider)

# estimate models
naive = {'model': naive_model(), 'parameters': None}
forecast_handler_params = { ... }

# error measurements
err_m = [mae(), rmse(), mse(), mdac()]

models = [
    {'model': naive_model()},
    {'model': random_model()}
]

# create forecast handler

```

```

fh = forecast_handler(data_provider, forecast_handler_params, naive, models, err_m)

```

```

# get a benchmark result

```

```

bench_result = fh.get_benchmark_result()

```

```

# get error measurements for benchmark model

```

```

bench_measurements = fh.get_error_measurements(bench_result)

```

```

# perform grid search

```

```

grid_search_result = fh.grid_search()

```

Далее для получения прогноза по выбранным моделям используется метод *grid search*, в котором реализован цикл по моделям и параметрам. *Grid search* реализован следующим образом: на верхнем уровне идет цикл по колонкам данных, уровнем ниже – цикл по используемым моделям для прогнозирования и на самом нижнем уровне – цикл по параметрам. Расчет оценки точности производится методом *get\_error\_measurements*.

**Формирование отчета.** На данный момент формирование результирующего отчета реализовано с использованием формата LaTeX. Представим код, отвечающий за данный процесс формирования отчета:

```

from framework.data import csv_data_provider as dp
from framework.reporting import latex_report_generator
from framework.visualization import visualizer as vs

# create data provider
data_provider_parameters = {
    'filepath': 'data/demo1.csv',
    'testing_set_size': 96
}
data_provider = dp(data_provider_parameters)

# create charts builder
vis_params = {
    'output_directory': 'report/images/', # directory, where
    charts will be placed
    'timestamp_column': 'DateTime', # timestamp column
    'timestamp_parameters': {'dayfirst': True} # parameters
    for parsing timestamp column
}
v = vs(vis_params, data_provider)

generator_parameters = {
    'path_to_report': 'report/body.tex', # file, where to write
    report
    'include_timestamp_info': True # if true, try to find
    timestamp columns automatically
}

# create latex report generator
rg = latex_report_generator(generator_parameters,
    data_provider, v)

# generate report
rg.data_loading()
rg.charts(True)

```

В корне репозитория с кодом доступны файлы с примерами использования данного фреймворка. Тестовые примеры описывают основной функционал фреймворка и разделены на несколько файлов:

- *example\_load\_data.py* представляет информацию по загрузке данных из демонстрационного примера;
- *example\_grid\_search.py* выполняет работу по формированию прогноза и оценке точности используемой модели;
- *example\_create\_charts.py* представляет функционал по визуализации данных;
- *example\_generate\_report.py* описывает работу по генерации отчета в формате LaTeX, включая все предыдущие пункты.

### Пример использования

Настоящий фреймворк использовался для анализа временных рядов о потреблении электроэнергии в офисных зданиях [27]. Анализ проводился с целью повышения эффективности энергетического менеджмента за счет внедрения принципов проактивного управления [28]. В частности, требуется разработать модели прогнозирования для планирования бюджета на потребление энергии и выявления неэффективных режимов работы. Несмотря на

существование алгоритмов автоматического прогнозирования временных рядов [19, 29], процедура совершенствования моделей прогнозирования осуществляется аналитиком данных.

Данные содержатся в CSV-файле, который представляет собой результат выгрузки из SCADA системы EcoSCADA [27]. По умолчанию в выгрузке содержатся данные по всем измерениям для здания.

На рисунке 2 представлены графики, полученные после визуализации данных.

Для генерации отчета пользователь задает параметры провайдера данных, параметры *forecast\_handler*, параметры визуализации данных, используемые модели и меры точности, параметры генератора отчетов.

Выполнив скрипт *example\_generate\_report.py*, получим отчет в формате LaTeX, который необходимо собрать вручную, используя соответствующую среду, например TexLive. В результате получаем документ *report.pdf*.

Формируется отчет из 103 страниц, включающий секции Data Loading, Charts и Forecast result.

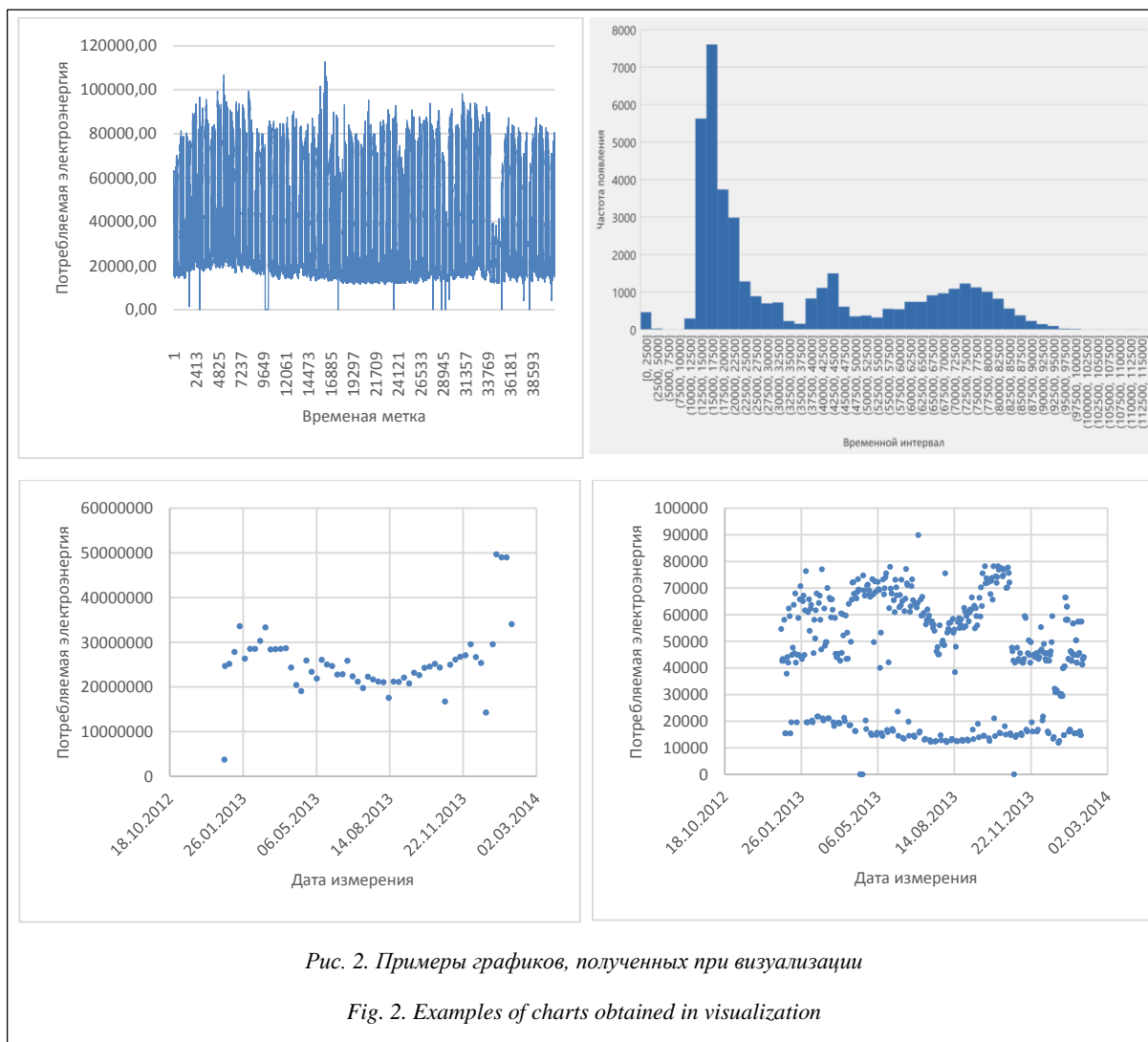


Рис. 2. Примеры графиков, полученных при визуализации

Fig. 2. Examples of charts obtained in visualization

## Naive model

Forecast horizon: 96

Model doesn't have any parameters



Testing set forecast



Forecast



Testing set forecast absolute error distribution

	RMSE	MSE	MdAE	MAE
No parameters	8019.03984277	64305000.0	2100.0	5181.25

Рис. 3. Страница из отчета по результату прогнозирования с использованием наивной модели

Fig. 3. A page from the report on a naive model forecasting result

На рисунке 3 представлена страница из отчета прогнозирования временного ряда с использованием наивной модели.

### Заключение

Для решения задачи автоматического анализа временных рядов, данные которых представлены в формате таблиц (в частности, в текстовом формате с разделителями запятых), предложено программное решение в виде фреймворка на языке Python с использованием модульной архитектуры. Основное назначение – формирование отчета о результатах анализа временных рядов в формате LaTeX.

Все функции фреймворка разделены на определенные группы. В результате исследователь получает возможность вести журнал исследований (экспериментов), получать отчет в автоматическом режиме, адаптировать отчет под его нужды, а также формировать пояснительную записку к техническому заданию в автоматическом режиме. Фреймворк также предусматривает возможность расши-

рения функциональной составляющей, то есть исследователь может работать и с другими форматами, а также может дополнить его необходимыми методами для анализа и прогнозирования временных рядов.

Работа частично поддержана РФФИ (проект № 16-37-60066\_мол\_дк) и грантом Президента РФ (проект № МД-6964.2016.9).

### Литература

1. Muller P. The CIO of 2020. The future of the chief information officer. URL: <http://www.slideshare.net/HPESoftwareSolutions/the-cio-of-2020> (дата обращения: 18.01.2017).
2. Engel Y., Etzion O. and Feldman Z. 2012. A basic model for proactive event-driven computing. Proc. 6th ACM Intern. Conf. on Distributed Event-Based Systems (DEBS '12). ACM, NY, USA, 2012, pp. 107–118. DOI: 10.1145/2335484.2335496.
3. IBM SPSS Modeler CRISP-DM Guide. URL: [ftp://public.dhe.ibm.com/software/analytics/spss/documentation/modeler/14.2/en/CRISP\\_DM.pdf](ftp://public.dhe.ibm.com/software/analytics/spss/documentation/modeler/14.2/en/CRISP_DM.pdf) (дата обращения: 18.01.2017).
4. Hyndman R.J., Athanasopoulos G. Forecasting: principles and practice. Paperback. October 17, 2013. URL: <https://www.otexts.org/fpp> (дата обращения: 18.01.2017).
5. Yarushev S.A., Averkin A.N. Review of studies on time se-

ries forecasting based on hybrid methods, neural networks and multiple regression // Программные продукты и системы. 2016. № 1. С. 75–82 (англ.).

6. Лесик И.А. Решение задачи прогнозирования с использованием нейронных сетей прямого распространения на примере построения прогноза роста курса акций // Программные продукты и системы. 2015. № 2. С. 70–74.

7. Kovalev S.M., Sukhanov A.V. Anomaly detection based on Markov chain model with production rules // Программные продукты и системы. 2014. № 3. С. 40–43 (англ.).

8. STATISTICA. URL: <http://statsoft.ru/> (дата обращения: 18.01.2017).

9. EViews. URL: <http://www.eviews.com/home.html> (дата обращения: 18.01.2017).

10. SPSS Statistics. URL: <https://www-03.ibm.com/software/products/ru/spss-statistics> (дата обращения: 18.01.2017).

11. SAS. URL: <http://www.sas.com> (дата обращения: 18.01.2017).

12. STATA. URL: <https://www.stata.com/> (дата обращения: 18.01.2017).

13. MathWorks Matlab. URL: <https://www.mathworks.com/> (дата обращения: 18.01.2017).

14. Wolfram Mathematica. URL: <https://www.wolfram.com/mathematica/> (дата обращения: 18.01.2017).

15. Scilab. URL: <http://www.scilab.org/> (дата обращения: 18.01.2017).

16. MAXIMA. URL: <http://maxima.sourceforge.net/ru/> (дата обращения: 18.01.2017).

17. The Comprehensive R Archive Network. URL: <https://cran.r-project.org/> (дата обращения: 18.01.2017).

18. Tavish Srivastava. A complete tutorial on time series modeling in R. URL: <https://www.analyticsvidhya.com/blog/2015/12/complete-tutorial-time-series-modeling/> (дата обращения: 18.01.2017).

19. Rob J. Hyndman, Yeasmin Khandakar. Automatic time series forecasting: the forecast package for R, Jour. of Statistical Software, 2008, vol. 27, no. 1, pp. 1–22.

20. Rob J. Hyndman. R package ‘forecast’. URL: <https://cran.r-project.org/web/packages/forecast/forecast.pdf> (дата обращения: 18.01.2017).

21. The Python Package Index. URL: <https://pypi.python.org/pypi> (дата обращения: 18.01.2017).

22. Aarshay Jain. A comprehensive beginner’s guide to create a Time Series Forecast. URL: <https://www.analyticsvidhya.com/blog/2016/02/time-series-forecasting-codes-python> (дата обращения: 18.01.2017).

23. Python Data Analysis Library. URL: <http://pandas.pydata.org/> (дата обращения: 18.01.2017).

24. Matplotlib. URL: <http://matplotlib.org/> (дата обращения: 18.01.2017).

25. NumPy. URL: <http://www.numpy.org/> (дата обращения: 18.01.2017).

26. Shcherbakov M.V., Brebels A., Shcherbakova N.L., Tyukov A.P., Janovsky T.A., and Kamaev V.A. A survey of forecast error measures. World Applied Sciences Jour., 2013, vol. 24, iss. 24, pp. 171–176.

27. Kamaev V.A., Scherbarov M.V., Panchenko D.P., Scherbakova N.L., Brebels A. Using connectionist systems for electric energy consumption forecasting in shopping centers. Automation and Remote Control, 2012, vol. 73, no. 6, pp. 1075–1084.

28. Tennenhouse D.L. Proactive computing. Communications of the ACM, 2000, vol. 43, pp. 43–50.

29. Golubev A., Shcherbakov M., Shcherbakova N., Kamaev V. Automatic multi-steps forecasting method for multi seasonal time series based on symbolic aggregate approximation and grid search approaches. Jour. of Fundamental and Applied Sciences, 2016, vol. 8, no. 3S, pp. 2429–2441.

Software & Systems

DOI: 10.15827/0236-235X.119.439-446

Received 20.01.17

2017, vol. 30, no. 3, pp. 439–446

# A FRAMEWORK FOR ANALYSIS AND FORECASTING OF TIME SERIES IN THE DEVELOPMENT OF PROACTIVE DECISION SUPPORT SYSTEM COMPONENTS

**M.V. Shcherbakov**<sup>1</sup>, Dr.Sc. (Engineering), Chief Researcher, [maxim.shcherbakov@vstu.ru](mailto:maxim.shcherbakov@vstu.ru)

**K.S. Zadiran**<sup>1</sup>, Student, [konstantin.zadiran@gmail.com](mailto:konstantin.zadiran@gmail.com)

**A.V. Golubev**<sup>1</sup>, Postgraduate Student, [ax.golubev@gmail.com](mailto:ax.golubev@gmail.com)

**Al-Gunaid Mohammed Amin**<sup>1</sup>, Lecturer, [mohammadalgunaid@gmail.com](mailto:mohammadalgunaid@gmail.com)

<sup>1</sup> Volgograd State Technical University, Lenin Ave. 28, Volgograd, 400005, Russian Federation

**Abstract.** The article describes the developed framework for automation research on software design for decision support proactive systems. In particular, it considers the problem of analysis and forecasting time series to form automation components in predicting various processes.

In practice, there are various data analysis libraries (R, Python) used for prototyping the component uses. The main problem is a lack (or insufficiency) of implementation of analysis methods, i.e. the sequence of actions during analysis. In addition, there is a number of limiting factors that affect the efficiency of analysis and forecasting components design: (i) time-consuming routine operations during time series manual analysis; (ii) developer’s insufficient qualifications as a deterrent in implementing proactive system components; (iii) a frequent need in the analysis of a large number of similar data.

The proposed software solution (a software in Python) automates the process of time series analysis and generate a LaTeX report. Formation is automatic in accordance with CRISP-DM methodology. The report provides a comprehensive analysis of time series in accordance with published and accepted methods.

All functions of the framework can be attributed to one of the following groups: data loading and the internal data frame; descriptive time series analysis with visualization; prediction and report making. The paper shows an example of using a

framework to solve the problem of proactive management decisions making support in energy management systems (forecasting of electricity consumption).

**Keywords:** proactive systems, software framework, time series analysis, time series forecasting, data visualization.

**Acknowledgements.** The work has been partially supported by RFBR (project no. 16-37-60066\_мол\_ош) and a grant of the President of the Russian Federation (project no. МД-6964.2016.9).

### References

1. Muller P. *The CIO of 2020: The future of the Chief Information Officer*. Available at: <http://www.slideshare.net/HPE-SoftwareSolutions/the-cio-of-2020> (accessed January 18, 2017).
2. Engel Y., Etzion O., Feldman Z. A basic model for proactive event-driven computing. *Proc. 6th ACM Int. Conf. on Distributed Event-Based Systems (DEBS '12)*. ACM, NY, USA, 2012, pp. 107–118.
3. *IBM SPSS Modeler CRISP-DM Guide*. Available at: [ftp://public.dhe.ibm.com/software/analytics/spss/documentation/modeler/14.2/en/CRISP\\_DM.pdf](ftp://public.dhe.ibm.com/software/analytics/spss/documentation/modeler/14.2/en/CRISP_DM.pdf) (accessed January 18, 2017).
4. Hyndman R.J., Athanasopoulos G. *Forecasting: Principles and Practice*. 2013. Available at: <https://www.otexts.org/fpp> (accessed January 18, 2017).
5. Yarushev S.A., Averkin A.N. Review of studies on time series forecasting based on hybrid methods, neural networks and multiple regression. *Programmnye produkty i sistemy* [Software & Systems]. 2016, no. 1 (113), pp. 75–82.
6. Lesik I. A. Meeting the challenge of forecasting using neural networks of direct distribution by the example of the construction of the forecast growth rate of shares. *Programmnye produkty i sistemy* [Software & Systems]. 2015, no. 2 (110), pp. 70–74 (in Russ.).
7. Kovalev S.M., Sukhanov A.V. Anomaly detection based on Markov chain model with production rules. *Programmnye produkty i sistemy* [Software & Systems]. 2014, no. 3 (107), pp. 40–43.
8. STATISTICA. Available at: <http://statsoft.ru/> (accessed January 18, 2017).
9. EViews. Available at: <http://www.eviews.com/home.html> (accessed January 18, 2017).
10. SPSS Statistics. Available at: <https://www-03.ibm.com/software/products/ru/spss-statistics> (accessed January 18, 2017).
11. SAS. Available at: <http://www.sas.com> (accessed January 18, 2017).
12. STATA. Available at: <https://www.stata.com/> (accessed January 18, 2017).
13. MathWorks Matlab. Available at: <https://www.mathworks.com/> (accessed January 18, 2017).
14. Wolfram Mathematica. Available at: <https://www.wolfram.com/mathematica/> (accessed January 18, 2017).
15. Scilab. Available at: <http://www.scilab.org/> (accessed January 18, 2017).
16. MAXIMA. Available at: <http://maxima.sourceforge.net/ru/> (accessed January 18, 2017).
17. The Comprehensive R Archive Network. Available at: <https://cran.r-project.org/> (accessed January 18, 2017).
18. Tavish Srivastava. *A Complete Tutorial on Time Series Modeling in R*. Available at: <https://www.analyticsvidhya.com/blog/2015/12/complete-tutorial-time-series-modeling/> (accessed January 18, 2017).
19. Hyndman R.J., Khandakar Y. Automatic Time Series Forecasting: The forecast Package for R. *Jour. of Statistical Software*. 2008, vol. 27, no. 1, pp. 1–22.
20. Hyndman R.J. *R Package 'forecast'*. Available at: <https://cran.r-project.org/web/packages/forecast/forecast.pdf> (accessed January 18, 2017).
21. The Python Package Index. Available at: <https://pypi.python.org/pypi> (accessed January 18, 2017).
22. Aarshay Jain. *A comprehensive beginner's guide to create a Time Series Forecast*. Available at: <https://www.analyticsvidhya.com/blog/2016/02/time-series-forecasting-codes-python> (accessed January 18, 2017).
23. Python Data Analysis Library. Available at: <http://pandas.pydata.org/> (accessed January 18, 2017).
24. Matplotlib. Available at: <http://matplotlib.org/> (accessed January 18, 2017).
25. NumPy. Available at: <http://www.numpy.org/> (accessed January 18, 2017).
26. Shcherbakov M.V., Brebels A., Shcherbakova N.L., Tyukov A.P., Yanovsky T.A., Kamaev V.A. A Survey of Forecast Error Measures. *World Applied Sciences Jour. (WASJ)*. 2013, vol. 24, spec. iss. 24: Information Technologies in Modern Industry, Education & Society, pp. 171–176.
27. Kamaev V.A., Shcherbakov M.V., Panchenko D.P., Shcherbakova N.L., Brebels A. Using connectionist systems for electric energy consumption forecasting in shopping centers. *Automation and Remote Control*. 2012, vol. 73, no. 6, pp. 1075–1084.
28. Tennenhouse D. Proactive computing. *Communications of the ACM*. 2000, vol. 43, pp. 43–50.
29. Golubev A.V., Shcherbakov M.V., Shcherbakova N.L., Kamaev V.A. Automatic multi-steps forecasting method for multi seasonal time series based on symbolic aggregate approximation and grid search approaches. *Jour. of Fundamental and Applied Sciences*. 2016, vol. 8, no. 3, pp. 2429–2441.



УДК 004.5

DOI: 10.15827/0236-235X.119.447-455

Дата подачи статьи: 20.01.17

2017. Т. 30. № 3. С. 447–455

## **СОВРЕМЕННЫЕ ТЕНДЕНЦИИ В АВТОМАТИЗИРОВАННОЙ ОЦЕНКЕ ЮЗАБИЛИТИ И ПОВЕДЕНЧЕСКИЕ ФАКТОРЫ В АЛГОРИТМАХ ПОИСКОВЫХ СИСТЕМ**

*М.А. Бакаев, к.т.н., старший научный сотрудник, taxis81@gmail.com  
(Новосибирский государственный технический университет,  
просп. К. Маркса, 20, г. Новосибирск, 630073, Россия)*

В настоящее время сохраняются неопределенность относительно количественного измерения юзабилити, а также ненужное противопоставление его объективных составляющих (продуктивности и эффективности) и субъективного аспекта удовлетворенности пользователя.

В данной обзорной статье отмечается, что практическая цель заключается не столько в измерении юзабилити веб-сайта, сколько в улучшении качества его использования или его оценки со стороны соответствующих алгоритмов поисковых систем, которые все более повышают важность поведенческих факторов в ранжировании. Описаны традиционные методы оценки и проектирования юзабилити веб-сайтов и современные инструменты, вносящие в них элементы автоматизации.

В рамках автоматизированной оценки юзабилити выделены подходы, основанные на взаимодействиях, метриках и моделировании, указаны возможности и ограничения каждого из них. Кроме того, дан краткий обзор гибридных подходов, которые все в большей степени используют методы искусственного интеллекта и машинного обучения. Так, эффективным методом для оценки качества взаимодействия с веб-сайтом являются нейронные сети, применяемые, в частности, в рамках эволюционных алгоритмов проектирования для вычисления фитнес-функции для решений-кандидатов. Описаны методы автоматизированной оценки юзабилити сайтов со стороны глобальных поисковых систем (Яндекс, Гугл), приводится перечень некоторых значимых поведенческих факторов. Фактически поисковые системы рассматривают юзабилити как «черный ящик» и активно применяют методы машинного обучения для аппроксимации его оценки.

Таким образом, концептуализация и практическая оценка юзабилити со стороны поисковых систем, которые в значительной степени формируют веб-среду на сегодняшний день, постоянно эволюционируют и требуют соответствующих гибких и адаптивных подходов со стороны веб-дизайнеров и разработчиков.

**Ключевые слова:** человеко-компьютерное взаимодействие, проектирование юзабилити, веб-дизайн, искусственный интеллект, поисковая оптимизация.

В настоящее время все меньше организаций стремятся лишь создать сайт: большинство из них понимают, что важнее то, сколько потенциальных клиентов посещают сайт и с какой целью. Это, пожалуй, лучше всего отражает формула, предложенная Я. Нильсеном:  $B \sim V * C * L$ , где  $B$  – бизнес-эффект от сайта;  $V$  – количество уникальных посетителей;  $C$  – показатель конверсии (превращение посетителей в клиентов);  $L$  – уровень лояльности (отражает повторные визиты, средний размер заказа и т.п.) [1]. Традиционно считается, что юзабилити (здесь и далее этим термином будем обозначать эргономичность, удобство использования) веб-сайта влияет на  $C$  и  $L$ , хотя для лояльности эффект не совсем прямой и касается примерно 1/3 изменений этого показателя [2]. В наши дни, однако, юзабилити влияет и на  $V$ , поскольку поисковые системы начали учитывать так называемые поведенческие факторы в ранжировании веб-сайтов. Таким образом, неудивительно, что интерес к юзабилити растет со стороны не только дизайнеров-практиков, но и исследователей.

Понятие «юзабилити» внесено в стандарты относительно давно, например, в ISO 9241-11 (1998) оно определяется как степень, с которой продукт может быть использован определенными пользователями при определенном контексте использования для достижения определенных целей с долж-

ной эффективностью, продуктивностью и удовлетворенностью. Юзабилити – это качественное понятие, специфичное для пользователей, их задач (целей) и контекста взаимодействия. Продуктивность означает полноту и точность достижения пользователями своих целей и, как правило, измеряется следующими параметрами:

- соотношение между успешными и ошибочными действиями пользователя;
- доля используемого функционала и команд (чем выше, тем лучше);
- когнитивная нагрузка на пользователя, выражающаяся в потреблении ресурсов памяти, внимания, воображения и т.д. (чем меньше, тем лучше).

Из этих трех параметров когнитивная нагрузка является, пожалуй, наименее очевидной, и, соответственно, нет общепринятого подхода к ее измерению. Так, в [3] в ходе сравнения результатов в трех различных условиях обучения был введен индекс когнитивной нагрузки и соотношен с относительной продуктивностью. Некоторые исследовательские работы в этой области основывались на показателях жизнедеятельности организма человека, таких как кровяное давление или движение глаз и реакция зрачков [4], хотя в рамках таких подходов трудно различить когнитивные и физические аспекты нагрузки. В других исследованиях время

рассматривается как показатель производительности и отражение когнитивной сложности задачи (см. [5]).

Эффективность (минимизация объема ресурсов для выполнения задачи) для веб-сайтов обычно понимается как скорость, с которой пользователи могут выполнить свои задачи. Соответственно, декомпозиция может быть выполнена следующим образом:

- изучаемость (легкость, с которой пользователи могут начать выполнение своих задач на незнакомом сайте, или время, необходимое для того, чтобы научиться им пользоваться; вообще пользователи редко тратят время на изучение работы с конкретным веб-сайтом, поскольку их много);
- время на выполнение задачи;
- ошибки (сколько ошибок допускает пользователь, насколько они серьезны, сколько времени необходимо для их исправления и т.д.);
- запоминаемость (насколько легко можно вернуться к результирующей работе с веб-сайтом после значительного перерыва).

На практике при оценке юзабилити изучаемость и запоминаемость измеряются довольно редко, а начиная с 1990-х годов Я. Нильсен ввел и обосновал подход, называемый «бюджетным юзабилити» и основанный на простых, быстрых и дешевых методах проектирования юзабилити, которые, как правило, не подразумевают количественные измерения.

Наконец, под удовлетворенностью подразумевается то, насколько приятно использовать веб-сайт. Это субъективный эмоциональный аспект, который тоже является специфичным с точки зрения пользователя, задачи и контекста (к примеру, удовлетворенность при редких визитах на сайт электронного магазина и при ежедневной работе с интранет-системой проистекают из различных факторов хотя бы с точки зрения визуального дизайна). Следует заметить, что простота понятия «приятно» во многом обманчива и этот компонент юзабилити на настоящий момент остается в некотором противопоставлении остальным, поскольку юзабилити «для использования» и юзабилити «чтобы было» существенно различаются [2]. Сравнительная важность компонентов юзабилити также не определена (например, может быть только лишь удовлетворенность, то есть то, что пользователь помнит после использования сайта или продукта, имеет значение), и, соответственно, количественное выражение юзабилити остается проблематичным.

В данной статье приведены обзор измерений и оценка юзабилити в настоящее время с использованием автоматизированных и автоматических способов. Охвачен также ряд концептуальных моментов, касающихся юзабилити, поскольку обычно считается, что улучшение какого-либо показателя (а оптимизация человеко-компьютерного взаимо-

действия – цель проектирования юзабилити) вряд ли возможно без понимания того, как он может измеряться. Кроме того, в работе описываются методы оценки юзабилити.

### Традиционные методы оценки юзабилити

Практически любой процесс оценки юзабилити включает в себя сбор и обработку данных о взаимодействии с пользователем (будь оно реальным, совершаемым в ходе тестирования, или же умозрительным), их анализ и интерпретацию результатов, на основании которых вырабатываются рекомендации по улучшению юзабилити. В отличие от показателей эргономичности для интерфейсов человеко-машинных систем, которые зачастую оцениваются количественно в ходе специально поставленных экспериментов (см., например, [6] о применении в авиации), в большинстве веб-проектов применяются подходы на качественном уровне. Традиционные методы проектирования веб-юзабилити, которые начали активно использоваться еще в 1990-х годах, были направлены на повышение конверсии, снижение отскока, улучшение очевидных юзабилити-метрик и общее создание полезного, привлекательного и современного веб-дизайна. Далее описываются основные методы в этой области, классифицируемые в основном по их отношению к триаде пользователь–задача–контекст, и говорится о том, как в настоящее время осуществляется распределение работы между специалистами-людьми и компьютерами.

**Явное и опосредованное наблюдение за пользователями.** Наблюдение за пользователями – пожалуй, наиболее очевидный метод в инструментарии проектировщика юзабилити, поскольку он подразумевает, что реальные пользователи выполняют реальные задачи в реальном контексте, и это практически полностью соответствует определению юзабилити, за исключением возможного эффекта наблюдателя. В отсутствие непосредственного доступа к пользователям наблюдение может производиться с использованием различных существующих инструментов, которые предоставляют визуальное отображение визитов пользователей на сайт, теплокарты кликов и т.д. Например, сервис Яндекс.Вебвизор позволяет просматривать видео поведения пользователей на сайте – с траекториями движения мыши, кликами, навигацией между страницами, выбором текста, прокруткой – таким образом давая множество данных для анализа юзабилити. Существуют также более специализированные инструменты для отображения и подробного анализа теплокарт, такие как FengGui или ClickHeat.

Инструменты для логирования поведения пользователей и анализа логов серверов могут быть отнесены к этой категории, поскольку они фактиче-

ски документируют действия пользователей на сайте. Крупные сервисы веб-аналитики, такие как Яндекс.Метрика или Google.Analytics (GA), предоставляют все больше средств для оценки метрик, связанных с юзабилити, и отслеживания взаимодействий с пользователем (в GA называемых событиями) на бесплатной или условно-бесплатной основе. Так, в GA исследователями [7] был выделен набор метрик, которые наиболее интересны для проектировщиков взаимодействия в сфере электронной коммерции: average page views per visit, percentage of time spent visits, percentage of click depth visits, bounce rate, order conversion rate, average searches per visit, percent of visits using search, search results to site exits ratio, cart start rate, cart completion rate, checkout start rate, checkout completion rate и, наконец, information find conversion rate. Среди отчетов это Audience Overview (Mobile), Social Network Referrals, Goals Overview (Conversions), Path Length (to conversion), Top Events [8]. Однако все инструменты, упомянутые выше, в лучшем случае позволяют частичную автоматизацию, поскольку подразумевают участие юзабилити-экспертов в интерпретации данных, полученных в ходе наблюдения за пользователем, и в выработке рекомендаций по улучшению сайтов.

**Юзабилити-тестирование** – это один из наиболее эффективных в проектировании юзабилити метод [9]. Фактически тестирование проходит в лабораторных условиях, но для его проведения стремятся выбрать участников, максимально похожих на реальных пользователей, и дать им специально разработанные реалистичные задания. Успешность в значительной степени зависит от квалификации юзабилити-эксперта [10], а реалистичность контекста использования остается в большинстве случаев недостижимой.

В связи с развитием широкополосных каналов связи, способных передавать видео в реальном времени, в том числе на основе пиринговых сетей [11], появился ряд сервисов для дистанционного юзабилити-тестирования, таких как Userlytics, OpenHallway и другие. Многие из них помогают с подбором участников (которые могут оставаться в привычной для себя обстановке) и, как правило, способны снизить финансовые и временные затраты, особенно для случаев, когда целевые пользователи рассеяны по миру. Среди минусов – более строгие требования к понятности заданий, предлагаемых пользователям, а также риск набора участников, которые слишком много раз участвовали в юзабилити-тестировании и потому более не похожи на реальных пользователей [12]. Существуют специализированные программные инструменты, способные автоматизировать некоторые аспекты юзабилити-тестирования, такие как запись видео и ввод с клавиатуры и мыши, организация заданий, хранение и организация данных и т.д. (например, ПО Morae компании TechSmith способно делать

многое из перечисленного). Однако ни один из инструментов среди попавших в наше поле зрения не способен осуществлять юзабилити-тестирование без активного участия юзабилити-экспертов, и прежде всего в разработке заданий для участников, качественном анализе результатов оценки юзабилити и формулировании рекомендаций по повышению юзабилити.

**Опросы и обследования.** Несмотря на давно известное первое правило юзабилити, которое гласит: «Не слушайте, что говорят пользователи» (но смотрите, что они делают) [13], опросы пользователей все-таки имеют место в оценке юзабилити. Ранее наиболее распространенным подходом было просто попросить пользователя заполнить форму обратной связи на сайте (например, автоматически показывая ее, когда пользователь собирается закрыть вкладку браузера) в надежде на его добросовестность, но в настоящее время бесплатные или условно-бесплатные сервисы, такие как Usabilityhub, Usabilla, Askusers.ru и др., предлагают иным образом организовать процесс. Владелец веб-сайта или юзабилити-специалист дает ссылку на сайт или скриншот, описывает задания и вопросы, а затем, основываясь на ответах потенциальных пользователей или экспертов, получает оценку юзабилити. Этот метод обычно имеет тенденцию собирать по большей части негативные ответы, но он все же может быть полезен для определения юзабилити-проблем, хотя нет гарантии, что респонденты отражали совокупность целевых пользователей и добросовестно следовали предложенным заданиям.

Еще одна популярная и тоже «непрямая» группа методов – это обследование юзабилити, осуществляемое экспертами, а не пользователями. Юзабилити веб-сайтов проверяется на соответствие некоторым эвристическим рекомендациям, принципам или задачам – в любом случае обследующие должны адекватно представлять особенности и потребности целевых пользователей. Помимо того, что некоторые из сервисов юзабилити-опросов, которые упоминались выше, могут быть использованы для сбора мнений экспертов, существуют и специализированные инструменты для проверки соответствия рекомендациям. Они, однако, чаще направлены на проверку доступности для всех категорий пользователей, поскольку рекомендации, касающиеся этой сферы, хорошо описаны в соответствующих стандартах (W3C's Web Content Accessibility Guidelines, US Section 508 и многих других – например, для пользователей с нарушениями зрения рекомендации существуют более десяти лет [14]) и зачастую могут быть проверены прямо по программному коду веб-сайта. Автоматическая проверка рекомендаций, связанных с юзабилити структуры сайта, сценариев взаимодействия, элементов интерфейса и пр., остается гораздо менее достижимой (относительно свежий и полный обзор дается в [15]).

Следует отметить, что сбор субъективных мнений пользователей остается основным методом для измерения такого аспекта юзабилити, как удовлетворенность, который находится в некоторой изоляции от продуктивности и эффективности, обычно воспринимаемых как более объективные. И хотя активные дискуссии на тему, что же такое удовлетворенность пользователя, продолжаются (см. [16]), в Японии с 1980-х годов началось использование вполне прагматичного подхода в этой области – так называемой эмоциональной (кансей) инженерии, которая, можно сказать, рассматривает удовлетворенность пользователя от взаимодействия с продуктом как «черный ящик» [17]. Этот метод позволяет идентифицировать успешный веб-сайт или другой артефакт через набор субъективных оценок по нескольким эмоциональным шкалам, а затем выработать проектные решения для нового продукта с целью произвести желаемое впечатление на пользователей. Применение эмоциональной инженерии в веб-инженерии, однако, пока не носит системный характер [18], и случаи широко используемой на практике автоматизации этого метода неизвестны.

#### **Автоматизация оценки юзабилити**

**Основные типы подходов к автоматизации оценки юзабилити.** Среди преимуществ автоматической оценки юзабилити веб-сайтов в основном отмечают следующие [19]: снижение затрат и требований к квалификации экспертов, более полный и последовательный охват, возможность прогнозировать потери от юзабилити-проблем, оперативно пробовать различные версии дизайна и т.д. Соответствующие подходы были заложены изначально еще в начале 2000-х годов (см., например, [20]) и в настоящее время могут быть разделены на три большие группы.

Прежде всего это *оценка юзабилити, основанная на взаимодействии*. Можно сказать, это наиболее прямолинейный подход, поскольку он использует данные, возникающие в результате взаимодействия (реального или постановочного) с сайтом, юзабилити которого оценивается. В частности, в ходе анализа могут рассматриваться движение курсора мыши или характеристики ввода с клавиатуры, сравниваться лог реального взаимодействия с некоторым описанным идеалом или изучаться потенциальные юзабилити-угрозы [21]. Однако отмечается, что при помощи таких технологий затруднительно получить достоверные количественные значения, учитывая разнообразие контекстов использования [2]. Фундаментальным недостатком подхода является также то, что для проведения оценки требуется достаточный объем данных о реальных взаимодействиях. Предоставление клиен-

там потенциально низкокачественного веб-интерфейса влечет за собой риск для репутации или даже финансовые потери.

Соответственно, *оценка юзабилити, основанная на метриках*, пытается определить и квантифицировать характеристики веб-сайтов, отражающие их юзабилити – от соответствия рекомендациям по дизайну [22] и визуальной сложности веб-интерфейсов [5] до высокоуровневых факторов дизайна, например объема содержимого страницы или соотношения объемов графики и текста [23]. Полагаем, что уровень юзабилити веб-сайта (особенно, если он низкий) действительно возможно предсказать, основываясь на некоторых факторах, и современные инструменты автоматизации в этой области показывают неплохую корреляцию с результатами оценщиков-людей [15]. Однако этот подход имеет пока не разрешенную проблему с учетом влияния различных категорий пользователей и контекстов использования на сравнительную значимость метрик.

В рамках *оценки юзабилити, основанной на моделировании*, преодоление описанной выше проблемы представляется возможным, поскольку методы искусственного интеллекта (например симуляция), применяемые для оценки юзабилити без реального взаимодействия, могут учитывать модель задач, как и модели контекста использования (пользователя, платформы и среды), которые обычно и создаются в рамках модели ориентированной парадигмы разработки интерфейсов [24]. Например, система MASP+MeMo [25] моделирует взаимодействие с учетом контекста использования и задач (хотя модель пользователя не выделяется, будучи включенной в контекст), в то время как в [26] были предложены настраиваемые модели. В целом подход выглядит многообещающим, особенно с учетом современного бурного развития методов искусственного интеллекта. Однако следует заметить, что вычислительная сложность моделей может оказаться слишком высокой (если только веб-интерфейсы не создаются для относительно простых задач или особых контекстов использования), а адаптация моделей к меняющимся тенденциям в проектировании юзабилити слишком затратной.

**Гибридные подходы к оценке юзабилити на основе методов искусственного интеллекта.** Описанные выше основные подходы к автоматизированной оценке юзабилити можно обобщить так, как показано в таблице. Из нее видно, что никакой отдельный подход или инструмент не является универсальным решением (отсутствие общепринятого количественного выражения юзабилити, безусловно, способствует такой ситуации) и их комбинация часто оказывается более выигрышной, в том числе с точки зрения экономической эффективности.

## Обобщение автоматизированных подходов к оценке юзабилити

## Generalization of automated approaches to assessing usability

Подход	Пользователи, веб-сайты, взаимодействие	Ограничения
<b>Основанный на взаимодействии</b>	Реальные пользователи или их представители. Реальные веб-сайты или их прототипы (динамические). Реальные взаимодействия (отраженные в логах)	В покрытии – в основном «собственные» веб-сайты. В затратах времени – требуется накопить достаточное количество взаимодействий. В плохом сервисе – потенциальные потери от низкого качества
<b>Основанный на метриках</b>	Без пользователей. Реальные веб-сайты (как правило, статический анализ). Без взаимодействий (анализ кода или дизайна)	В обобщении – метрики и их сравнительные веса зависят от контекста использования. В специалистах – могут требоваться эксперты или пользователи-аннотаторы
<b>Основанный на моделировании</b>	Модель пользователей или их симуляция. Модель, прототип или реальный сайт. Симулированное взаимодействие, анализ данных других взаимодействий	В ресурсах – требуются собранные данные, актуальные модели, вычислительные ресурсы. В знаниях – модели должны быть созданы и обучены

Гибридные подходы к оценке юзабилити зачастую основываются на методах искусственного интеллекта и машинного обучения, что может включать в себя обработку данных о взаимодействии в реальном времени, основанные на моделях эволюционные подходы [27], юзабилити-модели, обученные на данных взаимодействий или опросов [2], и т.п. Результаты могут варьироваться от только лишь полученной оценки качества взаимодействия до определения потенциальных юзабилити-проблем [28], указания на релевантные рекомендации по дизайну [29] и даже до регенерации интерфейса на лету [30]. Особо можно отметить новый и довольно многообещающий инструмент под названием WaPPU [2], который совмещает анализ взаимодействий, сбор субъективных юзабилити-метрик и обучаемые юзабилити-модели. Этот гибкий подход позволяет как осуществлять количественную оценку юзабилити, так и предсказывать статистически значимые отличия между юзабилити различных версий пользовательских интерфейсов; правда, для этого требуется значительное количество взаимодействий, а интеграция с системами веб-аналитики пока не предусмотрена.

Поскольку известно, что проектирование веб-юзабилити – это итеративный процесс, в котором стратегия «часто проверяй, часто изменяй» является одной из лучших, неудивительно, что в последнее время в этой области расширяется применение эволюционных алгоритмов, основанных на методах искусственного интеллекта [31]. Преимущества эволюционных алгоритмов по сравнению, например, с более широко используемым методом А/В тестирования в реальной среде включают в себя потенциально более быстрые итерации, меньшую необходимость вовлечения дизайнера-человека при создании решений-кандидатов, а также возможность избежать работы реальных пользователей с неудачными версиями интерфейсов. Проблемы с применением эволюционных алгоритмов в проектировании взаимодействия в основном свя-

заны с формулировкой и вычислением фитнес-функции: поскольку оно должно производиться на каждом шаге, его вычислительная сложность критична для общей производительности алгоритма. Поэтому для вычисления фитнес-функции в эволюционных алгоритмах все чаще применяются нейронные сети, фактически симулирующие особенности и предпочтения пользователей (см., например, [32, 33]), что позволяет осуществлять выбор кандидатов на основе аппроксимации фитнес-функции – только приемлемые дизайны показываются пользователям и участвуют в сборе данных в ходе реальных взаимодействий. Нейронные сети рассматривают юзабилити как «черный ящик» и могут адаптироваться к меняющимся требованиям или среде [34], например, к эволюционирующему пониманию юзабилити со стороны поисковых систем.

**Оценка поведенческих факторов поисковыми системами.** Как упоминалось ранее, крупные поисковые системы имеют собственные алгоритмы оценки юзабилити веб-сайтов, и оно сейчас является значимым фактором, влияющим на ранжирование в результатах поиска. С учетом постоянно идущей битвы против черной и серой поисковых оптимизаций алгоритмы полностью не разглашаются и подвергаются постоянным модификациям. Поэтому специалисты в этой области постоянно ищут малейшие намеки (например, цитаты из выступлений представителей Гугла или Яндекс), чтобы оптимизировать свои сайты под них. С этой точки зрения поисковые системы все более ответственны за формирование практического определения юзабилити, пока оно остается объектом теоретических дискуссий в исследовательском сообществе. Хотя компании время от времени патентуют свои технологии для оценки юзабилити и перечни учитываемых факторов (так, Yahoo! получила патент «Количественный анализ загроможденности веб-страницы и ее влияния на субъективные впечатления» еще в 2006 г.), они публикуются с суще-

ственной задержкой и таким образом не имеют особой практической значимости. Например, заявка на патент Гугл «Модификация ранжирования в результатах поиска в зависимости от неявной реакции пользователей и модель ошибки представления» (US 8938463 B1) была подана в 2007 г., а дата его публикации – 2015 г. В этой ситуации специалисты по оптимизации работают над реальными веб-проектами исходя из собственного представления о значимых юзабилити-факторах, или же подобно сервисам типа WooRank или GetGoodRank стараются непосредственно конвертировать в деньги свои знания (адекватность которых, однако, проверить трудно).

Фактически алгоритмы поисковых систем соответствуют гибридным подходам к оценке юзабилити, хотя в основном полагаются на измерения в рамках взаимодействий (что понятно, поскольку они имеют доступ к огромным объемам данных о поведении пользователей), и оптимизация под поисковые системы в настоящий момент скорее направлена на такие поведенческие факторы. Хотя строгое обоснование и точные количественные оценки отсутствуют, широко признается, что перечисленные далее юзабилити-факторы, основывающиеся на взаимодействии, важны для ранжирования [35].

- *Высокий естественный показатель кликабельности (CTR)* для ключевых слов, на который влияет контент веб-сайта, отображаемый в результатах поиска.

- *Низкий показатель отскоков:* веб-страницы, с которых пользователи быстро уходят, вероятно, оказались им не очень полезны (хотя это необязательно означает, что у этих страниц низкое юзабилити).

- *Высокая посещаемость:* если веб-сайт популярен, особенно с высоким количеством повторных посещений, это может свидетельствовать о его юзабилити.

- *Большое время пребывания на сайте:* если посетители проводят на нем долгое время, это считается свидетельством в пользу его высокого юзабилити, хотя вообще-то обычно показатель юзабилити «время выполнения задачи» стремятся минимизировать, а не максимизировать.

- *Высокая активность пользователей:* если пользователи активно комментируют содержимое сайта, значит, они смогли как оценить его полезность, так и успешно осуществить взаимодействие с сайтом, вводя данные и отправляя их для публикации.

В настоящее время есть ряд свидетельств, что поисковые системы в своей оценке юзабилити начинают прибегать и к моделированию, например, патент Гугл (US 8903690 B2) «Связь между инструментами тестирования графических пользовательских интерфейсов и моделированием производительности человека для оценки юзабилити»,

опубликованный в 2014 г. Конечно же, поисковые системы хотели бы иметь и возможность оценивать сайты на основе их программного кода, и уже в 2006 г. в упомянутом ранее патенте Yahoo! было предложено строить линейную регрессию с использованием метрик контента (текста), ссылок, дизайна и т.п. Системно элементы оценки юзабилити, основанной на метриках, были впервые введены Яндексом в районе 2009 г., когда были запущены алгоритмы семейства MatrixNet, позволившие этой системе на тот момент получить преимущество перед Гугл, запустившим свой алгоритм машинного обучения Panda лишь в 2011 г. [36]. В таких алгоритмах позитивные оценки юзабилити, полученные от людей-оценщиков, сопоставляются очень большому количеству возможных метрик с постоянным пересчетом их сравнительной значимости – так что, однозначный ответ на вопрос о поведенческих факторах, наиболее важных для ранжирования сайтов в результатах поиска, при всем желании не могут дать даже представители поисковых систем.

Таким образом, следует сделать вывод, что даже крупные поисковые системы не определяют юзабилити в явном виде, а скорее рассматривают его как «черный ящик» (что соответствует подходу, используемому в нейронных сетях) и аппроксимируют оценку, поскольку для точной оценки юзабилити требуется способность полного понимания текстовой информации, которой не достигли еще даже поисковые системы. Следовательно, концептуализация и практическая оценка юзабилити со стороны поисковых систем, которые в значительной степени формируют веб-среду на сегодняшний день, постоянно эволюционируют и требуют соответствующих гибких и адаптивных подходов со стороны веб-дизайнеров и инженеров.

## Заключение

Экономический эффект от проектирования юзабилити остается высоким, однако явно требуются дальнейшие инновации в этой области, поскольку все очевидные и легко устранимые ошибки в основном уже были исправлены [1]. Существуют инструменты, которые совмещают несколько подходов к оценке юзабилити, стремясь обеспечить более полную оценку и тесную связь с проектированием юзабилити, то есть не только измерение и идентификацию юзабилити-проблем, но и помощь в их исправлении, например, посредством связи с релевантными рекомендациями или шаблонами проектирования.

Результаты данного обзора позволяют предположить, что барьеры на пути к более полной и эффективной автоматизации оценки юзабилити включают в себя сохраняющиеся неясности в количественном выражении юзабилити в сочетании с меняющимися стандартами веб-индустрии и ожи-

даниями пользователей. На практике в настоящее время поисковые системы осуществляют оценку юзабилити на основе комбинации подходов, основанных на взаимодействии, метриках и моделях. Новейшие методы в этой области также включают в себя эволюционные алгоритмы для проектирования взаимодействия (как, например, было предложено в [31]), которые, в частности, могут работать с меняющейся фитнес-функцией, адаптирующейся к постоянно меняющемуся концептуальному представлению об юзабилити.

*Исследование выполнено при поддержке РФФИ в рамках научного проекта № 16-37-60060 мол\_а\_дк.*

#### Литература

- Nielsen J. Usability ROI declining, but still strong. 2008. URL: <https://www.nngroup.com/articles/usability-roi-declining-but-still-strong/> (дата обращения: 20.01.2017).
- Speicher M., Both A., and Gaedke M. Ensuring web interface quality through usability-based split testing. LNCS, 2014, vol. 8541, pp. 93–110.
- Paas F.G. and Van Merriënboe J.J. The efficiency of instructional conditions: an approach to combine mental effort and performance measures. Jour. of the Human Factors and Ergonomics Society, 1993, vol. 35, no. 4, pp. 737–743.
- Buettner R. Cognitive workload of humans using artificial intelligence systems: towards objective measurement applying eye-tracking technology. LNCS, 2013, vol. 8077, pp. 37–48.
- Bakaev M. and Avdeenko T. A quantitative measure for information transfer in human-machine control systems. Proc. IEEE Int. Siberian Conf. on Control and Communications (SIBCON), 2015, pp. 1–4.
- Карпов А.А., Ронжин А.Л., Усов В.М. Инструментальная методика тестирования интерактивного бесконтактного человеко-машинного взаимодействия при использовании шлема виртуальной реальности // Пилотируемые полеты в космос. № 3. 2015. С. 43–53.
- Hasan L., Morris A. and Proberts S. Using Google Analytics to evaluate the usability of e-commerce sites. LNCS, 2009, vol. 5619, pp. 697–706.
- Cardello J. Five essential analytics reports for UX strategists. 2014. URL: <https://www.nngroup.com/articles/analytics-reports-ux-strategists/> (дата обращения: 20.01.2017).
- Nielsen J. The most important usability activity. 2012. URL: <https://www.nngroup.com/articles/the-most-important-usability-activity/> (дата обращения: 20.01.2017).
- Nielsen J. Anybody can do usability. 2009. URL: <https://www.nngroup.com/articles/anybody-can-do-usability/> (дата обращения: 20.01.2017).
- Савельев А.И. Алгоритмы обработки данных в контролируемых аккаунтах системы видеоконференцсвязи // Информационно-управляющие системы. 2016. Т. 82. № 3. С. 906–913.
- Schade A. Remote usability tests: moderated and unmoderated. 2013. URL: <https://www.nngroup.com/articles/remote-usability-tests/> (дата обращения: 20.01.2017).
- Nielsen J. First rule of usability? Don't listen to users. 2001. URL: <https://www.nngroup.com/articles/first-rule-of-usability-dont-listen-to-users/> (дата обращения: 20.01.2017).
- Leporini B. and Paternò F. Increasing usability when interacting through screen readers. Universal access in the information society, 2004, vol. 3, no. 1, pp. 57–70.
- Dingli A. and Cassar S. An intelligent framework for web-site usability. Advances in Human-Computer Interaction, 2014, vol. 5, pp. 1–13.
- Speicher M., Both A., and Gaedke M. SOS: does your search engine results page (SERP) need help? Proc. 33rd Annual ACM Conf. on Human Factors in Computing Systems, 2015, pp. 1005–1014.
- Schütte S.T.W., Eklund J., Axelsson J.R.C., Nagamachi M. Concepts, methods and tools in Kansei engineering. Theoretical Issues in Ergonomics Science, 2004, vol. 5, no. 3, pp. 214–231.
- Bakaev M., Gaedke M., Khvorostov V., Heil S. Extending Kansei engineering for requirements consideration in web interaction design. LNCS, 2016, vol. 9671, pp. 513–518.
- Dingli A. and Mifsud J. Useful: A framework to mainstream web site usability through automated evaluation. IHCI, 2011, vol. 2, no. 1, pp. 10–30.
- Vanderdonckt J., Beirekdar A., Noirhomme-Fraiture M. Automated evaluation of web usability and accessibility by guideline review. LNCS, 2004, vol. 3140, pp. 17–30.
- Grigera J., Garrido A., and Rivero J.M. A tool for detecting bad usability smells in an automatic way. LNCS, 2014, vol. 8541, pp. 490–493.
- Gay G. and Qi Li C. AChecker: open, interactive, customizable, web accessibility checking. Proc. 2010 Intern. Cross Disciplinary Conf. on Web Accessibility (W4A'10), 2010, p. 23.
- Lin Y.C., Yeh C.H., and Wei C.C. How will the use of graphics affect visual aesthetics? A user-centered approach for web page design. Int. Jour. of Human-Computer Studies, 2013, vol. 71, no. 3, pp. 217–227.
- Kieras D. Model-based evaluation. In Jacko, J.A. & Sears, A. (eds) The human-computer interaction handbook. NJ: Lawrence Erlbaum Associates, 2003, pp. 1139–1151.
- Feuerstack S., et al. Automated usability evaluation during model-based interactive system development. LNCS, 2008, vol. 5247, pp. 134–141.
- Ladry J., Palanque P., Barboni E., and Navarre D. Model-based usability evaluation and analysis of interactive techniques. Proc. 5th Intern. Workshop on Model Driven Development of Advanced User Interfaces at ACM CHI, 2010, pp. 21–24.
- Pleuss A., Wollny S., and Botterweck G. Model-driven development and evolution of customized user interfaces. Proc. 5th ACM SIGCHI Symposium on Engineering Interactive Computing Systems, 2013, pp. 13–22.
- Grigera J., Garrido A., Rivero J.M., Rossi G. Automatic detection of usability smells in web applications. Intern. Jour. of Human-Computer Studies, 97, 2017, pp. 129–148.
- Viswanathan S. and Peters J.C. Automating UI guidelines verification by leveraging pattern based UI and model based development. ACM CHI'10 EA on Human Factors in Computing Systems, 2010, pp. 4733–4742.
- Barrera-León L., et al. TUKUCHIY: a dynamic user interface generator to improve usability. Intern. Jour. of Web Information Systems, 2016, vol. 12, iss. 2, pp. 150–176.
- Bakaev M., Gaedke M. Application of Evolutionary Algorithms in Interaction Design. Proc. IEEE ElConRusNW, 2016, pp. 125–130.
- Qu Q.X. Kansei knowledge extraction based on evolutionary genetic algorithm: an application to e-commerce web appearance design. Theoretical Issues in Ergonomics Science, 2015, vol. 16 (3), pp. 299–313.
- Guo F., et. al. Optimization design of a webpage based on Kansei engineering. Human Factors and Ergonomics in Manufacturing & Service Industries, 2016, vol. 26, no. 1, pp. 110–126.
- Yao X. Evolving artificial neural networks. Proc. IEEE, 1999, vol. 87, no. 9, pp. 1423–1447.
- Dean B. Google's 200 Ranking Factors: The Complete List. 2016. URL: <http://backlinko.com/google-ranking-factors> (дата обращения: 20.01.2017).
- Siotos M. A panda wandering in the Siberian steppe... 2011. URL: <http://www.freshegg.co.uk/blog/technical-seo/search-algorithms/panda-wondering-siberian-steppe> (дата обращения: 20.01.2017).

**MODERN TRENDS IN AUTOMATED USABILITY EVALUATION AND BEHAVIORAL FACTORS  
IN SEARCH ENGINE ALGORITHMS**

**M.A. Bakaev**<sup>1</sup>, Ph.D. (Engineering), Senior Researcher, maxis81@gmail.com

<sup>1</sup>Novosibirsk State Technical University, K. Marks Ave. 20, Novosibirsk, 630073, Russian Federation

**Abstract.** Nowadays, there are persisting ambiguities in usability conceptualization for quantitative measurement and a certain gap between its more objective components (effectiveness and efficiency) and a fairly subjective aspect of user satisfaction.

The paper mentions that the practical goal is not necessarily measuring website's usability, but rather improving its quality-in-use or assessment by respective algorithms of major search engines, which increase the importance of "behavioral factors" in their rankings. The paper describes traditional methods in usability evaluation and design, as well as up-to-date tools that introduce certain automation into them.

The authors identify approaches within automated usability evaluation and summarize possibilities and limitations of each one. The approaches are based on interactions, metrics and modeling. Furthermore, there is an overview of hybrid approaches that use more and more AI and machine learning methods. Therefore, an effective method for website interaction quality evaluation is neural networks, which are applied within evolutionary web design algorithms to calculate fitness function for candidate solutions. The final part of the paper describes how global search engines (such as Yandex and Google) perform automated website usability evaluation and provides a list of the most important behavioral factors. In fact, they mostly consider usability as a "black box" and actively apply machine-learning methods to approximate its calculation.

Thus, usability conceptualization and practical assessment by search engines, which nowadays shape the web to a great extent, is constantly evolving and, in our opinion, calls for respectively flexible approaches from web designers and engineers.

**Keywords:** human-computer interaction, usability engineering, web design, artificial intelligence, search engines optimization.

**Acknowledgements.** The reported study has been funded by RFBR within the research project no. 16-37-60060 mol\_a\_dk.

**References**

1. Nielsen J. *Usability ROI Declining, But Still Strong*. 2008. Available at: <https://www.nngroup.com/articles/usability-roi-declining-but-still-strong/> (accessed January 20, 2017).
2. Speicher M., Both A., Gaedke M. Ensuring web interface quality through usability-based split testing. *LNCS*. 2014, vol. 8541 (Web Engineering), pp. 93–110.
3. Paas F.G., Van Merriënboe J.J. The efficiency of instructional conditions: An approach to combine mental effort and performance measures. *Jour. of the Human Factors and Ergonomics Society*. 1993, vol. 35 (4), pp. 737–743.
4. Buettner R. Cognitive Workload of Humans Using Artificial Intelligence Systems: Towards Objective Measurement Applying Eye-Tracking Technology. *LNCS*. 2013, vol. 8077 (KI 2013: Advances in Artificial Intelligence), pp. 37–48.
5. Bakaev M., Avdeenko T. A quantitative measure for information transfer in human-machine control systems. *IEEE Int. Siberian Conf. on Control and Communications (SIBCON)*. 2015, pp. 1–4.
6. Karpov A.A., Ronzhin A.L., Usov V.M. Instrumental Methods of Testing an Interactive Contactless Human-Machine Interaction When Using a Helmet-Mounted Display. *Pilotiruemye polety v kosmos [Manned Spaceflight]*. 2015, no. 3 (16), pp. 43–53.
7. Hasan L., Morris A., Proberts S. Using Google Analytics to evaluate the usability of e-commerce sites. *LNCS*. 2009, vol. 5619 (Human centered design), Springer Berlin Heidelberg Publ., pp. 697–706.
8. Cardello J. *Five Essential Analytics Reports for UX Strategists*. 2014. Available at: <https://www.nngroup.com/articles/analytics-reports-ux-strategists/> (accessed January 20, 2017).
9. Nielsen J. *The Most Important Usability Activity*. 2012. Available at: <https://www.nngroup.com/articles/the-most-important-usability-activity/> (accessed January 20, 2017).
10. Nielsen J. *Anybody Can Do Usability*. 2009. Available at: <https://www.nngroup.com/articles/anybody-can-do-usability/> (accessed January 20, 2017).
11. Savelev A.I. Algorithms of Data Processing in Supervised Accounts of a Videoconferencing System. *Informatsionno-upravliaiushchie sistemy [Information and Control Systems]*. 2016, vol. 82 (3), pp. 906–913 (in Russ.).
12. Schade A. *Remote Usability Tests: Moderated and Unmoderated*. 2013. Available at: <https://www.nngroup.com/articles/remote-usability-tests/> (accessed January 20, 2017).
13. Nielsen J. *First Rule of Usability? Don't Listen to Users*. 2001. Available at: <https://www.nngroup.com/articles/remote-usability-tests/> (accessed January 20, 2017).
14. Leporini B., Paternò F. Increasing usability when interacting through screen readers. *Universal access in the information society*. 2004, vol. 3 (1), pp. 57–70.
15. Dingli A., Cassar S. An intelligent framework for website usability. *Advances in Human-Computer Interaction*. 2014, vol. 5, pp. 1–13.
16. Speicher M., Both A., Gaedke M. SOS: Does Your Search Engine Results Page (SERP) Need Help? *Proc. 33rd Annual ACM Conf. on Human Factors in Computing Systems*. 2015, pp. 1005–1014.



17. Schütte S.T.W., Eklund J., Axelsson J.R.C., Nagamachi M. Concepts, methods and tools in Kansei Engineering. *Theoretical Issues in Ergonomics Science*. 2004, vol. 5 (3), pp. 214–231.
18. Bakaev M., Gaedke M., Khvorostov V., Heil S. Extending Kansei Engineering for Requirements Consideration in Web Interaction Design. *LNCS*. 2016, vol. 9671, pp. 513–518.
19. Dingli A., Mifsud J. Useful: A framework to mainstream web site usability through automated evaluation. *IJHCI*. 2011, vol. 2 (1), pp. 10–30.
20. Vanderdonckt J., Beirekdar A., Noirhomme-Fraiture M. Automated evaluation of web usability and accessibility by guideline review. *LNCS*. 2004, vol. 3140 (Web Engineering), pp. 17–30.
21. Grigera J., Garrido A., Rivero J.M. A tool for detecting bad usability smells in an automatic way. *LNCS*. 2014, vol. 8541 (Web Engineering), pp. 490–493.
22. Gay G., Qi Li C. AChecker: open, interactive, customizable, web accessibility checking. *Proc. 2010 Int. Cross Disciplinary Conf. on Web Accessibility (W4A'10)*. 2010, no. 23, p. 23.
23. Lin Y.C., Yeh C.H., Wei C.C. How will the use of graphics affect visual aesthetics? A user-centered approach for web page design. *Int. Journ. of Human-Computer Studies*. 2013, vol. 71 (3), pp. 217–227.
24. Kieras D. Model-based evaluation. *The human-computer interaction handbook*. In Jacko, J.A. & Sears, A. (Eds.) NJ, Lawrence Erlbaum Associates Publ., 2003, pp. 1139–1151.
25. Feuerstack S., Blumendorf M., Kern M., Kruppa M., Quade M., Runge M., Albayrak S. Automated usability evaluation during model-based interactive system development. *LNCS*. 2008, vol. 5247 (Engineering Interactive Systems), pp. 134–141.
26. Ladry J., Palanque P., Barboni E., Navarre D. Model-based usability evaluation and analysis of interactive techniques. *Proc. 5th Int. Workshop on Model Driven Development of Advanced User Interfaces at ACM CHI*. 2010, pp. 21–24.
27. Pleuss A., Wollny S., Botterweck G. Model-driven development and evolution of customized user interfaces. *Proc. 5th ACM SIGCHI Symp. on Engineering Interactive Computing Systems*. 2013, pp. 13–22.
28. Grigera J., Garrido A., Rivero J.M., Rossi G. Automatic detection of usability smells in web applications. *Int. Jour. of Human-Computer Studies*. 2017, vol. 97, pp. 129–148.
29. Viswanathan S., Peters J.C. Automating UI guidelines verification by leveraging pattern based UI and model based development. *ACM CHI'10 EA on Human Factors in Computing Systems*. 2010, pp. 4733–4742.
30. Barrera-León L., Mejia-Molina N., Carrillo-Ramos A., Flórez-Valencia L., Pavlich-Mariscal J.A. TUKUCHIY: A Dynamic User Interface Generator to Improve Usability. *Int. Jour. of Web Information Systems*. 2016, vol. 12, iss. 2, pp. 150–176.
31. Bakaev M., Gaedke M. Application of Evolutionary Algorithms in Interaction Design. *Proc. IEEE ElConRusNW*. 2016, pp. 125–130.
32. Qu Q.X. Kansei knowledge extraction based on evolutionary genetic algorithm: an application to e-commerce web appearance design. *Theoretical Issues in Ergonomics Science*. 2015, vol. 16 (3), pp. 299–313.
33. Guo F., Liu W.L., Cao Y., Liu F.T., Li M.L. Optimization design of a webpage based on Kansei Engineering. *Human Factors and Ergonomics in Manufacturing & Service Industries*. 2016, vol. 26 (1), pp. 110–126.
34. Yao X. Evolving artificial neural networks. *Proc. IEEE*. 1999, vol. 87, no. 9, pp. 1423–1447.
35. Dean B. *Google's 200 Ranking Factors: The Complete List*. Available at: <http://backlinko.com/google-ranking-factors> (accessed January 20, 2017).
36. Siotos M. *A Panda Wandering in the Siberian Steppe... The Fresh Egg Blog*. 2011. Available at: <http://www.freshegg.co.uk/blog/technical-seo/search-algorithms/panda-wondering-siberian-steppe> (accessed January 20, 2017).

## Примеры библиографического описания статьи

1. Бакаев М.А. Современные тенденции в автоматизированной оценке юзабилити и поведенческие факторы в алгоритмах поисковых систем // Программные продукты и системы. 2017. Т. 30. № 3. С. 447–455. DOI: 10.15827/0236-235X.119.447-455.
2. Bakaev M.A. Modern trends in automated usability evaluation and behavioral factors in search engine algorithms. *Programmnye produkty i sistemy* [Software & Systems]. 2017, vol. 30, no. 3, pp. 447–455 (in Russ.). DOI: 10.15827/0236-235X.119.447-455.

УДК 519.767.2

DOI: 10.15827/0236-235X.119.456-460

Дата подачи статьи: 05.06.17

2017. Т. 30. № 3. С. 456–460

## ВЕРИФИКАЦИЯ МОДЕЛЕЙ СИСТЕМ НА БАЗЕ ЭКВАЦИОНАЛЬНОЙ ХАРАКТЕРИСТИКИ ФОРМУЛ LTL

Ю.П. Кораблин, д.т.н., профессор, y.p.k@mail.ru;

А.С. Кочергин, аспирант, kocherginalexandr@gmail.com;

А.А. Шипов, к.т.н., a-j-a-1@yandex.ru

(Российский государственный социальный университет,  
ул. Вильгельма Пика, 4, г. Москва, 129226, Россия)

Метод верификации на моделях Model Checking уже давно получил широкое признание в области, связанной с оценкой качества работы программных и технических систем. Такие ключевые компании в области IT-индустрии, как Intel, Microsoft, Amazon и другие, активно применяют его на этапах разработки и сопровождения своих продуктов. Успех Model Checking, безусловно, не является случайным, поскольку именно его появление и развитие позволили решить множество проблем в области верификации, а именно: проблемы унифицированного представления программных и технических систем, формального задания требований, автоматизации этапов верификации, верификации больших распределенных программных систем и другие.

Однако постоянное развитие современных технологий и темпы роста сложности современных программных систем ставят перед Model Checking все новые проблемы, которые могут стать непреодолимым препятствием на пути эффективной верификации. Поэтому необходимо постоянное совершенствование теории и инструментов данного метода.

В статье авторами подробно рассматривается реализация алгоритма верификации метода Model Checking для формул логики линейного времени LTL на базе новой нотации RLTL (Recursive Linear Temporal Logic), которая является рекурсивным представлением формул логики линейного времени. Поскольку на базе RLTL могут быть заданы как модель верифицируемой системы, так и требования к ней, можно избежать необходимости их предварительного преобразования к автоматам Бюхи и сразу приступить к процессу верификации, что упростит алгоритм метода и повысит его эффективность.

**Ключевые слова:** верификация, Model Checking, эквивалентная характеристика RLTL, модель Крипке, автомат Бюхи, формула временной логики, LTL, CTL.

Верификация программных и технических систем является сегодня одним из важнейших этапов их жизненного цикла. Для этих целей уже создано и успешно апробировано множество различных средств и методов разного типа сложности и применимости. Однако наиболее часто используемыми и востребованными, как правило, остаются средства формальной верификации, работа которых может быть в той или иной мере автоматизирована в отличие от, например, динамических методов верификации и экспертиз.

Среди формальных методов верификации наибольшую популярность за свою практичность и эффективность получил так называемый метод проверки на моделях Model Checking [1, 2]. Базовая идея алгоритма этого метода для требований к системе, заданных в виде формул логики линейного времени LTL [3, 4], представлена в виде схемы на рисунке 1.

Использование данной схемы в чистом виде при верификации моделей систем больших размеров (более  $2^{20}$  состояний) во многих случаях может приводить к тому, что верификация либо будет выполняться очень долго, либо вообще не будет выполнена. Для решения данных проблем и достижения более высокой производительности сегодня существует ряд специальных средств и методов, которые за счет определенных манипуляций над моделью системы и/или над требованиями к ней позволяют добиться приемлемой производи-

тельности [5–7]. В частности, одним из инструментов повышения эффективности выполнения верификации является предложенный в работе [8] метод абстракции и унификации RLTL-моделей (Recursive Linear Temporal Logic).

В данной статье авторами предложен принципиально иной подход к повышению производительности процесса верификации, который явля-

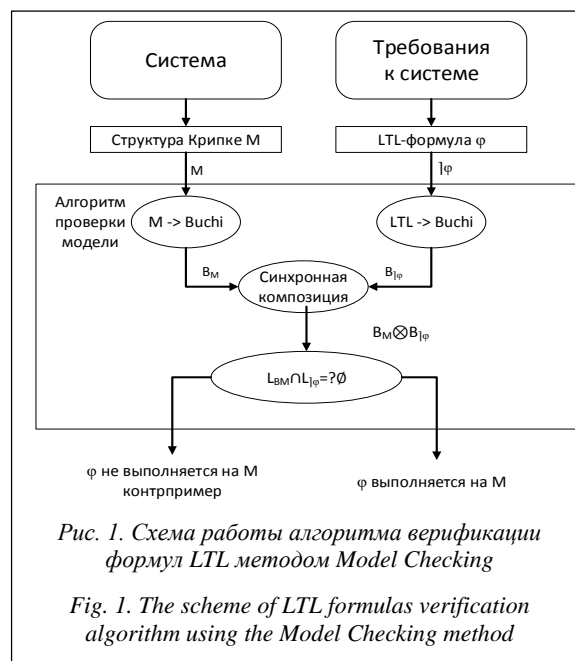
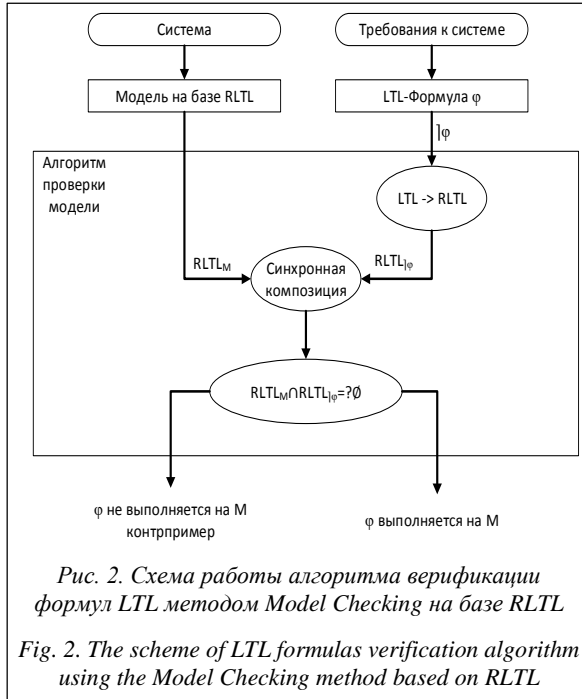


Рис. 1. Схема работы алгоритма верификации формул LTL методом Model Checking

Fig. 1. The scheme of LTL formulas verification algorithm using the Model Checking method

ется результатом использования предложенной в [9] новой нотации LTL, обозначаемой как RLTL. Показана возможность выполнения на базе RLTL и самого процесса верификации, что позволит упростить алгоритм верификации методом Model Checking и привести его к виду, представленному на рисунке 2.



### RLTL-нотация

Рассмотрим представления основных базовых конструкций логики LTL, заданных в RLTL-нотации:

LTL	RLTL
$\varphi_1 \vee \varphi_2$	$\varphi_1 + \varphi_2$
$\varphi_1 \wedge \varphi_2$	$\{\varphi_1, \varphi_2\}$
$X\varphi$	$\Delta \circ \varphi$

Запись  $\{\varphi_1, \varphi_2\}$  означает множество  $\mu$ , состоящее из предикатов  $\varphi_1$  и  $\varphi_2$ .

В RLTL приняты следующий обозначения.

○ – оператор продолжения (конкатенация выражений), что позволяет использовать оператор X в неявном виде, а также упростить его восприятие, в частности, запись  $\mu_1 \circ \mu_2$  будет пониматься как « $\mu_2$  следует за  $\mu_1$ ».

$\mu$  – множество выполнимых на конкретном этапе вычислительного процесса атомарных предикатов, задающих те или иные свойства системы.

$\Delta$  – предикат неопределенности, задающий неопределенное множество свойств системы  $\mu$ , выполнимых на конкретном этапе вычислительного процесса. Отрицанием предиката неопределенности является некоторое другое неопределенное подмножество свойств  $\mu'$ , а отрицанием отрицания – некоторое третье неопределенное подмноже-

ство  $\mu''$ . Таким образом, отрицание предиката неопределенности является неполным, а каждое последующее отрицание дает неопределенное подмножество атомарных предикатов. Каждое из этих подмножеств ввиду своей неопределенности также может быть обозначено через  $\Delta$ .

$\perp$  – оператор остановки вычислительного процесса. Это означает, что, как только данный оператор встречается в конкретный момент времени вычислительного процесса, в дальнейшем ни одно свойство системы не будет выполнено.

Сформулируем основные аксиомы нотации RLTL:

A1.	$FF\varphi = F\varphi$	(FF)
A2.	$GG\varphi = G\varphi$	(GG)
A3.	$\neg F\varphi = G\neg\varphi$	( $\neg F$ )
A4.	$\neg G\varphi = F\neg\varphi$	( $\neg G$ )
A5.	$G\varphi = \varphi \circ G\varphi$	(G)
A6.	$F\varphi = \varphi + \Delta \circ F\varphi$	(F)
A7.	$G\varphi + \neg G\varphi = \Delta$	(+ G)
A8.	$F\varphi + \neg F\varphi = \Delta$	(+ F)
A9.	$G\varphi_1 \circ \varphi_2 = G\varphi_1$	(G ○)
A10.	$F\varphi_1 \circ \varphi_2 = F(\varphi_1 \circ \varphi_2)$	(F ○)
A11.	$U(\varphi_1, \varphi_2) = \varphi_2 + \varphi_1 \circ U(\varphi_1, \varphi_2)$	(U)
A12.	$\neg(\mu \circ \mu_1) = \neg\mu + \Delta \circ \neg\mu_1$	( $\neg \mu \circ$ )
A13.	$\{\mu, \Delta\} = \mu$	( $\Delta$ )
A14.	$\{\varphi_1, \Delta \circ \varphi_2\} = \varphi_1 \circ \varphi_2$	( $\Delta \circ$ )
A15.	$\neg\mu = (\Delta \setminus \mu)$	$\neg\mu$
A16.	$\perp \circ \mu = \mu \circ \perp = \perp$	$\perp \circ$
A17.	$\perp + \mu = \mu$	$\perp +$

**Правила вывода в RLTL:** R1.  $\neg\Delta \rightarrow \Delta$ .

**Лемма 1.** Оператор F логики LTL, рекурсивное представление которого задано как  $Fq = q \vee X(Fq)$ , может быть представлен в RLTL-нотации в виде  $F' = q + \Delta \circ F'$ .

**Доказательство.** Обозначим  $Fq$  через  $F'$ . Тогда  $F' = Fq = q \vee X(Fq) = q \vee X(F')$ . Отсюда, в соответствии с правилами представления формул LTL в RLTL, получаем:  $F' = q + \Delta \circ F'$ . ■

**Лемма 2.** Оператор G логики LTL, рекурсивное представление которого задано как  $Gq = q \wedge X(Gq)$ , может быть представлен в RLTL-нотации в виде  $F' = q \circ F'$ .

**Доказательство.** Обозначим  $Gq$  через  $F'$ . Тогда  $F' = Gq = q \wedge X(Gq) = q \wedge X(F')$ . Отсюда, в соответствии с правилами представления формул LTL в RLTL, получаем

$$F' = \{q, \Delta \circ F'\} = [A14] = q \circ F'. \blacksquare$$

**Лемма 3.** Оператор U логики LTL, рекурсивное представление которого задано как  $pUq = q \vee (p \wedge X(pUq))$ , может быть представлен в RLTL-нотации в виде  $F' = q + p \circ F'$ .

**Доказательство.** Обозначим  $pUq$  через  $F'$ . Тогда  $F' = pUq = q \vee (p \wedge X(pUq)) = q \vee (p \wedge X(F'))$ .

Отсюда, в соответствии с правилами представления формул LTL в RLTL, получаем:  $F' = q + \{p, \Delta \circ F'\} = [A14] = q + p \circ F'$ . ■

### Построение синхронной композиции двух систем рекурсивных уравнений

Идея верификации моделей систем, заданных на базе RLTL-нотации, практически ничем не отличается от верификации систем, заданных в виде автоматов Бюхи. Для того чтобы проверить, выполняется ли целевое требование, также заданное с помощью RLTL, на модели системы или нет, необходимо запустить две системы синхронно (модель и требование к ней) и следить за процессом их синхронной работы. Система, работающая синхронно, осуществляет переход в следующее состояние по некоторому символу  $p$  тогда и только тогда, когда вторая система также может выполнить переход из своего текущего состояния в некоторое новое состояние по символу  $p$ .

Сформулируем основные аксиомы, необходимые для построения синхронной композиции двух систем рекурсивных уравнений:

- S1.  $\mu_1 \otimes \mu_2 = \mu_1$ , если  $\mu_1 \subseteq \mu_2$ ,  
 $\mu_2$ , если  $\mu_2 \subseteq \mu_1$ ,  
 $\perp$  в противном случае,  
 S2.  $\mu \otimes \perp = \perp$ ,  
 S3.  $\mu \otimes \Delta = \mu$ ,  
 S4.  $(\mu_1 + \mu_2) \otimes \mu_3 = (\mu_1 \otimes \mu_3) + (\mu_2 \otimes \mu_3)$ ,  
 S5.  $(\mu_1 \circ \mu_2) \otimes (\mu_3 \circ \mu_4) = (\mu_1 \otimes \mu_3) \circ (\mu_2 \otimes \mu_4)$ .

Под символом  $\otimes$  будем понимать оператор синхронной композиции. Все, что стоит справа или слева от данного оператора, должно выполняться синхронно, если это возможно.

### Пример построения синхронной композиции

**Дано.**

Система A:

$$F_{A1} = a \circ F_{A1} + b \circ F_{A2},$$

$$F_{A2} = c \circ F_{A1}.$$

Система B:

$$F_{B1} = c + \Delta \circ F_{B1}.$$

Для построения синхронной композиции расширим систему A следующим образом, полагая, что в случае наступления события «с» система продолжит работать бесконечно долго, то есть заикнется:  $F_{B1} = c \circ F_{B2} + \Delta \circ F_{B1}$ ,

$$F_{B2} = \Delta \circ F_{B2}.$$

**Решение.**

Построение синхронной композиции  $A \otimes B$ :

$$\begin{aligned} P_1 &= F_{A1} \otimes F_{B1} = \\ &= (a \otimes c) \circ (F_{A1} \otimes F_{B2}) + (a \otimes \Delta) \circ (F_{A1} \otimes F_{B1}) + \\ &+ (b \otimes c) \circ (F_{A2} \otimes F_{B2}) + (b \otimes \Delta) \circ (F_{A2} \otimes F_{B1}) = \\ &= \perp \circ (F_{A1} \otimes F_{B2}) + (a \otimes \Delta) \circ P_1 + \perp \circ (F_{A2} \otimes F_{B2}) + \\ &+ (b \otimes \Delta) \circ P_2 = \perp + a \circ P_1 + \perp + b \circ P_2 = a \circ P_1 + b \circ P_2. \\ P_2 &= F_{A2} \otimes F_{B1} = (c \otimes c) \circ (F_{A1} \otimes F_{B2}) + (c \otimes \Delta) \circ (F_{A1} \otimes F_{B1}) = \\ &= (c \otimes c) \circ P_3 + (c \otimes \Delta) \circ P_1 = c \circ P_3 + c \circ P_1. \\ P_3 &= F_{A1} \otimes F_{B2} = (a \otimes \Delta) \circ (F_{A1} \otimes F_{B2}) + (b \otimes \Delta) \circ (F_{A2} \otimes F_{B2}) = \\ &= a \circ P_3 + b \circ P_4. \end{aligned}$$

$$P_4 = F_{A2} \otimes F_{B2} = (c \otimes \Delta) \circ (F_{A1} \otimes F_{B2}) = c \circ P_3.$$

Конечная синхронная композиция:

$$P_1 = a \circ P_1 + b \circ P_2,$$

$$P_2 = c \circ P_3 + c \circ P_1,$$

$$P_3 = a \circ P_3 + b \circ P_4,$$

$$P_4 = c \circ P_3.$$

**Теорема 1.** Пусть заданы две системы рекурсивных уравнений A и B, где  $P_1, P_2, \dots, P_n$  – множество метaperменных системы A;  $P_2, P_2, \dots, P_{2m}$  – множество метaperменных системы B.

$$P_{1i} = \sum_{j \in N} \alpha_{ij} \circ P_{1ij}, \text{ где } N = \{1, 2, \dots, n\}.$$

$$P_{2i} = \sum_{k \in M} \alpha_{ik} \circ P_{2ik}, \text{ где } M = \{1, 2, \dots, m\}.$$

$$\forall i \forall j \exists l \in N \text{ такое, что } P_{1ij} = P_{1l}$$

$$\forall i \forall k \exists r \in M \text{ такое, что } P_{2ik} = P_{2r}.$$

Тогда синхронная композиция двух систем уравнений A и B, где  $P_1 = P_{11} \otimes P_{21}$ , конечна.

**Доказательство.**

$$\text{Обозначим через } \xi(u, v) = P_{1u} \otimes P_{2v}, \quad (*)$$

где  $u = 1, 2, \dots, n$ ;  $v = 1, 2, \dots, m$ .

Количество выражений  $\xi(u, v)$  конечно. По построению синхронной композиции имеем

$$\begin{aligned} \xi(u, v) &= \sum_{j \in N} \alpha_{uj} \circ P_{1uj} \otimes \sum_{k \in M} \alpha_{vk} \circ P_{2vk} = \\ &= \sum_{\substack{j \in N, \\ k \in M}} (\alpha_{uj} \otimes \alpha_{vk}) \circ (P_{1uj} \otimes P_{2vk}) = \\ &= \sum_{\substack{j \in N, \\ k \in M}} (\alpha_{uj} \otimes \alpha_{vk}) \circ P_{jk}, \text{ где все } P_{ij} \text{ входят в множе-} \end{aligned}$$

ство (\*). Таким образом, получается, что система уравнений для синхронной композиции A и B конечна. ■

### Верификация

Для верификации с помощью RLTL необходимо наличие модели M на базе RLTL, а также проверяемое относительно нее требование  $\phi$ , заданное на базе LTL. Для проверки выполнимости в системе заданного требования  $\phi$ , как и в алгоритме верификации формул логики линейного времени, будет использовано обратное требование  $[\phi]$ . Главным и самым важным отличием предлагаемого алгоритма от алгоритма верификации формул LTL является то, что верификация на базе RLTL более производительная, поскольку не требуется преобразовывать модель и обратное к ней требование в автоматы Бюхи. При этом осуществляется лишь преобразование обратного требования, заданного с помощью LTL, в RLTL-представление. Переход от LTL к RLTL является тривиальным и выполняется по описанным выше правилам и аксиомам в отличие от алгоритмов перехода от LTL к автоматам Бюхи, которые, как сказано в [1], зачастую выдают автоматы Бюхи экспоненциальной сложности относительно длины формулы  $\phi$ .

Отметим также, что, если формулировать требования к системе сразу на базе RLTL-нотации,

можно в принципе избавиться от всех промежуточных этапов превращения и выполнять процесс верификации сразу после создания модели системы и требований к ней. Указанные выше отличия позволяют существенно повысить производительность процесса верификации по сравнению с верификацией на основе автоматов Бюхи и получить более наглядные представления как модели системы, так и задаваемых требований в виде системы рекурсивных уравнений.

Ключевым процессом в RLTL-верификации, как и в алгоритме верификации формул LTL, является процесс построения синхронной композиции. После того как будет построена композиция двух систем в соответствии с методом, представленным ранее, необходимо выполнить оценку результатов данного процесса. Чтобы исходное требование  $\varphi$  оказалось невыполнимым для модели  $M$ , достаточно найти любую ветку вычислительного процесса  $M$ , на которой выполняется обратное требование  $\neg\varphi$ . Если такая ветка существует, она и будет искомым контрпримером. Обратное требование  $\neg\varphi$  выполнимо для композиции  $M \otimes \varphi$  на некотором ее вычислительном пути тогда и только тогда, когда в  $M \otimes \varphi$  присутствуют все уравнения из  $\neg\varphi$ .

#### Пример 2. Верификация модели.

**Дано.**

Модель системы ( $M$ ):

$$FA_1 = a \circ FA_1 + b \circ FA_2,$$

$$FA_2 = c \circ FA_1.$$

Проверяемое требование ( $\varphi$ ) LTL:

$$\varphi = Fc \text{ — когда-нибудь наступит «с»}.$$

Обратное требование ( $\neg\varphi$ ) LTL:

$$\neg\varphi = G\neg c \text{ — всегда будет «!с»}.$$

Обратное требование ( $\neg\varphi$ ) RLTL:

$$F_{\neg\varphi} = \neg c \circ F_{\neg\varphi}.$$

**Решение.**

Построение синхронной композиции  $M \otimes \neg\varphi$ :

$$\begin{aligned} P_1 &= FA_1 \otimes F_{\neg\varphi} = \\ &= (a \otimes \neg c) \circ (FA_1 \otimes F_{\neg\varphi}) + (b \otimes \neg c) \circ (FA_2 \otimes F_{\neg\varphi}) = \\ &= (a \otimes (\Delta \setminus c)) \circ (FA_1 \otimes F_{\neg\varphi}) + \\ &+ (b \otimes (\Delta \setminus c)) \circ (FA_2 \otimes F_{\neg\varphi}) = \\ &= a \circ (FA_1 \otimes F_{\neg\varphi}) + b \circ (FA_2 \otimes F_{\neg\varphi}) = a \circ P_1 + b \circ P_2. \end{aligned}$$

$$P_2 = FA_2 \otimes F_{\neg\varphi} = (c \otimes \neg c) \circ (FA_1 \otimes F_{\neg\varphi}) =$$

$$= \perp \circ FA_1 \otimes F_{\neg\varphi} = \perp.$$

Конечная синхронная композиция:

$$P_1 = a \circ P_1 + b \circ P_2,$$

$$P_2 = \perp.$$

**Результат:**  $P_1 = a \circ P_1$ .

Проверяемое свойство  $\neg\varphi$  выполняется на  $M$ , поскольку в  $M \otimes \neg\varphi$  в полной мере было реализовано свойство  $\neg\varphi$ , а именно, его уравнение на одном из путей вычислительного процесса. Из этого следует, что на этом пути не реализуется  $\varphi$ . Значит, путь  $P_1 \rightarrow a \rightarrow P_1 \rightarrow a \rightarrow P_1 \dots \rightarrow a \rightarrow P_1$  является контрпримером.

Рассмотрим более содержательный пример, в котором будет выполняться проверка работы дымового извещателя. Для наглядности на рисунке 3 представлена его модель в виде автомата Бюхи, где

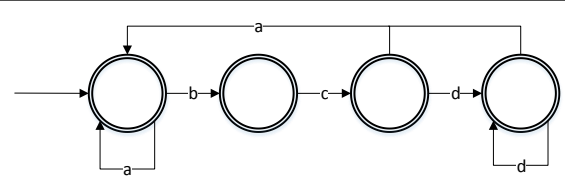


Рис. 3. Модель дымового извещателя на базе автомата Бюхи

Fig. 3. A smoke detector model based on Buchi automaton

$a$  – режим готовности;  $b$  – срабатывание датчика дыма;  $c$  – опрос соседних датчиков;  $d$  – режим тревоги.

Модель дымового извещателя на базе нотации RLTL будет иметь следующий вид:

$$FA_1 = a \circ FA_1 + b \circ FA_2,$$

$$FA_2 = c \circ FA_3,$$

$$FA_3 = d \circ FA_4 + a \circ FA_1,$$

$$FA_4 = d \circ FA_4 + a \circ FA_1.$$

Проверяемое требование ( $\varphi$ ) LTL:  $\varphi = GF(b \rightarrow Xc)$  – всегда, в случае наступления в будущем в некоторый момент времени события  $b$ , на следующем шаге будет выполнено событие  $c$ .

Обратное требование ( $\neg\varphi$ ) LTL:  $\neg\varphi = FG(b \wedge \neg Xc)$  – когда-нибудь наступит такой момент, что всегда будет наступать  $b$  на текущем шаге и на следующем шаге будет  $\neg c$ .

Обратное требование ( $\neg\varphi$ ) RLTL:

$$FB_1 = b \circ FB_3 + \Delta \circ FB_1,$$

$$FB_2 = b \circ FB_3,$$

$$FB_3 = \neg c \circ FB_2.$$

Построение синхронной композиции  $M \otimes \neg\varphi$ :

$$\begin{aligned} P_1 &= FA_1 \otimes FB_1 = (a \otimes b) \circ (FA_1 \otimes FB_3) + (b \otimes b) \circ (FA_2 \otimes FB_3) + \\ &+ (a \otimes \Delta) \circ (FA_1 \otimes FB_1) + (b \otimes \Delta) \circ (FA_2 \otimes FB_1) = \\ &= \perp \circ (FA_1 \otimes FB_3) + b \circ (FA_2 \otimes FB_3) + a \circ (FA_1 \otimes FB_1) + \\ &+ b \circ (FA_2 \otimes FB_1) = \\ &= b \circ (FA_2 \otimes FB_3) + a \circ (FA_1 \otimes FB_1) + b \circ (FA_2 \otimes FB_1) = \\ &= b \circ P_2 + a \circ P_1 + b \circ P_3. \end{aligned}$$

$$P_2 = (FA_2 \otimes FB_3) = (c \otimes \neg c) \circ (FA_3 \otimes FB_2) = \perp \circ (FA_3 \otimes FB_2) = \perp.$$

$$P_3 = (FA_2 \otimes FB_1) = (c \otimes b) \circ (FA_3 \otimes FB_3) +$$

$$+ (c \otimes \Delta) \circ (FA_3 \otimes FB_1) =$$

$$= \perp \circ (FA_3 \otimes FB_3) + c \circ (FA_3 \otimes FB_1) =$$

$$= c \circ (FA_3 \otimes FB_1) = c \circ P_4.$$

$$P_4 = (FA_3 \otimes FB_1) = (d \otimes b) \circ (FA_4 \otimes FB_3) +$$

$$+ (d \otimes \Delta) \circ (FA_4 \otimes FB_1) + (a \otimes b) \circ (FA_1 \otimes FB_3) +$$

$$+ (a \otimes \Delta) \circ (FA_1 \otimes FB_1) =$$

$$= \perp \circ (FA_4 \otimes FB_3) + d \circ (FA_4 \otimes FB_1) + \perp \circ (FA_1 \otimes FB_3) +$$

$$+ a \circ (FA_1 \otimes FB_1) = d \circ (FA_4 \otimes FB_1) + a \circ (FA_1 \otimes FB_1) =$$

$$= d \circ P_5 + a \circ P_1.$$

$$P_5 = (FA_4 \otimes FB_1) = (d \otimes b) \circ (FA_4 \otimes FB_3) +$$

$$+ (d \otimes \Delta) \circ (FA_4 \otimes FB_1) + (a \otimes b) \circ (FA_1 \otimes FB_3) +$$

$$+ (a \otimes \Delta) \circ (FA_1 \otimes FB_1) =$$

$$= \perp \circ (FA_4 \otimes FB_3) + d \circ (FA_4 \otimes FB_1) + \perp \circ (FA_1 \otimes FB_3) +$$

$$+ a \circ (FA_1 \otimes FB_1) = d \circ P_5 + a \circ P_1.$$

Конечная синхронная композиция:

$$P_1 = FA_1 \otimes FB_1 = b \circ \perp + a \circ (FA_1 \otimes FB_1) + b \circ (FA_2 \otimes FB_1) =$$

$$= a \circ P_1 + b \circ P_3.$$

$$P_3 = (FA_2 \otimes FB_1) = c \circ (FA_3 \otimes FB_1) = c \circ P_4.$$

$$P_4 = (FA_3 \otimes FB_1) = d \circ (FA_4 \otimes FB_1) + a \circ (FA_1 \otimes FB_1) =$$

$$= d \circ P_5 + a \circ P_1.$$

$$P_5 = (FA_4 \otimes FB_1) = d \circ (FA_4 \otimes FB_1) + a \circ (FA_1 \otimes FB_1) =$$

$$= d \circ P_5 + a \circ P_1.$$

**Результат.** Проверяемое свойство  $\varphi$  не выполняется на  $M$ , поскольку в композиции  $M \otimes S$  присутствуют не все уравнения из  $\varphi$ , а именно, нет уравнений  $F_{B2}$  и  $F_{B3}$ . Таким образом, исходное требование  $\varphi$  выполнимо на  $M$ .

### Заключение

В статье предложен и подробно рассмотрен новый алгоритм верификации для метода Model Checking на базе нотации RLTL для верификации формул логики линейного времени. Данный алгоритм во многом схож с алгоритмом верификации формул LTL, поскольку использует аналогичные принципы и подход к решению задачи верификации. Однако он имеет особенность – позволяет сократить число подготовительных этапов, необходимых для верификации, тем самым способствуя повышению производительности метода.

Работа предложенного алгоритма подробно рассмотрена на конкретных примерах с описанием всех этапов и их теоретической базы. Авторы считают, что он может выступить в качестве более производительной альтернативы существующему, а RLTL-нотация может использоваться в качестве основной структуры данных, на базе которой будут задаваться как модели верифицируемых систем,

так и требования к ним. В целом же такой подход должен способствовать общему повышению быстродействия метода Model Checking за счет снижения сложности алгоритма верификации и использования единой структуры данных.

### Литература

1. Карпов Ю.Г. Model Checking. Верификация параллельных и распределенных программных систем. СПб: БХВ-Петербург, 2010. 552 с.
2. Кораблин Ю.П. Семантика языков распределенного программирования; [под ред. В.П. Кутепова]. М.: Изд-во МЭИ, 1996. 102 с.
3. Kroger F., Merz S. Temporal logic and state systems. Springer, 2008, 436 p.
4. Manna Z., Pnueli A. The temporal logic of reactive and concurrent systems: specification. Springer, 1992, 427 p.
5. Кларк Э.М., Грамберг О., Пелед Д. Верификация моделей программ. Model Checking. М.: Изд-во МЦНМО, 2002. 416 с.
6. Gerard J. Holzman. An analysis of bitstate hashing. Proc. 15th Int. Conf. on Protocol Specification, Testing, and Verification, 1998, pp. 301–314.
7. Ernst-Rudiger Olderog, Krzysztof R. Apt. Fairness in parallel programs: the transformational approach. ACM Trans. Program. Lang. Syst., 1988, vol. 10, no. 3, pp. 420–455.
8. Шипов А.А., Кораблин Ю.П. Построение моделей систем на базе эквациональной характеристики формул LTL // Программные продукты и системы. 2017. № 1. С. 61–66.
9. Шипов А.А., Кораблин Ю.П. Эквационная характеристика формул LTL // Программные продукты и системы. 2015. № 4. С. 175–179.

Software &amp; Systems

DOI: 10.15827/0236-235X.119.456-460

Received 05.06.17

2017, vol. 30, no. 3, pp. 456–460

### SYSTEM MODEL VERIFICATION BASED ON EQUATIONAL CHARACTERISTICS OF LTL FORMULAS

**Yu.P. Korablin**<sup>1</sup>, Dr.Sc. (Engineering), Professor, y.p.k@mail.ru

**A.S. Kochergin**<sup>1</sup>, Postgraduate Student, kocherginaalexandr@gmail.com

**A.A. Shipov**<sup>1</sup>, Ph.D (Engineering), a-j-a-1@yandex.ru

<sup>1</sup> Russian State Social University, Vilgelm Pik St. 4, Moscow, 129226, Russian Federation

**Abstract.** For a long time, the Model Checking method has been widely used in the field related with software and technical systems quality evaluation. Such top IT companies as Intel, Microsoft, Amazon, etc. actively use it in the processes of development and maintenance of their products. Such a success of this method is certainly not accidental. It helped to solve a lot of problems in the field of verification, namely: the problems of unified representation of software and technical systems, the problem of formal requirements representation, the automation of verification phases, the verification of large distributed software systems, etc.

However, such challenges as the continuous development of modern technologies and the growth rates of modern software systems complexity can become an insurmountable obstacle for effective verification with Model Checking. Therefore, it is necessary to make permanent improvements of its theory and tools.

The article demonstrates a new verification algorithm of Linear Temporal Logic formulas by means of Model Checking based on the new RLTL notation (Recursive Linear Temporal Logic), which is a recursive representation of LTL formulas. This algorithm can avoid necessary conversions into Büchi automata of the system model and verifying statements since RLTL can be used to define both of them. This allows beginning a verification process immediately and increases Model Checking efficiency.

**Keywords:** verification, Model Checking, RLTL the equational characteristics of LTL, Kripke structure, Buchi automaton, temporal logic formula, LTL, CTL.

### References

1. Karpov Yu.G. Model Checking. Verifikatsiya paralelnykh i raspredelennykh programmnykh sistem [Model Checking. Verification of Parallel and Distributed Software Systems]. BHV-Peterburg Publ., 2010, 552 p.
2. Korablin Yu.P. Semantika yazykov raspredelennogo programmirovaniya [Semantics of Distributed Programming Languages]. V.P. Kutevov (Ed.). Moscow, MEI Publ., 1996, 102 p.
3. Kroger F., Merz S. Temporal Logic and State Systems. Springer Publ., 2008, 436 p.
4. Manna Z., Pnueli A. The Temporal Logic of Reactive and Concurrent Systems: Specification. Springer Publ., 1992, 427 p.
5. Klark E.M., Gramberg O., Peled D. Verifikatsiya modeley programm. Model Checking [Verification of Program Models. Model Checking]. MTsNMO Publ., 2002, 416 p.
6. Holzman G.J. An Analysis of Bitstate Hashing. Proc. 15th Int. Conf. Protocol Specification, Testing, and Verification. 1998, pp. 301–314.
7. Ernst-Rudiger Olderog, Krzysztof R. Apt. Fairness in Parallel Programs: The Transformational Approach. ACM Trans. on Programming Languages and Systems. 1988, vol. 10, no. 3, pp. 420–455.
8. Shipov A.A., Korablin Yu.P. System models construction based on LTL formula equational characteristics. Programmnye produkty i sistemy [Software & Systems]. 2017, no. 1, pp. 61–66 (in Russ.).
9. Shipov A.A., Korablin Yu.P. Equational characteristics of LTL formulas. Programmnye produkty i sistemy [Software & Systems]. 2015, no. 4 (112), pp. 175–179 (in Russ.).

УДК 004.048

DOI: 10.15827/0236-235X.119.461-468

Дата подачи статьи: 15.05.17

2017. Т. 30. № 3. С. 461–468

## РАЗРАБОТКА СИСТЕМЫ РАЗРЕШЕНИЯ АНАФОРЫ НА ОСНОВЕ МЕТОДОВ МАШИННОГО ОБУЧЕНИЯ

А.В. Соколов, магистрант, a23sokolov@gmail.com

(Новосибирский государственный университет, ул. Пирогова, 2, г. Новосибирск, 630090, Россия);

Т.В. Батура, к.ф.-м.н., ведущий научный сотрудник, tatiana.v.batura@gmail.com

(Новосибирский государственный университет, ул. Пирогова, 2, г. Новосибирск, 630090, Россия);

Институт систем информатики им. А.П. Ершова СО РАН,

просп. Акад. Лаврентьева, 6, г. Новосибирск, 630090, Россия)

В работе предложен и реализован метод разрешения анафоры местоимений третьего лица в текстах на русском языке.

Задача нахождения искомых пар «анафор–антецедент» рассматривается как задача бинарной классификации. Предварительно осуществляется морфологический и синтаксический анализ текста. Для морфологического анализатора использовалась библиотека *ru morphology2*. Синтаксический анализ производился при помощи *MaltParser*. Непосредственно сам алгоритм разрешения анафоры состоит из трех этапов. Сначала происходит поиск всех местоимений, затем составляется множество потенциальных антецедентов и осуществляется выбор наиболее подходящего кандидата. Компонент создания множества кандидатов в антецеденты основан на применении дистанционного, морфологического и синтаксического фильтров. В качестве алгоритма классификации выбран алгоритм машинного обучения *Random Forest*. Классификатор выбора наиболее вероятного кандидата учитывает 78 различных признаков.

Проведены исследования по проверке эффективности данного метода, доказывающие применимость разработанного подхода. По результатам тестирования можно заметить, что качество анализатора улучшается, когда не учитывается морфологический фактор падежа. Также можно отметить, что на наборе признаков, не содержащем падежи, меньшее влияние на конечный результат оказывает количество взятых для расчета деревьев.

При создании систем разрешения анафоры основные проблемы заключаются в следующем. Во-первых, поиск анафорических отношений лежит в области семантики и поэтому трудно поддается формализации. Во-вторых, существуют особенности русского языка, такие как развитая морфология, морфологические и синтаксические неоднозначности, которые отрицательно сказываются на результате.

**Ключевые слова:** анафора, антецедент, методы классификации, машинное обучение, анафорический классификатор, анализ текстовой информации.

Из-за увеличивающихся объемов информации большую значимость приобретают такие задачи, как извлечение информации, машинный перевод и информационный поиск, для которых важна автоматизация тех или иных аспектов референции. Но, несмотря на актуальность данной темы, развитых разработок для русского языка существует не так много, и они значительно уступают по качеству ручной работе лингвистов. В связи с этим необходимо создание новых, более эффективных методов решения данной проблемы.

**Анафора** – использование языковых выражений, которые могут быть проинтерпретированы лишь с учетом другого, как правило предшествующего, фрагмента текста. В данном случае речь идет не о повторе всего выражения в целом, а о сокращенном воспроизведении в тексте того, что было представлено ранее в виде полного языкового выражения. Например, *Космонавт вернулся на борт станции. Он сообщил, что чувствует себя нормально*. Антецедентом называется первый член анафорического отношения, а анафором (или анафорическим элементом) – второй. В приведенном примере антецедентом является слово «космонавт», а анафором – местоимение «он».

На основании того, какой частью речи выражен анафор, выделяют именную и местоименную анафоры. В случае именной анафоры анафор выражен

именной группой. Например, *Юрий Гагарин вернулся на борт станции. Космонавт сообщил, что чувствует себя нормально*. Если анафор выражен местоимением или местоименным словом, то такой вид анафоры является местоименным. Пример местоименной анафоры уже был приведен.

Процесс создания системы разрешения анафоры трудоемкий, так как существуют несколько проблем: во-первых, поиск анафорических отношений лежит в области семантики и поэтому трудно поддается формализации; во-вторых, русский язык имеет такие особенности, как развитая морфология, морфологические и синтаксические неоднозначности и прочее. Как справедливо отмечено в работе [1], при анализе текстов наибольшую трудность представляет не разбор слов по отдельности, с этим отлично справляются большинство морфологических анализаторов для большинства естественных языков, а анализ отношений, то есть связей между словами.

В настоящее время ведется работа над созданием системы автоматического разрешения анафоры для местоимений третьего лица, поскольку именно этот вид является наиболее распространенным. Обработка текста в данной системе осуществляется в несколько этапов: морфологический анализ, синтаксический анализ и непосредственно разрешение анафоры.

### Существующие средства автоматического разрешения анафоры

Один из подходов к разрешению анафоры, основанный на правилах, предложен в работе [2]. Основная идея состоит в том, чтобы, помимо разметки частей речи, учитывать информацию об именных группах, предшествующих анафоре на расстоянии двух предложений. Выбираются только те именные группы, которые согласуются с анафорой в роде и числе. Далее к ним последовательно применяется набор правил.

Для русского языка принцип обнаружения анафорических отношений с использованием правил описан в работе [3]. Однако исследование в основном направлено на изучение типов замещения, использующихся в различных видах общественно-политических текстов, а не на сравнение точности определения анафорических связей.

Для решения задачи определения анафоры в работах [4–6] описаны эксперименты с применением методов машинного обучения, в частности метода опорных векторов. Так, в работе [5] отмечается: если, помимо SVM, дополнительно использовать набор правил, то достигается наилучшая точность 0,52–0,59. В исследовании [6] обнаружено, что дополнительные знания о семантических ролях анафоры и антецедента способны улучшить качество решения задачи на 0,1–6,6 %.

В [7] описан подход к разрешению местоименной анафоры при анализе мнений пользователей на основе машинного обучения. Авторы использовали 16 признаков, разделив их на три категории: анафорическое местоимение, кандидат в антецеденты и характеристика отношения. Исследование проводилось для баскского языка на сравнительно небольшом корпусе, содержащем 50 000 слов и 249 анафорических местоимений. В эксперименте сравнивались различные методы машинного обучения: метод опорных векторов (SVM), многослойный перцептрон (MLP), метод Байеса (NB), метод k-ближайших соседей (kNN), ансамбль решающих деревьев (RF), комбинированный подход на основе метода Байеса и деревьев решений (NB-Tree).

Сложность автоматического разрешения анафоры заключается в том, что процесс предварительной обработки текста, как правило, включает в себя последовательность действий: определение частей речи и морфологических характеристик слов, выделение именных групп, синтаксический и поверхностно-семантический разбор. Каждый шаг удается автоматизировать с определенной точностью, погрешности накапливаются, а это, в свою очередь, плохо сказывается на конечном результате. Поэтому повышение качества работы подобных алгоритмов затруднительно. В то же время интерес к данной проблеме в последние годы не ослабевает как в зарубежной, так и в отечественной научной среде.

Авторами данной статьи предложен и реализован метод разрешения анафоры местоимений третьего лица, проведено экспериментальное исследование для оценки эффективности предложенного метода.

### Построение частей системы, отвечающих за морфологический и синтаксический анализ

Для морфологического анализатора была использована распространенная библиотека *morphy2* [8], написанная на языке Python (версии 2.7 и 3.3+). Решающими факторами при выборе библиотеки были поддержка русского языка, высокая скорость работы (100 000 слов/сек.) и использование словаря *OpenCorpora* для реализации.

Библиотека предоставляет широкий функционал:

- приводит слово к нормальной форме, например, «люди → человек», «гулял → гулять»;
- ставит слово в нужную форму, например, заменяя форму единственного числа на множественное, меняет падежные формы слова и т.д.;
- возвращает грамматическую информацию о слове (число, род, падеж, часть речи и т.д.).

Опишем процесс построения синтаксического анализатора подробнее.

#### Построение синтаксического анализатора.

Существуют два основных подхода к созданию синтаксических парсеров: подход, основанный на правилах (*rule based*), и машинное обучение с учителем (*supervised machine learning*).

Рассмотрим каждый из них подробнее. Метод, в основе которого лежат правила, дает наибольшую точность, вследствие чего он применяется в большинстве коммерческих систем. Основная идея заключается в создании набора правил, позволяющих определить, как проставлять связи в предложении. Например, в предложении *Папа мыл машину* можно применять следующие правила (предположим, что мы уже сняли все неоднозначности и точно знаем грамматические категории слов).

1. Слово «*мыл*» имеет характеристики: *единственное число, прошедшее время, мужской род*. Вершина предложения (в английской терминологии – *root*).

2. Слово «*папа*» имеет характеристики: *именный падеж, мужской род, единственное число*. Должно зависеть от глагола «*мыл*», тип связи между словами должен быть «субъект».

3. Слово «*машину*» имеет характеристики: *винительный падеж, женский род, единственное число*. Должно зависеть от глагола «*мыл*», тип связи должен быть «объект».

В системе может быть множество подобных приведенному выше правил, а также антиправил, указывающих, когда не нужно проставлять связи. Например, если у существительного и глагола раз-



личаются род или число, то между ними нет связи. Также существуют правила третьего типа, которые указывают, какую пару слов следует предпочесть, если возможны несколько вариантов. Например, в предложении *Папа мыл окно* и «*папа*», и «*окно*» могут выступать в качестве субъекта, однако мы можем предпочесть предстоящее глаголу слово, а не последующее. Из описанного выше можно заметить, что такой подход очень ресурсоемок, так как для создания парсера необходима хорошая команда лингвистов, которой потребуется буквально описать весь русский язык. Поэтому в работе будем применять второй подход – машинное обучение с учителем.

Идея парсера, использующего машинное обучение, заключается в следующем. На вход классификатору подается много примеров с правильными ответами, на которых система должна обучиться самостоятельно. Чтобы обучить синтаксические классификаторы, в качестве данных для обучения используют специально размеченные корпуса, коллекции текстов, в которых размечена синтаксическая структура. Предложение, взятое для примера, может быть размечено так:

- |   |        |                     |   |         |
|---|--------|---------------------|---|---------|
| 1 | Папа   | сущ.им.ед.муж.      | 2 | субъект |
| 2 | мыл    | глагол.ед.муж.прош. | 0 | –       |
| 3 | машину | сущ.вин.ед.жен.     | 2 | объект  |

В этом формате каждая строка описывает отдельное слово в виде записей. Для каждого слова нужно хранить следующие данные:

- номер слова в предложении (1);
- словоформа (папа);
- грамматические категории (сущ.им.ед.муж.);
- номер главного слова (2);
- тип связи (субъект).

Существует несколько открытых парсеров, которые можно обучить для работы с русским языком. Был выбран один из них, а именно MaltParser [9, 10]. Этот пакет включает в себя различные алгоритмы синтаксического анализа, в том числе алгоритмы Нивре и Ковингтона, а также реализации нескольких методов машинного обучения для предсказателей переходов. Благодаря своей эффективности и производительности MaltParser на данный момент является одним из наиболее широко используемых синтаксических анализаторов. Исследования по применению MaltParser проводились для многих языков, в том числе и для русского [11, 12]. Пакет может использоваться с различными алгоритмами обучения, с версии 1.3 по умолчанию поддерживаются

два встроенных алгоритма обучения: LIBSVM [13] и LIBLINEAR [14]. По результатам обучения и тестирования второй алгоритм проявил себя заметно лучше, поэтому впоследствии все вычисления производились с ним.

Для обучения синтаксического классификатора использовался размеченный корпус СинТагРус, который входит в состав НКРЯ (данный корпус закрыт и предоставляется только для исследовательских целей). Корпус представляет собой набор текстов, размеченных в формате XML, с информацией о принадлежности каждого слова к части речи, соответствующих грамматических признаках, нормальной форме слова, синтаксических связях внутри предложения.

Для обучения синтаксического анализатора необходимо перевести размеченные данные в формат maltpab. Для этих целей был написан скрипт, который читает XML и выдает нужный формат, при этом нормализуя грамматические категории. Результат преобразования следующий:

Здесь	ADV	4	обст
уместнее	ADV.comp	4	обст
всего	S.gen.n.sg	2	сравнит
провести	VINF	0	ROOT
аналогию	S.acc.f.sg	4	1-компл
с	PR	5	1-компл
законами	S.ins.m.pl	6	предл
физики	S.f.gen.sg	7	квазиагент

Обучение MaltParser осуществляется на данных такого вида. В результате обучения получается файл \*.mco, который содержит обученную модель и необходимые конфигурационные данные. Этот файл будет использоваться классификатором для синтаксического анализа неразмеченных данных. Общая схема анализатора и вспомогательных модулей представлена на рисунке 1.

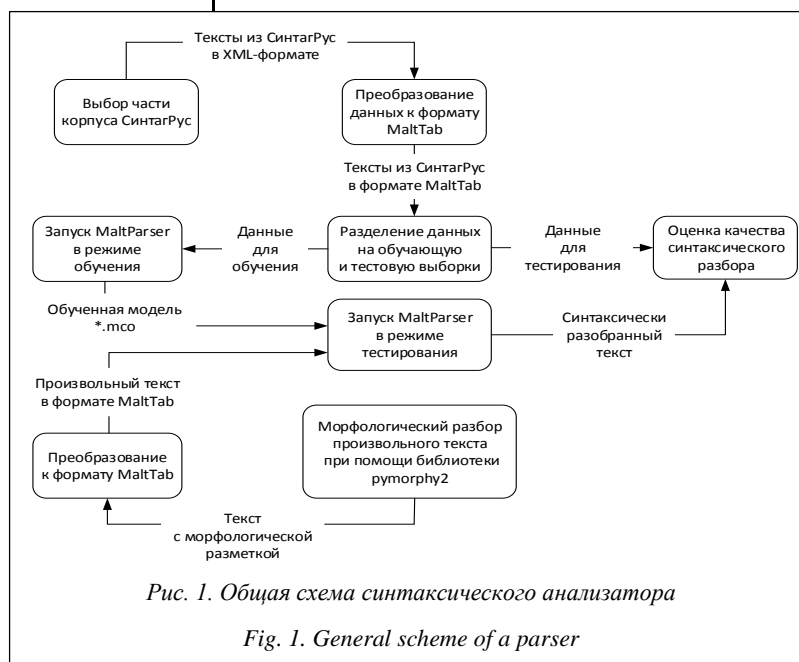


Fig. 1. General scheme of a parser

Одним из ключевых факторов, влияющих на оценку качества синтаксического анализатора, является результат морфологического парсера с использованием *rumorphy2*. Например, слово «ученый» в зависимости от контекста может быть как прилагательным, так и существительным, то есть является субстантивированным прилагательным (когда прилагательные переходят в разряд существительных). В подобных случаях неправильно определенная часть речи на уровне морфологического анализа влечет за собой ошибки на синтаксическом уровне. Также есть вероятность встретить в текстах слова-омонимы, частным случаем которых являются омоформы. Яркий пример – слово «печь», которое в зависимости от контекста выступает как существительное или глагол. Приняв ошибки такого рода, на основе предложенной модели получим результаты, представленные в таблице 1.

В таблице показаны результаты синтаксического анализатора *MaltParser*, обученного на 1/7 части корпуса *СинТагРус*. Для оценки качества результатов произведено несколько замеров, в таблице отражены усредненные значения. Для оценки полученной модели произведены замеры на морфологической разметке *СинТагРус*, которые дали неплохие результаты. Затем произведены замеры с использованием морфологической разметки из *rumorphy2*. На основании результатов, отраженных в таблице 1, можно сделать вывод, что точность синтаксической разметки ухудшается.

### Построение анафорического классификатора

В первую очередь происходит предобработка входного текста: он разбивается на предложения, после этого каждое предложение анализируется морфологическим и синтаксическим парсерами, результаты анализа для предложений сохраняются в файл и используются на следующих этапах. Затем анализатор просматривает текст слева направо, останавливаясь на каждом найденном местоимении. Для каждого местоимения текст просматривается в обратном направлении, в процессе чего составляется множество потенциальных антецедентов (кандидатов). В качестве кандидатов выступают морфологические интерпретации слов, то есть само слово и его морфологические характери-

стики. Как только такое множество составлено, анализатор продолжает просмотр текста в поисках следующего местоимения, затем анализатор выбирает из набора признаков единственного кандидата, которого считает верным антецедентом для данного местоимения. Таким образом, описываемый метод состоит из следующих этапов:

- поиск местоимения;
- составление множества потенциальных антецедентов;
- выбор наиболее подходящего кандидата.

**Создание множества кандидатов.** Основная цель данного этапа – создание множества морфологических интерпретаций, которые могут при некоторых условиях быть антецедентом для данного местоимения. Из полученного множества на следующих этапах выбирается единственный антецедент, он и будет признан верным. Такое множество должно удовлетворять критерию полноты, содержать в себе верный антецедент, чтобы у алгоритма выбора на последующих этапах была возможность его выбрать. В то же время представляется разумным ограничить пространство выбора и не включать в множество заведомо неподходящих кандидатов. Данный этап будем рассматривать как набор фильтров, которые будут применяться ко всем морфологическим интерпретациям, предшествующим в тексте местоимению. Каждый из фильтров должен либо отфильтровать кандидата, либо оставить его в рассматриваемом множестве.

**Фильтр расстояния.** Зачастую местоимение и антецедент находятся достаточно близко друг к другу. Хотя при тестировании и написании этого правила были зафиксированы случаи, когда антецедент и местоимение находились на расстоянии 8 предложений, тем не менее, большинство антецедентов укладываются в довольно маленькую окрестность местоимения. После проведения ряда экспериментов было решено ограничить рассматриваемое множество предложений на поиск претендентов до трех.

**Морфологический фильтр.** Морфологический фильтр, реализованный в данной работе, можно сформулировать с помощью набора правил:

- претендентами могут стать только существительные;
- кандидат и претендент должны быть согласованы в роде и числе.

Таблица 1

Результаты морфологического анализатора

Table 1

The results of a morphological parser

Выборка	Морфологическая разметка СинТагРус		Морфологическая разметка Rumorphy2	
	Точность (accuracy, %)	Время (сек.)	Точность (accuracy, %)	Время (сек.)
<b>Выборка 2014 года</b> Количество слов = 58 617 Количество предл. = 4 182	75	6,19	63	6,25
<b>Выборка 2015 года</b> Количество слов = 4 926 Количество предл. = 399	76	2,85	65	2,45

**Синтаксический фильтр.** Синтаксический фильтр, реализованный в данной работе, основан на обобщенных синтаксических ограничениях из статьи [15]. В работе показана их эффективность. Ограничения сформулированы как набор условий на взаимное расположение в синтаксическом дереве местоимения и кандидата. Условия описывают невозможные для местоимения и антецедента отношения, то есть, если хотя бы одно из условий выполняется, кандидат отвергается.

Для упрощения работы фильтра было решено объединить эти правила в одно. Чтобы претендент удовлетворял условиям фильтра, нужно, чтобы антецедент и местоимения принадлежали к разным Root-группам.

Пример, удовлетворяющий условиям фильтра:

Маша любит мороженое. Потому что оно вкусное.

1) Глагольная группа *любит*.

2) Root-группа *вкусное*.

Пример, не удовлетворяющий условиям фильтра:

Маше она нравится.

1) Глагольная группа *нравится*.

2) Root-группа *нравится*.

На этом этапе получен набор претендентов, удовлетворяющих условиям фильтра, это множество имеет небольшие размеры для конкурентоспособной работы классификатора.

**Преобразование набора признаков к векторному виду.** Затем на полученном множестве кандидатов ставится задача преобразования множества признаков в векторную форму. Набор признаков переводится в векторный формат по определенным правилам.

**Расстояние в словах.** Для каждого претендента вычисляется расстояние в словах до местоимения. В зависимости от этого расстояния вектор заполняется единицами. Были выделены три расстояния для фиксации их в векторе:

- от 10 слов включительно; вектор [1, 0, 0];
- от 10 до 30 слов включительно; вектор [0, 1, 0];
- от 30 слов; вектор [0, 0, 1].

Претенденту может соответствовать только один вектор с описанием в векторной форме.

**Морфологические признаки.** Из морфологических признаков, используемых в модели, для классификации учитывается только падеж:

- именительный [1, 0, 0, 0, 0, 0];
- родительный [0, 1, 0, 0, 0, 0];
- дательный [0, 0, 1, 0, 0, 0];
- винительный [0, 0, 0, 1, 0, 0];
- творительный [0, 0, 0, 0, 1, 0];
- предложный [0, 0, 0, 0, 0, 1].

**Синтаксические признаки.** В СинТагРус представлены четыре группы синтаксических отношений (СинтОтн) [16].

1. Актантные СинтОтн. Главной особенностью актантных СинтОтн является то, что они

связывают предикатное слово [X] со словом [Y], заполняющим некоторую синтаксическую валентность этого предикатного слова. Всего 25 вариантов отношений.

2. Атрибутивные СинтОтн. Главной особенностью атрибутивных СинтОтн является то, что они связывают некоторое слово [X] со словом [Y], которое выражает при X значение невалентного атрибута – в самом широком смысле этого слова. Всего 31 вариант отношений.

3. Сочинительные СинтОтн. В принятой в данном корпусе системе синтаксических отношений сочинительные отношения принципиально не отличаются от подчинительных – и те, и другие связывают главный член конструкции с зависимым. Всего 5 вариантов отношений.

4. Служебные СинтОтн. Служебные СинтОтн связывают два элемента, синтаксически тесно связанные друг с другом. Часто члены таких конструкций – фактически части одного сложного слова. Всего 8 вариантов отношений.

Суммарно мы получаем набор из 69 видов отношений. Так как каждое слово может иметь только один тип отношений в предложении, вектор синтаксических признаков X будет иметь вид  $X = [x_1,$

$x_2, \dots, x_{69}]$ , где  $x_i \in \{0, 1\}$  и  $\sum_{i=1}^{69} x_i = 1$ .

Порядковый номер  $i$  определяется с помощью позиции в массиве отношений для каждого слова в зависимости от его синтаксических отношений внутри предложения.

Далее ставится задача комбинирования всех перечисленных признаков путем конкатенации векторов. Итоговый вектор Z представляется в виде

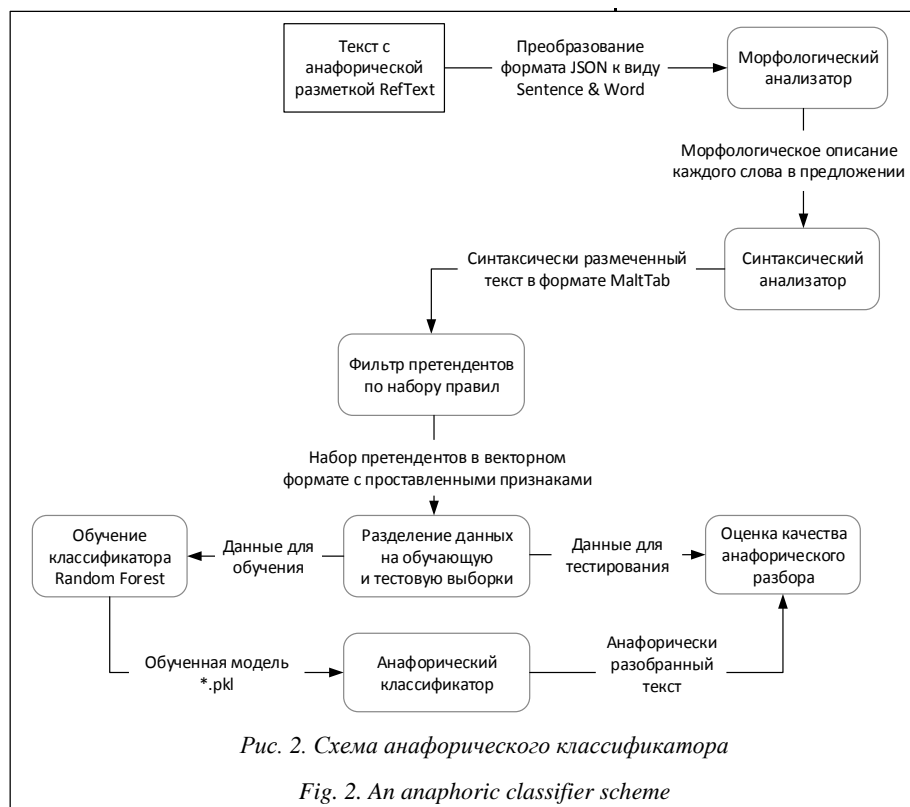
$Z = [z_1, z_2, \dots, z_{78}]$ , где  $z_i \in \{0, 1\}$  и  $\sum_{i=1}^{78} z_i = 3$ .

Таким образом, суммарное количество рассматриваемых признаков равно 78.

**Описание работы классификатора.** Для обучения анафорического классификатора использовался анафорически размеченный корпус текстов на русском языке, разработанный в Институте системного анализа ФИЦ ИУ РАН [6].

На основе размеченного корпуса текстов составляется обучающая выборка и строится бинарный классификатор, относящий с некоторой вероятностью кандидата к одному из двух классов: является антецедентом, не является антецедентом.

Обучающая выборка составляется следующим образом. Размеченные тексты просматриваются анализатором, для каждого местоимения составляется множество потенциальных антецедентов, для каждого кандидата из множества вычисляются описанные выше признаки. Далее вектор признаков каждого из кандидатов заносится в обучающую выборку с классом 1, если данный кандидат является антецедентом, и с классом 0 в противном случае.



классификации решение о выборе лучшего признака, по которому будет происходить разбиение, выбирается голосованием по большинству:

$$f(x) = \text{sign} \frac{1}{K} \sum_{k=1}^K g_k(x).$$

Данный подход к построению ансамбля моделей иногда называют бэггингом (bagging от Bootstrap Aggregating). Основная сложность его применения заключается в том, чтобы построенные модели действительно оказались независимыми.

Общая схема работы анафорического классификатора представлена на рисунке 2.

### Полученные результаты и выводы

В качестве алгоритма классификации был выбран алгоритм машинного обучения «решающий лес» (Random Forest) [17], основанный на деревьях принятия решений.

Сам по себе процесс построения каждого дерева можно описать следующим образом. Рассмотрим выборку  $X$ . Пусть  $x_j$  –  $j$ -й признак;  $t$  – порог значения признака;  $x_j \leq t$  – условие разбиения;  $Q(X, j, t)$  – критерий, характеризующий ошибку разбиения. Требуется найти наилучшие параметры  $j$  и  $t$ , при которых ошибка разбиения будет минимальной.

Решающий лес (Random Forest) – ансамблевый алгоритм, представляющий собой множество решающих деревьев. Идея случайного леса состоит в том, чтобы выбрать наилучший признак  $j$  не из всех возможных признаков, а из случайного подмножества признаков некоторой заданной размерности. Выбор этого подмножества признаков осуществляется каждый раз при очередном разбиении вершины. Таким образом, каждое дерево  $g_k(x)$  строится на своей обучающей выборке  $X'$ . В задаче

Суммарная точность считалась как процент правильно предсказанных антецедентов для местоимений, на которых производилось тестирование (табл. 2). Исследования проводились на следующих наборах признаков: Набор 1. Все признаки; Набор 2. Без падежей.

Проведенное тестирование показало, что описанный в работе метод может успешно применяться для разрешения местоименной анафоры, а также то, что качество анализатора улучшается, когда не учитывается морфологический фактор падежа.

### Заключение

В данной работе предложен и реализован метод разрешения анафоры местоимений третьего лица в текстах на русском языке. Разработан алгоритм разрешения местоименной анафоры, включающий в себя следующие компоненты:

### Результаты анафорического классификатора

Таблица 2

#### The results of an anaphoric classifier

Table 2

Количество деревьев	Аккуратность (accuracy, %)		F-мера (f1 score, %)		Точность (precision, %)		Полнота (recall, %)	
	Набор 1	Набор 2	Набор 1	Набор 2	Набор 1	Набор 2	Набор 1	Набор 2
1	57	64	53	58	45	52	64	66
5	64	64	56	59	55	53	57	67
10	62	64	55	56	50	52	60	66

– компонент создания множества кандидатов в антецеденты, основанный на применении набора фильтров: дистанционного, морфологического, синтаксического;

– классификатор выбора наиболее вероятного кандидата из множества предложенных на основе ряда признаков: дистанционного, морфологического, синтаксического.

Были проведены исследования по проверке эффективности данного метода, доказывающие применимость разработанного подхода.

### **Благодарность**

Авторы работы выражают благодарность коллективу создателей синтаксически размеченного корпуса СинТагРус, разработанного в Институте проблем передачи информации РАН, и коллективу создателей анафорически размеченного корпуса текстов на русском языке, разработанного в Институте системного анализа РАН.

### **Литература**

1. Шелманов А.О. Исследование методов автоматического анализа текстов и разработка интегрированной системы семантико-синтаксического анализа: дисс. ... канд. тех. наук. М.: 2015. 210 с.
2. Barbu C., Mitkov R. Evaluation tool for rule-based anaphora resolution methods. Proc. 39th Annual Meeting on Association for Computational Linguistics, 2001, pp. 34–41.
3. Абрамов В.Е., Абрамова Н.Н., Некрасова Е.В., Росс Г.Н. Статистический анализ связности текстов по общественно-политической тематике // Электронные библиотеки: перспективные методы и технологии, электронные коллекции: тр. 13 Всерос. науч. конф. Воронеж, 2011. С. 127–133.
4. Толпегин П.В. Автоматическое разрешение кореференции местоимений третьего лица русскоязычных текстов: автореф. дисс. ... канд. тех. наук. М., 2008. 241 с.
5. Protopopova E.V., Bodrova A.A., Volskaya S.A., Krylova I.V., Chuchunkov A.S., Alexeeva S.V., Bocharov V.V., Granovsky D.V. Anaphoric Annotation and Corpus-Based Anaphora Resolution: An Experiment, Computational Linguistics and Intellectual Technologies // Диалог-2014: сб. тр. Междунар. науч. конф. по компьютер. лингвистике. 2014. Вып. 13. С. 562–571 (англ.).
6. Kamenskaya M.A., Khramoin I.V., Smirnov I.V. Data-driven Methods for Anaphora Resolution of Russian // Диалог-2014: сб. тр. Междунар. науч. конф. по компьютер. лингвистике. 2014. Вып. 13. С. 241–250 (англ.).
7. Arregi O., et al. Determination of features for a machine learning approach to pronominal anaphora resolution in basque. Procesamiento del Language Natural, 2010, no. 45, pp. 291–294.
8. Korobov M. Morphological analyzer and generator for russian and ukrainian languages. Analysis of Images, Social Networks and Texts. 2015, pp. 320–332.
9. Nivre J., Hall J., Nilsson J. et al. MaltParser: A language-independent system for data-driven dependency parsing. Natural Language Engineering. 2007, vol. 13, no. 2, pp. 95–135.
10. Nivre J., Hall J., Nilsson J. MaltParser: A data-driven parser-generator for dependency parsing. Proc. Intern. Conf. LREC, 2006, vol. 6, pp. 2216–2219.
11. Sharoff S., Nivre J. The proper place of men and machines in language technology: Processing Russian without any linguistic knowledge // Диалог-2011: сб. тр. Междунар. науч. конф. по компьютер. лингвистике. 2014. Вып. 10. С. 591–604 (англ.).
12. Смирнов И.В., Шелманов А.О., Кузнецова Е.С., Храмоин И.В. Семантико-синтаксический анализ естественных языков. Ч. II. Метод семантико-синтаксического анализа текстов // Искусственный интеллект и принятие решений. М.: Изд-во ИСА РАН, 2014. № 1. С. 11–24.
13. Chang C.C., Lin C.J. LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology, 2011, vol 2, iss. 3, art. no. 27.
14. Fan R.E., Chang K.W., Hsieh C.J., Wang X.R. and Lin C.J. LIBLINEAR: A library for large linear classification Jour. of Machine Learning Research. 2008, vol. 9, pp. 1871–1874.
15. Lappin S., Leass H.J. An algorithm for pronominal anaphora resolution. Computational Linguistics. 1994, vol. 20, no. 4, pp. 535–561.
16. Национальный корпус русского языка. URL: <http://www.ruscorpora.ru/instruction-syntax.html> (дата обращения: 14.05.2017).
17. Breiman L. Random forests. Machine learning. 2001, vol. 45, no. 1, pp. 5–32.

Software & Systems

DOI: 10.15827/0236-235X.119.461-468

Received 15.05.17

2017, vol. 30, no. 3, pp. 461–468

## **ANAPHOR RESOLUTION SYSTEM DEVELOPMENT BASED ON MACHINE LEARNING METHODS**

**A.V. Sokolov**<sup>1</sup>, Graduate Student, [a23sokolov@gmail.com](mailto:a23sokolov@gmail.com)

**T.V. Batura**<sup>1,2</sup>, Ph.D. (Physics and Mathematics), Leading Researcher, Senior Researcher, [tatiana.v.batura@gmail.com](mailto:tatiana.v.batura@gmail.com)

<sup>1</sup> Novosibirsk State University, Pirogov St. 2, Novosibirsk, 630090, Russian Federation

<sup>2</sup> 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 proposes and implements a method for the anaphora resolution of third person pronouns in Russian texts.

The problem of finding the true pairs “anaphor-antecedent” is considered as a binary classification problem. Initially, the authors perform morphological and syntactic analysis of the text. The morphological analyzer used the pymorphy2 library. The parsing has been performed using MaltParser. The algorithm of anaphora resolution itself consists of three stages. First stage includes searching for all pronouns, then there is a compilation of many potential antecedents, and finally the most suitable candidate is selected. The component of creating a set of candidates for antecedents is based on using distance, morphological and syntactic filters. Classification uses the Random Forest algorithm. The anaphoric classifier takes into account 78 different features.

The authors performed a series of experiments in order to prove the effectiveness of the proposed method. They showed that the quality of the analyzer improves if we do not take into account the morphological case. It can also be noted that the number of trees taken for calculation has a lesser effect on the final result when taking a feature set without cases.

The paper considers the main difficulties in developing the anaphora resolution systems. First, the search for anaphoric relations is in the semantic domain, and therefore it is difficult to formalize. Second, there are some features of the Russian language, such as developed morphology, morphological and syntactic ambiguities, which adversely affect the result.

**Keywords:** anaphora, antecedent, natural language processing, classification methods, machine learning, anaphoric classifier, analysis of text information.

**Acknowledgements.** The authors express their gratitude to the team of creators of the syntactically marked corpus SinTagRus, which has been developed at the Institute for Information Transmission Problems of the Russian Academy of Sciences, and to the team of creators of the Anaphorically Marked Corpus of Texts in Russian developed at the Institute of System Analysis of the Russian Academy of Sciences.

### References

1. Shelmanov A.O. *Issledovanie metodov avtomaticheskogo analiza tekstov i razrabotka integrirovannoy sistemy semantiko-sintaksicheskogo analiza* [A Research on Automatic Text Analysis Methods and a Development of an Integrated System of Semantic-Syntactic Analysis]. PhD Thesis. Moscow, 2015, 210 p.
2. Barbu C., Mitkov R. Evaluation tool for rule-based anaphora resolution methods. *Proc. 39th Annual Meeting on Association for Computational Linguistics*. 2001, pp. 34–41.
3. Abramov V.E., Abramova N.N., Nekrasova E.V., Ross G.N. Statistical analysis of the connectivity of texts on social and political topics. *Elektronnye biblioteki: perspektivnye metody i tekhnologii, elektronnye kolleksii: tr. 13 Vseros. nauch. konf.* [Proc. 13th All-Russian Science Conf. "Digital Libraries: Advanced Methods and Technologies, Digital Collections" (RCDL'2011)]. 2011, pp. 127–133 (in Russ.).
4. Tolpegin P.V. *Avtomaticheskoe razreshenie koreferentsii mestoimeny tretyego litsa russkoyazychnykh tekstov* [Automatic resolution of the co-reference of third person pronouns in Russian-language texts]. PhD Thesis. Moscow, 2008, 214 p. (in Russ.).
5. Protopopova E.V., Bodrova A.A., Volskaya S.A., Krylova I.V., Chuchunkov A.S., Alexeeva S.V., Bocharov V.V., Granovsky D.V. Anaphoric annotation and corpus-based anaphora resolution: an experiment, computational linguistics and intellectual technologies. *Dialog-2014: sb. tr. Mezhdunar. nauch. konf. po kompyuter. lingvistike* [Proc. Annual Int. Science Conf. on Computer Linguistics "Dialogue-2014"]. 2014, iss. 13 (20), pp. 562–571 (in Russ.).
6. Kamenskaya M.A., Khramoin I.V., Smirnov I.V. Data-driven Methods for Anaphora Resolution of Russian. *Dialog-2014: sb. tr. Mezhdunar. nauch. konf. po kompyuter. lingvistike* [Proc. Annual Int. Science Conf. on Computer Linguistics "Dialogue-2014"]. 2014, iss. 13 (20), pp. 241–250 (in Russ.).
7. Arregi O., Ceberio K., Diaz de Illaraza A., Goenaga I., Sierra B., Zelaia A. Determination of Features for a Machine Learning Approach to Pronominal Anaphora Resolution in Basque. *Procesamiento del lenguaje natural* [Natural Language Processing]. 2010, no. 45, pp. 291–294.
8. Korobov M. Morphological Analyzer and Generator for Russian and Ukrainian Languages. *Analysis of Images, Social Networks and Texts*. 2015, pp. 320–332.
9. Nivre J., Hall J., Nilsson J., Chanev A., Eryigit G., Kübler S., Marinov S., Marsi E. MaltParser: A language-independent system for data-driven dependency parsing. *Natural Language Engineering*. 2007, vol. 13, no. 2, pp. 95–135.
10. Nivre J., Hall J., Nilsson J. MaltParser: A data-driven parser-generator for dependency parsing. *Proc. Int. Conf. on Language Resources and Evaluation (LREC)*. 2006, vol. 6, pp. 2216–2219.
11. Sharoff S., Nivre J. The proper place of men and machines in language technology: Processing Russian without any linguistic knowledge. *Dialog-2011: sb. tr. Mezhdunar. nauch. konf. po kompyuter. lingvistike* [Proc. Annual Int. Science Conf. on Computer Linguistics "Dialogue-2011"]. 2011, no. 10, pp. 591–604.
12. Smirnov I.V., Shelmanov A.O., Kuznetsova E.S., Khramoin I.V. Semantic-syntactic analysis of natural languages. Part II. Method of semantic-syntactic analysis of texts. *Iskusstvenny intellekt i prinyatie resheny* [Artificial intelligence and decision-making]. Moscow, 2014, no. 1, pp. 11–24 (in Russ.).
13. Chang C.C., Lin C.J. LIBSVM: A Library for Support Vector Machines. *ACM Trans. on Intelligent Systems and Technology*. 2011, vol. 2, iss. 3, art. no. 272001.
14. Fan R.E., Chang K.W., Hsieh C.J., Wang X.R., Lin C.J. *LIBLINEAR: A library for large linear classification* *Journal of Machine Learning Research*. 2008, vol. 9, pp. 1871–1874.
15. Lappin S., Leass H.J. An Algorithm for Pronominal Anaphora Resolution. *Computational Linguistics*. 1994, vol. 20, no. 4, pp. 535–561.
16. *Natsionalny korpus russkogo yazyka* [Russian National Corpus]. Available at: <http://www.ruscorpora.ru/instruction-syntax.html> (accessed May 14, 2017).
17. Breiman L. Random forests. *Machine learning*. 2001, vol. 45, no. 1, pp. 5–32.

УДК 681.322

DOI: 10.15827/0236-235X.119.469-473

Дата подачи статьи: 01.03.17

2017. Т. 30. № 3. С. 469–473

## **ИНФОРМАЦИОННО-АНАЛИТИЧЕСКАЯ СИСТЕМА ОЦЕНИВАНИЯ ТРУДОЗАТРАТ И СТОИМОСТИ СОЗДАНИЯ ПРОГРАММНЫХ СРЕДСТВ**

*С.А. Котов, к.э.н., доцент, директор, info@gicpsvt.ru;*

*А.А. Демирский, к.т.н., начальник отдела, info@gicpsvt.ru*

*(Главный испытательный сертификационный центр безопасности программных средств  
и вычислительной техники, ул. Ржевская, 10, г. Тверь, 170023, Россия)*

В статье рассматривается метод оценки размера создаваемого программного средства на ранних этапах жизненного цикла его разработки. Для программных средств одного класса наибольшее влияние на трудоемкость разработки оказывает их размер. Обоснованная оценка размера создаваемого программного средства необходима для адекватной оценки трудозатрат и стоимости разработки. В статье дается краткий обзор существующих методов оценки размера создаваемого программного средства. Среди них наиболее распространенными являются метод конструктивной модели стоимости (СОСОМО, СОСОМО II) и метод функционально-балльной оценки (FPA). Указаны недостатки данных методов.

Для формирования оценки размера создаваемого программного средства предлагается использовать метод, являющийся развитием FPA и СОСОМО II. Он основан на подсчете программных объектов: экранных форм, отчетов, файлов данных (таблиц СУБД), функций импорта/экспорта данных. Для оценки размера создаваемых программных объектов используется база нечетких продукционных правил, параметры которой определяются с помощью искусственной нейронной сети (нейронная продукционная сеть Такаги–Сугено–Канга). Приводится математическое описание используемых моделей оценки. Метод был адаптирован для оценки размера создаваемого специального ПО автоматизированных систем. Выделены преимущества применения предложенного метода оценки размера создаваемого программного средства.

С использованием данного подхода была разработана информационно-аналитическая система оценивания трудозатрат и стоимости создания программных средств. В статье рассмотрена функциональная структура информационно-аналитической системы, дается перечень режимов ее функционирования.

В заключение приведены результаты применения предложенного метода для оценки размера создаваемого специального ПО автоматизированных систем.

**Ключевые слова:** оценка трудозатрат на создание программного обеспечения, управление программными проектами, нечеткие модели.

Проекты по созданию крупных программных средств (ПС) традиционно начинаются с анализа и разработки технико-экономического обоснования затрат, связанных с их созданием и развитием. Заказчику необходимо оценить объем требующихся ресурсов, а потенциальному разработчику – возможность реализации проекта в условиях и ресурсах, предлагаемых заказчиком.

Следствием низкого качества технико-экономического обоснования стоимости выполнения работ по созданию и развитию ПС зачастую являются конфликты между заказчиками и разработчиками, что отрицательно сказывается на результате.

Большей части этих негативных последствий можно избежать, используя современные методы управления проектами для их успешного завершения. При этом важное значение имеет технология формирования точных оценок затрат ресурсов, необходимых для создания и развития ПС.

При создании ПС основные затраты связаны с непосредственным или овлеществленным интеллектуальным трудом различных специалистов (аналитиков, проектировщиков, программистов и т.п.). Поэтому для измерения затрат по работам, связанным с разработкой ПС, наиболее универсальной единицей стала трудоемкость выполняемых работ, выраженная в человеко-днях или человеко-месяцах. Стоимость работ вычисляется как произведе-

ние трудозатрат на стоимость нормо-часа специалистов, выполняющих данную работу [1].

Современные методы оценки трудозатрат на создание и развитие ПС можно разделить на следующие классы:

- методы структурированной экспертной оценки;
- методы оценки по математическим моделям, построенным на основе обработки статистических данных.

Экспертные оценки применяются при отсутствии дискретных эмпирических данных. Оценки, получаемые подобным образом, представляют собой результат анализа известных эксперту проектов, в которых он принимал участие. Основным непреодолимым недостатком данной группы методов является слабое обоснование полученных оценок. Фактически единственным обоснованием оценки является авторитет эксперта (или группы экспертов).

Масштабные исследования в области математического моделирования расходов, связанных с разработкой и эксплуатацией ПС, начались в США в 1965 году. Эти разработки легли в основу нескольких примитивных моделей, созданных в конце 60–начале 70-х годов двадцатого столетия. Потом появились более сложные модели: SLIM [2], метод контрольных точек (Checkpoint) [3], PRICE-S [4],

SEER [5] и COSOMO [6]. Разработчики этих моделей оценки затрат столкнулись с одной и той же проблемой: по мере развития технологий разработки ПС происходил значительный рост размеров этого средства и увеличивалась его сложность, что существенно затрудняло точную оценку расходов на разработку [7]. Для построения математических моделей оценки в основном используется регрессионный анализ. Данную группу методов характеризует достаточно высокая точность получаемых оценок. Однако для калибровки моделей требуется большая база статистических данных по уже завершенным проектам. Кроме того, не всегда заранее известен вид зависимости входа и выхода модели.

В настоящее время наиболее распространены методами оценки на основе математических моделей являются методы конструктивной модели стоимости (COSOMO, COSOMO II) [6] и функционально-балльной оценки (FPA) [8].

Опыт использования упомянутых выше математических моделей оценки позволяет выделить основные факторы, влияющие на затраты, связанные с созданием и развитием ПС. Для ПС одного класса наибольшее влияние на трудоемкость разработки оказывает его размер. Данный параметр может изменяться в большом диапазоне: на три-четыре порядка от  $10^4$  до  $10^7$  строк исходного кода программ [6, 8, 9]. Именно размер ПС определяет основной объем работ по проекту. Поэтому при оценке затрат на разработку ПС размер программы используется в качестве базового доминирующего параметра. Так как точность оценок размера ПС определяет точность оценок прочих стоимостных параметров проекта, методам его оценивания уделяется большое внимание. В качестве единицы измерения размера ПС обычно используется строка исходного кода (SLOC – source line of code). Данная единица измерения требует точного описания понятия «строка кода» [9]. Другая проблема использования единиц измерения размера ПС, основанных на коде, заключается в том, что на ранних стадиях жизненного цикла ПС информация об исходном коде недоступна.

Применяемые подходы для оценки размера создаваемого ПС в методах COSOMO II и FPA имеют ряд недостатков, затрудняющих их применение.

- Для использования метода функциональных баллов необходимы высококвалифицированные сертифицированные специалисты. При несоблюдении данного условия точность оценок резко снижается.

- Метод функциональных баллов требует значительного времени на сбор данных о создаваемом ПС. При этом состав собираемых данных подразумевает детальное понимание требований к ПС.

- Метод функциональных баллов ориентирован в большей степени на процедурное программирование, так как оперирует многими понятиями, не

используемыми при объектно-ориентированном подходе.

- COSOMO II использует метод оценки размера ПС с сильно ограниченным набором программных объектов. Сложность реализации каждого программного объекта определяется с помощью экспертной оценки.

- Статистике, используемой при калибровке модели COSOMO II, уже более 15 лет. Как следствие, в ней плохо учтена специфика создания ПС с использованием технологии объектно-ориентированного программирования (которая сейчас широко применяется при создании ПС).

Для формирования оценки трудозатрат на работы по созданию и развитию ПС предлагается использовать метод, являющийся развитием методов оценки размера создаваемого ПС, применяемых в FPA и COSOMO II. Это позволит частично или полностью устранить указанные недостатки. Для оценки трудозатрат на создание ПС на основе его размера могут быть использованы откалиброванные модели, применяемые в методах COSOMO II и FPA. Основной задачей является повышение точности и обоснованности оценок размера создаваемого ПС, полученных на ранних этапах жизненного цикла.

В предлагаемом методе размер ПС – это сумма оценок размеров всех программных объектов, из которых состоит ПС. Под программными объектами будем понимать конструктивные элементы ПС, на создание которых требуются значительные трудозатраты. Однотипные программные объекты можно объединить в классы. Для каждого класса разрабатывается модель для оценки размера в зависимости от значений общих характеристик (свойств) данного класса программных объектов. При использовании данного подхода процесс оценки значительно упрощается. На первом этапе аналитики занимаются определением количества и свойств создаваемых программных объектов ПС. Затем в зависимости от значений свойств этих программных объектов вычисляется их вероятный размер. Важно, чтобы в качестве свойств программных объектов выступали такие характеристики, которые могут быть доступны для оценки на начальных этапах жизненного цикла ПС. В качестве единиц измерения размера программных объектов будет использоваться SLOC. Это позволит аккумулировать оценки отдельных программных объектов для формирования итоговой оценки размера создаваемого ПС.

Для построения моделей оценки размера классов программных объектов были использованы методы искусственного интеллекта: нейронные сети и нечеткая логика (нейро-нечеткие, или гибридные модели). Особенностью этих методов является учет в них априорных сведений об объекте или о процессе исследований в форме нечетких продукционных правил «ЕСЛИ–ТО». Параметры правил



определяются по имеющимся статистическим данным с помощью алгоритма обратного распространения ошибки, генетических алгоритмов [10] и др.

Основные преимущества использования нейро-нечетких методов моделирования:

- позволяют строить модель, используя малые выборки данных (по сравнению со статистическими методами);
- не требуют предварительного определения вида зависимости входов и выходов модели;
- дают возможность моделировать сложные процессы и системы (многомерные, нелинейные);
- позволяют достаточно легко интерпретировать получаемые модели.

При использовании нейро-нечетких методов моделирования размеры экземпляров программных объектов ПС определяются с помощью *базы знаний* (БЗ), представляющей собой набор правил нечеткого вывода. Основное назначение БЗ – идентификация нелинейной зависимости  $y = \varphi(X)$ , где  $X$  – вектор значений количественных свойств класса программных объектов ПС;  $y$  – значение размера кода программного объекта ПС в SLOC.

Построение моделей оценки размера программных объектов ПС осуществляется в два этапа. На первом этапе формируется нечеткая БЗ. Данная БЗ связывает входы и выходы модели с помощью лингвистических правил «ЕСЛИ–ТО». Набор правил создается на основе извлечения знаний из имеющейся статистики. На следующем этапе производится настройка параметров нечеткой БЗ для достижения минимального отклонения модельных результатов от имеющихся статистических данных. Подобная настройка реализуется с помощью нечетких нейронных сетей, способных одновременно формировать нечеткие правила и адаптировать функции принадлежности путем модификации весов связей в процессе обучения.

Особенность применяемых моделей состоит в том, что заключения правил представляются в форме функциональных зависимостей (нейронная продукционная сеть Такаги–Сугено–Канга):

П<sub>і</sub>: ЕСЛИ  $x_1$  есть  $A_{i1}$  И ... И  $x_j$  есть  $A_{ij}$  И ... И  $x_m$  есть  $A_{im}$ , ТО  $y = c_{i0} + \sum_{j=1}^m c_{ij}x_j$ ,  $i = 1, \dots, n$ , где  $A_{ij}$  –

нечеткое множество, отражающее значение характеристики программного объекта  $x_j$ , с функцией принадлежности  $\mu(x_j)$ ;  $x_j$  – количественная характеристика программного объекта ПС, влияющая на размер кода, который необходимо написать для его создания;  $y$  – четкая выходная переменная, равная искомому размеру кода программного объекта ПС;  $n$  – количество правил;  $m$  – количество входных переменных.

При малом объеме обучающих данных возможно использование упрощенной модели, когда заключения правил заданы константами:  $y = c_i$ ,  $i = 1, 2, \dots, n$ .

Полученная БЗ может быть интерпретирована как некоторое разбиение пространства входных характеристик на нечеткие подобласти, в каждой из которых значение выходной переменной рассчитывается как линейная комбинация входов. Переключение с одного линейного закона «входы–выход» на другой осуществляют правила БЗ. Так как границы подобластей входных переменных размыты, одновременно могут выполняться несколько правил, но с разными весами. Итоговое значение выходной переменной  $y$  определяется как суперпозиция линейных зависимостей:

$$y = \frac{\sum_{i=1}^n \left( \left( c_{i0} + \sum_{j=1}^m c_{ij}x_j \right) \prod_{j=1}^m \exp \left[ - \left( \frac{x'_j - a_{ij}}{b_{ij}} \right)^2 \right] \right)}{\sum_{i=1}^n \prod_{j=1}^m \exp \left[ - \left( \frac{x'_j - a_{ij}}{b_{ij}} \right)^2 \right]}.$$

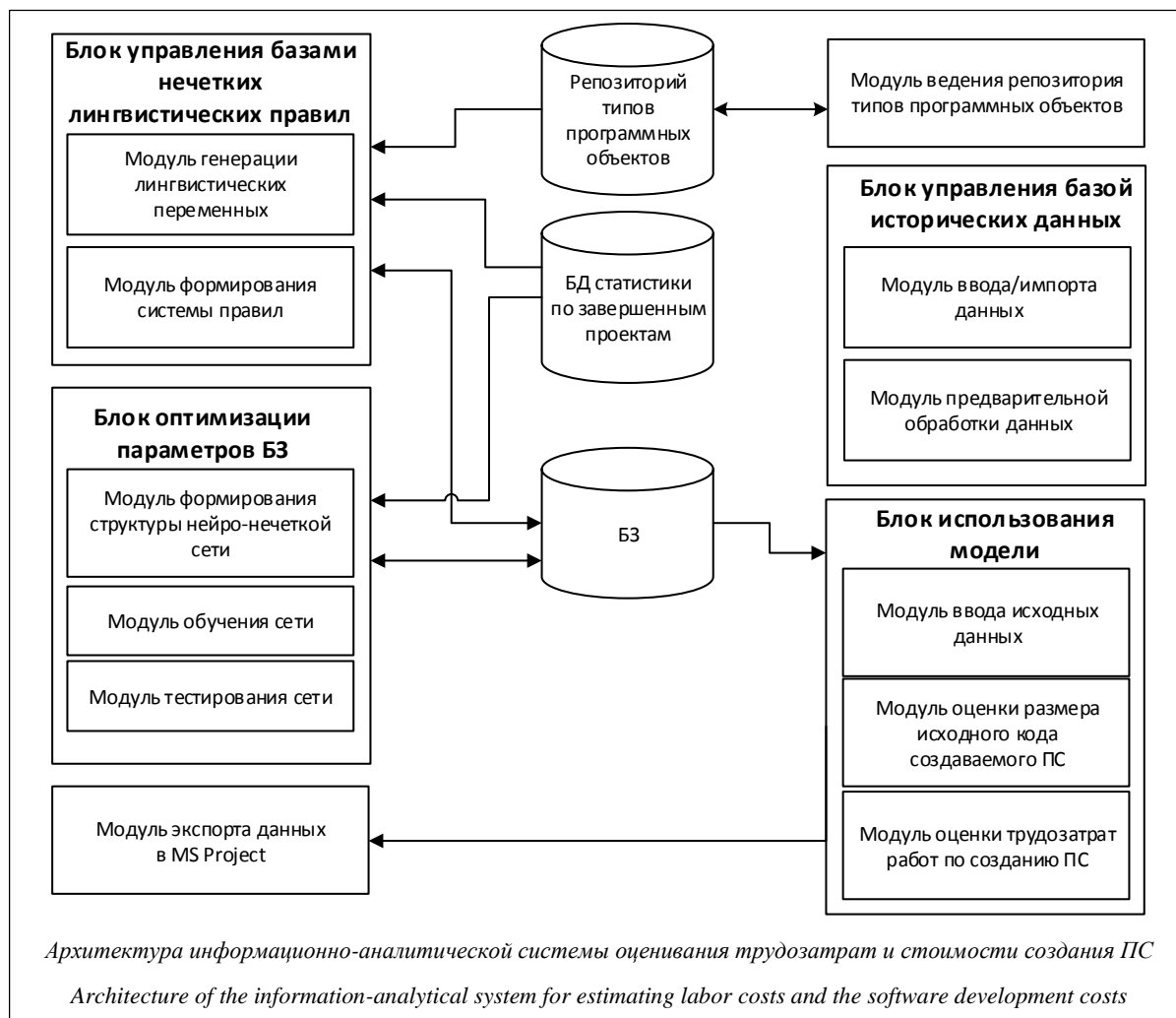
В данном выражении для задания функций принадлежности используется функция Гаусса, где  $a$  и  $b$  – параметры этой функции.

С использованием данного подхода была разработана информационно-аналитическая система оценивания трудозатрат и стоимости создания ПС. Цель такой системы – обеспечение руководителя проекта инструментарием для оперативной оценки трудозатрат на разработку ПС (как всего проекта, так и его отдельных этапов) и автоматизированной подготовки всех необходимых данных для получения этой оценки.

Информационно-аналитическая система оценивания трудозатрат и стоимости создания ПС реализует следующие функции:

- ведение БД статистики по завершенным проектам разработки ПС;
- описание структуры проекта в разрезе классов программных объектов ПС;
- выбор вида модели нечеткого вывода формируемой БЗ;
- генерация структуры лингвистических переменных, входящих в БЗ, на основе имеющейся статистики по завершенным проектам;
- генерация базы нечетких решающих правил;
- решение задачи оптимизации баз нечетких правил;
- оптимизация параметров баз нечетких решающих правил на основе имеющейся статистики по завершенным проектам;
- оценка технико-экономических параметров проекта с использованием БЗ;
- экспорт данных с результатами оценки технико-экономических параметров проекта во внешние приложения (ПО календарного планирования проектов).

Архитектура информационно-аналитической системы оценивания трудозатрат и стоимости создания ПС представлена на рисунке. По своему составу она близка к экспертным системам.



Информационно-аналитическая система оценивания трудозатрат и стоимости создания ПС может работать в одном из четырех режимов:

- ввод исходных данных для построения моделей оценки размера кода программных объектов ПС;
- построение математических моделей оценки размера кода программных объектов ПС;
- настройка (адаптация) параметров полученных моделей;
- использование сгенерированных моделей оценки размера кода программных объектов ПС для оценивания трудозатрат и стоимости их создания.

С помощью системы были построены модели оценки размера программных объектов специального ПО информационных систем. Данный класс ПС характеризуется наличием БД и глубоко проработанного графического интерфейса пользователя. В качестве типов программных объектов, характеристики которых доступны на ранних этапах жизненного цикла, выделялись следующие классы: экранные формы, диалоги, отчеты, файлы данных (таблицы СУБД), функции импорта/экспорта данных.

Разработанная информационно-аналитическая система апробирована в Главном испытательном сертификационном центре безопасности программных средств и вычислительной техники (г. Тверь). Исходные данные для формирования моделей оценки получены исходя из анализа ПС, поступивших на сертификационные испытания. Полученные результаты оценки размера ПС сравнивались с оценками, полученными по методу функциональных баллов. Были проанализированы завершённые проекты с размером ПС от  $10^4$  до  $10^6$  в строках исходного кода (SLOC). Результаты оценки размера ПС, полученные с использованием нейро-нечетких методов моделирования, отличаются от фактических на 5–10 %. Данные, полученные с использованием метода функциональных баллов, отличаются от фактических на 10–23 %. Оценка размера создаваемого ПС вышеперечисленными методами производилась на этапе проектирования интерфейса пользователя, после чего результат сравнивался с фактическим размером ПС после завершения его разработки.

На данный момент результаты исследования были апробированы и успешно внедрены в Банке России и Федеральной таможенной службе.

**Литература**

1. Липаев В.В. Технико-экономическое обоснование проектов сложных программных средств. М.: СИНТЕГ, 2004. 284 с.
2. Jensen Dr. Randall W., Putnam Lawrence H. Sr., Roetzheim William. Software estimating models: three viewpoints. Software Technology Support Center, 2006, pp. 23–29.
3. Jones C., Bonsignour O. The economics of software quality. Addison-Wesley, 2012, pp. 105–109.
4. Shermon D. System cost engineering. Gower Publ., 2009, 326 p.
5. Fischman Lee, McRitchie Karen, Galorath Daniel D. Inside SEER-SEM, CROSSTALK The Jour. of Defense Soft. Eng., 2005, pp. 26–28.
6. Boehm B., Abts C., Brown A.W., Chulani S., Clark B.K., Horowitz E., Madachy R., Reifer D.J., Steece B. Software Cost Estimation with COCOMO II. Englewood Cliffs, NJ, Prentice-Hall, 2000, 544 p.
7. Котов С.Л. Нормирование жизненного цикла программной продукции. М.: ЮНИТИ, 2002. 143 с.
8. Garmus D., Herron D. Functional point analysis. Addison-Wesley, 2000, 400 p.
9. McConnell S. Software estimation: demystifying the black art. Microsoft Press, 2006, 308 p.
10. Борисов В.В. Нечеткие модели и сети. М.: Горячая линия–Телеком, 2007. 284 p.

Software &amp; Systems

DOI: 10.15827/0236-235X.119.469-473

Received 01.03.17

2017, vol. 30, no. 3, pp. 469–473

### AN ENTERPRISE INFORMATION SYSTEM FOR ESTIMATING LABOR AND SOFTWARE DEVELOPMENT COSTS

S.L. Kotov<sup>1</sup>, Ph.D. (Economics), Associate Professor, info@gicpsvt.ruA.A. Demirsky<sup>1</sup>, Ph.D. (Engineering), Head of Department, info@gicpsvt.ru<sup>1</sup> Main Testing Certification Center of Security Software and Computer Engineering, Rzevskaya St. 10, Tver, 170023, Russian Federation

**Abstract.** This article aims to study the process of software size estimation and its impact on the software development process. Software size has the greatest impact on the complexity of development. A software estimation process, which is integrated with the software development process, can help to establish realistic and credible plans to implement the project requirements and satisfy commitments. It also can support other management activities by providing accurate and timely planning information.

The article provides an overview of existing methods for estimating the size of the software created. The most common methods are: functional point analysis (FPA) and constructive cost model (COCOMO, COCOMO II). The article indicates disadvantages of these methods.

The paper presents a new approach of software size estimation. The fuzzy model Takagi-Sugeno-Kang is described by fuzzy IF-THEN rules, which represent local input-output relations of a nonlinear system. The main feature of the Takagi-Sugeno-Kang fuzzy model is to express a local dynamics of each fuzzy implication (rule) by a linear system model. The fuzzy model Takagi-Sugeno-Kang is used to estimate the size of software objects.

The paper illustrates how the new method can be used in the enterprise information system for estimating labor and software development costs. The result shows that the approach is better than FPA method.

**Keywords:** labor cost estimation, software size, software estimation, project management, fuzzy, fuzzy modeling.

**References**

1. Lipaev V.V. *Tekhniko-ekonomicheskoe obosnovanie proektov slozhnykh programmnykh sredstv* [A Feasibility Report for Complex Software Projects]. Moscow, SINTEG Publ., 2004, 284 p.
2. Jensen Dr. Randall W., Putnam Lawrence H. Sr., Roetzheim W. *Software Estimating Models: Three Viewpoints*. Software Technology Support Center, 2006, pp. 23–29.
3. Jones C., Bonsignour O. *The Economics of Software Quality*. Addison-Wesley Publ., 2012, pp. 105–109.
4. Shermon D. *System cost engineering*. Gower Publ., 2009, 326 p.
5. Fischman L., McRitchie K., Galorath D.D. Inside SEER-SEM, CROSSTALK. *The Jour. of Defense Software Engineering*. 2005, pp. 26–28.
6. Boehm B., Abts C., Winsor Brown A., Chulani S., Clark B.K., Horowitz E., Madachy R., Reifer D.J., Steece B. *Software Cost Estimation with COCOMO II*. Englewood Cliffs, NJ, Prentice-Hall Publ., 2000, 544 p.
7. Kotov S.L. *Normirovanie zhiznennogo tsikla programmoynoy produktsii* [Normalization of Software Products Life Cycle]. Moscow, Yuniti Publ., 2002, 143 p.
8. Garmus D., Herron D. *Functional Point Analysis*. Addison-Wesley Publ., 2000, 400 p.
9. McConnell S. *Software estimation: demystifying the black art*. Microsoft Press, 2006, 308 p.
10. Borisov V.V. *Nechetkie modeli i seti* [Fuzzy Models and Networks]. Moscow, Goryachaya liniya–Telekom Publ., 2007, 284 p.

УДК 330.47  
DOI: 10.15827/0236-235X.119.474-477

Дата подачи статьи: 27.03.17  
2017. Т. 30. № 3. С. 474–477

## **РАСЧЕТ ОПТИМАЛЬНОЙ ЗАВИСИМОСТИ ЦЕНЫ ИНФОРМАЦИОННОГО РЕСУРСА ПРИ ЕГО РЕАЛИЗАЦИИ НА НЕСКОЛЬКИХ РЫНКАХ**

*С.А. Амелкин, к.т.н., руководитель центра, sergey.a.amelkin@gmail.com  
(Исследовательский центр системного анализа Института программных систем  
им. А.К. Айламазяна РАН, ул. Петра I, 4а, г. Переславль-Залесский, 152021, Россия);*

*О.С. Иванова, инженер, olety@yandex.ru  
(Исследовательский центр медицинской информатики Института программных систем  
им. А.К. Айламазяна РАН, ул. Петра I, 4а, г. Переславль-Залесский, 152021, Россия)*

В последнее время наряду с материальными ресурсами все большее значение приобретают информационные, нематериальные ресурсы. Они имеют ряд особенностей, которые следует учитывать при решении задач оптимального управления бизнес-процессами. Одна из наиболее ключевых проблем информационного рынка – определение стоимости информации. Однако здесь возникает ряд трудностей из-за специфики товара.

В данной статье под информационным товаром будем понимать некий программный продукт. Продажа информационных ресурсов характеризуется тем, что его тиражирование может быть осуществлено без существенных дополнительных затрат. Это приводит к увеличению объема ресурса в ходе процессов его распространения. При этом установление цены является одним из ключевых моментов, определяющих дальнейшее распространение и продвижение товара, которое должно учитывать множество факторов, в том числе и способ его распространения.

В работе рассмотрена задача определения ценовой политики фирмы, реализующей на рынке информационный ресурс. Решена задача определения оптимальной зависимости цены информационного ресурса от времени для случая, когда фирма-продавец не имеет конкурентов на рынке, рынок насыщается, а покупатели лишены возможности дальнейшего распространения ресурса. Получены условия оптимального выбора цены информационного ресурса, а для линейной функции спроса рассчитан максимально возможный доход фирмы, а также получены условия, определяющие продолжительность и объем продаж, обеспечивающих максимум среднего дохода фирмы.

При известной функции спроса, которую можно найти аппроксимацией данных эксперимента, полученные условия позволяют выбрать оптимальное изменение цены продаж в функции, определяющей число проданных экземпляров информационного ресурса.

**Ключевые слова:** информационный ресурс, оптимальная цена, функция спроса, интенсивность продаж, необходимые условия оптимальности.

Информацию в экономике рассматривают как товар (информационный продукт, услуга), а также как ресурс, используемый в процессе хозяйственной деятельности. К информационным товарам и услугам относятся ПО, БД, образовательные услуги, консультирование, результаты НИОКР и прочие [1, 2]. Информацию стали воспринимать как новый экономический ресурс, как новое экономическое благо, что привело к появлению нового вида рынка – информационного.

Информационный рынок – это система экономических, правовых и организационных отношений по торговле товарами, созданными информационной индустрией. Как всякий рынок, он характеризуется определенной номенклатурой продуктов и услуг, ценами, спросом и предложением, поставщиками и потребителями [3]. Информационные продукты и услуги обмениваются на информационном рынке и имеют большое количество особенностей как на стадиях разработки и производства, так и на этапе обращения.

В данной статье под информационным товаром будем понимать некий программный продукт – комплекс взаимосвязанных программ для решения определенной проблемы (задачи) массового спроса, подготовленный к реализации как любой вид промышленной продукции [4].

Одна из наиболее ключевых проблем информационного рынка – определение стоимости информации. Однако здесь возникает ряд трудностей из-за специфики товара. Сложность формирования цен на программный продукт объясняется следующими его особенностями:

- конечная стоимость информации зависит от ее полезности для отдельного потребителя;
- минимальные материальные затраты существенно снижают стоимость тиражирования информации, что дает возможность наращивать объемы продукции без существенных расходов ресурсов и времени; к тому же характер потребления информации неразрушающий, благодаря чему она многократно может быть использована производителем;
- информационные продукты со временем теряют свою первичную потребительскую стоимость;
- производители несут риски возможности разглашения содержания или несанкционированного копирования и распространения информационных продуктов, которые еще не прошли процедуру правовой защиты;
- при реализации продукции на различных рынках производители не имеют возможности на каждом из этих рынков назначать свою цену (от-

существование ценовой диверсификации), что связано с небольшим количеством, а все чаще с полным отсутствием посредников при продажах информационных ресурсов (например, при продажах через веб-сайт производителя) [5, 6].

Кроме того, продвижение программы не может быть эффективным, если нет четкого представления о сегменте рынка, на который нацелен продукт. Поэтому после создания программного продукта необходимо определиться с сегментом рынка, на который будет поставляться продукт, с видом продажи и, конечно, с ценой. Разработчики или перепродавцы, как правило, устанавливают на продукт не какую-то одну цену, а создают целую систему ценообразования, которая отражает отличия в спросе на разных сегментах рынка, в гарантиях, договорах на сопровождение и проч. Какой бы хороший функционал ни имела программа, если цена на нее неоправданна, если она не удовлетворяет покупателя или занимает неправильное место на рынке, продаваться она не будет [7, 8].

Таким образом, продавец должен определить такую цену на программный продукт, которая обеспечит ему наибольший средний доход с учетом различия спроса на различных рынках, где реализуется этот продукт.

### Постановка задачи

Рассмотрим ситуацию, когда продавец (фирма) является единственным владельцем информационного ресурса. Имеется  $v$  рынков с заданным числом потенциальных покупателей  $n_{0i}$ , где  $i = 1, \dots, v$ . Обозначим через  $m$  суммарную интенсивность продаж. Эта величина зависит от цены товара  $p$ . Цена товара одинакова на всех рынках. Будем называть зависимость  $m(p)$  функцией спроса. Издержки будем считать равными нулю.

Изначально цена  $p$  назначается продавцом, а спрос на товар  $q_i$ , где  $i = 1, \dots, v$ , зависит от рынка, что может быть описано параметрами рынка  $a_i$  (например, эластичность спроса на  $i$ -м рынке).

Продавец, управляя ценой, стремится максимизировать свой доход. Отсюда вытекает задача выбора оптимальной цены продажи информационных ресурсов с учетом способа ее распространения.

### Формализация задачи

Функцию спроса  $m$ , которую предполагаем известной, обозначим как  $m(p, n)$ , где  $n = \sum_{i=1}^v n_i$ , а  $n_i$  – проданное число товара на каждом из рынков. Эта функция положительна для всех  $p < p_0$  и равна нулю при достижении максимальной цены  $p_0$ :

$$m(p) = \sum_{i=1}^v q_i(p, a_i(n_i)), \quad m \geq 0, \quad \frac{\partial q_i}{\partial p} < 0.$$

**Постановка задачи:** продать за заданное время  $\tau$   $N$  экземпляров продукта так, чтобы получить максимальный доход  $\Pi$ :

$$\Pi = \int_0^{\tau} p(t) q_i(p, a_i(n_i)) dt \rightarrow \max_p. \quad (1)$$

Число проданных программ изменяется в соответствии с уравнением

$$\frac{dn}{dt} = m(p, n) = \sum_i q_i(p, a_i(n_i)), \quad (2)$$

$$n(0) = 0, \quad n(\tau) = N, \quad 0 \leq t \leq \tau.$$

### Решение задачи

С учетом того, что насыщение рынка происходит монотонно,  $\frac{dn}{dt} > 0$ , можно путем замены переменных перейти от задачи с условием в виде дифференциального уравнения к изопериметрической задаче, которую будем решать традиционным методом Лагранжа [9]. Проведем замену переменной:

$$dt = \frac{dn}{\sum_i q_i(p, a_i(n_i))}, \quad t(0) = 0, \quad t(N) = \tau.$$

При этом задача (1), (2) примет вид

$$\Pi = \int_0^N p(n) dn \rightarrow \max_p \quad (1a)$$

при условии

$$\int_0^N \sum_i q_i(p(n), a_i(n_i)) dn = \tau. \quad (3)$$

Решим задачу (1a), (3) с интегральным критерием оптимальности и интегральным ограничением методом Лагранжа. Подынтегральное выражение функционала Лагранжа для невырожденного решения ( $\lambda$  не равно нулю) имеет вид

$$L = p(n) - \frac{\lambda}{\sum_i q_i} \rightarrow \max_p \min_{\lambda}. \quad (4)$$

Выражение (4) представляет собой текущий доход с учетом ограничений по времени.

Необходимые условия оптимальности при отсутствии ограничений на  $p$  и непрерывной и непрерывно дифференцируемой по  $p$  функции спроса приводят к требованию стационарности функции  $L$ :

$$\frac{\partial L}{\partial p} = 0 \rightarrow 1 + \frac{\lambda}{(\sum_i q_i)^2} \sum_i \frac{\partial q_i}{\partial p} = 0. \quad (5)$$

Отсюда получим условия оптимального выбора цены продажи в зависимости от числа  $n$  проданных экземпляров продукта:

$$\frac{(\sum_i q_i)^2}{\sum_i \frac{\partial q_i}{\partial p}} = \lambda = \text{const} = \sum_i q_i \Rightarrow \sqrt{-\sum_i \frac{\partial q_i}{\partial p} \lambda}. \quad (6)$$

Это равенство совместно с условием (3) определяет  $\lambda^*$  и  $p^*(n)$ .

Максимальная прибыль зависит от объема  $N$  ресурса, намеченного к продаже, от продолжительности продаж  $\tau$  и вида функции  $m(p, n)$ .

Покажем характер этой зависимости на примере.

### Пример

Рассмотрим задачу оптимальной продажи информационного ресурса (ПО), задав функцию спроса в форме

$$m(p, n) = \sum_i q_i = \sum_i \alpha_i (p_{0i} - p)(n_{0i} - n), \quad (7)$$

где  $\alpha$  – постоянная;  $p_0$  – максимальная цена, при которой спрос отсутствует;  $n_0$  – общее число потенциальных покупателей.

Экономический смысл (7) заключается в следующем:

- при любом  $n$  функция спроса линейна;
- чем больше  $n$ , тем меньше потенциальных покупателей;

- $\alpha(p_0 - p)$  – кривая индивидуального спроса.

1. С учетом того, что для функции спроса (7)

$$\sum \frac{\partial q_i}{\partial p} = - \sum_i \alpha_i (n_{0i} - n), \quad (8)$$

условие оптимальности (6) примет вид

$$\sum_i \alpha_i (p_{0i} - p)(n_{0i} - n) = \sqrt{\sum_i \alpha_i \lambda (n_{0i} - n)}. \quad (9)$$

Отсюда оптимальная ценовая политика производителя ПО равна

$$p^*(n) = \frac{\sum_i \alpha_i p_{0i} (n_{0i} - n) - \sqrt{\sum_i \alpha_i \lambda (n_{0i} - n)}}{\sqrt{\alpha_i (n_{0i} - n)}}. \quad (10)$$

Прибыль от продажи равна

$$\int_0^N \frac{dn}{\sqrt{\sum_i \alpha_i \lambda (n_{0i} - n)}} = \tau \rightarrow \frac{-2 \sqrt{\sum_i \alpha_i n_{0i} - \sum_i \alpha_i n}}{\sum_i \alpha_i} = \tau.$$

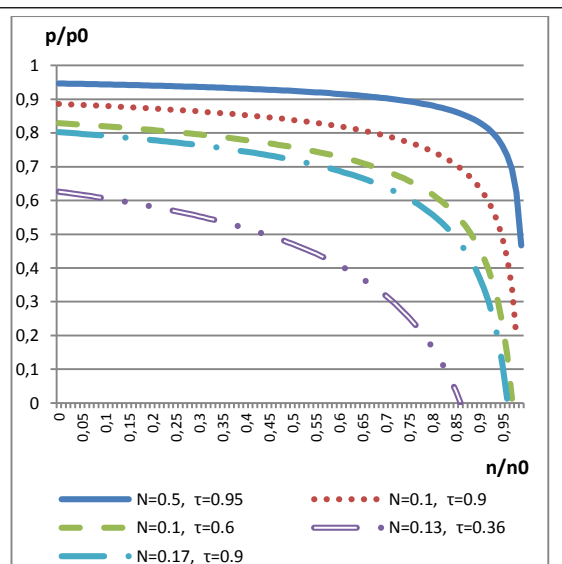
Для удобства представления результатов введем безразмерные величины:

- безразмерная цена  $\hat{p}^* = \frac{p^*}{p_0}$ ;
- безразмерное количество проданного ПО  $\hat{n} = \frac{n}{n_0}$ .

Оптимальные траектории безразмерной цены в зависимости от интенсивности продаж показаны на рисунке.

### Выводы

Продажа информационных ресурсов характеризуется тем, что тиражирование такого ресурса может быть осуществлено без существенных до-



Зависимость оптимальной цены от величины продаж

The dependence of an optimal price on an amount of sales

полнительных затрат. Это приводит к увеличению объема ресурса в ходе процессов его распространения. При этом установление цены является одним из ключевых моментов, определяющих дальнейшее распространение и продвижение товара, которое должно учитывать множество факторов, в том числе и способ распространения товара. В данной работе для максимизации дохода учитывается продажа через разных посредников. Оптимальным правилом определения цены при функции спроса, заданной выражением (7), является следующее: надо так назначать и корректировать цену на информационный ресурс, чтобы обеспечить в течение всего времени продаж линейно возрастающую интенсивность продаж. При известной функции спроса, которую можно найти аппроксимацией данных эксперимента, полученные условия позволяют выбрать оптимальное изменение цены продаж в функции, определяющей число проданных экземпляров информационного ресурса.

### Литература

1. Климов С.М. Интеллектуальные ресурсы общества. СПб: Изд-во ИВЭСЭП, Знание, 2002. 56 с.
2. Шуть О.Н. Информация как экономический ресурс. URL: <http://www.cfin.ru/bandurin/article/sbrn02/08.shtml> (дата обращения: 12.03.2017).
3. Ясенов В.Н. Информационные системы и технологии в экономике. М.: Юнити-Дана, 2008. 560 с.
4. Сафиуллин Л.Н., Улесов Д.В. Предпринимательская деятельность на рынке услуг: сущность и особенности развития в экономике знаний. Казань: Центр инновационных технологий, 2010. 201 с.
5. Соловьев В.И. Стратегия и тактика конкуренции на рынке программ: опыт экономико-математического моделирования: монография. М.: Вега Инфо, 2010. 200 с.
6. Прошлякина А.Н. Особенности ценообразования на информационные продукты и услуги, бизнес // Вестн. Волгоградского ин-та бизнеса. 2014. № 2. С. 80–85.
7. Данилова У.В. Ценообразование на рынке програм-

мных продуктов: автореф. дисс. ... канд. экон. наук. Владивосток: ИАПУ ДВО РАН, 2003. 25 с.

8. Mehrmann J. Seven steps to success: marketing, promotion and selling software. Business Know-How. URL: <https://www.businessknowhow.com/marketing/salessteps.htm> (дата обращения: 26.03.2017).

businessknowhow.com/marketing/salessteps.htm (дата обращения: 26.03.2017).

9. Кремер Н.Ш. Высшая математика для экономики. М.: Юнити-Дана, 2007. 128 с.

Software & Systems

DOI: 10.15827/0236-235X.119.474-477

Received 27.03.17

2017, vol. 30, no. 3, pp. 474–477

### CALCULATION OF THE OPTIMAL DEPENDENCY OF INFORMATION RESOURCE PRICE DURING ITS MARKETING AT SEVERAL MARKETS

S.A. Amelkin<sup>1</sup>, Ph.D. (Engineering), Head of the Center, [sergey.a.amelkin@gmail.com](mailto:sergey.a.amelkin@gmail.com)

O.S. Ivanova<sup>2</sup>, Engineer, [olety@yandex.ru](mailto:olety@yandex.ru)

<sup>1</sup> System Analysis Research Center, Program System Institute of RAS, Petr I Ave. 4a, Pereslavl-Zalesskiy, 152021, Russian Federation

<sup>2</sup> Medical Informatics Research Center, Program System Institute of RAS, Petr I Ave. 4a, Pereslavl-Zalesskiy, 152021, Russian Federation

**Abstract.** The paper considers the problem of determining the pricing policy of the company that implements a market information resource. One of the key challenges of the information market – information cost determination. However, this raises a number of difficulties due to the nature of the goods.

The paper considers the information goods as a certain software product. The sale of information resources is characterized by the fact that replication of this resource can be without any significant additional costs. This leads to increasing in resource volume during its dissemination. At the same time, price setting is one of the key factors determining further distribution and promotion of goods, which takes into account a variety of factors including the way it is distributed.

The paper solves the problem of determining the optimal dependence of the information resource price on time for the case when the retailer has no competitors on the market, the market is saturated, and buyers do not have the possibility of further dissemination of the resource. The authors obtain the conditions for optimal selection of the information resource price, calculate the maximum possible income of the company for a linear demand function, and describe the conditions governing the duration and volume of sales ensuring a high average company income.

When the function of demand is known (it can be found using an approximation of the experimental data), the obtained conditions allow selecting the optimal change in sell prices as a function of the number of sold copies of the information resource.

**Keywords:** information resource, optimal price, demand function, the intensity of sales, necessary optimality conditions.

### References

1. Klimov S.M. *Intellectualnye resursy obshchestva* [Intellectual Resources of Society]. St. Petersburg, Znanie Publ., 2006, 56 p.
2. Shut O.N. *Informatsiya kak ekonomichesky resurs* [Information as an Economic Resource]. Available at: <http://www.cfin.ru/bandurin/article/sbrn02/08.shtml> (accessed March 12, 2017).
3. Yasenev V.N. *Informatsionnye sistemy i tekhnologii v ekonomike* [Information Systems and Technologies in Economy], Moscow, Yuniti-Dana Publ., 2008, 560 p.
4. Safiullin L.N., Ulesov D.V. *Predprinimatelskaya deyatel'nost' na rynke uslug: sushchnost' i osobennosti razvitiya v ekonomike znaniy* [Entrepreneurial Activity in the Services Market: the Nature and Development Features in Knowledge Economy]. Monograph. Kazan, 2010, 201 p.
5. Solovov V.I. *Strategiya i taktika konkurentsii na rynke program: opyt ekonomiko-matematicheskogo modelirovaniya* [Competition Strategy and Tactics in the Software Market: Economic and Mathematical Modeling Experience]. Monograph. Moscow, Vega Info, 2010, 200 p.
6. Proshchalykina A.N. Pricing features of information products and services, business. *Vestnik Volgogradskogo instituta biznesa* [The Bulletin of the Volgograd Business Institute]. 2014, no. 2 (27), pp. 80–85 (in Russ.).
7. Danilova U.V. *Tsenoobrazovanie na rynke programnykh produktov* [Pricing in the Software Products Market]. PhD Thesis, Vladivostok, IACP FEB RAS Publ., 2003.
8. *Seven Steps to Success: Marketing, Promotion and Selling Software Online*. Chado Software, Inc. Available at: <https://www.businessknowhow.com/marketing/salessteps.htm> (accessed March 26, 2017)
9. Kremer N.Sh. *Vysshaya matematika dlya ekonomiki* [Higher Mathematics for Economics]. Moscow, Yuniti-Dana, 2007, 128 p.

УДК 004.89, 004.91

DOI: 10.15827/0236-235X.119.478-486

Дата подачи статьи: 22.05.17

2017. Т. 30. № 3. С. 478–486

**МЕТОД ЧАСТОТНО-МОРФОЛОГИЧЕСКОЙ КЛАССИФИКАЦИИ ТЕКСТОВ**

А.А. Осочкин, аспирант, osa585848@bk.ru;

В.В. Фомин, д.т.н., профессор, v\_v\_fomin@mail.ru;

А.В. Флегонтов, д.т.н., профессор, flegontoff@yandex.ru

(Российский государственный педагогический университет им. А.И. Герцена,  
наб. реки Мойки, 48, г. Санкт-Петербург, 191186, Россия)

Появление централизованных хранилищ данных и накопление в них информации в виде как структурированных таблиц, так и слабоструктурированных текстов стали следствием растущего внимания к методам анализа данных. Анализ данных в перспективе позволяет получать важную информацию, на основе которой можно принять верное управленческое решение или спрогнозировать дальнейшее развитие событий. Одним из важных направлений этого анализа является автоматическая классификация накопленных данных в электронном виде, упрощенная модель которой сводится к считыванию, обработке текста и присвоению документу темы из заранее заданного списка. Все чаще работы зарубежных коллег посвящаются классификации данных в области медицины для последующего прогноза развития болезни на основе статистики или постановки диагноза на основе истории болезни. Главную сложность в классификации представляют тексты на естественном языке, которые в силу лингвистических особенностей языка и поддержки частью методов классификации исключительно числовых данных трудно поддаются классификации.

В настоящей работе исследуется научная активность в сфере классификации данных на естественном языке на основе ежегодной публикации научных трудов в данной сфере, а также предлагается на рассмотрение метод классификации русскоязычных текстов, интегрирующий в себе алгоритмы частотного, морфологического и интеллектуального анализов.

Процедура классификации текстов предполагает применение частотных, морфологических показателей и регрессионных деревьев. Также в данной работе представлены результаты ряда экспериментов по идентификации метода классификации с наиболее высокой точностью. Классификация осуществлялась по функциональным, литературным и авторским стилям.

**Ключевые слова:** классификация текстов, частотный анализ, морфологический анализ, деревья решений, data mining, text mining.

В среде информационно-коммуникационных технологий и систем происходят устойчивый рост и накопление текстовой слабоструктурированной информации [1], увеличивается объем хранилищ данных (библиотек, банков данных, репозиторий и т.д.). Потребность в эффективном извлечении ценных знаний из текстовых массивов влечет за собой усложнение и появление новых методов обработки информации – интеллектуального анализа текстов (text mining), в том числе за счет применения ресурсоемких статистических алгоритмов, алгоритмов интеллектуального анализа данных (data mining) [2], семантического поиска, использования сетевых и интернет-технологий и т.д. Вследствие роста объема данных и времени их обработки из-за сложности алгоритмов растут затраты на повышение производительности вычислительной техники.

Задачей развития методов text mining является извлечение полезных знаний из информационных массивов с учетом особенностей обработки естественного языка (ЕЯ), в том числе классификация текстов, извлечение информации, реферирование, информационный поиск и т.д. [3, 4]. Методы text mining используются в различных программных и информационных технологиях и как отдельные приложения, библиотечные модули, и в составе инструментария интеллектуального анализа данных, систем бизнес-аналитики, корпоративного управления и т.д. Одной из ключевых задач text mining является классификация текстов на ЕЯ [5].

Рост объема массивов данных и потребность в эффективном извлечении из них ценной информации обуславливают усложнение и появление новых методов обработки информации [6], в том числе за счет применения ресурсоемких статистических алгоритмов, алгоритмов семантического поиска, нейронных и интернет-технологий и т.д. Из-за увеличения массивов информации, подлежащих обработке, и применения для анализа все более сложных и глубоких методов растут требования к вычислительным ресурсам.

В борьбе с растущей сложностью и увеличивающимися затратами на технологии обработки текстов прослеживается тенденция возврата к классическим методам частотно-морфологического анализа. Поисково-текстовые методы и алгоритмы [7] акцентируют внимание на попытке использования небольшого, минимального арсенала теоретико-лингвистических изысков и делают акцент на формальных методах статистической обработки упрощенных словоформ. Авторами статьи разработан специальный комбинированный метод классификации с целью решения вышеназванной проблемы. Он включает в себя этап извлечения из текста числового набора показателей, что позволяет применить для классификации методики data mining, тем самым расширяя спектр возможных методик, применяемых для классификации.

Ключевой особенностью комбинированного метода классификации является наличие коллек-



ции алгоритмов, позволяющей выбрать из нее наиболее эффективные для классификации анализируемого набора текстов. Коллекция алгоритмов не статична: появляющиеся новые алгоритмы классификации могут быть включены в комбинированную методику классификации текстов, что делает ее еще более универсальной.

### Обзор исследований в области классификации данных на ЕЯ

Исследования в области классификации данных на ЕЯ начались в середине XX века. Динамика развития публикационной научной активности по классификации текстов проиллюстрирована на рисунке 1, где представлены результаты исследования количества научных работ, опубликованных в период с 1960 по 2008 гг. [8].

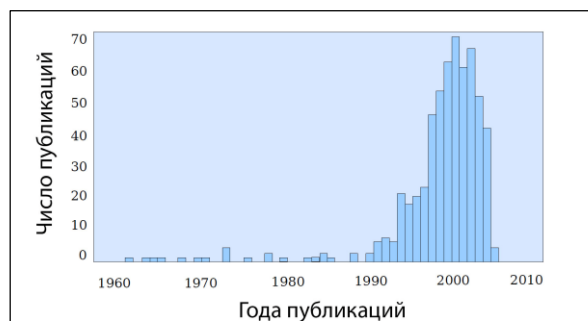


Рис. 1. Публикации в сфере text mining и обработки текстов на ЕЯ в 1960–2008 гг.

Fig. 1. Publications in text mining and natural language text processing in 1960–2008

На основе интернет-поисковика Base [9] был проведен обзор публикаций, книг и других работ в данной области в период с 2008 по 2016 гг. Результаты представлены на рисунке 2.

Для отбора работ использовались ключ «Классификация текстов» и фильтры classification и text mining.

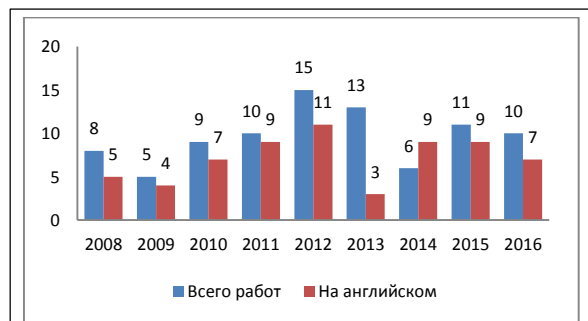


Рис. 2. Публикации в сфере text mining и обработки текстов на ЕЯ в 2008–2016 гг.

Fig. 2. Publications in text mining and natural language text processing in 2008–2016

При этом за период с 2008 по 2016 гг. учитывались не только изданные книги, но и публикации в различных известных журналах [9], находящихся в БД. На рисунке 2 в левых колонках отражено общее количество публикаций в данной сфере независимо от языка, в правых – количество публикаций только на английском языке. Средняя доля публикаций в период с 2008 по 2016 гг. на английском языке от общего числа публикаций за данный период составляет 76 %. Это говорит о том, что данное направление активно развивается англоязычными авторами и, как следствие, рассматривается классификация данных на английском языке.

Теперь сузим тему. Для этого выполним поиск работ по ключу text classification и используем два фильтра: classification и machine learning (рис. 3).



Рис. 3. Публикации в сфере text mining и обработки текстов на ЕЯ (английском) с 2008 по 2016 гг.

Fig. 3. Publications in text mining and natural language processing in English from 2008 to 2016

Графики на рисунке 3 отражают тенденцию спада интереса к данной задаче и, как следствие, количества работ в данной области с 2005 года.

В современных работах по классификации текстов авторы все чаще опираются на статистические способы классификации данных, полученных при помощи синтаксического и семантического анализа.

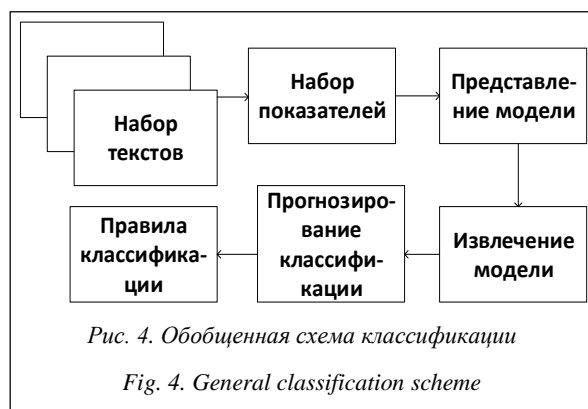
### Классификация наборов текстов

В классификации текстовых данных при опоре на статистические методы важным этапом является процесс извлечения набора статистических показателей (рис. 4).

Набор показателей из данных на ЕЯ может быть получен при помощи различных процедур анализа, которые условно можно разделить на три вида: частотный, морфологический и морфологическо-семантический анализ.

Именно этап извлечения набора показателей является наиболее ресурсоемким и важнейшим фактором, влияющим на эффективность классификации в целом.

Поэтому для дальнейшего развития таких направлений text mining и обработки текстов на ЕЯ необходимо искать методы классификации не только высокоэффективные по точности, но и поз-



воляющие сократить использование вычислительных ресурсов при классификации данных.

Сокращение ресурсоемкости при классификации данных на ЕЯ может быть достигнуто путем применения различных методов классификации, в том числе основанных на теории вероятности и распространенной теореме Байеса, а также упрощенных статистических мерах [10] (частота вхождения слова в текст). Эти два подхода демонстрируют проблематику баланса между ресурсоемкостью и точностью классификации.

Большой потенциал заложен в методах, основанных на деревьях решений (регрессионных деревьях). В классификации данных деревья решений используются не так часто, как другие методы. Это обусловлено вытеснением метода деревьев решений другими методиками классификации данных, особенно основанных на теории вероятности, которые в частных случаях показали более точный результат классификации.

Однако есть работы, экспериментально доказывающие эффективность метода дерева решений для классификации данных. Метод дерева решений основан на машинном обучении, поэтому для него обязательна учебная выборка. Значительным положительным эффектом деревьев решений является их способность снижать множество признаков, оставляя только значимые. Важным преимуществом деревьев решений является логический аппарат интерпретации и пояснения результата. Использование метода дерева решений позволяет классифицировать данные, с высокой точностью используя минимальный набор показателей, тем самым уменьшая ресурсоемкость классификации.

Рассмотрим гипотезу о том, что часть задач классификации текстов можно успешно осуществить с небольшими затратами вычислительных ресурсов путем манипуляции минимальным набором частотных характеристик, ограниченным формализованным набором ЕЯ и алгоритмами классификации. Решение данной задачи можно свести к применению различного рода алгоритмов морфологического, частотного анализа для получения минимального набора данных и к использованию эффективного метода классификации.

Уменьшение ресурсоемкости процесса возможно путем использования частотно-морфологического анализа, позволяющего получить минимальный набор теоретико-лингвистических показателей и статистически упрощенных словоформ, извлекаемых из текстов. При таком подходе уменьшается алгоритмическая и вычислительная нагрузка на процесс оценки и классификации текстов, но ограничивается интерпретация смыслового контекста извлекаемой информации.

Полученный в результате морфологического анализа теоретико-лингвистический набор показателей можно условно разделить на морфологические словоформы и их характеристики, а также синтаксические показатели.

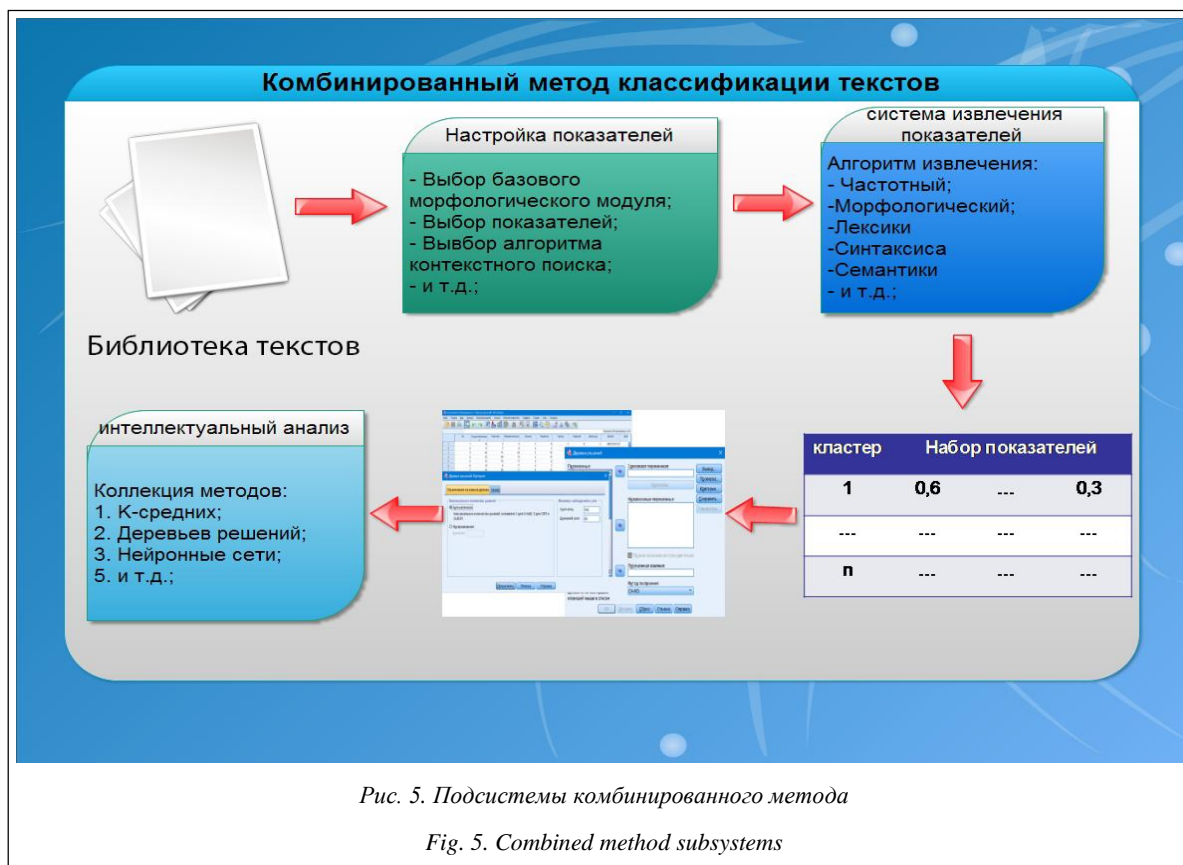
Синтаксис – форма или структура выражений, предложений и программных единиц. Исторически сложилось так, что синтаксическая единица больше слова и представляет собой словосочетание или предложение. Извлечение синтаксиса из текста позволяет осуществлять синтаксический анализ текста и на его основе строить деревья синтаксического анализа, которые часто дублируются, так как предложения построены на общих синтаксических правилах ЕЯ.

Результаты ряда экспериментов [11] показали, что тексты в целом содержат не более 3–10 отличающихся друг от друга видов синтаксических деревьев. Их вариативность может быть объяснена множеством факторов [12], таких как использование различных стилей речи при написании текста (официальный, разговорный и т.д.), стилистическая особенность автора, которая выражается в нарушении грамматических правил, построения предложений и т.д. Таким образом, последовательность частей речи позволяет получить множество различных показателей, которые отображают не всегда очевидные, но характерные для какого-либо кластера характеристики.

Число последовательностей частей речи для каждого ЕЯ разное, так как число частей речи в них варьируется, но для большинства языков в среднем оно равно 10. Таким образом, количество различных цепочек частей речи может составлять  $10^{10}$ . Большинство частей речи имеют индивидуальные характеристики, например, для существительных это падеж, одушевленное, неодушевленное и т.д. При учете индивидуальных характеристик еще больше обостряется проблема роста вычислительных ресурсов.

#### Комбинированный метод классификации текста

Комбинированный метод классификации текста – это комплекс применяемых алгоритмов, приемов, операций и инструментария к анализируемому набору текста для его классификации. В комбинированный метод заложены концепция



минимальности действий для использования и замены модулей и компонентов, а также возможности расширения функций за счет привлечения сторонних программ и библиотек.

Система реализации text mining на базе комбинированного метода состоит из трех блоков (см. рис. 5).

1. Библиотека текстов – поиск, извлечение, каталогизация, хранение текстов. Предполагает развитую систему управления: навигация, добавление, удаление, просмотр, форматирование и пр.

2. Извлечение параметров – извлечение числовых параметров текста, морфологический и синтаксический анализ текста, ведение БД параметров и т.д.

3. Интеллектуальный анализ данных – обучение, кластеризация, идентификация объектов-текстов, формирование статистических данных, обучающей выборки в форматах алгоритмов интеллектуального анализа данных. Включает коллекцию алгоритмов для выбора наиболее эффективных методов классификации.

Ключевой особенностью комбинированного метода text mining является возможность выбора наиболее эффективного алгоритма data mining (из коллекции алгоритмов) для классификации анализируемого набора текстов. Коллекция алгоритмов не статична, и по мере развития исследований в нее могут быть включены новые алгоритмы классификации.

### Описание программы

Для реализации комбинированного метода классификации было разработано специальное программное средство – Frequency and morphological analysis, или сокращенно FaM. Основная цель программы – извлечение при помощи частотного и морфологического анализа из анализируемых текстов информации о частоте употребления слов и их характеристиках, расчет на их основе коэффициентов для последующего анализа в интеллектуальных пакетах данных.

Для частотного и морфологического анализов используются интегрированные модули сторонних авторов, работающие только в 64-битных версиях Windows. Исходя из этого факта, анализ текста возможен только под управлением ОС Windows 64-битной версии.

Программа разрабатывается на объектно-ориентированном языке C#, среда разработки – Microsoft Visual Studio 2013 SP3. Этот выбор обусловлен отсутствием поддержки у некоторых модулей анализа text mining ОС Linux.

Результатом анализа текста является набор показателей, который содержится в БД программы, а также может быть продублирован в Excel для быстрого экспорта в интеллектуальные пакеты анализа данных.

На рисунке 6 представлен интерфейс программы версии 1.01. Цифрами на рисунке обозна-

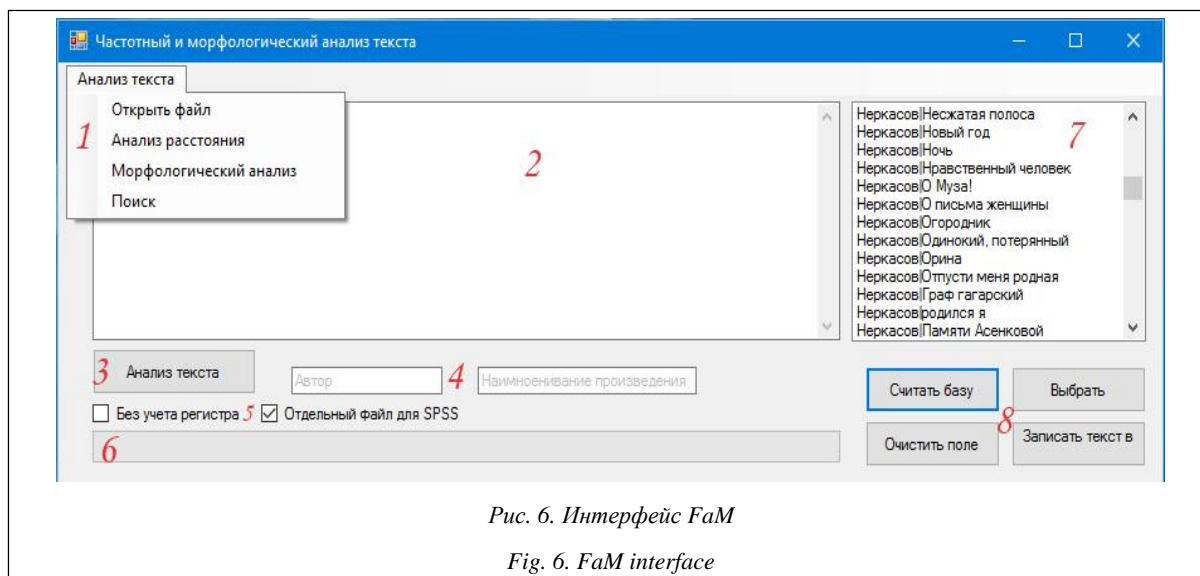


Рис. 6. Интерфейс FaM

Fig. 6. FaM interface

чено следующее: 1 – меню, позволяющие перейти к дополнительным видам анализа; 2 – основное поле работы, в которое пользователь может ввести текст вручную или импортировать из документов; 3 – кнопка для осуществления частотного и морфологического анализов; 4 – два поля для ввода автора и наименования произведения для удобства навигации по архиву, ввод необязателен; 5 – две опциональные настройки анализа: а) анализ с учетом регистра нижних и верхних букв, б) экспорт результатов анализа в специальный лист Excel, позволяющий сразу же импортировать его в SPSS; 6 – шкала прогресса, отображающая во время работы завершенность процедуры; 7 – список текстов в базе; 8 – кнопки для работы с базой (считать базу, выбрать, записать текст), очистить поле – быстрая очистка основного поля.

#### Реализация частотного анализа в комбинированном методе классификации текста

В комбинированном методе классификации текста наиболее сложными этапами являются частотный и морфологический анализы.

При частотном анализе была поставлена задача не только получить точные частотные данные, но и оптимизировать процесс анализа, уменьшив время, необходимое для контекстного поиска. Схема частотного анализа в комбинированном методе классификации текста изображена на рисунке 7.

Инициализация алгоритма частотного анализа начинается с процедуры считывания текста.

Базовыми в частотном анализе являются алгоритм подсчета всех символов, определения содержания иностранных слов в тексте и необходимость использования морфологических модулей для других языков.

Следующим этапом является очистка текста от спама, под которым подразумеваются скобки, ка-

вычки, слитные с текстом знаки препинания и т.п. Данная процедура позволяет очистить слова от знаков препинания, тем самым повысив точность при поиске словоформ в морфологическом анализе. Чтобы не испортить оригинальный текст, создается дубликат, к которому применяется фильтрация от спама.

После очистки программа приступает непосредственно к частотному анализу, используя алгоритм контекстного поиска слов. Для обеспечения максимальной скорости определена процедура подключения алгоритма контекстного поиска из двух алгоритмов – Боейра–Мура или brute force. Выбор алгоритмов не случаен, он основан на результатах экспериментов по скорости поиска. Если объем текста превышает 800 символов, программа

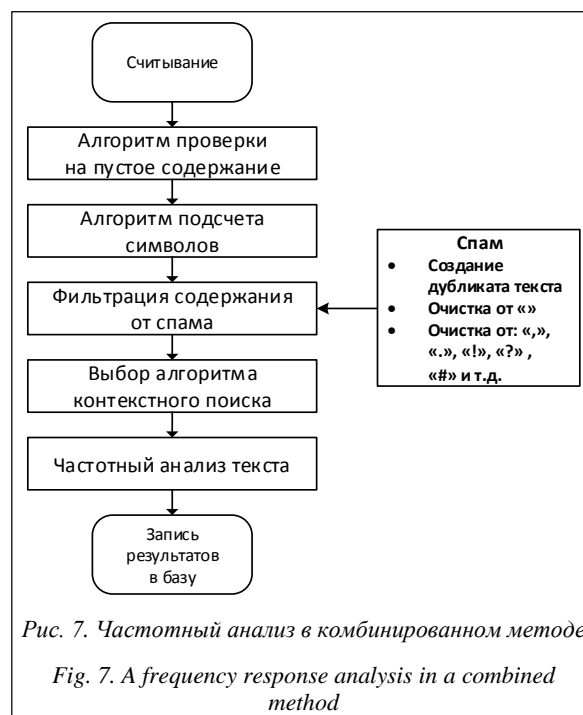


Рис. 7. Частотный анализ в комбинированном методе

Fig. 7. A frequency response analysis in a combined method

автоматически использует алгоритм Боейра–Мура, если меньше – brute force.

Описанные этапы частотного анализа готовят текст для дальнейшего морфологического анализа.

### Реализация морфологического анализа

Ключевым элементом комбинированного метода классификации текста является морфологический анализ, структура которого показана на рисунке 8.

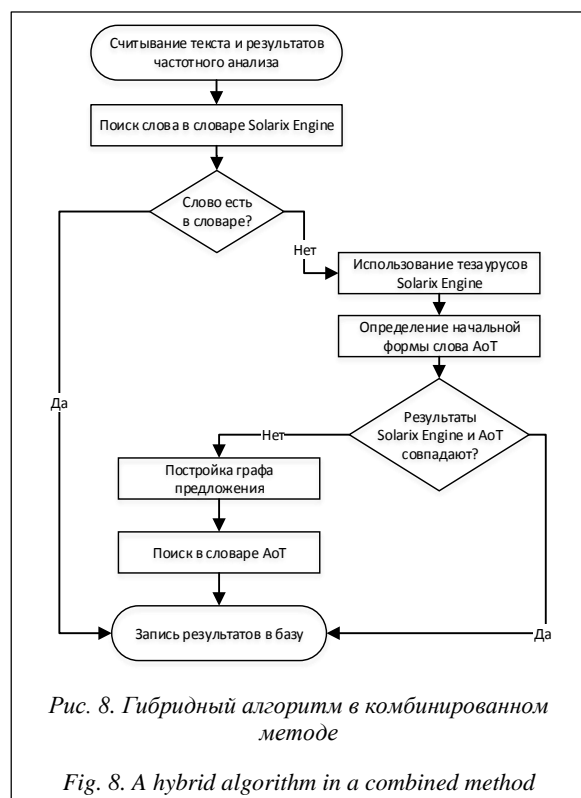


Рис. 8. Гибридный алгоритм в комбинированном методе

Fig. 8. A hybrid algorithm in a combined method

Морфологический анализ осуществляется специальным гибридным алгоритмом с использованием двух встроенных модулей – автоматической обработки текста, или AoT, и Solarix Engine. Морфологический модуль AoT [13] содержит русский морфологический словарь, включающий около 161 000 слов с различными формами, а также синтаксический и семантический анализ текста. Solarix Engine [14] – модуль морфологического анализа, включающий в себя словарь из 1 800 000 слов и 218 000 статей; тезаурус содержит информацию о возможных отношениях подчинения и ассоциативных отношениях между словами для машинного обучения.

В гибридном алгоритме осуществляется двойная проверка на часть речи, что позволяет получить более точные результаты. Если результаты двух модулей противоречат друг другу, строится граф предложения, в процессе построения идентифицируются взаимосвязи между словами и частями речи.

Таким образом, использование в морфологическом анализе двух морфологических модулей разных типов является одной из особенностей гибридного алгоритма.

Был проведен эксперимент по оценке точности морфологического анализа. Анализировались пять простых текстов длиной не более 200 слов, причем заранее было известно, какой частью речи является каждое слово. Точность анализа определялась по формуле  $Qx = \frac{\sum X_i}{\sum Y_i}$ , где  $X_i$  – количество слов, от-

несенных морфологическим модулем к части речи  $i$ ;  $Y_i$  – количество слов, которые действительно относятся к части речи  $i$ .

В таблице 1 показано, что в среднем общая точность двух модулей одинаковая. Стоит обратить внимание, что модуль AoT почти всегда ошибался при определении деепричастия, а модуль Solarix Engine – при анализе причастий и прилагательных. Однако при анализе всего предложения целиком (тезаурусом) точность повысилась на 9 %.

Таблица 1

Оценка точности модулей, %

Table 1

Module accuracy estimation, %

Текст	AoT	Solarix Engine	Solarix Engine с тезаурусами	AoT + Solarix Engine с тезаурусами
№ 1	76	76	82	86
№ 2	73	81	89	96
№ 3	71	74	79	93
№ 4	77	77	86	91
№ 5	79	68	82	88
Ср. рез.	75	75	84	91

Применение к 16 % слов, которые не смог идентифицировать Solarix Engine, метода AoT с лексикализацией (приведением слов в начальную форму) позволило повысить точность еще на 7 %.

### Набор показателей

После получения в результате частотного и морфологического анализов набора числительных показателей алгоритм выполнения комбинированного метода рассчитывает на основе формул относительные показатели. Набор состоит из 76 показателей, 55 из которых пользователь может отключать или подключать в настройках для уменьшения времени, необходимого для расчета коэффициентов. Опишем базовые формулы, используемые для расчета основных показателей.

$$D_i = \frac{k_i}{\sum_{j=1}^n k_j} - \text{доля частей речи (глаголы, суще-}$$

ствительные, прилагательные, наречия, частицы, союзы и т.д.), где  $n$  – количество частей речи;  $k_j$  – количество слов в тексте, принадлежащих к  $j$ -й ча-

сти речи. Коэффициент, отображающий долю  $i$ -й части речи от общего числа других частей речи в тексте.

$$L_i = \frac{C}{K} - \text{средняя длина} - \text{усредненный коэффициент, где } C - \text{общее количество единиц информации; } K - \text{общее количество слов, предложений или абзацев. Отображает } i\text{-е отношение числа единиц информации (символов, слов, предложений) к количеству синтаксически более сложных образований в тексте (слов, предложений, абзацев). К таким показателям относятся среднее количество символов на слово, слов на предложение, предложений на абзац.}$$

или гласных, или согласных, или числительных и т.д.;  $K_{\text{сим}}$  – общее количество символов в тексте. Коэффициент, отображающий среднее число гласных, согласных, числительных и других знаков, приходящихся на общее количество символов.

$$C_{\text{сим}} = \frac{c}{K_{\text{сим}}} - \text{доля символов, где } c - \text{количество}$$

или гласных, или согласных, или числительных и т.д.;  $K_{\text{сим}}$  – общее количество символов в тексте. Коэффициент, отображающий среднее число гласных, согласных, числительных и других знаков, приходящихся на общее количество символов.

$$N_{\text{пад}} = \frac{n_{\text{пад}}}{N} - \text{количество существительных в}$$

падеже, где  $n_{\text{пад}}$  – существительных в падеже;  $N$  – общее количество существительных. Коэффициент, отображающий число существительных в различных падежах, приходящихся на общее число существительных.

Возможность расчета большого количества коэффициентов увеличивает вероятность того, что один из показателей отображает уникальное свойство как объекта, так и класса. После завершения расчета коэффициентов они записываются в базу Excel, которая экспортируется в интеллектуальный пакет анализа данных.

Полученные результаты могут быть экспортированы в разные интеллектуальные пакеты анализа данных. В данной работе в качестве интеллектуальной программы анализа данных были выбраны три программы, имеющие множество методов классификации данных: IBM SPSS Statistic 23, Rapid miner Studio Free 7.2, SciKit-Learn 0.18rc2.

На вход интеллектуальному пакету анализа данных с алгоритмом классификации подается набор показателей, полученных в результате анализа 100 текстов (на русском языке), принадлежащих четырем функциональным стилям: литературному – 20 текстов, новостному – 20, научному – 40, официальному – 20.

С целью обучения в набор показателей добавлен условный кластер, который принимает четыре значения: литературный, новостной, научный, официальный.

Также стоит отметить общие настройки некоторых методов:

– количество конечных кластеров – 4, контрольная выборка – 40 %;

– евклидово расстояние, используемое для измерения расстояний в методе ближайшего соседа;

– в методе дерева решений количество итераций равно 100, количество отношений между деревом-отцом и деревом-сыном – min 2.

Результаты классификации текстов при помощи различных алгоритмов в интеллектуальных пакетах данных представлены в таблице 2.

Таблица 2

## Тестовая классификация

Table 2

## Test classification

Программа	Метод классификации	Точность, %
Rapid miner	Метод ближайшего соседа	68
Rapid miner	Дерево решений	63
Rapid miner	Многоуровневое дерево решений	95
Rapid miner	Случайных деревьев (лучший результат)	95
SciKit-Learn	Дерево решений	88
SciKit-Learn	Метод Байеса	88
SciKit-Learn	Байес с опорными векторами	87
SciKit-Learn	Роше	50
SciKit-Learn	Метод ближайшего соседа	76
IBM SPSS	Метод ближайшего соседа	73
IBM SPSS	Дерево решений (исчерпывающий CHAID)	98
IBM SPSS	Дерево решений (метод CHAID)	94
IBM SPSS	Дерево решений (метод CRT)	83

Основным алгоритмом интеллектуального анализа для дальнейших экспериментов был выбран SPSS (дерево решений и исчерпывающий CHAID). Как видно из таблицы 2, этот алгоритм обеспечил наиболее точную классификацию текстов – 98 %, что на 3 % выше, чем показал метод деревьев решений в программе Rapid miner.

## Эксперименты

С целью классификации русскоязычных текстов был проведен ряд экспериментов, результаты которых также были описаны в работах [15, 16]. В таблице 3 представлены итоги основных экспериментов. Следует отметить, что в экспериментах варьируются размеры выборки и свойства классифицируемых объектов.

Первый и второй эксперименты (проза) – классификация литературных произведений русских классиков; эксперименты направлены на определение сложной структуры стиля автора за счет структуры предложения и частей речи в нем.

Третий эксперимент – классификация художественных произведений современников по литературным жанрам.

Четвертый эксперимент – классификация текстов по функциональным стилям. Цель экспери-

## Результаты экспериментов

Таблица 3

## Experimental results

Table 3

Эксперимент	Автор	Контрольная выборка 20 %	Контрольная выборка 50 %	Перекрестная проверка
№1	<b>Произведения (6 кластеров)</b>			
	Д.А. Гранин	75,00 %	66,70 %	<b>89,40 %</b>
	Ф.М. Достоевский	100,00 %	94,10 %	
	А.И. Куприн	100,00 %	88,20 %	
	Л.Н. Толстой	100,00 %	78,90 %	
	А.П. Чехов	100,00 %	100,00 %	
	М.А. Шолохов	87,50 %	100,00 %	
	<b>Общая точность</b>	91,40 %	86,70 %	
№ 2	Количество текстов	36	90	180
	<b>Произведения (5 кластеров)</b>			
	Ф.М. Достоевский	75,00 %	73,70 %	<b>94,70 %</b>
	А.И. Куприн	88,90 %	83,30 %	
	Л.Н. Толстой	100,00 %	82,60 %	
	А.П. Чехов	100,00 %	100,00 %	
	М.А. Шолохов	100,00 %	100,00 %	
	<b>Общая точность</b>	91,20 %	85,90 %	
	Количество текстов	30	75	150
№ 3	<b>Жанры литературы (4 кластера)</b>			
	Детектив	66,70 %	87,50 %	<b>69,90 %</b>
	Люб. роман	50,00 %	66,70 %	
	Фэнтези	80,00 %	75,00 %	
	Хоррор	100,00 %	63,60 %	
	<b>Общая точность</b>	77,80 %	72,20 %	
№ 4	Количество текстов	16	40	<b>80</b>
	<b>Функциональный стиль (4 кластера)</b>			
	Литературный	100,00 %	100,00 %	<b>85,00 %</b>
	Новостной	71,00 %	30,00 %	
	Научный	100,00 %	95,50 %	
	Официальный	100,00 %	92,90 %	
	<b>Общая точность</b>	90,00 %	84,20 %	
	Количество текстов	16	40	<b>80</b>

мента – определить возможность такой классификации данных на ЕЯ.

Результаты экспериментов подтвердили, что использование комбинированного метода позволяет с небольшой ресурсоемкостью и высокой достоверностью осуществить классификацию текстов по ряду задач семантической направленности.

### Заключение

Благодаря использованию при анализе текста интегральной технологии частотного, морфологического и интеллектуального анализов исследование позволило выявить закономерности классификации слабоформализуемой информации.

Представленная процедура классификации текста ограничена полноценной обработкой текстов только на русском языке, что делает ее более узкоспециализированной. Однако использование частотно-морфологического анализа, гибкость набора показателей, скорость выполнения анализа, вариативность, применение большого числа методов классификации, а также результаты классификации набора текстов из разных функциональных

стилей и их высокая достоверность распознавания позволяют говорить о возможности использования предложенной технологии как одного из эффективных инструментов для анализа естественно-языковой информации.

### Литература

1. Soergel D. Organizing information: principles of data base a. retrieval systems. Orlando: Acad. press, 1985, vol. 14, 450 p.
2. Weiss S., Indurkha N. Predictive data mining: a practical guide. SF, Morgan Kaufmann, 1998, 228 p.
3. Bird S., Klein E., Loper E. Natural language processing with Python. Sebastopol: O'Reilly Media, 2015, 479 p.
4. Islam M.Z., Rahm A., Mehler R. Text readability classification of textbooks of a low-resource language. Proc. 26th Pacific Asia Conf. on Language, Information, and Computation, 2012, pp. 545–553.
5. Manning D., Schutze H. Foundations of statistical natural language processing. Cambridge: MIT Press, 2000, 680 p.
6. Dumais S., Chen H. Hierarchical classification of web content. Proc. 23rd Annual Intern/ ACM SIGIR Conf. on Research and Development in Information Retrieval, 2002, pp. 256–263.
7. Aggarwal C. Data mining: the textbook. NY, Springer, 2015, 734 p.
8. Medlock B.W. Investigating classification for natural language processing tasks. Univ. Cambridge Publ., 2008, 138 p.
9. Internet papers base "Base". URL: <https://www.base-search.net> (дата обращения: 07.05.2017).
10. Weiss M.S. et al. Maximizing Text-mining performance.



Jour. Intelligent Information Retrieval, 1999, vol. 14, pp. 63–69.

11. Hellwig O. Improving the morphological analysis of classical Sanskrit. Düsseldorf Univ. URL: <http://aclweb.org/anthology/W/W16/W16-3715.pdf> (дата обращения: 07.05.2017).

12. Caropreso M.F., Matwin S., Sebastiani F. Learner-independent evaluation of the usefulness of statistical phrases for automated text categorization: in Publ. *Text databases and document management: theory and practice*. Virginia Commonwealth Univ. Publ., 2001, pp. 78–102.

13. Official website of the program automatic text processing «AoT». Chart. Russian morphological dictionary. URL: <http://aot.ru/index.html> (дата обращения: 07.05.2017).

14. Official website of the program Solarix Engine, Chart. Computer Russian grammar. URL: <http://www.solarix.ru/index-ru.shtml> (дата обращения: 07.05.2017).

15. Фомин В.В., Осочкин А.А. Классификация текстов на основе частотного и морфологического анализов с применением алгоритмов Data-mining // Информатизация образования и науки. 2016. Вып. 3. С. 137–152.

16. Фомин В.В., Фомина И.К., Осочкин А.А. Классификация текстов на основе частотного и морфологического анализов с применением алгоритмов дата-мининг // Актуальные вопросы и перспективы развития математических и естественных наук. 2016. № 3. С. 64–69.

Software & Systems

DOI: 10.15827/0236-235X.119.478-486

Received 22.05.17

2017, vol. 30, no. 3, pp. 478–486

## METHOD OF FREQUENCY-MORPHOLOGICAL CLASSIFICATION OF TEXTS

A.A. Osochkin<sup>1</sup>, Postgraduate Student, [osa585848@bk.ru](mailto:osa585848@bk.ru)

V.V. Fomin<sup>1</sup>, Dr.Sc. (Engineering), Professor, [v\\_v\\_fomin@mail.ru](mailto:v_v_fomin@mail.ru)

A.V. Flegontov<sup>1</sup>, Dr.Sc. (Engineering), Professor, [flegontoff@yandex.ru](mailto:flegontoff@yandex.ru)

<sup>1</sup> Herzen State Pedagogical University of Russia, Reki Moyki Quay 48, St. Petersburg, 191186, Russian Federation

**Abstract.** Appearing of centralized data storages and information accumulation as structured tables or semistructured texts is a result of growing attention to data analysis techniques. The analysis of similar data in the long term allows obtaining important information, which can become a basis for making right management decisions or predicting further development of events in many fields. One of the important directions of such analysis is automatic classification of collected data in electronic form. Its simplified model is reduced to reading, text processing and assigning a topic to a document from a given list of topics. Foreign papers more and more often are devoted to medical data classification for further disease progression forecast on the basis of statistics or diagnosis based on medical history. The main difficulty in classification are natural language texts. They are difficult to classify due to linguistic features of language and support by a part of classification methods of exclusively numerical data.

The paper studies scientific activity in the field of NLP based on the annual publication of scientific papers in this field. It also offers the method of Russian-language texts classification that integrates the algorithms of frequency, morphological and intellectual analysis. The paper presents the results of some experiments on the identification method of classification with high classification accuracy. The classification was carried out according to functional, literary, and authorial styles.

**Keywords:** text classification, frequency analysis, morphological analysis, trees of decisions, data mining, text mining.

## References

1. Soergel D. *Organizing information: principles of data base and retrieval systems*. Orlando, Acad. press, 1985, vol. 14, 450 p.
2. Weiss S., Indurkha N. *Predictive data mining: a practical guide*. SF, Morgan Kaufmann Publ., 1998, 228 p.
3. Bird S., Klein E., Loper E. *Natural language processing with Python*. Sebastopol, O'Reilly Media Publ., 2015, 479 p.
4. Islam M.Z., Rahm A., Mehler R. Text readability classification of textbooks of a low-resource language. *Proc. 26th Pacific Asia Conf. Language, Information, and Computation*. 2012, pp. 545–553.
5. Manning D., Schütze H. *Foundations of Statistical Natural Language Processing*. Cambridge, MIT Press, 2000, 680 p.
6. Dumais S., Chen H. Hierarchical classification of web content. *Proc. 23rd Annual Int. ACM SIGIR Conf. Research and Development in Information Retrieval*. 2002, pp. 256–263.
7. Aggarwal C. *Data mining: the textbook*. NY, Springer Publ., 2015, 734 p.
8. Medlock B.W. *Investigating classification for natural language processing tasks*. Univ. Cambridge Publ., 2008, 138 p.
9. *Internet papers base "Base"*. Available at: <https://www.base-search.net> (accessed May 7, 2017).
10. Weiss M.S., Apte C., Damerau F.J., Johnson D.E., Oles F.J., Goetz T., Hampp T. Maximizing Text-mining performance. *Jour. Intelligent Information Retrieval*. 1999, vol. 14, pp. 63–69.
11. Hellwig O. *Improving the morphological analysis of classical Sanskrit*. Düsseldorf Univ. Available at: <http://aclweb.org/anthology/W/W16/W16-3715.pdf> (accessed May 7, 2017).
12. Caropreso M.F., Matwin S., Sebastiani F. Learner-independent evaluation of the usefulness of statistical phrases for automated text categorization: in Publ. *Text databases and document management: theory and practice*. Virginia Commonwealth Univ. Publ., 2001, pp. 78–102.
13. *Official website of the program automatic text processing "AoT"*. Chart. Russian morphological dictionary. Available at: <http://aot.ru/index.html> (accessed May 7, 2017).
14. *Solarix Engine*. Chart. Computer Russian grammar. Available at: <http://www.solarix.ru/index-ru.shtml> (accessed May 7, 2017).
15. Fomin V.V., Osochkin A.A. Text classification based on frequency and morphological analysis using data-mining algorithms. *Informatizatsiya obrazovaniya i nauki* [Informatization of Education and Science]. 2016, iss. 3, pp. 137–152 (in Russ.).
16. Fomin V.V., Fomina I.K., Osochkin A.A. Texts classification based on a frequency and morphological analysis with date-mining algorithms. *Aktualnye voprosy i perspektivy razvitiya matematicheskikh i estestvennykh nauk* [Actual Issues and Prospects for the Development of Mathematical and Natural Sciences]. 2016, no. 3, pp. 64–69 (in Russ.).



УДК 519.688:004.93

DOI: 10.15827/0236-235X.119.487-495

Дата подачи статьи: 15.06.17

2017. Т. 30. № 3. С. 487–495

## **АЛГОРИТМЫ КОНЦЕПТУАЛЬНОГО МОДЕЛИРОВАНИЯ И КЛАССИФИКАЦИИ ТЕКСТОВ В КОРПУСЕ ТУВИНСКОГО ЯЗЫКА**

*В.В. Быкова, д.ф.-м.н., профессор, bykvalen@mail.ru*

*(Сибирский федеральный университет,*

*Свободный просп., 79, г. Красноярск, 660041, Россия);*

*Ч.М. Монгуш, аспирант, преподаватель, mongushchod91@yandex.ru*

*(Сибирский федеральный университет, Свободный просп., 79, г. Красноярск, 660041, Россия;*

*Тувинский государственный университет, ул. Ленина, 36, г. Кызыл, 667000, Россия)*

Корпус языка – это информационно-лингвистическая система, основанная на собрании оцифрованных текстов некоторого языка. На сегодняшний день корпус тувинского языка включает официально-деловые документы и произведения тувинской художественной литературы.

Работы по расширению корпуса тувинского языка и углублению уровня обработки текстов продолжаются. Они приводят к необходимости решения задач анализа естественно-языковых текстов. Основными из этих задач являются классификация по прецедентам и концептуальное моделирование.

Для их решения в статье используется алгебраический подход, называемый анализом формальных понятий. Предлагаются алгоритмы и программы для построения концептуальной модели коллекции литературных произведений и решения задачи бинарной классификации по прецедентам. Указаны приемы снижения вычислительной сложности рассматриваемых алгоритмов.

В работе представлены результаты вычислительных экспериментов, подтверждающие результативность предложенных приемов по снижению сложности вычислений. Приведены результаты концептуального моделирования и бинарной классификации произведений тувинского фольклора.

**Ключевые слова:** корпус языка, анализ формальных понятий, концептуальные модели текстов, алгоритм классификации, алгоритмы снижения размерности контекстов.

В настоящее время активно создаются корпуса языков народов Российской Федерации для сохранения национального литературного наследия и проведения научных исследований по изучению этих языков. Работа над созданием Национального корпуса тувинского языка ведется сотрудниками, аспирантами и студентами Тувинского государственного и Сибирского федерального университетов [1]. Под корпусом понимается информационно-лингвистическая система, основанная на собрании оцифрованных текстов. Корпус включает в себя различные типы текстов, представленных в языке, а также разметку – информацию о свойствах текстов. В рамках корпусов решаются многие задачи анализа естественно-языковых текстов, возникающие в филологических и лингвистических исследованиях [2]. Основными из них являются классификация по прецедентам и концептуальное моделирование. Классификация по прецедентам, как правило, направлена на установление жанра и автора текста, определение пространственно-временного периода написания произведения. Цель концептуального моделирования – структурное представление знаний, извлеченных из текстов произведений (например, особенности использования языковых клише и диалектных вариантов эпических выражений).

Традиционными подходами для решения задачи классификации являются методы машинного обучения и математической статистики, основанные на количественных мерах близости рассматриваемых текстов [3]. В этом случае, как правило, се-

мантическая составляющая анализируемых текстов не рассматривается. Для построения концептуальных моделей текстов обычно используются концептуальные графы и решетки понятий [3]. Однако существует формализованный подход, известный в литературе как анализ формальных понятий, который позволяет выполнять классификацию, концептуальное моделирование и решать многие другие задачи анализа текстов на основе алгебраической теории решеток [4, 5]. В рамках этого подхода тексты анализируются на семантическом уровне, моделируются и представляются бинарными контекстами, отражающими наличие или отсутствие признаков, характерных для исследуемых произведений. Использование методов анализа формальных понятий позволяет не включать в ПО корпусов пакеты программ, ориентированные на решение отдельных задач анализа данных.

*Анализ формальных понятий* (АФП) является прикладной ветвью алгебраической теории решеток, в рамках которой возможна формализация терминов «понятие» и «иерархия понятий».

Основные идеи АФП были сформулированы в работе Р. Вилле и Б. Гантера [6] и развиты в исследованиях С.О. Кузнецова, Д.И. Игнатов, С.И. Гурова [5, 7]. В АФП формальные понятия определяются с помощью соответствий Галуа и представляют собой пары множеств вида (объем, содержание). Основным достоинством такого определения является абсолютное соответствие традиционной трактовке термина «понятие», используемого в гуманитарных науках [8].

В статье предлагаются алгоритмы и программы для построения концептуальной модели и решения задачи бинарной классификации по прецедентам с помощью математического аппарата АФП. Указаны приемы снижения вычислительной сложности рассматриваемых алгоритмов. Представлены результаты вычислительных экспериментов, подтверждающие результативность предложенных приемов по снижению сложности вычислений. Приведены результаты концептуального моделирования и бинарной классификации по прецедентам произведений тувинского фольклора.

### Основные термины и обозначения

Приведем термины и обозначения, применяемые в АФП [6].

Пусть для некоторой предметной области определены два непустых конечных множества  $G$  и  $M$  объектов и признаков соответственно (от немецких слов *Gegenstände* – объект, *Merkmale* – признак). Пусть также задано отношение инцидентности между множествами  $G$  и  $M$ :  $I \subseteq G \times M$ . Существование в  $I$  пары  $(g, m)$ ,  $g \in G$  и  $m \in M$ , означает, что объект  $g$  имеет признак  $m$  и, наоборот, признак  $m$  присущ объекту  $g$ . Тройка  $K = (G, M, I)$  называется формальным контекстом (или кратко контекстом) для рассматриваемой предметной области. Если множества  $G$  и  $M$  линейно упорядочены (например лексикографически), то контекст можно однозначно (с точностью до «материальной» природы объектов и признаков) задать бинарной матрицей  $T$ , отражающей отношение инцидентности  $I$ .

Выберем два произвольных элемента  $g \in G$  и  $m \in M$ . Определим для них отображения  $\varphi$  и  $\psi$ :

$\varphi(g) = \{m \in M \mid (g, m) \in I\}$  – множество признаков, присущих объекту  $g \in G$ ;

$\psi(m) = \{g \in G \mid (g, m) \in I\}$  – множество объектов, обладающих признаком  $m \in M$ .

Отображения  $\varphi$  и  $\psi$  можно обобщить на произвольные  $A \subseteq G$  и  $B \subseteq M$  следующим образом:

$$\varphi(A) = \{m \in M \mid \forall g \in A (g, m) \in I\},$$

$$\psi(B) = \{g \in G \mid \forall m \in B (g, m) \in I\}.$$

Здесь  $\varphi(A)$  – множество признаков, общих для всех объектов из  $A$ , а  $\psi(B)$  – множество объектов, обладающих всеми признаками из  $B$ . При этом считается, что  $\varphi(\emptyset) = M$  и  $\psi(\emptyset) = G$ , то есть пустому множеству объектов присущи все признаки из  $M$ , и каждый объект рассматриваемого контекста обладает пустым множеством признаков. Отображения  $\varphi$  и  $\psi$  определены так, что для любых  $A_1, A_2 \subseteq G$  и  $B_1, B_2 \subseteq M$  верны равенства:

$$\varphi(A_1 \cup A_2) = \varphi(A_1) \cap \varphi(A_2),$$

$$\psi(B_1 \cup B_2) = \psi(B_1) \cap \psi(B_2).$$

Обычно в анализе формальных понятий для отображений  $\varphi$  и  $\psi$  применяется единое обозначение  $(\cdot)'$ , а приведенные выше формулы для  $\varphi(A)$ ,  $\psi(B)$  записываются так:

$$A' = \bigcap_{g \in A} g' = \{m \in M \mid \forall g \in A (g, m) \in I\}, \quad (1)$$

$$B' = \bigcap_{m \in B} m' = \{g \in G \mid \forall m \in B (g, m) \in I\}. \quad (2)$$

Если  $g \in G$  и  $m \in M$ , то обозначения  $g'$  и  $m'$  служат сокращенной формой записи множеств  $\varphi(g) = \{g'\}$  и  $\psi(m) = \{m'\}$  соответственно. Отображения «'» удовлетворяют свойствам, вытекающим из их определения и вполне реалистичного и постулируемого в анализе данных положения: расширение (сокращение) множества признаков уменьшает (увеличивает) число объектов, обладающих этими признаками. Формально эти свойства можно выразить в виде следующих утверждений.

**Утверждение 1.** Для всякого контекста  $K = (G, M, I)$  и любых  $B_1, B_2 \subseteq M$  верны свойства:

– антимонотонность:

если  $B_1 \subseteq B_2$ , то  $(B_2)' \subseteq (B_1)'$ ;

– экстенсивность:

$$B_1 \subseteq (B_1)'', \text{ где } (B_1)'' = ((B_1)')' \subseteq M.$$

**Утверждение 2.** Для всякого контекста  $K = (G, M, I)$  и любых  $A_1, A_2 \subseteq G$  верны свойства:

– антимонотонность:

если  $A_1 \subseteq A_2$ , то  $(A_2)' \subseteq (A_1)'$ ;

– экстенсивность:

$$A_1 \subseteq (A_1)'', \text{ где } (A_1)'' = ((A_1)')' \subseteq G.$$

Множество  $(B_1)'' = \varphi(\psi(B_1))$  можно трактовать как набор признаков, которые всегда появляются в объектах контекста  $K = (G, M, I)$  вместе с признаками из  $B_1$ , причем это множество является наибольшим по включению в пределах этого контекста. Множество  $(A_1)'' = \psi(\varphi(A_1))$  можно интерпретировать как наибольшее по включению множество объектов, которые обладают всеми признаками, характерными для объектов  $A_1$ . Согласно утверждениям 1 и 2, отображения  $\varphi$  и  $\psi$  составляют пару соответствий Галуа между множествами  $2^G$  и  $2^M$ , частично упорядоченными по включению [9]. Здесь традиционно  $2^G$  и  $2^M$  – совокупность всех подмножеств рассматриваемых множеств  $G$  и  $M$  соответственно. Двойное применение отображения «'» определяет оператор замыкания «''» на  $2^G$  или  $2^M$  в алгебраическом смысле [9].

Множество признаков  $B \subseteq M$ , для которого  $B = B''$ , называется замкнутым в контексте  $K = (G, M, I)$ . Принято говорить, что множество  $B''$  является замыканием для  $B \subseteq M$  в контексте  $K = (G, M, I)$ . Исходя из (1) и (2), при  $B' \neq \emptyset$  замыкание для  $B \subseteq M$  можно вычислить по формуле

$$B'' = \bigcap_{g \in G} \{g' \mid B \subseteq g'\}. \quad (3)$$

Если  $B' = \emptyset$ , то всегда  $B'' = \varphi(\psi(B)) = \varphi(\emptyset) = M$ . Важно отметить, что применение формулы (3) позволяет за один просмотр контекста  $K = (G, M, I)$  найти замыкание для заданного множества признаков.

### Концептуальное моделирование и АФП

Концептуальное моделирование направлено на структурное представление знаний об исследуемой

предметной области. Концептуальная модель как семантическая структурная модель предметной области представляет группы однородных объектов, связанных между собой отношениями. Однородность объектов в данном случае понимается как наличие у них одних и тех же свойств. Эти группы однородных объектов традиционно называют сущностями, концептами или понятиями. Объекты, образующие понятия, могут иметь различную природу. В концептуальных моделях используются не только бинарные отношения, но и отношения произвольной ариности. Такая универсальность позволяет применять концептуальное моделирование в самых разных областях, в том числе в анализе текстов. В общем случае описание концептуальной модели содержит перечень взаимосвязанных понятий предметной области вместе с их свойствами и характеристиками.

Для описания концептуальных моделей используются различные средства, но в основном графические (диаграммы «сущность–связь», концептуальные графы, решетки понятий). В системах обработки текстовых данных применяются преимущественно концептуальные графы и решетки [3]. Анализ формальных понятий позволяет не только представить (описать и визуализировать) концептуальную модель, но и построить ее, исходя из формальных контекстов. Формальный контекст является абстрактной моделью исследуемой предметной области, отражающей отношение инцидентности между объектами и их свойствами. Имея в наличии формальный контекст, с помощью математических методов АФП можно выявить основные понятия предметной области, установить между ними отношение частичного порядка и объединить формальные понятия в решетку. Такая решетка понятий – математическое описание концептуальной модели, допускающее ее исследование математическими методами.

Дадим определение решетки формальных понятий. Пусть для некоторой предметной области задан формальный контекст  $K = (G, M, I)$ , где  $G$  – множество объектов;  $M$  – множество признаков;  $I$  – отношение инцидентности между множествами  $G$  и  $M$ . Пара  $(A, B)$ ,  $A \subseteq G$ ,  $B \subseteq M$ , такая, что  $A' = B$  и  $B' = A$ , называется формальным понятием с объемом  $A$  и содержанием  $B$ . Другими словами, пара множеств  $(A, B)$  является формальным понятием для контекста  $K = (G, M, I)$  тогда и только тогда, когда  $A = A''$  и  $B = B''$ , то есть когда  $A, B$  – замкнутые множества относительно оператора «'». Если контекст  $K = (G, M, I)$  представлен матрицей  $T$ , то формальному понятию  $(A, B)$  соответствует ее максимальная подматрица, заполненная единицами. Строки этой подматрицы отвечают элементам из  $A$ , а столбцы – элементам из  $B$ .

Пусть  $FC_K$  – множество всех формальных понятий контекста  $K = (G, M, I)$ . Введем на  $FC_K$  отношение частичного порядка  $\sqsubseteq$  следующим образом:

$(A_1, B_1) \sqsubseteq (A_2, B_2)$ , если  $A_1 \subseteq A_2$  (или  $B_2 \subseteq B_1$ ), (4) где  $A_1, A_2 \subseteq G$  и  $B_1, B_2 \subseteq M$ . Заметим, что в высказывании (4) достаточно указать лишь одно из двух включений  $A_1 \subseteq A_2$  или  $B_2 \subseteq B_1$ , поскольку в силу антимонотонности отображений «'» из одного из них всегда следует другое. Согласно (4), если  $(A_1, B_1) \sqsubseteq (A_2, B_2)$ , формальное понятие  $(A_2, B_2)$  можно считать более общим, чем понятие  $(A_1, B_1)$ , поскольку оно имеет меньший набор характерных признаков, а значит, большее число объектов, обладающих этими признаками.

Определим на  $FC_K$  операции пересечения  $\sqcap$  и объединения  $\sqcup$  через одноименные теоретико-множественные операции  $\cap$  и  $\cup$  следующим образом:

$$(A_1, B_1) \sqcap (A_2, B_2) = (A_1 \cap A_2, (A_1 \cap A_2)'), \quad (5)$$

$$(A_1, B_1) \sqcup (A_2, B_2) = ((B_1 \cap B_2)', B_1 \cap B_2). \quad (6)$$

Тогда частично упорядоченное множество  $(FC_K, \sqsubseteq)$  образует решетку  $L_K = (FC_K, \sqcap, \sqcup)$ . Операции  $\sqcap$  и  $\sqcup$ , установленные соотношениями (5) и (6), удовлетворяют всем необходимым для решеток законам ассоциативности, коммутативности, идемпотентности и поглощения [9]. Эта решетка называется решеткой формальных понятий контекста  $K = (G, M, I)$ . Известно, что  $L_K$  является полной решеткой [6]. Нулем решетки  $L_K$  является формальное понятие  $(M', M)$ , содержащее все признаки контекста  $K = (G, M, I)$ , а единицей – формальное понятие  $(G, G')$ , в котором объем – множество всех объектов рассматриваемого контекста.

Решетка  $L_K$  связывает все элементы частично упорядоченного множества  $FC_K$  в определенную иерархическую структуру. Чем выше уровень расположения формального понятия в  $L_K$ , тем более общим по отношению к формальным понятиям, находящимся ниже в  $L_K$ , оно является. Таким образом, решетка  $L_K$  – это формализованное представление множества формальных понятий и связей между ними в смысле отношения (4). Причем каждое формальное понятие этой решетки определяет множество однородных объектов исследуемой предметной области со своим специфичным набором признаков. На основе решетки формальных понятий и методов АФП можно решать задачу бинарной классификации по прецедентам.

### Задача бинарной классификации по прецедентам и алгоритм ее решения

Известны различные формулировки задачи классификации [3, 7]. Задача бинарной классификации по прецедентам традиционно формулируется следующим образом. Пусть задано конечное множество объектов  $G$ , разделенное на два класса  $G^+$  и  $G^-$ ,  $G^+ \cap G^- = \emptyset$ ,  $G^+ \cup G^- = G$ . Такое разбиение определено с помощью некоторой обучающей выборки и целевого бинарного признака  $z$ . Элементы множеств  $G^+$  и  $G^-$  называют положительными и отрицательными прецедентами соответственно. Все

объекты из  $G$  описаны через конечное множество признаков  $M$ , которое задается  $(0, 1)$ -матрицей  $T$ , кодирующей наличие или отсутствие признака  $m \in M$  для объекта  $g \in G$ . Пусть задан некоторый объект  $x \notin G$ . Считается, что он обладает множеством признаков  $M_x \subseteq M$ . Требуется найти решающее правило, которое для объекта  $x$  определяет класс принадлежности. Решающее правило должно приводить к отказу от классификации, когда принадлежность объекта  $x$  к тому или иному классу не может быть однозначно определена.

Для описания данной задачи в терминах АФП достаточно лишь уточнить вид представления классов  $G^+$  и  $G^-$ . С этой целью сопоставим классу  $G^+$  положительный контекст  $K^+ = (G^+, M, I^+)$ , а классу  $G^-$  – отрицательный контекст  $K^- = (G^-, M, I^-)$ . Существование в  $I^+$  пары  $(g, m)$  означает, что объект  $g \in G^+$  имеет признак  $m \in M$ . Аналогично принадлежность пары  $(g, m)$  к  $I^-$  говорит о том, что объекту  $g \in G^-$  присущ признак  $m \in M$ . Таким образом, бинарная матрица  $T$  разбивается на две подматрицы, соответствующие отношениям инцидентности  $I^+$  и  $I^-$ .

Существуют различные алгоритмы классификации на основе АФП. К ним относятся алгоритмы Rulearner, GALOIS, GRAND, CITREC, CLNN & CLNB и LEGAL, использующие всю решетку понятий или ее некоторое подмножество [10], и алгоритмы, основанные на гипотезах [7]. В данной статье задача бинарной классификации по прецедентам решается с помощью гипотез.

Гипотезой называется некоторый набор признаков, который присутствует в описании объектов одного класса и не присутствует в описании объектов другого класса. Гипотезы извлекаются из решеток формальных понятий  $L_{K^+}$  и  $L_{K^-}$ , построенных для контекстов  $K^+$  и  $K^-$  соответственно. Содержание  $B^+$  формального понятия  $(A^+, B^+) \in L_{K^+}$  называется положительной гипотезой, если не существует такого формального понятия  $(A^-, B^-) \in L_{K^-}$ , что  $B^+ \subseteq B^-$ . В противном случае множество признаков  $B^+$  называется фальсифицированной положительной гипотезой. Аналогичным образом определяются отрицательные гипотезы и фальсифицированные отрицательные гипотезы: содержание  $B^-$  формального понятия  $(A^-, B^-) \in L_{K^-}$  считается отрицательной гипотезой, если не существует такого формального понятия  $(A^+, B^+) \in L_{K^+}$ , что  $B^- \subseteq B^+$ , иначе  $B^-$  является фальсифицированной отрицательной гипотезой.

Решающее правило бинарной классификации по прецедентам для объекта  $x$  можно сформулировать следующим образом [7]:

– объект  $x$  относится к классу  $G^+$ , если множество  $M_x$  включает хотя бы одну положительную гипотезу и не включает ни одной отрицательной гипотезы; в противном случае объект  $x$  относится к классу  $G^-$ ;

– отказ от классификации происходит, если  $M_x$  не включает в качестве подмножеств ни положительные, ни отрицательные гипотезы, или если  $M_x$  включает как положительные, так и отрицательные гипотезы.

Процесс решения задачи бинарной классификации на основе гипотез состоит из пяти этапов:

- 1-й этап – предобработка исходных контекстов;
- 2-й этап – нахождение формальных понятий в  $K^+$  и  $K^-$ ;
- 3-й этап – построение решеток  $L_{K^+}$  и  $L_{K^-}$ ;
- 4-й этап – выявление гипотез;
- 5-й этап – применение решающего правила бинарной классификации для объекта  $x \notin G$ .

На первом этапе производится предобработка исходных контекстов с целью уменьшения их размеров. Предобработка выполняется так, чтобы не изменились число и состав формальных понятий в  $L_{K^+}$  и  $L_{K^-}$ . Сокращение может затрагивать как множество объектов, так и множество признаков. Возможные случаи и алгоритмы их обработки рассмотрим применительно к положительному контексту и соответствующей ему матрице инцидентности.

#### Случай 1 (дубликаты строк).

Пусть в  $K^+ = (G^+, M, I^+)$  существует множество объектов  $A = \{g_1, g_2\}$ , таких, что  $g_1' = g_2' = B$ . Тогда  $A'' = (g_1' \cap g_2')' = (B \cap B)' = (B)' = A$ , то есть  $A$  является замкнутым множеством. Следовательно, объект  $g_2$  можно удалить из  $K^+$  и не учитывать при вычислении положительных формальных понятий. При построении решетки  $L_{K^+}$  объект  $g_2$  необходимо добавить в объемы тех формальных понятий, в которые вошел объект  $g_1$ .

#### Случай 2 (нулевые строки и столбцы).

Если в  $K^+ = (G^+, M, I^+)$  существует такой объект  $g$ , что  $g' = \emptyset$ , то  $g'' = (g')' = (\emptyset)' = G^+$ . Аналогично, если в  $K^+ = (G^+, M, I^+)$  имеется признак  $m \in M$ , такой, что  $m' = \emptyset$ , то  $m'' = (m')' = (\emptyset)' = M$ . Поэтому на момент вычисления положительных формальных понятий объект  $g$  и признак  $m$  следует отбросить, а затем при построении  $L_{K^+}$  объект  $g$  добавить в единицу, а признак  $m$  – в ноль этой решетки.

#### Случай 3 (единичные строки и столбцы).

Если в контексте  $K^+ = (G^+, M, I^+)$  существует такой объект  $g \in G^+$ , что  $g' = M$ , то  $g'' = (g')' = (M)' = g$ . Поэтому объект  $g$  надо опустить при нахождении формальных понятий, но затем добавить в решетку  $L_{K^+}$  новое формальное понятие  $(g, M)$ , а объемы всех ранее полученных положительных формальных понятий пополнить объектом  $g$ . Аналогично, если имеется такой признак  $m$ , что  $m' = G^+$ ,  $m$  вначале необходимо опустить и потом добавить в содержание всех формальных понятий решетки  $L_{K^+}$ .

На втором и третьем этапах выявляются формальные понятия в исходных контекстах  $K^+$  и  $K^-$ , прошедших предобработку. Простейшим способом осуществления этих действий является пере-

бор всех различных подмножеств множества признаков (их число, как правило, значительно меньше числа объектов) с вычислением для каждого из них замыкания по формуле (3). Затем на основе (4)–(6) строятся решетки  $L_{K^+}$  и  $L_{K^-}$  для контекстов  $K^+$  и  $K^-$  соответственно.

На четвертом и пятом этапах выявляются положительные гипотезы, фальсифицированные положительные гипотезы, отрицательные гипотезы и фальсифицированные отрицательные гипотезы путем проверки отношения включения содержаний соответствующих формальных понятий. После этого в соответствии с приведенным выше решающим правилом классификации принимается решение о том, чтобы или отнести объект  $x$  к  $G^+$  или к  $G^-$ , или констатировать отказ от классификации. Следует отметить, что на четвертом и пятом этапах процесса решения задачи бинарной классификации может быть использован не только алгоритм классификации на основе гипотез, но и любой другой алгоритм, базирующийся на АФП.

#### Проблема построения решетки формальных понятий и приемы снижения сложности вычислений

Рассмотренные выше задачи концептуального моделирования и бинарной классификации опираются на решетки формальных понятий. Известно, что задача порождения для заданного контекста всех формальных понятий и построения решетки формальных понятий является NP-трудной. Обоснование этого факта дано в [5]. Высокая вычислительная сложность данной задачи объясняется тем, что число формальных понятий может экспоненциально зависеть от размера контекста. Например, это имеет место для контекстов вида  $K = (G, M, \neq)$ . Поэтому время, необходимое на выявление формальных понятий в контексте  $K = (G, M, I)$  и построение решетки, в худшем случае составляет  $O(|FC_K| \cdot |G|^2 \cdot |M|)$ , где  $|FC_K|$  – число формальных понятий. Далее предлагаются два приема снижения вычислительной сложности этого процесса.

**Прием 1:** уменьшение размера величин  $|G|$  и  $|M|$  с помощью алгоритмов обработки случаев 1–3.

Эти случаи описаны выше, там же доказана корректность их применения. Время реализации приема 1 составляет  $O(|G| \cdot |M|)$ .

**Прием 2:** декомпозиция контекста – разделение контекста на полиномиальное число боксов (с последующим поиском формальных понятий в каждом из выделенных боксов).

Введем понятие бокса через объектные и признаковые формальные понятия контекста  $K = (G, M, I)$ . Назовем объектным понятием формальное понятие вида  $(g'', g')$ , где  $g \in G$ , а признаковым понятием – формальное понятие вида  $(m', m'')$ , где  $m \in M$ . Таким образом, каждому объекту из  $G$  соответствует одно объектное понятие, и каждому

признаку из  $M$  – одно признаковое понятие. Следовательно, для контекста  $K = (G, M, I)$  число объектных понятий равно  $|G|$ , а число признаковых понятий составляет  $|M|$ . Заметим, что объектное понятие  $(g'', g')$  имеет самое большое по размеру содержание  $g'$  среди других формальных понятий, имеющих в объеме объект  $g$ , а признаковое понятие  $(m', m'')$  – самый большой объем  $m'$  среди других понятий, имеющих в содержании признак  $m$ . Это следует из антимонотонности соответствий Галуа, указанных в утверждениях 1 и 2.

Обозначим через  $O_K = \{(g'', g') \mid \forall g \in G\} \subseteq FC_K$  множество всех объектных понятий и через  $S_K = \{(m', m'') \mid \forall m \in M\} \subseteq FC_K$  множество всех признаковых понятий контекста  $K = (G, M, I)$ . Заметим, что множества  $O_K$  и  $S_K$  могут иметь непустое пересечение. Выберем два формальных понятия  $(g'', g') \in O_K$  и  $(m', m'') \in S_K$ . Если для них верно отношение порядка  $(g'', g') \sqsubseteq (m', m'')$  или, то же самое, выполняются условия

$$g'' \subseteq m' \text{ и } m'' \subseteq g', \quad (7)$$

то пару  $(m', g')$  назовем боксом контекста  $K = (G, M, I)$ , образованным элементами  $g \in G$  и  $m \in M$ . Пусть формальное понятие  $(A, B) \in FC_K$  вложено в бокс  $(m', g')$  контекста  $K = (G, M, I)$ , если  $A \subseteq m'$  и  $B \subseteq g'$ . Всякий бокс  $(m', g')$  не является пустым, поскольку согласно (7) в него всегда вложены формальные понятия  $(g'', g') \in O_K$  и  $(m', m'') \in S_K$ .

Рассмотрим некоторый бокс  $(m', g')$ , образованный элементами  $g \in G$  и  $m \in M$  контекста  $K = (G, M, I)$ . Очевидно, что данный бокс определяет некоторую подматрицу матрицы  $T$  и образует подконтекст  $(G, M, C)$  контекста  $K = (G, M, I)$ , где  $C \subseteq I$ . При этом  $(x, y) \in C$ , если и только если  $x \in m'$  и  $y \in g'$ . Соответствие между боксами и формальными понятиями контекста  $K = (G, M, I)$  устанавливает утверждение 3 [11], подтверждающее корректность приема 2.

**Утверждение 3.** Для всякого контекста  $K = (G, M, I)$  и любой пары множеств  $(A, B)$ , где  $\emptyset \neq A \subseteq G$ ,  $\emptyset \neq B \subseteq M$ , справедливы высказывания:

а) если  $(A, B)$  – формальное понятие контекста  $K = (G, M, I)$ , то всегда в этом контексте существует бокс  $(m', g')$ , образованный элементами  $g \in G$  и  $m \in M$ , причем, возможно, не единственный, в который это формальное понятие вкладывается;

б) если  $(A, B)$  – формальное понятие подконтекста  $(G, M, C)$ , соответствующего некоторому боксу  $(m', g')$  контекста  $K = (G, M, I)$ , то оно также является формальным понятием контекста  $K = (G, M, I)$ .

Если в контексте  $K = (G, M, I)$  имеются формальные понятия  $(G, \emptyset)$  и  $(\emptyset, M)$ , то для них невозможно установить признаковые и объектные понятия, поэтому они не вкладываются ни в один из боксов данного контекста. Их наличие необходимо просто учитывать при построении решеток.

Однократное формирование боксов для контекста  $K = (G, M, I)$  включает в себя следующие действия: нахождение всех объектных и признаковых понятий; проверка условия (7) для каждой пары таких формальных понятий и формирование боксов. Число анализируемых пар, проверок и полученных боксов всегда не более чем  $|I| = |G| \cdot |M|$ . Поэтому время формирования боксов составляет  $O(|I| \cdot (|G| + |M|))$ . В худшем случае может быть найден только один бокс, совпадающий с исходным контекстом, и тогда декомпозиция контекста на боксы не дает эффекта. Это возможно, например, для контекста, полностью заполненного единицами. Однако реальные контексты, как правило, разлагаются на разумное число боксов. Важно отметить, что процесс разбиения контекста на боксы может быть организован итерационно, ведь каждый выявленный бокс может быть вновь разбит на боксы. Но если данный процесс продолжать до тех пор, пока все боксы выродятся в формальные понятия, это может привести к экспоненциальному числу боксов, а значит, и к экспоненциальному времени их построения. Для получения полиномиального числа боксов рекомендуется ограничиваться константным числом итераций.

#### Описание программы и результаты вычислительных экспериментов

Рассмотренные выше алгоритмы решения задач концептуального моделирования и бинарной классификации по прецедентам, а также приемы снижения вычислительной сложности этих алгоритмов реализованы в программе FCACorpus (язык программирования Delphi). Функция визуализации решеток формальных понятий не была включена в программу FCACorpus, так как ее применение целесообразно только для контекстов сравнительно небольших размеров. Визуализировать решетку можно всегда с помощью специальных программных средств [12].

Для оценки результативности приемов снижения сложности вычислений, реализованных в программе FCACorpus, были выполнены вычислительные эксперименты. Использовались контексты с числом объектов 15, 18, 20 и числом признаков 15. Эти контексты были сформированы на основе паспортов фольклорных произведений, взятых из Национального корпуса тувинского языка. Для каждого контекста  $K = (G, M, I)$  осуществлялся поиск множества  $FC_K$  всех формальных понятий без разбиения и с однократным разбиением этого контекста на боксы. Результаты вычислительных экспериментов приведены в таблице 1, где  $|G|$  – количество объектов контекста;  $|FC_K|$  – число найденных формальных понятий;  $N$  – количество образованных боксов;  $t$  – время выполнения программы. Вычислительные эксперименты выполнялись на компьютере с процессором Intel® Core™ i7-720QM

Processor (6M Cache, 1.60 GHz) и ОЗУ размером 4 ГБ.

Таблица 1

#### Результаты экспериментов

Table 1

#### The experimental results

Вычисление всех формальных понятий контекста	$ G $	$ FC_K $	$N$	$t$ , мс
Без разбиения на боксы	15	36	–	480
С разбиением на боксы		36	12	66
Без разбиения на боксы	18	73	–	12480
С разбиением на боксы		73	23	120
Без разбиения на боксы	20	98	–	30519
С разбиением на боксы		98	40	150

Как видно из таблицы 1, количество и состав полученных формальных понятий в обоих случаях (без разбиения на боксы, с разбиением на боксы) полностью совпадают. Однако применение боксов дает значительный выигрыш во времени – время выполнения программы в этом случае уменьшается в 10–20 раз. Эксперименты на случайно сгенерированных контекстах различной размерности, показали, что, чем больше объектов и признаков содержит анализируемый контекст, тем больше выигрыш во времени.

#### Результаты концептуального моделирования и классификации произведений тувинского фольклора

Рассмотрим пример применения АФП и разработанных программ для филологических исследований, направленных на концептуальное моделирование произведений тувинского фольклора и определение их принадлежности к жанру героического эпоса. Для решения этих задач необходимо сформировать соответствующие контексты на основе паспортов произведений. Паспорт произведения – это набор признаков, характеризующих семантические, синтаксические и морфологические особенности этого произведения. Например, паспорт произведения тувинского фольклора содержит информацию о сказителе, сведения о про странственно-временном периоде написания, жанровые и сюжетные особенности произведения. Всего выделено 14 признаков, приведенных в таблице 2.

В таблице 3 представлен бинарный контекст  $K^+ = (G^+, M, I^+)$  для четырех фольклорных произведений, где  $G^+ = \{\text{«Арзылаң-Кара аъттыг Хунан-Кара»}, \text{«Мөрүн-Хүлүк»}, \text{«Өлээдей-Мерген»}, \text{«Элестей ашак»}\}$ ;  $M = \{s_1, s_2, s_3, s_4, a_1, a_2, q_1, q_2, c_1, c_2, c_3, t_1, t_2, t_3\}$ ;  $I^+$  – отношение инцидентности между  $G^+$  и  $M$ .

Известно, что все произведения из  $G^+$  относятся к жанру тувинского героического эпоса [13]. В таблице 3 названия произведений заменены их поряд-

ковыми номерами. Единичный (нулевой или пустой) элемент этой таблицы указывает на то, что соответствующее литературное произведение обладает (не обладает) тем или иным признаком.

Таблица 2

## Паспорт произведения тувинского фольклора

Table 2

## Passport of a Tuvan folklore work

Идентификатор признака	Значение признака
$s_1$	Сказитель Кашкак
$s_2$	Сказитель Хертек
$s_3$	Сказитель Ооржак
$s_4$	Другой сказитель или народ
$a_1$	Горный ареал
$a_2$	Степной ареал
$q_1$	Есть богатырь
$q_2$	Нет богатыря
$c_1$	Сюжет «Сватовство»
$c_2$	Сюжет «Сестра добывает брату суженую»
$c_3$	Другой сюжет
$t_1$	Зачин «Эрте шагның экинде, бурун шагның мурнунда»
$t_2$	Зачин «Шыянам, эрте бурунгу шагда»
$t_3$	Зачин «Шыянам, эртенгиниң эртезинде бурунгуң мурнунда»

Таблица 3

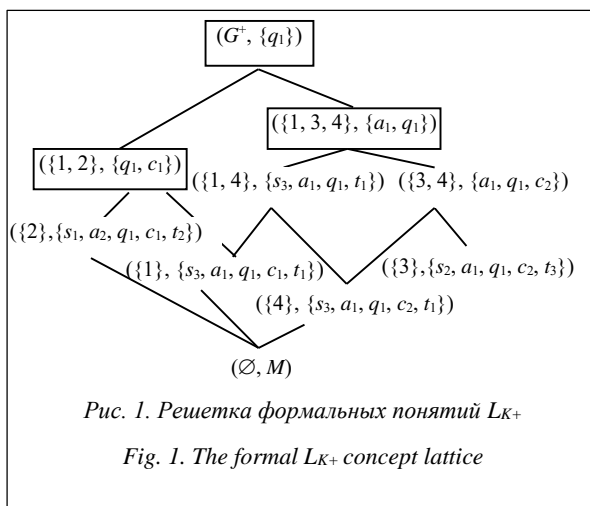
Контекст  $K^+$  произведений тувинского героического эпоса

Table 3

The context  $K^+$  of Tuvan heroic epic works

№	$s_1$	$s_2$	$s_3$	$s_4$	$a_1$	$a_2$	$q_1$	$q_2$	$c_1$	$c_2$	$c_3$	$t_1$	$t_2$	$t_3$
1			1		1		1		1			1		
2	1					1	1		1				1	
3		1			1		1			1				1
4			1		1		1			1		1		

Заметим, что контекст  $K^+$  допускает преобразование согласно описанным выше случаям 2 и 3. Всего контекст  $K^+$  порождает 10 формальных понятий, которые образуют решетку  $L_{K^+}$  (рис. 1). Еди-

Рис. 1. Решетка формальных понятий  $L_{K^+}$ Fig. 1. The formal  $L_{K^+}$  concept lattice

ницей этой решетки является формальное понятие  $(G^+, \{q_1\})$ , а нулем  $(\emptyset, M)$ .

Решетка  $L_{K^+}$  – концептуальная модель для множества произведений тувинского героического эпоса  $G^+$ , построенная и представленная в терминах АФП. Выявленные формальные понятия и семантические связи между ними позволяют сделать следующие выводы.

1. Для произведений, относящихся к жанру тувинского героического эпоса, характерно прежде всего наличие богатыря, так как единицей решетки  $L_{K^+}$  является формальное понятие  $(G^+, \{q_1\})$ . Это понятие – самое общее по отношению ко всем другим формальным понятиям этой решетки. Ведь по определению решетки, чем выше уровень расположения формального понятия в  $L_{K^+}$ , тем более общим по отношению к формальным понятиям, находящимся ниже в  $L_{K^+}$ , оно является.

2. Произведениям тувинского героического эпоса присущи признаки Горный ареал или Сюжет «Сватовство». Эти признаки входят в содержание формальных понятий  $(\{1, 3, 4\}, \{a_1, q_1\})$ ,  $(\{1, 2\}, \{q_1, c_1\})$ , расположенных в решетке  $L_{K^+}$  уровнем ниже, чем формальное понятие  $(G^+, \{q_1\})$ , и выше всех других понятий.

3. Согласно формальному понятию  $(\{1, 4\}, \{s_3, a_1, q_1, t_1\})$ , для произведений героического эпоса, сказителем которых является Ооржак, специфичным является зачин «Эрте шагның экинде, бурун шагның мурнунда».

Экспертами установлено, что указанные выводы соответствуют действительности, то есть являются филологически правильными. Каждый из указанных выводов – определенные знания о произведениях тувинского героического эпоса, представленных в  $K^+$ . Очевидно, что увеличение числа произведений в  $K^+$  углубляет эти знания.

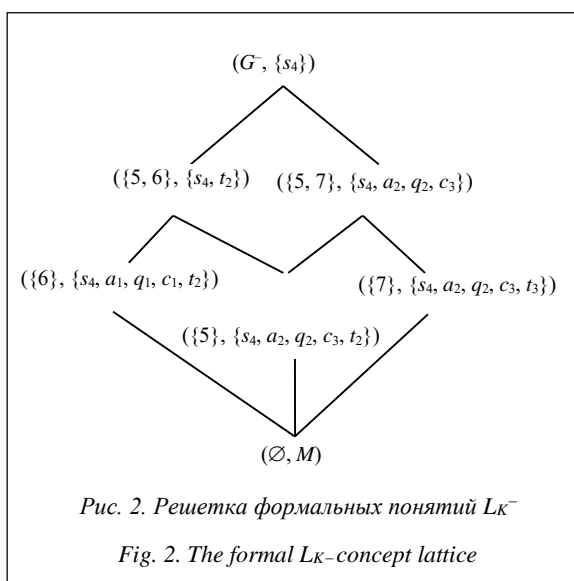
Рассмотрим теперь задачу бинарной классификации по прецедентам. Для этого сформируем отрицательный контекст  $K^- = (G^-, M, I^-)$ , состоящий из трех литературных произведений, которые не относятся к жанру тувинского героического эпоса (табл. 4). Здесь  $G^- = \{\text{«Чечен-Маанай и Тенек-Тулун», «Караты-Хаан биле Алдын-кыс», «Кыс-Халыыр»}\}$  или с использованием порядковых номеров произведений  $G^- = \{5, 6, 7\}$ . Следует отметить, что контекст  $K^-$  также допускает преобразование. Контекст  $K^-$  порождает 7 формальных понятий, которые образуют решетку  $L_{K^-}$  (рис. 2). Единицей этой решетки является формальное понятие  $(G^-, \{s_4\})$ , а нулем  $(\emptyset, M)$ .

Контексты  $K^+ = (G^+, M, I^+)$ ,  $K^- = (G^-, M, I^-)$  соответствуют двум классам  $G^+$  и  $G^-$  произведений, разделенных по целевому бинарному признаку  $z = \text{«произведение относится (не относится) к жанру героического эпоса»}$ . Пусть задано новое произведение  $x$  с множеством признаков  $M_x = \{s_2, a_1, q_1, c_2, t_1\}$ . Требуется для  $x$  определить класс, к которому его можно отнести.

Таблица 4  
Контекст  $K^-$  произведений тувинского фольклора  
Table 4  
The context  $K^-$  of Tuvan folklore works

№	$s_1$	$s_2$	$s_3$	$s_4$	$a_1$	$a_2$	$q_1$	$q_2$	$c_1$	$c_2$	$c_3$	$t_1$	$t_2$	$t_3$
5				1		1		1			1			
6				1		1		1			1			
7				1		1		1			1			

В решетке  $L_{K^+}$  множества признаков  $\{s_3, a_1, q_1, t_1\}$ ,  $\{a_1, q_1, c_2\}$ ,  $\{s_1, a_2, q_1, c_1, t_2\}$ ,  $\{s_3, a_1, q_1, c_1, t_1\}$ ,  $\{s_2, a_1, q_1, c_2, t_3\}$ ,  $\{s_3, a_1, q_1, c_2, t_1\}$  являются положительными гипотезами, а  $\{q_1\}$ ,  $\{a_1, q_1\}$ ,  $\{q_1, c_1\}$  – фальсифицированными положительными гипотезами. В решетке  $L_{K^-}$  множества признаков  $\{s_4\}$ ,  $\{s_4, t_2\}$ ,  $\{s_4, a_2, q_2, c_3\}$ ,  $\{s_4, a_2, q_2, c_3, t_3\}$ ,  $\{s_4, a_1, q_1, c_1, t_2\}$ ,  $\{s_4, a_2, q_2, c_3, t_2\}$  определяют отрицательные гипотезы.



По правилу бинарной классификации произведение  $x$  с набором признаков  $M_x = \{s_2, a_1, q_1, c_2, t_1\}$  будет отнесено к классу  $G^+$ , то есть к жанру героического эпоса, так как  $M_x$  включает положительную гипотезу  $\{a_1, q_1, c_2\}$  и не содержит отрицательных гипотез. Если  $M_x = \{s_4, t_2\}$ , то произведение  $x$  будет отнесено к классу  $G^-$ . При  $M_x = \{q_1\}$  произойдет отказ от классификации. Применение метода скользящего контроля к используемому алгоритму

классификации показало его вполне удовлетворительное качество [14].

Программа FCASCorpus в настоящее время успешно используется в научно-образовательном центре «Тюркология» Тувинского государственного университета для филологических и лингвистических исследований естественно-языковых текстов, представленных в Национальном корпусе тувинского языка. В дальнейшем предполагается расширить функциональные возможности программы FCASCorpus с целью повышения эффективности используемых в ней алгоритмов.

### Литература

1. Салчак А.Я., Байыр-оол А.В. Электронный корпус тувинского языка: состояние, проблемы // Мир науки, культуры, образования. 2013. № 6. С. 408–409.
2. Бавуу-Сюрюн М.В. Вопросы создания электронных ресурсов тувинского языка: некоторые итоги, неотложные задачи и перспективы // Новые исследования Тувы. 2016, № 4. URL: <http://nit.tuva.asia/nit/article/view/610> (дата обращения: 14.06.2017).
3. Барсегян А.А., Куприянов М.С., Степаненко В.В., Холлод И.И. Технологии анализа данных: Data Mining, Visual Mining, Text Mining, OLAP. СПб: БХВ-Петербург, 2008. 384 с.
4. Богатырев М.Ю., Нуриахметов В.Р., Вакурин В.С. Методы анализа формальных понятий в информационных системах технической поддержки // Изв. ТулГУ: Технич. науки. 2013. Вып. 2. С. 25–36.
5. Кузнецов С.О. Автоматическое обучение на основе анализа формальных понятий // Автоматика и телемеханика. 2001. № 10. С. 3–27.
6. Ganter B., Wille R. Formal concept analyses: mathematical foundations. Springer Science and Business Media, 2012, 284 p.
7. Гуров С.И., Онищенко А.А. Классификация на основе АФП и биклассификации: возможности подхода // Прикладная математика и информатика: тр. факульт. ВМК МГУ. 2011. Т. 38. С. 77–87.
8. Vlasov D.V. The methods of forming the theoretical concepts. Jour. of Buryat State Univ., 2009, no. 6, pp. 37–41.
9. Биргоф Г. Теория решеток. М.: Наука, 1984. 568 с.
10. Meddouri N., Meddouri M. Classification methods based on formal concept analysis. CLA 2008, pp. 9–16.
11. Bykova V.V., Mongush Ch.M. On Algebraic Approach of R. Wille and B. Ganter in the Investigation of Texts. Jour. of Siberian Federal Univ.: Math. and Physics. 2017, no. 3, pp. 372–384.
12. Евтушенко С.А. Система анализа данных CONCEPT EXPLORER // КИИ-2000: сб. тр. VII Национальн. конф. по искусств. интеллекту М.: Физматлит, 2000. С. 127–134.
13. Орус-оол С.М. Тувинские героические сказания (текстология, поэтика, стиль). М.: МАКС Пресс, 2001. 422 с.
14. Воронцов К.В. Комбинаторный подход к оценке качества обучаемых алгоритмов // Математические вопросы кибернетики. М.: Физматлит, 2004. Т. 13. С. 5–36.

### ALGORITHMS OF CONCEPTUAL MODELING AND TEXT CLASSIFICATION IN THE TUVAN LANGUAGE CORPUS

V.V. Bykova<sup>1</sup>, Dr. Sc. (Physics and Mathematics), Professor, [bykvalen@mail.ru](mailto:bykvalen@mail.ru)  
Ch.M. Mongush<sup>1,2</sup>, Postgraduate Student, Lecturer, [mongushchod91@yandex.ru](mailto:mongushchod91@yandex.ru)

<sup>1</sup> Siberian Federal University, Svobodny Ave. 79, Krasnoyarsk, 660041, Russian Federation

<sup>2</sup> Tuvan State University, Lenina St. 36, Kyzyl, 667000, Russian Federation



**Abstract.** The corpus is an information-linguistic system based on the collection of digitized texts in some language. Nowadays, the corpus of Tuvan language includes official and business documents and Tuvan literary works.

Expanding of the Tuvan corpus and deepening of the text processing level are continuing. These works lead to the tasks of a natural language text analysis. The main task is classification by precedents and conceptual modeling.

In order to solve these problems, the paper uses an algebraic approach, which is called the analysis of formal concepts. The paper proposes algorithms and programs for constructing a conceptual model of literary works collection and solving the problem of a binary classification by precedents. There are methods of reducing computational complexity of the considered algorithms.

The paper presents the results of computational experiments, which confirm the effectiveness of the proposed methods for reducing computation complexity. Finally, there are the results of conceptual modeling and binary classification of Tuvan folklore works.

**Keywords:** corpus, formal concept analysis, conceptual models of texts, classification algorithm, algorithms of reducing a context dimension.

### References

1. Salchak A.Ya., Bayyr-ool A.V. Electronic housing of tuvan language: condition, issues. *Mir nauki, kultury, obrazovaniya* [The World of Science, Culture and Education]. 2013, no. 6, pp. 408–409 (in Russ.).
2. Bavuu-Syuryun M.V. Creating electronic resources on tuvan language: preliminary results, current challenges and prospects. *Novye issledovaniya Tuvy* [The New Research of Tuva]. 2016, no. 4. Available at: <http://nit.tuva.asia/nit/article/view/610> (accessed June 14, 2017).
3. Barsegyan A.A., Kupriyanov M.S., Stepanenko V.V., Kholod I.I. *Tekhnologii analiza dannykh: Data Mining, Visual Mining, Text Mining, OLAP* [Data Analysis Technologies: Data Mining, Visual Mining, Text Mining, OLAP]. St. Petersburg, BHV-Peterburg Publ., 2008, 384 p.
4. Bogatyrev M.Yu., Nuriakhmetov V.R., Vakurin V.S. Methods of formal notion analysis in technical support information systems. *Izvestiya TulGU. Tekhnicheskie nauki* [News of the Tula State University. Technical Sciences]. 2013, iss. 2, pp. 25–36 (in Russ.).
5. Kuznetsov S.O. Machine Learning on the Basis of Formal Concept Analysis. *Avtomatika i telemekhanika* [Automation and Remote Control]. 2001, vol. 62, iss. 10, pp. 1543–1564.
6. Ganter B., Wille R. *Formal concept analyses: mathematical foundations*. Springer Science and Business Media Publ., 2012, 284 p.
7. Gurov S.I., Onishchenko A.A. Classification based on Formal Concept Analysis and biclusterization: the opportunities of the approach. *Prikladnaya matematika i informatika: Tr. fakulteta VMK MGU* [Applied Mathematics and Computer Science: Faculty of Computational Mathematics and Cybernetics of the Lomonosov MSU]. 2011, vol. 38, pp. 77–87.
8. Vlasov D.V. The methods of forming the theoretical concepts. *Jour. of the Buryat State Univ.* 2009, no. 6, pp. 37–41.
9. Birgof G. *Teorii reshetok* [Category Lattice]. Moscow, Nauka Publ., 1984, 568 p.
10. Meddouri N., Meddouri M. Classification Methods Based on Formal Concept Analysis. *CLA 2008*, pp. 9–16.
11. Bykova V.V., Mongush Ch.M. On Algebraic Approach of R. Wille and B. Ganter in the Investigation of Texts. *Jour. of Siberian Federal Univ.: Math. and Physics*. 2017, no. 3, pp. 372–384.
12. Evtushenko S.A. CONCEPT EXPLORER data analysis system. *Proc. 7th National Conf. on Artificial Intelligence KII-2000*. Moscow, Fizmatlit Publ., 2000, pp. 127–134 (in Russ.).
13. Orus-ool S.M. *Tuvinskie geroicheskie skazaniya (tekstologiya, poetika, stil)* [Tuvan Heroic Folk Tales (Textology, Poetics, Style)]. Moscow, Maks Press, 2001, 422 p.
14. Vorontsov K.V. A combinatorial approach to qualitative assessment of learning algorithms. *Matematicheskie voprosy kibernetiki* [Mathematical Problems of Cybernetics]. Moscow, Fizmatlit Publ., 2004, vol. 13, pp. 5–36 (in Russ.).

### Примеры библиографического описания статьи

1. Быкова В.В., Монгуш Ч.М. Алгоритмы концептуального моделирования и классификации текстов в корпусе тувинского языка // Программные продукты и системы. 2017. Т. 30. № 3. С. 487–495. DOI: 10.15827/0236-235X.119.487-495.
2. Bykova V.V., Mongush Ch.M. Algorithms of conceptual modeling and text classification in the tuvan language corpus. *Programmnye produkty i sistemy* [Software & Systems]. 2017, vol. 30, no. 3, pp. 487–495 (in Russ.). DOI: 10.15827/0236-235X.119.487-495.

УДК 004.89; 004.827; 621.73  
DOI: 10.15827/0236-235X.119.496-503

Дата подачи статьи: 24.04.17  
2017. Т. 30. № 3. С. 496–503

## **КОНЦЕПЦИЯ ПОЛУЧЕНИЯ ИНТЕРСУБЪЕКТИВНЫХ ОТВЕТОВ НА НЕЧЕТКИЕ ЗАПРОСЫ К БАЗЕ ДАННЫХ САПР ТЕХНОЛОГИЧЕСКИХ ПРОЦЕССОВ КОВКИ**

*С.И. Канюков, к.т.н., старший научный сотрудник, ksi@imach.uran.ru;  
А.В. Коновалов, д.т.н., профессор, зав. лабораторией, avk@imach.uran.ru;  
О.Ю. Муйземнек, к.т.н., старший научный сотрудник, olga@imach.uran.ru  
(Институт машиноведения Уральского отделения РАН,  
ул. Комсомольская, 34, г. Екатеринбург, 620049, Россия)*

Результаты эксплуатации САПР технологических процессовковки различных поковок на молотах и прессах представляют собой набор значений технологических параметров, получаемых в процессе решения технологических задач и записываемых в реляционную БД.

Современные СУБД позволяют работать, как правило, только с четкими БД и четкими запросами. Под четкой БД понимается совокупность записей – признаков интересующих пользователя объектов, значения которых представляют собой строки или числа, однозначно понимаемые пользователями. Четкий запрос предполагает перечисление значений таких признаков либо границ этих значений. Такой подход приводит к тому, что объекты, признаки которых почти удовлетворяют требованиям запроса, не будут выбраны и информация о них будет утеряна. С подобными потерями информации помогает справиться теория нечетких запросов к реляционному БД, механизмы которой, базирующиеся на теории нечетких множеств Л. Заде, впервые были предложены в 1984 году и впоследствии получили развитие в работах Д. Дюбуа и Г. Прада.

Разработана концепция применения механизма получения интерсубъективных ответов на нечеткие запросы к реляционной БД САПР технологических процессовковки. На примере решения задачи выбора из БД САПР технологических процессовковки с хорошей проработкой металла в конечной поковке и высокой степенью использования металла слитка определены лингвистические переменные, термы и их числовые характеристики, построены функции принадлежности числовых характеристик соответствующим термам и сформулированы правила выборки с учетом значимости поставленных целей.

Предложенная концепция может быть использована для разработки механизма получения ответов на нечеткие запросы и в других предметных областях.

**Ключевые слова:** САПР, технологияковки, БД, нечеткие запросы, интерсубъективные ответы, лингвистическая переменная.

Результатом эксплуатации САПР технологических процессов (ТП)ковки различных поковок на молотах и прессах является набор значений технологических параметров, получаемых в процессе решения технологических задач по заложенным в систему алгоритмам и записываемых в реляционную БД. Каждой записи соответствует технологическая картаковки – документ, содержащий всю информацию (текстовую, числовую, графическую), необходимую при изготовлении поковки кузнечной бригадой. По мере накопления таких записей на предприятиях обычно возникает необходимость формирования различных выборок из уже спроектированных ТП, которая реализуется путем запросов к этой базе на языке SQL.

Следует отметить, что современные СУБД позволяют работать, как правило, только с четкими БД и четкими запросами. Под четкой БД понимается совокупность записей – признаков интересующих пользователя объектов, значения которых представляют собой строки или числа, однозначно понимаемые пользователями. Четкий запрос предполагает перечисление значений таких признаков либо границ этих значений. Данный подход приводит к тому, что объекты, признаки которых почти удовлетворяют требованиям запроса, не будут выбраны и информация о них будет утеряна.

С подобными потерями информации помогает справиться теория нечетких запросов к реляционному БД, механизмы которой, базирующиеся на теории нечетких множеств Л. Заде [1], впервые были предложены в 1984 г. и впоследствии получили развитие в работах Д. Дюбуа и Г. Прада [2, 3].

Поскольку в реальности мы часто оперируем неопределенными, неточными категориями (*молодой, не старый, недалеко, близко, не очень дорогой* и т.п.), что обусловлено наличием в языке слов, имеющих неоднозначное и схожее значение, в нечетких запросах к БД, которые пытается сформулировать человек, нередко присутствуют неточности и неопределенности. Другими словами, нечеткий запрос в отличие от четкого может содержать признаки, которым обычно ставятся в соответствие лингвистические переменные с нечеткими значениями. Множество нечетких значений лингвистической переменной называют терм-множеством, а каждое отдельное значение из этого множества – лингвистическим термом [4]. Например, с лингвистической переменной *ВЕС* можно связать терм-множество лингвистических термов: *ВЕС* = {*малый, небольшой, средний, большой, очень большой*}.

Реализация нечетких запросов, появившихся в связи с необходимостью смягчить булеву логику

в реляционных БД, является перспективным направлением в современных системах обработки информации, поскольку дает возможность пользователям формулировать запросы на естественном языке. Для этой цели разработаны нечеткая реляционная алгебра и специальные расширения языка SQL для нечетких запросов [4–9].

Следует отметить, что нечеткие запросы активно используются при решении задач информационного поиска и мониторинга [10, 11], а также в системах планирования и проектирования производства, например, для выбора наилучшего по соотношению цены и качества оборудования [12]. Нечеткие запросы, построенные на основе методов нечеткой логики, находят применение при решении проблем, связанных с электронным документооборотом [13]. В последние годы изучаются вопросы создания экспертных систем, разрабатываемых на основе теории нечетких множеств, которые используют нечеткие запросы для поиска информации в образовательном процессе [14].

Не вдаваясь в подробности нечеткой реляционной алгебры и языка SQL для нечетких запросов, рассмотрим предлагаемую методологию организа-

ции таких запросов в САПР ТПковки на примерековки валов на прессах.

На рисунке 1 представлен фрагмент технологической картыковки поковки «Вал упорный» из стали 38ХМ, спроектированной САПР ТПковки.

Проектирование начинается с выбора слитка из номенклатуры предприятия. Основной характеристикой каждого слитка является его масса, с которой однозначно связаны все его размеры и выход годного – доля металла в слитке, пригодная для изготовления поковки. Так, для поковки с массой 1,73 т, изображенной на рисунке 1 (нижний эскиз), выбран слиток массой 5,55 т с выходом годного 0,61 (61 % от массы слитка).

После нагрева выбранного слитка обычно производится его осадка и протяжка в цилиндрическую заготовку, промежуточный подогрев и ковка конечной поковки.

В левой колонке на рисунке 1 дано описание действий кузнечной бригады, в следующей колонке приведены эскизы промежуточных заготовок и конечной поковки, на которых, помимо размеров, указываются характеристики степени деформации (уковы) – числа над эскизами, и далее

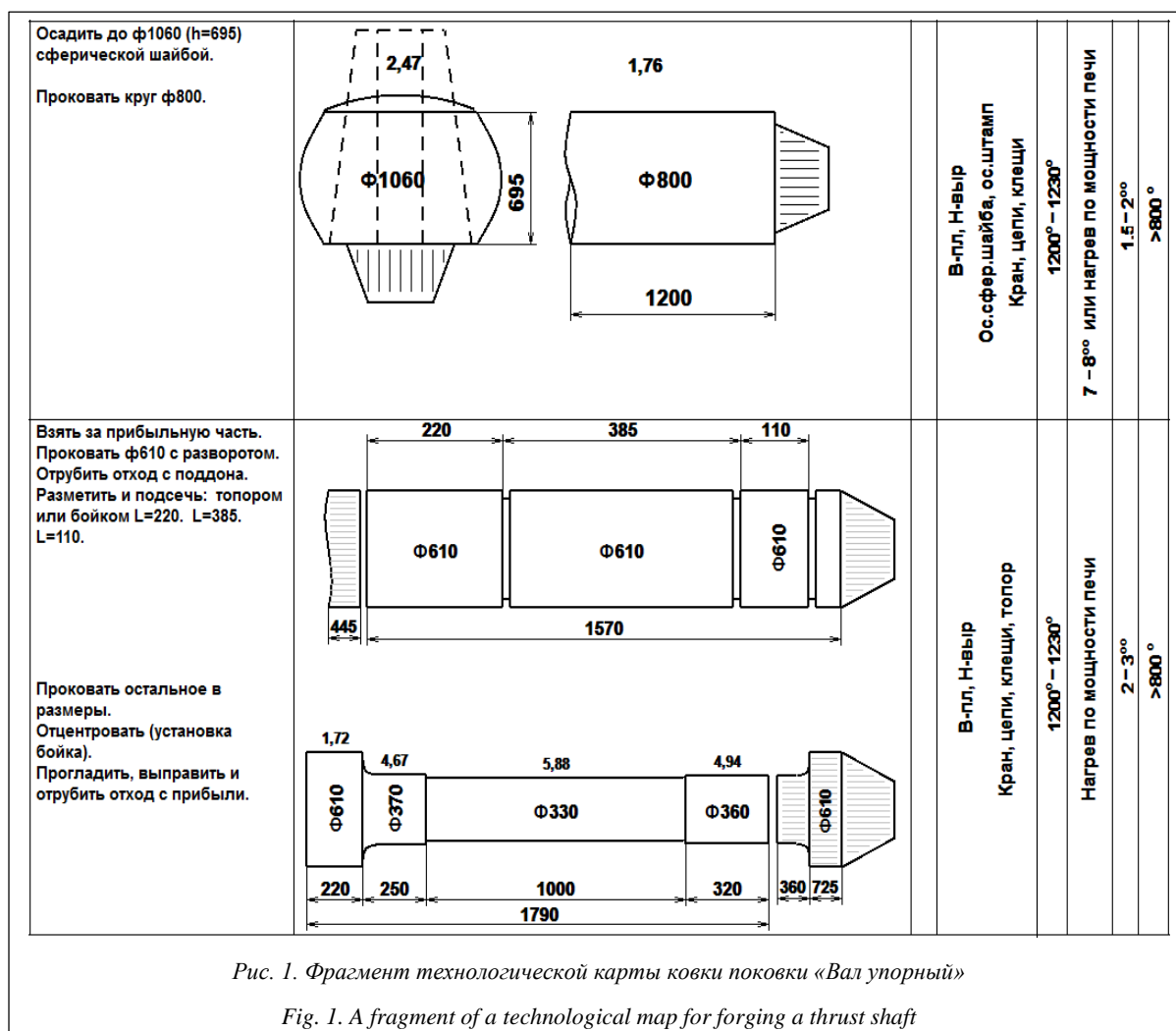


Рис. 1. Фрагмент технологической картыковки поковки «Вал упорный»

Fig. 1. A fragment of a technological map for forging a thrust shaft

слева направо – сведения об основных и вспомогательных инструментах, используемых в процессековки, температура нагрева печи, время нагрева, время выдержки, температура концаковки.

Допустим, реляционная БД САПР ТПковки уже сформирована и пользователь решил сделать следующий запрос: «Выбрать из БД технологические процессы с **хорошей** проработкой металла в конечной поковке и **высокой** степенью использования металла слитка». Такой запрос по существу включает в себя две нечеткие субъективные цели:

СУБЪЕКТИВНАЯ ЦЕЛЬ 1. Хорошая проработка металла поковки;

СУБЪЕКТИВНАЯ ЦЕЛЬ 2. Высокая степень использования металла слитка.

Задача выбора рационального (интерсубъективного) ответа на сформулированный нечеткий запрос, преследующий совокупность субъективных нечетких целей, может быть успешно решена с использованием основных положений теории нечетких множеств и нечеткого управления [15, 16] и теории принятия решений в нечетких условиях по схеме Беллмана–Заде [17].

Методология решения поставленной задачи состоит в выполнении следующих этапов.

**Этап 1. Для формализации нечетких признаков будущей выборки необходимо ввести лингвистические переменные с соответствующими терм-множествами и с каждой лингвистической переменной связать некоторую числовую характеристику.**

Числовой характеристикой качества проработки металла в теории обработки металлов давлением принято считать общий уков  $U$ .

$$U = \frac{D^2}{D_p^2}, \quad (1)$$

где  $D$  – диаметр осаженного слитка;  $D_p$  – габаритный диаметр конечной поковки.

Так, для поковки на рисунке 1 общий уков

$$U = \frac{1060^2}{610^2} = 3,02.$$

Поэтому с качеством проработки металла свяжем лингвистическую переменную  $УКОВ = \{\text{низкий, хороший, высокий}\}$ .

Числовой характеристикой  $K$  степени использования металла слитка будем считать *коэффициент использования металла* (КИМ).

$$K = \frac{M_p}{M_s \times V_s}, \quad (2)$$

где  $M_p$  – масса поковки;  $M_s$ ,  $V_s$  – масса и выход годного слитка соответственно.

Для поковки на рисунке 1 коэффициент использования металла  $K = \frac{1,73}{5,55 \times 0,61} = 0,51$ .

Поэтому с использованием металла слитка свяжем лингвистическую переменную  $КИМ = \{\text{низкий, хороший, высокий}\}$ .

Таким образом, нечеткий запрос «Выбрать технологические процессы с **хорошей** проработкой металла в конечной поковке и **высокой** степенью использования металла слитка» можно перефразировать в нечеткий запрос:

ЗАПРОС: «Выбрать технологические процессы по признакам»; (3)

СУБЪЕКТИВНАЯ ЦЕЛЬ 1.  $УКОВ = \{\text{хороший}\}$ ;

СУБЪЕКТИВНАЯ ЦЕЛЬ 2.  $КИМ = \{\text{высокий}\}$ .

**Этап 2. Для числовой характеристики каждого терма каждой лингвистической переменной необходимо установить область определения.**

Допустим,  $X$  – числовая характеристика некоторого терма. В общем случае область определения  $X$  состоит из трех частей. Например, для лингвистической переменной  $ВОЗРАСТ$  человека и терма *средний* можно принять утверждение: если  $X \in [35, 45]$ , то  $ВОЗРАСТ = \text{средний}$ . Однако можно утверждать и то, что  $X \in [30, 50] \Rightarrow ВОЗРАСТ = \text{средний}$ . Следовательно, область определения в данном случае можно разбить на три зоны:  $[30, 35]$ ,  $[35, 45]$ ,  $[45, 50]$ , границами которых служат значения  $a = 30$ ,  $b = 35$ ,  $c = 45$ ,  $d = 50$ , и в дальнейшем считать:  $X \in [35, 45] \Rightarrow ВОЗРАСТ = \text{средний}$  (безусловно средний),  $X \in [30, 35] \vee X \in [45, 50] \Rightarrow ВОЗРАСТ = \text{средний}$  (с некоторым допущением).

Аналогично и в рассматриваемой задаче. В таблице 1 приведены значения границ зон определения числовых характеристик термов введенных лингвистических переменных, принятых в САПР

Таблица 1

Границы зон определения числовых характеристик

Table 1

Boundaries of quantitative characterization intervals

Переменная	Терм	Числовая характеристика	Границы зон определения числовых характеристик			
			$a$	$b$	$c$	$d$
УКОВ	низкий	$U$	2	2,4	2,6	3
	хороший		2,8	3,5	5	5,5
	высокий		5	5,5	6	–
КИМ	низкий	$K$	0,1	0,3	0,4	0,6
	хороший		0,5	0,6	0,7	0,9
	высокий		0,8	0,9	1	–

ТПковки. Они получены в результате анализа реальных технологических картковки и на основе физических представлений о рассматриваемой области.

**Этап 3. Для каждого лингвистического термина каждой лингвистической переменной необходимо построить функцию принадлежности числовой характеристики этому термину.**

Определение нечеткого множества не накладывает ограничений на вид функции принадлежности элемента множеству. Основное требование, предъявляемое к построению функции принадлежности, – значение функции должно быть больше нуля хотя бы для одного лингвистического термина. Определение конкретного вида функции принадлежности  $\mu$  в каждой предметной области находится за пределами теории нечетких множеств и больше относится к теории экспертного оценивания и методов обработки экспертной информации. На практике наиболее часто применяют трапециевидальную функцию принадлежности [15].

Например, трапециевидальные функции принадлежности  $\mu_1$  числовой характеристики  $U$  лингвистическим терминам переменной  $УКОВ$  могут быть записаны следующим образом.

- Для лингвистического термина *низкий*:

$$\mu_1^{\text{низкий}}(U) = \begin{cases} \frac{U-a}{b-a}, & a \leq U \leq b, \\ 1, & b \leq U \leq c, \\ \frac{d-U}{d-c}, & c \leq U \leq d, \\ 0 & \text{в остальных случаях,} \end{cases} \quad (4)$$

где  $a, b, c, d$  – границы зон определения числовых характеристик, соответствующие этому термину, значения которых приведены в таблице 1.

- Для лингвистического термина *хороший*:  $\mu_1^{\text{хороший}}(U)$  аналогично выражению (4), но только для своих значений  $a, b, c, d$ .

- Для лингвистического термина *высокий*:

$$\mu_1^{\text{высокий}}(U) = \begin{cases} \frac{U-a}{b-a}, & a \leq U \leq b, \\ 1, & b \leq U, \\ 0 & \text{в остальных случаях.} \end{cases} \quad (5)$$

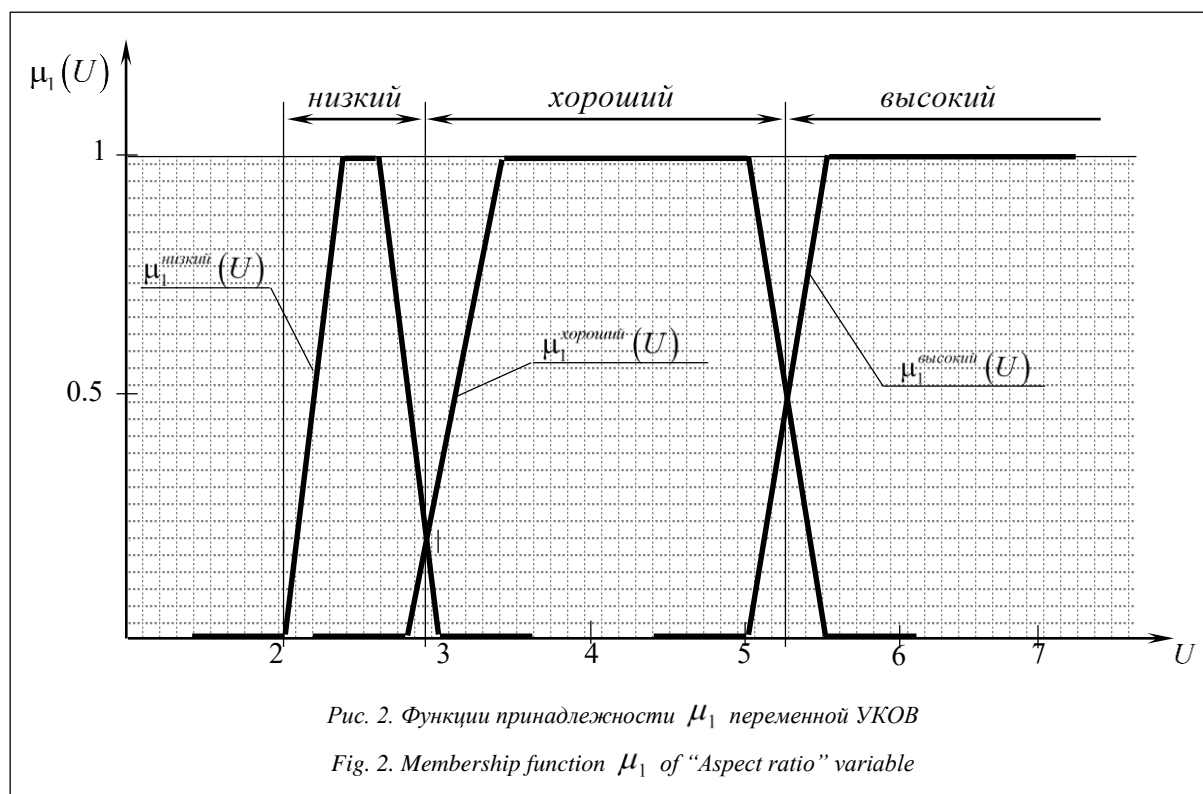
Аналогично строятся функции принадлежности и для переменной  $КИМ$ .

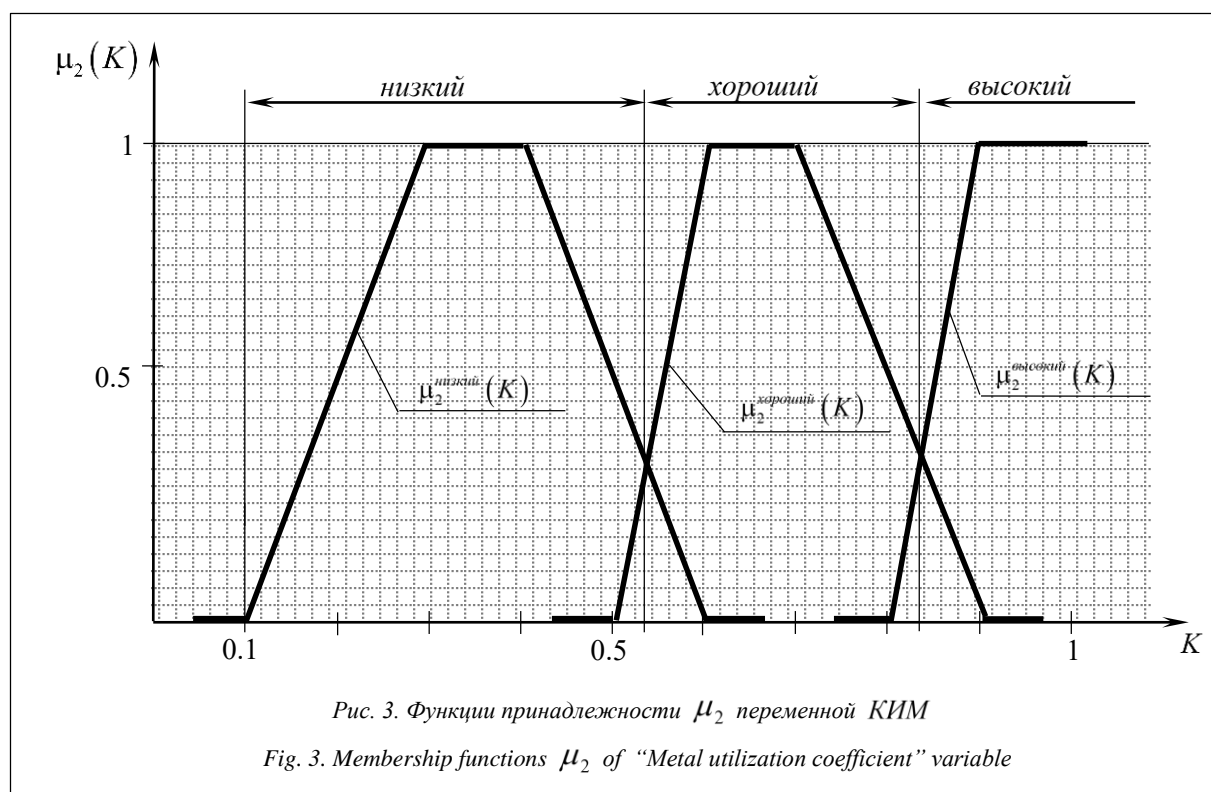
На рисунках 2 и 3 приведены графики трапециевидальных функций принадлежности числовых характеристик  $U$  и  $K$  лингвистическим терминам переменных  $УКОВ$  и  $КИМ$ , построенные на основе выражений (4, 5) и данных таблицы 1.

**Этап 4. Формирование интересубъективной выборки из реляционной БД в соответствии с запросом (3).**

Решение поставленной задачи начинается с расчета значений числовых характеристик  $U$  и  $K$  по формулам (1) и (2) для каждого ТП из БД.

Затем для каждого термина каждой лингвистической переменной по таблице 1 и выражениям (4) и (5) рассчитывается значение функции принадлежности числовых характеристик и решается поставленная задача выборки. Например, для поковки, изображенной на рисунке 1, для которой  $U = 3,02$ ,  $K = 0,51$ , такие расчеты будут следующими:





$$УКОВ = \{ \text{низкий} \} \Rightarrow \mu_1^{\text{низкий}}(3,02) = 0,$$

$$\begin{aligned} УКОВ = \{ \text{хороший} \} &\Rightarrow \mu_1^{\text{хороший}}(3,02) = \\ &= \frac{3,02 - 2,8}{3,5 - 2,8} = 0,31, \end{aligned}$$

$$УКОВ = \{ \text{высокий} \} \Rightarrow \mu_1^{\text{высокий}}(3,02) = 0.$$

$$\mu_1^{\text{хороший}} > \mu_1^{\text{низкий}} \vee \mu_1^{\text{хороший}} > \mu_1^{\text{высокий}} \Rightarrow УКОВ = \{ \text{хороший} \},$$

СУБЪЕКТИВНАЯ ЦЕЛЬ 1 достигнута.

$$КИМ = \{ \text{низкий} \} \Rightarrow \mu_2^{\text{низкий}}(0,51) =$$

$$= \frac{0,6 - 0,51}{0,6 - 0,4} = 0,45,$$

$$КИМ = \{ \text{хороший} \} \Rightarrow \mu_2^{\text{хороший}}(0,51) =$$

$$= \frac{0,51 - 0,5}{0,6 - 0,5} = 0,1,$$

$$КИМ = \{ \text{высокий} \} \Rightarrow \mu_2^{\text{высокий}}(0,51) = 0.$$

$$\mu_1^{\text{низкий}} > \mu_1^{\text{хороший}} \vee \mu_1^{\text{низкий}} > \mu_1^{\text{высокий}} \Rightarrow КИМ = \{ \text{низкий} \},$$

СУБЪЕКТИВНАЯ ЦЕЛЬ 2 не достигнута, и поковка, изображенная на рисунке 1, в выборку не попадет.

В качестве примера в таблице 2 приведен усеченный фрагмент реляционной БД САПР ТПковки, включающей в себя 10 ТП (записей) с расчетными числовыми характеристиками  $U_i$ ,  $K_i$  и значениями функций принадлежности  $\mu_1(U_i)$ ,  $\mu_2(K_i)$ ,  $i = 1, 2, \dots, 10$ .

Правило формирования интерсубъективной выборки по запросу (3) состоит в том, что в каждой  $i$ -й строке таблицы 2 ( $i = 1, 2, \dots, 10$ ) среди столбцов со значениями  $\mu_1(U_i)$  и  $\mu_2(K_i)$  выбираются соответствующие их максимальным значениям (в таблице 2 эти значения выделены жирным шрифтом), и, если выбранные столбцы соответствуют запросу (3),  $i$ -я строка попадает в выборку. Результаты запроса (3) отражены в таблице 3.

Нетрудно заметить, что недостатком такой выборки является то, что расположение строк в таблице 3 совпадает с порядком записей в БД и не отражает степени близости каждого ТП обоим поставленным субъективным целям. Для устранения этого недостатка необходимо ввести результирующую функцию принадлежности  $\mu(U_i, K_i)$ , характеризующую близость  $i$ -го ТП, попавшего в выборку, СУБЪЕКТИВНЫМ ЦЕЛЯМ 1 и 2 одновременно, а именно:

$$\mu(U_i, K_i) = \min [\mu_1^{\text{хороший}}(U_i), \mu_2^{\text{высокий}}(K_i)]. \quad (6)$$

Тогда таблица 3 преобразуется в таблицу 4, где строки уже расположены в порядке уменьшения значения  $\mu(U_i, K_i)$ .

Таким образом, наиболее близким по отношению к СУБЪЕКТИВНЫМ ЦЕЛЯМ 1 и 2 является ТП под номером  $i = 9$ , а наиболее далеким – под номером  $i = 2$ .

Отметим, что правило (6) справедливо, когда ЦЕЛИ 1 и 2 одинаково важны. Более общим является случай, когда важности целей различные [18].

Таблица 2

## Фрагмент реляционной БД САПР ТПковки

Table 2

## A fragment of relational database for forging CAPP software

$i$	Масса, т		$U_i$	$\mu_1(U_i)$			$K_i$	$\mu_2(K_i)$		
	поковки	слитка		низкий	хороший	высокий		низкий	хороший	высокий
1	1,73	5,55	3,02	0,00	<b>0,31</b>	0,00	0,51	<b>0,45</b>	0,10	0,00
2	3,36	7,00	3,00	0,00	<b>0,29</b>	0,00	0,85	0,00	0,25	<b>0,50</b>
3	59,73	103,00	5,20	0,00	<b>0,60</b>	0,40	1,00	0,0	0,0	<b>1,00</b>
4	8,04	16,50	3,72	0,00	<b>1,00</b>	0,00	0,84	0,00	0,30	<b>0,40</b>
5	8,92	24,20	5,31	0,00	0,38	<b>0,62</b>	0,64	0,00	<b>1,00</b>	0,00
6	12,17	50,00	2,23	<b>0,56</b>	0,00	0,00	0,42	<b>0,90</b>	0,00	0,00
7	18,06	32,80	4,71	0,00	<b>1,00</b>	0,00	0,86	0,00	0,20	<b>0,60</b>
8	25,88	53,00	5,30	0,00	0,40	<b>0,86</b>	0,84	0,00	0,30	<b>0,40</b>
9	43,19	83,72	4,78	0,00	<b>1,00</b>	0,00	0,89	0,00	0,05	<b>0,90</b>
10	72,70	140,00	5,97	0,00	0,00	<b>1,00</b>	0,90	0,00	0,00	<b>1,00</b>

Таблица 3

## Выборка по запросу (3)

Table 3

## Response to query (3)

$i$	Масса, т		$U_i$	$\mu_1(U_i)$	$K_i$	$\mu_2(K_i)$
	поковки	слитка		хороший		высокий
2	3,36	7,00	3,00	0.29	0,85	0,50
3	59,73	103,00	5,20	0,60	1,00	1,00
4	8,04	16,50	3,72	1.00	0,84	0,40
7	18,06	32,80	4,71	1,00	0,86	0,60
9	43.19	83.72	4.78	1.00	0.89	0.90

Таблица 4

## Выборка по запросу (3) и правилу (6)

Table 4

## A sample for query (3) respecting rule (6)

$i$	Масса, т		$U_i$	$\mu_1(U_i)$	$K_i$	$\mu_2(K_i)$	$\mu(U_i, K_i) \text{ (6)}$
	поковки	слитка		хороший		высокий	
9	43,19	83,72	4,78	1,00	0,89	0,90	0,90
7	18,06	32,80	4,71	1,00	0,86	0,60	0,60
3	59,73	103,00	5,20	0,60	1,00	1,00	0,60
4	8,04	16,50	3,72	1,00	0,84	0,40	0,40
2	3,36	7,00	3,00	0,29	0,85	0,50	0,29

Таблица 5

Выборка по правилу (7) при  $\alpha_1=1/3$ ,  $\alpha_2=2/3$ 

Table 5

A sample for rule (7) with  $\alpha_1=1/3$ ,  $\alpha_2=2/3$ 

$i$	Масса, т		$U_i$	$[\mu_1(U_i)]^{1/3}$	$K_i$	$[\mu_2(K_i)]^{2/3}$	$\mu(U_i, K_i)$
	поковки	слитка		хороший		высокий	
9	43,19	83,72	4,78	1,00	0,89	0,93	0,93
3	59,73	103,00	5,20	0,84	1,00	1,00	0,84
7	18,06	32,80	4,71	1,00	0,86	0,71	0,71
2	3,36	7,00	3,00	0,66	0,85	0,63	0,63
4	8,04	16,50	3,72	1,00	0,84	0,54	0,54

Пусть  $\alpha_1 \in [0, 1]$ ,  $\alpha_2 \in [0, 1]$  – относительные важности СУБЪЕКТИВНЫХ ЦЕЛЕЙ 1, 2 соответственно и  $\alpha_1 + \alpha_2 = 1$ . Тогда в соответствии с рекомендациями [18] результирующая функция принадлежности  $\mu(U_i, K_i)$  определяется выражением

$$\mu(U_i, K_i) = \min \left\{ [\mu_1^{\text{хороший}}(U_i)]^{\alpha_1}, [\mu_2^{\text{высокий}}(K_i)]^{\alpha_2} \right\}, \quad (7)$$

где  $\alpha_1$ ,  $\alpha_2$  – показатели степени, в которые возводятся  $\mu_1^{\text{хороший}}(U_i)$  и  $\mu_2^{\text{высокий}}(K_i)$ .

Например, пусть  $\alpha_1 = \frac{1}{3}$ ,  $\alpha_2 = \frac{2}{3}$ , то есть СУБЪЕКТИВНАЯ ЦЕЛЬ 2 в два раза важнее СУБЪЕКТИВНОЙ ЦЕЛИ 1. Результат выборки из таблицы 2 в соответствии с правилом (7) приведен в таблице 5.

Нетрудно заметить, что порядок расположения записей в таблице 5 по сравнению с таблицей 4 изменился.

Таким образом, разработана концепция применения механизма получения интересующих ответов на нечеткие запросы к реляционной БД САПР ТПКовки. Применительно к рассмотренной задаче выбора из БД ТПК с **хорошей** проработкой металла в конечной поковке и **высокой** степенью использования металла слитка определены лингвистические переменные, термы и их числовые характеристики, установлены границы зон определения числовых характеристик, построены функции принадлежности числовых характеристик соответствующим термам и сформулированы правила выборки с учетом важности поставленных целей. Результаты работы проиллюстрированы на конкретном примере.

*Работа выполнена при поддержке гранта РФФИ № 16-07-00597\_а в части формализации нечетких целей для получения интересующих ответов на нечеткие запросы и в рамках Комплексной программы УрО РАН, проект № 15-7-1-9, в части формализации лингвистических переменных, их термов и числовых характеристик.*

#### Литература

1. Заде Л. Понятие лингвистической переменной и его применение к принятию приближенных решений. М.: Мир, 1976. 166 с.
2. Дюбуа Д., Прад Г. Теория возможностей. Приложения к представлению знаний в информатике. М.: Радио и связь, 1990. 288 с.
3. Dubois D., Prade H. Using fuzzy sets in database systems: why and how? Proc. of 1996 Workshop on Flexible Query-Answering Systems (FQAS'96), Denmark, May 22–24, 1996, pp. 89–103.
4. Круглов В.В., Дли М.И. Интеллектуальные информационные системы: компьютерная поддержка систем нечеткой логики и нечеткого вывода. М.: Физматлит, 2002. 256 с.
5. Шукшина Н.С., Воронова Л.И. Разработка нечетких запросов к реляционной базе данных системы онлайн-платежей // Междунар. студ. науч. вестн. 2015. № 3–4. С. 522–524.
6. Штовба С.Д. Проектирование нечетких систем средствами MATLAB. М.: Горячая линия–Телеком, 2007. 288 с.
7. Рыжов А.П. О качестве поиска информации по нечетким описаниям // Изв. ЮФУ: Технич. науки. 2006. № 8. С. 10–17.
8. Коновалов Д.П. К вопросу нечетких запросов к реляционным базам данных // Перспективы развития информ. технологий. 2010. № 2. С. 87–92.
9. Целых А.Н., Котов Э.М., Целых А.А. Метод информационного поиска на основе нечеткого сходства ситуаций // Изв. ЮФУ: Технич. науки. 2014. № 6. С. 74–78.
10. Сонькин М.А., Лещик Ю.В. Применение алгоритмов нечеткого поиска в системах мониторинга лесопожарной обстановки // Изв. Томского политех. ун-та. 2012. № 5. С. 98–101.
11. Косоуров Д.А. Применение нечетких запросов в задачах планирования технического обслуживания и ремонта судов // Программные продукты и системы. 2013. № 2. С. 21–24.
12. Чуйкова Е.Н. Реализация нечеткого выбора оборудования в системе проектирования информационной сети // Вестн. Донского гос. технич. ун-та. 2014. № 3. С. 164–171.
13. Венцов Н.Н., Долгов В.В., Подколзина Л.А. Об одном способе построения запросов к базе данных на основе аппарата нечеткой логики // Инженер. вестн. Дона. 2015. № 3 (37). URL: [ivdon.ru/ru/magazine/archive/n3y2015/3172](http://ivdon.ru/ru/magazine/archive/n3y2015/3172) (дата обращения: 23.04.2017).
14. Чванова М.С., Киселева И.А., Молчанов А.А. Выбор проекта и оценка его эффективности на основе нечетких запросов и метода экспертных оценок // Вестн. Тамбовского ун-та: Гуманитар. науки. 2013. № 12. С. 138–150.
15. Рыжов А.П. Элементы теории нечетких множеств и ее приложений. М.: Диалог–МГУ, 1998. 81 с.
16. Шеридан Т.Б., Феррел У.Р. Системы человек–машина: модели обработки информации, управления и принятия решений человеком-оператором. М.: Машиностроение, 1980. 400 с.
17. Bellman R.E., Zadeh L.A. Decision-making in fuzzy environment. Management Science, 1970, vol. 17, no. 4, pp. 141–160.
18. Беллман Р., Заде Л. Принятие решений в расплывчатых условиях. Вопросы анализа и процедуры принятия решений. М.: Мир, 1976. С. 172–215.

#### A CONCEPT OF INTERSUBJECTIVE RESPONSE RECEIVING TO FUZZY QUERIES TO FORGING COMPUTER-AIDED PROCESS PLANNING DATABASE

**S.I. Kanyukov**<sup>1</sup>, Ph.D. (Engineering), Senior Researcher, [ksi@imach.uran.ru](mailto:ksi@imach.uran.ru)  
**A.V. Konovalov**<sup>1</sup>, Dr.Sc. (Engineering), Professor, Head of Laboratory, [avk@imach.uran.ru](mailto:avk@imach.uran.ru)  
**O.Yu. Muizemnek**<sup>1</sup>, Ph.D. (Engineering), Senior Researcher, [olga@imach.uran.ru](mailto:olga@imach.uran.ru)

<sup>1</sup> Institute of Engineering Science of the Ural Branch of the RAS, Komsomolskaya St. 34, Ekaterinburg, 620049, Russian Federation

**Abstract.** A usage experience of Computer-Aided Process Planning (CAPP) software for press and hammer forging is a set of technological parameter values obtained in the technological problem solution process. The values are stored in a relational database.

State-of-the-art database managing software generally use “crisp” (as opposing to “fuzzy”) databases and allow only “crisp” queries. We use “crisp database” as a term for a set of records that hold attributes of objects of user interest. The attribute values are represented by text or numeric values that are unambiguously understood by users. Crisp queries suppose enumeration of



the attribute values or their boundaries. Therefore, the objects with attributes that partly satisfy the query will be excluded from the query response, so the information will be lost. The fuzzy query theory allows overcoming this information loss problem. The theory of fuzzy query to relational database is based on the fuzzy set theory, which was introduced in 1984 by L. Zadeh and then developed by D. Dubois and H. Prade.

The paper considers the developed concept for acquiring intersubjective responses to fuzzy queries to relational database of a forging CAPP. The authors use case study of forging CAPP software database queries to define linguistic variables, terms and their quantitative characterizations. There are membership functions for terms mapped to quantitative characterizations. The paper formulates response construction rules according to relative importance of stated goals. The query requests the technological processes with “good” metal elaboration in final forged part and high utilization of metal from ingot.

The proposed concept can be applied in development of mechanism for acquiring responses to fuzzy queries in numerous subject areas.

**Keywords:** CAPP, forging, database, fuzzy query, intersubjective response, linguistic variable.

**Acknowledgements.** The authors express their gratitude for the financial support from the Russian Foundation for Basic Research (RFBR), project no. 16-07-00597\_a in a part of fuzzy goal formalization for intersubjective responses to fuzzy queries and by the Complex Program of UB RAS (project No. 15-7-1-9) in a part of formalizing linguistic variables, their terms and quantitative characterizations.

### References

1. Zadeh L.A. The Concept of a Linguistic Variable and its Application to Approximate Reasoning. *Information Sciences*. 1975, vol. 8, pp. 199–249 (Russ. ed.: Moscow, Mir Publ., 1976, 166 p.).
2. Dubois D., Prade H. *Possibility Theory*. NY, Plenum Press, 1988 (Russ.ed.: Moscow, Radio i svyaz Publ., 1990, 288 p.).
3. Dubois D., Prade H. Using Fuzzy Sets in Database Systems: Why and How? *Proc. 1996 Workshop on Flexible Query-Answering systems (FQAS'96)*. Denmark, 1996, pp. 89–103.
4. Kruglov V.V., Dli M.I. *Intellektualnye informatsionnye sistemy: kompyuternaya podderzhka sistem nechetskoy logiki i nechetkogo vyvoda* [Intelligent Information Systems: Computer Support of a Fuzzy Logic and Fuzzy Inference]. Moscow, Fizmatlit Publ., 2002, 256 p.
5. Shukshina N.S., Voronova L.I. The development of fuzzy queries to a relational database of an online payment system. *Mezhdunar. stud. nauch. vestn.* [European Student Scientific Jour.]. 2015, no. 3–4, pp. 522–524.
6. Shtovba S.D. *Proektirovanie nechetkikh sistem sredstvami MATLAB* [Fuzzy System Design Using MATLAB]. Moscow, Goryachaya liniya–Telekom Publ., 2007, 288 p.
7. Ryzhov A.P. On the quality of information search by fuzzy descriptions. *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences]. 2006, no. 8, pp. 10–17 (in Russ.).
8. Konovalov D.P. On the problem of fuzzy queries to relational databases. *Perspektivy razvitiya informatsionnykh tekhnologiy* [IT Development Prospects]. 2010, no. 2, pp. 87–92.
9. Tselykh A.N., Kotov E.M., Tselykh A.A. Method of information retrieval based on a fuzzy similarity of situations. *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences]. 2014, no. 6 (155), pp. 74–78 (in Russ.).
10. Sonkin M.A., Leshchik Yu.V. Application of fuzzy search algorithms in the systems for monitoring fire situation in the forests. *Izvestiya Tomskogo politekhnich. un-ta* [Bulletin of the Tomsk Polytechnic Univ.]. 2012, no. 5, pp. 98–101 (in Russ.).
11. Kosourov D.A. Fuzzy queries application in plan tasks of ships maintenance and repair. *Programmnye produkty i sistemy* [Software & Systems]. 2013, no. 2, pp. 21–24 (in Russ.).
12. Chuykova E.N. Realization of the fuzzy selection of equipment in the information network design system. *Vestn. DGTU* [Vestnik of DSTU]. 2014, no. 3 (78), pp. 164–171 (in Russ.).
13. Ventsov N.N., Dolgov V.V., Podkolzina L.A. One method of constructing database queries based on fuzzy logic. *Inzhenerny vestnik Dona* [Engineering journal of Don]. 2015, no. 3 (37). Available at: [ivdon.ru/ru/magazine/archive/n3y2015/3172](http://ivdon.ru/ru/magazine/archive/n3y2015/3172) (accessed April 23, 2017).
14. Chvanova M.S., Kiseleva I.A., Molchanov A.A. Project selection and evaluation of its effectiveness based on fuzzy request and method of expert evaluations. *Vestn. Tambovskogo un-ta. Seriya: gumanitarnye nauki* [Tambov Univ. Reports. Series Technical Sciences]. 2013, no. 12 (128), pp. 138–150 (in Russ.).
15. Ryzhov A.P. *Elementy teorii nechetkikh mnozhestv i ee prilozheny* [The Elements of the Fuzzy Sets Theory and Its Applications]. Moscow, Dialog–MGU Publ., 1998, 81 p.
16. Sheridan T.B., Ferrell W.R. *Man-Machine Systems: Information, Control, and Decision Models of Human Performance*. MIT Press, 1975, 452 p. (Russ.ed.: Moscow, Mashinostroenie Publ., 1980, 520 p.).
17. Bellman R.E., Zadeh L.A. Decision-Making in Fuzzy Environment. *Management Science*. 1970, vol. 17, no. 4, pp. 141–160.
18. Bellman R., Zadeh L.A. Decision-Making in Fuzzy Environment. *Voprosy analiza i protsedury prinyatiya resheny* [The Problems of Analysis and Decision-Making Procedures]. Moscow, Mir Publ., 1976, pp. 172–215 (in Russ.).

УДК 004.4

DOI: 10.15827/0236-235X.119.504-509

Дата подачи статьи: 28.02.17

2017. Т. 30. № 3. С. 504–509

## **АВТОМАТИЗАЦИЯ ПРОЦЕССОВ ПОДГОТОВКИ ПРОГРАММНОЙ ДОКУМЕНТАЦИИ НА ИЗДЕЛИЯ РАДИОЭЛЕКТРОННОЙ ПРОМЫШЛЕННОСТИ**

*А.В. Гурьянов, генеральный директор, postmaster@elavt.spb.ru*

*(ОКБ «Электроавтоматика», ул. Маршала Говорова, 40, г. Санкт-Петербург, 198095, Россия);*

*А.В. Шукалов, к.т.н., доцент, aviation78@mail.ru;*

*И.О. Жаринов, д.т.н., доцент, зав. кафедрой, igor\_rabota@pisem.net;*

*М.О. Костишин, к.т.н., доцент, maksim@kostishin.com;*

*С.А. Леоновец, аспирант, ser2694@ya.ru*

*(Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики, Кронверкский просп., 49, г. Санкт-Петербург, 197101, Россия)*

Рассматривается научная задача автоматизации подготовки текстовой программной документации с помощью специализированного программного средства.

Автоматизация процесса подготовки документации основана на обработке данных, содержащихся в нормативно-технической документации или в техническом задании. Обработка данных предполагает наличие структурированных электронных документов, подготовленных в распространенных форматах в соответствии с шаблонами на основе отраслевых стандартов, и генерацию автоматизированным способом текстового программного документа. Шаблоны могут меняться на любом этапе жизненного цикла проекта. Временные затраты на смену шаблона должны быть минимальными. Для их предварительного показа пользователю используется формат XPS.

Приведены в форме графа перечень разрабатываемых программных документов и пример обработки данных для генерации текстового документа, созданного на основании ГОСТ 19.101-77. Описаны новое разработанное ПО и инструментальные средства, доступные разработчику в проектной деятельности на приборостроительном предприятии. Показан пример изменения состояния атрибутов документов, а также приведен алгоритм создания проекта программного документа в разработанном ПО.

Разработка системы автоматизации оформления технической документации является экономически выгодной. Предлагаемая программа САПР позволяет автоматизировать на приборостроительном предприятии работу разработчиков программно-управляемых изделий, связанную с подготовкой программной документации.

Программное средство разработано и функционирует на базе инструментальной ЭВМ с характеристиками: процессор Intel(R) Core(TM) i5-6300HQ, 4 ядра, тактовая частота 2,3 ГГц, оперативная память 8 Gb под управлением операционной системы Windows 7.

**Ключевые слова:** *программная документация, автоматизация, текстовые документы, генерация.*

Процесс разработки программно-управляемых изделий сопровождается подготовкой *программной документации* (ПД) на различных этапах проектирования: эскизное проектирование, технический проект, технические предложения, рабочее конструкторское проектирование. Номенклатура разрабатываемых документов регламентируется соответствующими государственными стандартами, в частности ГОСТ 19.101-77, и зависит от сложности программы (компонента, комплекса) и области ее применения.

В арсенале разработчиков ПД сегодня имеются различные программные средства, автоматизирующие процесс ее подготовки. Широкое практическое применение получили системы 3SL Cradle 7.0 и ДС БАРС КТ-178В, предназначенные для контроля жизненного цикла ПО, а также текстовый процессор Word, используемый для оформления текстовой документации, и т.д. [1–4].

Основной особенностью применения таких программных средств является их узкая специализация, обеспечивающая разработчику возможность автоматизировать процесс подготовки ограниченного числа видов ПД в отрыве от общего числа до-

кументов, разрабатываемых на проект в целом. Таким образом, разработчики вынуждены устанавливать на инструментальную ЭВМ *автоматизированного рабочего места* (АРМ) различные виды САПР и оформлять техническую документацию проекта поочередно – сначала одну группу ПД с использованием одного вида САПР, затем другую группу ПД с использованием другого вида САПР и т.д. При этом эффективность автоматизации процесса разработки ПД от применения САПР резко снижается и в ряде случаев взаимосвязь между данными, являющимися составной частью различных документов, оказывается сосредоточенной только в памяти разработчика, что создает предпосылки для возникновения ошибок проектирования.

В связи с этим актуальной является задача разработки отраслевых САПР, позволяющих разработчикам осуществлять автоматизированную поддержку процесса сквозного проектирования ПД на программу. Основные методологические аспекты построения отраслевой САПР в области авиаприборостроения приведены в [5–10]. Цель настоящей статьи – представить широкому кругу читателей результаты исследований, направленных на разра-

ботку инструментального средства, реализуемого программным способом, для автоматизированного формирования взаимосвязанных текстовых программных документов на основе обработки данных проекта.

### Номенклатура разрабатываемых ПД

Основополагающим общетехническим документом, который регулирует на предприятии отношения заказчика (военного представителя) и разработчика в части объема и видов ПД, разрабатываемых на программно-управляемое изделие на определенном этапе проектирования, является протокол номенклатуры ПД. В нем указывается следующее: исчерпывающий перечень ПД, подготавливаемых на программу; используемые при этом САПР, включая версии ПО, установленные на инструментальной ЭВМ АРМ; подразделения – ответственные исполнители по разработке и согласованию (метрология, нормоконтроль, технологический контроль и т.д.) ПД; способ (электронный, на бумажных носителях) и сроки хранения документации в архиве предприятия.

На рисунке 1 приведен граф ПД (пример), разрабатываемых на программно-управляемое изделие. Стрелками схематично показаны взаимосвязи между документами – данные, являющиеся исходными для подготовки пошагово разрабатываемых видов ПД.

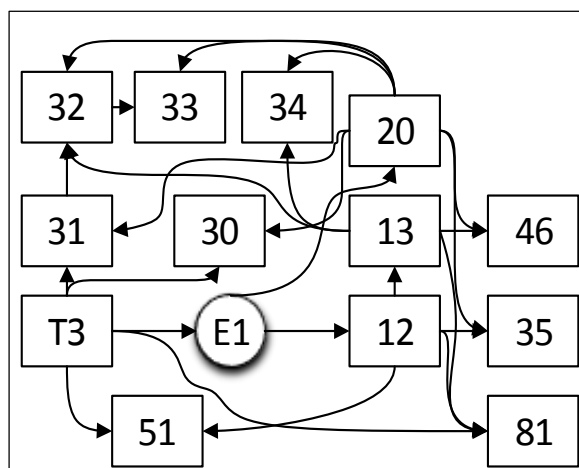


Рис. 1. Граф ПД, разрабатываемых на программно-управляемое изделие (пример)

Fig. 1. A program documentation graph developed for software-controlled product (example)

На рисунке 1 используются обозначения, принятые в отраслевых стандартах для определения различных кодов ПД: 12 – текст программы, 13 – описание программы, 20 – ведомость эксплуатационных документов, 30 – формуляр, 31 – общее описание, 32 – руководство системного программиста, 33 – руководство программиста, 34 – руководство

оператора, 35 – описание языка, 46 – руководство по техническому обслуживанию, 51 – порядок и методика испытаний, 81 – пояснительная записка, ТЗ – техническое задание, Е1 – схема деления изделия на составные части. Взаимосвязи программных документов показаны стрелками.

### Описание инструментального средства проектирования взаимосвязанных ПД

Инструментальное средство проектирования взаимосвязанных ПД представляет собой программный продукт, интерфейс пользователя которого показан на рисунке 2. Программа написана на объектно-ориентированном языке программирования высокого уровня C#, на платформе .NET Framework 4 с использованием встроенной презентационной графической подсистемы для построения клиентских приложений WPF. В качестве средства разработки использована интегрированная среда разработки Microsoft Visual Studio 2015 Professional Edition. В рабочем окне программы отображаются дерево разрабатываемого проекта (1), рабочая область окна предварительного просмотра (2), рабочая область окна редактирования (3), меню разработчика (4), панель инструментов разработчика (5) и строка состояния процесса проектирования (6).

Дерево разрабатываемого проекта представляет собой удобный для навигации по проекту программный инструмент, в котором отображаются нормативно-техническая документация, необходимая для генерации программной документации, документы служебной переписки разработчика и заказчика, а также разрабатываемые ПД.

Перечень разрабатываемых ПД сопровождается в рабочем окне программы указанием на их версию и принадлежность для каждого документа и состоянием разработки. Единица конфигурации (ЕК, configuration item, CI) – данные жизненного цикла, которые в целях управления конфигурацией рассматриваются как единое целое. В качестве ЕК документ имеет следующие обязательные атрибуты:

- уникальный идентификатор, не изменяемый в процессе жизненного цикла и позволяющий идентифицировать изменение документа на всем его протяжении; используется GUID;
- версия, состоящая из двух натуральных чисел (старшая и младшая части номера версии), увеличивающихся при начале изменения документа; старшая часть отражает номер базовой версии (в случае утверждения документа), младшая – номер редакции в процессе подготовки базовой версии;
- имя – строка, название документа;
- дата и время последнего изменения;
- состояние – одно из значений: «черновик» (draft), «проверка» (proposed), «утверждено» (approved), «отклонен» (declined).

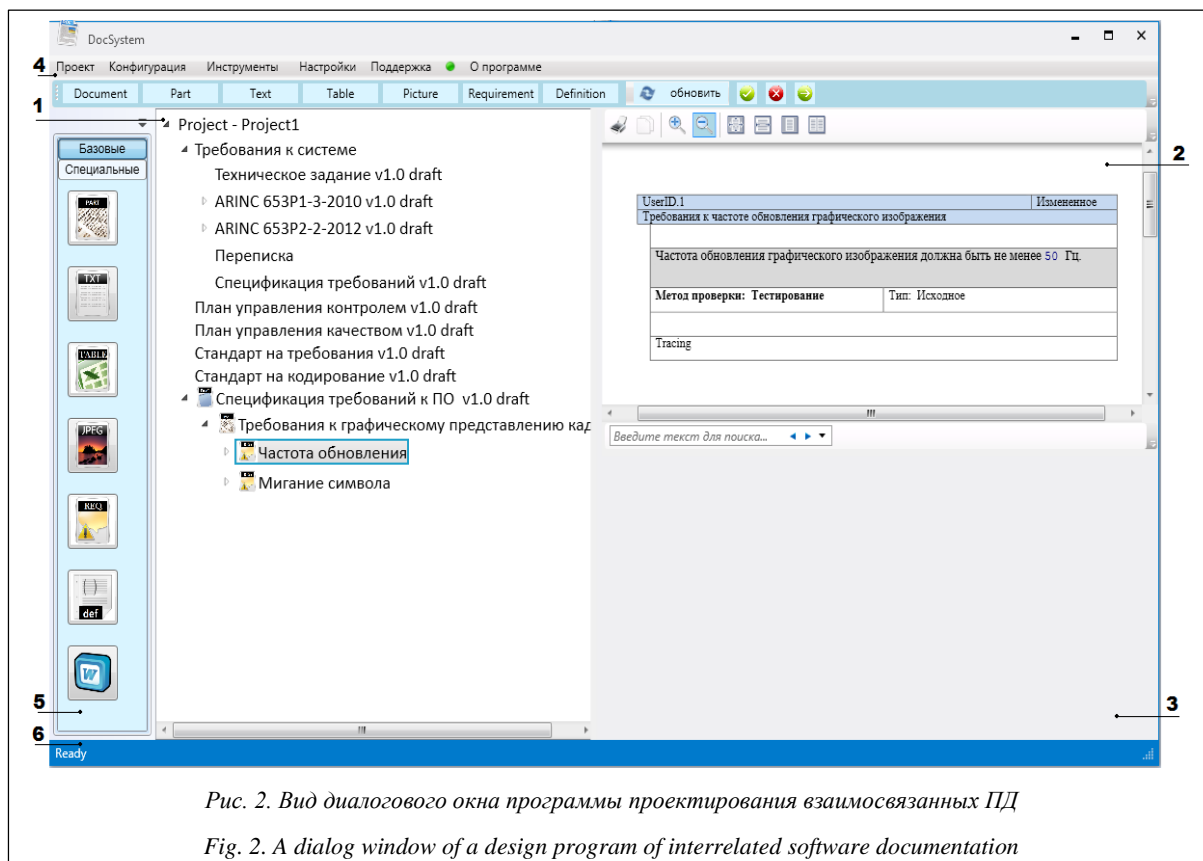


Рис. 2. Вид диалогового окна программы проектирования взаимосвязанных ПД

Fig. 2. A dialog window of a design program of interrelated software documentation

На рисунке 3 представлена схема взаимодействия изменяемых и постоянных атрибутов ЕК.

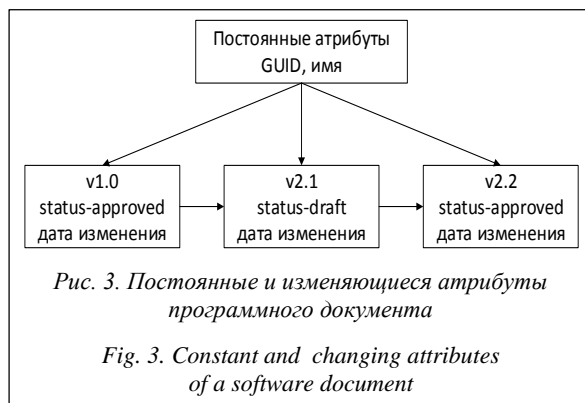


Рис. 3. Постоянные и изменяющиеся атрибуты программного документа

Fig. 3. Constant and changing attributes of a software document

Дерево проекта, представленное на рисунке 2, программно реализовано от API-функции (application programming interface) *System.Windows.Controls.TreeView* операционной системы Windows инструментальной ЭВМ АРМ. Каждый вид элемента разрабатываемого ПД представлен соответствующим видом ярлыка в рабочем окне программы в рабочем поле (1).

Рабочая область окна предварительного просмотра (2) позволяет разработчику оценивать содержание и оформление ПД при его автоматизированной генерации. Инструментальные средства САПР позволяют разработчику просматривать проект документа (режим предварительного про-

смотра) перед его сохранением на диске и при необходимости распечатать его на бумажный носитель. Реализовано несколько полезных возможностей:

- поиск текста в конструкторском документе по ключевым словам, заданным в строке поиска;
- масштабирование документа для удобства его предварительного просмотра.

Меню разработчика (4) в рабочем окне программы, показанном на рисунке 2, представлено следующими опциями.

• «Проект». Включает стандартные функции для работы с текущим проектом (создание нового проекта, загрузка существующего проекта из дискового накопителя, сохранение внесенных изменений в проекте). Опция позволяет выбрать шаблон документа, созданный на предприятии на основе требований отраслевых стандартов, а также сохранять документ на диске инструментальной ЭВМ АРМ в универсальном формате XML (eXtensible Markup Language) для поддержки просмотра и редактирования документа в стандартных текстовых редакторах.

• «Конфигурация». Включает функции для создания разработчиком конфигурации атрибутов элементов ПД, зависящих от типа документа и соответствующей нормативно-технической документации. Атрибуты документа могут быть одного из следующих типов: произвольный текст, список из нескольких элементов, который формирует разработчик конфигурации, логическое значение.

- «Инструменты». Включает функции автоматизированного контроля соответствия документа требованиям нормативно-технической документации, а также функции генерации документа в формате текстового процессора Word.

- «Настройки». Подменю, где разработчику доступно изменение настроек функционирования рабочей программы, а именно: частота автосохранения проекта на диске инструментальной ЭВМ, язык интерфейса программы и т.д.

- «Поддержка». Включает подменю с формой для обращения разработчика в службу технической поддержки, организованную разработчиком САПР (при необходимости).

- «О программе». Содержит сведения о версии и принадлежности САПР.

Панели инструментов (5) разработчика позволяют разработчику получить быстрый доступ к наиболее часто используемым инструментальным средствам программы создания взаимосвязанных ПД.

Строка состояния (6) процесса проектирования представляет разработчику справочные данные и допускает следующие виды состояний программы: Ready (программа готова к использованию), Loading (загрузка проекта или документа), Generation (генерация ПД), Save (сохранение проекта на диске).

#### Пример использования шаблонов конструкторских документов

Программа разработки текстовой программной документации использует шаблоны оформления, основанные на требованиях нормативно-технической документации (отраслевые стандарты). Программа

предполагает подготовку отчетных текстовых документов в электронном виде в формате файлов *dot* или *doc*. Алгоритм процесса создания проекта с использованием программных шаблонов представлен на рисунке 4.

При генерации ПД пользователь может выбрать готовый шаблон или задать и сохранить свой собственный, по которому будет оформлен документ. Автоматически в документе формируются следующие элементы: титульный лист, содержание, таб-

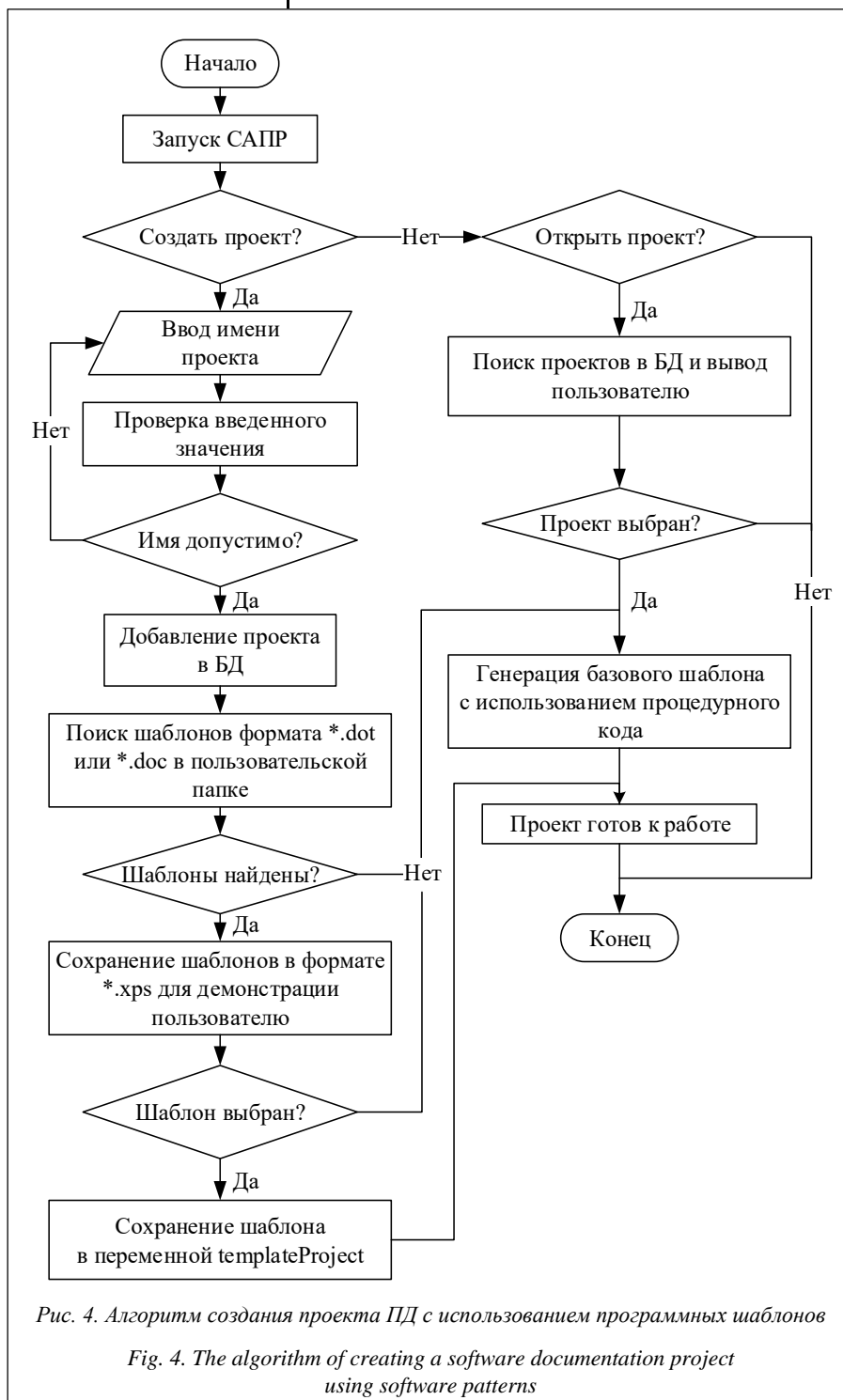


Рис. 4. Алгоритм создания проекта ПД с использованием программных шаблонов

Fig. 4. The algorithm of creating a software documentation project using software patterns

```

foreach (string nameDoc in Directory.GetFiles(MyDoc))
{
    // Отсеиваем файлы, не являющиеся шаблонами
    if (nameDoc.Remove(0, nameDoc.Length - 3) == ".dot")
    {
        // Сохранение в XPS для предпросмотра
        Word._Application oWord = new Word.Application();
        Word._Document oDoc = oWord.Documents.Add(@nameDoc);

        string pathXPS = nameDoc.Remove(nameDoc.Length - 3) + "XPS";
        object fileName = pathXPS;
        object FileFormat = Word.WdSaveFormat.wdFormatXPS;
        object oMissing = System.Reflection.Missing.Value;

        oWord.ActiveDocument.SaveAs(ref fileName,
            ref FileFormat, ref oMissing, ref oMissing, ref oMissing,
            ref oMissing, ref oMissing, ref oMissing, ref oMissing,
            ref oMissing, ref oMissing, ref oMissing, ref oMissing, ref oMissing);

        oDoc.Close(Word.WdSaveOptions.wdDoNotSaveChanges);
        oWord.Quit();

        string nameShort = nameDoc.Remove(nameDoc.Length - 4).Remove(0, MyDoc.Length);
        // Вызов метода, рисующего панель
        CreateTemplateView(nameDoc, pathXPS, nameShort);
    }
}

```

Рис. 5. Фрагмент кода программы

Fig. 5. A fragment of a program code

лицы требований, колонтитулы, а также при необходимости специальные элементы (таблица квалификационных методов, матрица прослеживаемости и пр.).

Фрагмент кода программы, написанной на высокоуровневом языке программирования C#, который обрабатывает документ в формате *dot*, содержащий необходимый шаблон, представлен на рисунке 5.

Таким образом, в программе САПР реализована автоматизированная обработка структурированных электронных текстовых ПД с целью извлечения из них инженерных данных, используемых в ПД, разрабатываемых на последующих этапах проектирования.

### Заключение

Разработка системы автоматизации оформления технической документации является экономически выгодной. Предлагаемая программа САПР позволяет автоматизировать на приборостроительном предприятии работу разработчиков программно-управляемых изделий, связанную с подготовкой программной документации.

Программа САПР имеет объективные преимущества перед известными программными продук-

тами аналогичного назначения, такими как 3SL Cradle 7.0 и ДС БАРС КТ-178В, за счет наличия встроенной опции контроля версий, позволяющей разработчику сопровождать изделия, разработанные по разным версиям ПД, в жизненном цикле продукции, и создавать архивные версии комплектов ПД в электронном виде.

Программное средство разработано и функционирует на базе инструментальной ЭВМ с характеристиками: процессор Intel(R) Core(TM) i5-6300HQ, 4 ядра, тактовая частота 2,3 ГГц, оперативная память 8 Gb под управлением Windows 7.

### Литература

1. Авдеева М., Чиркин А. Перевод бумажной документации в электронный вид // САПР и графика. 2004. № 1. С. 70–72.
2. Садовников Д., Ноздрин А., Ширяев Н. Система управления технической и проектно-конструкторской документацией // САПР и графика. 2002. № 5. С. 74–77.
3. Кукаренко Е., Молочко Д. Управление потоками знаний в техническом документообороте предприятия // САПР и графика. 2001. № 10. С. 35–37.
4. Бычков И., Ващук Ю. Конструкторская спецификация – информационная основа управления предприятием // САПР и графика. 2001. № 9. С. 90–95.
5. Брахутин А.Г. CALS выходит на федеральный уровень // Вестн. авиации и космонавтики. 2001. № 5. С. 26–27.
6. Богатырев В.А. Комбинаторно-вероятностная оценка надежности и отказоустойчивости кластерных систем // Приборы и системы. Управление, контроль, диагностика. 2006. № 6. С. 21–26.
7. Богатырев В.А. Надежность и эффективность резервирования компьютерных сетей // Информационные технологии. 2006. № 9. С. 25–30.
8. Парамонов П.П., Гатчин Ю.А., Жаринов И.О., Жаринов О.О., Дейко М.С. Принципы построения отраслевой системы автоматизированного проектирования в авиационном приборостроении // Науч.-технич. вестн. информ. технологий, механики и оптики. 2012. № 6. С. 111–117.
9. Богатырев В.А., Богатырев С.В. Объединение резервированных серверов в кластеры высоконадежной компьютерной системы // Информационные технологии. 2009. № 6. С. 41–47.
10. Жаринов И.О., Жаринов О.О., Шек-Иовсепянц Р.А., Суслов В.Д. Оценка снижения трудоемкости подготовки конструкторской документации с использованием CALS-технологии в приборостроении // Науч.-технич. вестн. информ. технологий, механики и оптики. 2012. № 4. С. 151–153.

## AUTOMATIZATION OF SOFTWARE DOCUMENTATION PREPARATION PROCESSES FOR RADIO-ELECTRONICS INDUSTRY DEVICES

A.V. Guryanov<sup>1</sup>, Director General, postmaster@elavt.spb.ru

A.V. Shukalov<sup>2</sup>, Ph.D. (Engineering), Associate Professor, aviation78@mail.ru

I.O. Zharinov<sup>2</sup>, Dr.Sc. (Engineering), Associate Professor, Head of Chair, igor\_rabota@pisem.net

M.O. Kostishin<sup>2</sup>, Ph.D. (Engineering), Associate Professor, maksim@kostishin.com

S.A. Leonovets<sup>2</sup>, Postgraduate Student, ser2694@ya.ru

<sup>1</sup> Design Bureau "Electroavtomatika", Marshala Govorova Str. 40, St. Petersburg, 198095, Russian Federation

<sup>2</sup> The National Research University of Information Technologies, Mechanics and Optics, Kronverksky Ave. 49, St. Petersburg, 197101, Russian Federation

**Abstract.** The paper considers a scientific task of automated preparation of text software documentation using a specialized software.

Automation of the process of document preparation is based on data processing from specifications and technical documentation or in the specification. Data processing assumes availability of structured electronic documents prepared in wide-spread formats according to templates on the basis of industry standards and generation by the automated method of the textual program document. Templates can change at any stage of a project lifecycle. Time expenditure on a template change shall be minimum. Their preview to the user is in the XPS format.

The paper provides a list of the developed software documents and the example of data processing to generate a textual document as a graph. The textual document is created based on the state standard 19.101-77. The paper describes a new developed software and tools that are available to a developer in project activities at the instrument-making entity. There is the example of changing a condition of attributes of documents. There is also the algorithm of creating a software document draft in the developed software.

The development of a system for automating the registration of technical documentation is cost-effective. The proposed CAD program allows automating the work of software-controlled product developers related to program documentation preparation in an instrument-making enterprise.

The software is developed and functions based on a computer with characteristics: Intel (R) Core (TM) i5-6300HQ processor, 4 cores, 2.3 GHz clock speed, 8 Gb RAM, Windows 7.

**Keywords:** program documentation, automatization, text document, generating.

### References

1. Avdeeva M., Chirkin A. Conversion of paper documents into electronic format. *SAPR i grafika* [CAD and Graphics]. 2004, no. 1, pp. 70–72 (in Russ.).
2. Sadovnikov D., Nozdrin A., Shiryayev N. Control system for technical and design documentation. *SAPR i grafika* [CAD and Graphics]. 2002, no. 5, pp. 74–77 (in Russ.).
3. Kukarenko E., Molochko D. Managing Knowledge Flows in an Enterprise Technical Document Workflow. *SAPR i grafika* [CAD and Graphics]. 2001, no. 10, pp. 35–37 (in Russ.).
4. Bychkov I., Vashchuk Yu. Design specification is the information basis of enterprise management. *SAPR i grafika* [CAD and Graphics]. 2001, no. 9, pp. 90–95 (in Russ.).
5. Brakhutin A.G. CALS rises to the federal level. *Vestn. aviatsii i kosmonavтики* [Aerospace Herald]. 2001, no. 5, pp. 26–27 (in Russ.).
6. Bogatyrev V.A. The combinatorial stochastic method of reliability evaluation and fault tolerance for networks with cluster architecture. *Pribory i sistemy. Upravlenie, kontrol, diagnostika* [Instruments and Systems: Monitoring, Control, and Diagnostics]. 2006, no. 6, pp. 21–26 (in Russ.).
7. Bogatyrev V.A. Reliability and Efficiency Reservations of Computer Networks. *Informatsionnye tekhnologii* [Information Technologies]. 2006, no. 9, pp. 25–30 (in Russ.).
8. Paramonov P.P., Gatchin Yu.A., Zharinov I.O., Zharinov O.O., Deyko M.S. Principles of constructing a branch system of computer-aided design in aircraft instrument making. *Nauch.-tekhnich. vestn. inform. tekhnology, mekhaniki i optiki* [Scientific and Technical Journal of Information Technologies, Mechanics and Optics]. 2012, no. 6, pp. 111–117 (in Russ.).
9. Bogatyrev V.A., Bogatyrev S.V. Association Reservation Servers in Clusters Highly Reliable Computersystem. *Informatsionnye tekhnologii* [Information Technologies]. 2009, no. 6, pp. 41–47 (in Russ.).
10. Zharinov I.O., Zharinov O.O., Shek-Iovsepyants R.A., Suslov V.D. Assessment of labor intensity reduction when preparing design documentation using CALS-technology in instrument engineering. *Nauch.-tekhnich. vestn. inform. tekhnology, mekhaniki i optiki* [Scientific and Technical Journal of Information Technologies, Mechanics and Optics]. 2012, no. 4, pp. 151–153 (in Russ.).

УДК 519.171.4

DOI: 10.15827/0236-235X.119.510-516

Дата подачи статьи: 13.02.17

2017. Т. 30. № 3. С. 510–516

## **ПРОСТРАНСТВЕННО-ВРЕМЕННАЯ МОДЕЛЬ ТРАНСПОРТНОЙ СЕТИ СИСТЕМЫ МАТЕРИАЛЬНО-ТЕХНИЧЕСКОГО ОБЕСПЕЧЕНИЯ ЭКСПЛУАТАЦИИ ГРУППИРОВКИ РАДИОЭЛЕКТРОННЫХ СРЕДСТВ**

*С.В. Игнатьев, д.т.н., профессор;**В.Б. Тихонов, к.т.н., доцент;**А.В. Красников, к.т.н., ст. преподаватель;**А.А. Осипов, адъюнкт, alexalexosipov@mail.ru**(Ярославское высшее военное училище противовоздушной обороны,  
Московский просп., 28, г. Ярославль, 150001, Россия)*

Поддержание надежности сложных радиоэлектронных средств на заданном уровне в первую очередь зависит от эффективности функционирования существующей системы технической эксплуатации, одной из составных частей которой является система материально-технического обеспечения.

Основной задачей данной системы является своевременное пополнение запасов материально-технических ресурсов, необходимых для правильной эксплуатации радиоэлектронных средств. При этом в связи со значительным пространственным разнесением комплексов специального назначения, основу которых составляют радиоэлектронные средства, от баз хранения и ремонта становится актуальной задача быстрого и правильного принятия решения при управлении доставкой необходимых ресурсов. Для этого требуется знать возможные (предпочтительные) маршруты доставки материально-технических ресурсов с учетом различных аварийных ситуаций, дорожных заторов, масштабных дорожных работ, природных катаклизмов и других факторов, а также оптимальные из них по критерию требуемого значения времени и (или) стоимости доставки материально-технических ресурсов.

В работе предложена пространственно-временная модель транспортной сети системы материально-технического обеспечения эксплуатации группировки радиоэлектронных средств, на основе которой может быть решена задача определения предпочтительных маршрутов доставки материально-технических ресурсов до потребителя и выбора из них оптимального маршрута по заданным критериям.

**Ключевые слова:** *система материально-технического обеспечения, радиоэлектронные средства, материально-технические ресурсы, пространственно-временная модель, транспортная сеть, граф.*

Основу системы технической эксплуатации составляют подсистемы технического обслуживания, диагностирования и *материально-технического обеспечения* (МТО). Функционирование первых двух систем направлено на увеличение времени наработки *радиоэлектронных средств* (РЭС) на отказ  $T_0$  и уменьшение времени их восстановления  $T_v$  соответственно. На сегодняшний день исследования ведутся по каждой системе, причем в разных направлениях. При этом все больше внимания уделяется экспертным системам или системам поддержки принятия решения обслуживающим персоналом в процессе технической эксплуатации сложных РЭС. Известны определенные результаты данных исследований в области технического диагностирования и обслуживания [1–4]. Однако они получены при определенных ограничениях, одним из которых является допущение того, что РЭС полностью укомплектованы требуемыми *материально-техническими ресурсами* (МТР) и последние не оказывают никакого влияния на  $T_0$  и  $T_v$ . Такая ситуация не в полной мере соответствует действительности, что приводит к необходимости более глубокого исследования *системы материально-технического обеспечения* (СМТО).

Под СМТО эксплуатации понимается совокупность материальных, транспортных, информационных и других компонентов, служащих для функционирования технической системы с заданной эф-

фективностью. Одной из основных задач МТО является своевременная доставка до потребителя МТР [5]. Ее решение характеризуется временем доставки  $t_d$  (составляющая  $T_v$ ) и определяется как

$$t_d = t_{\text{си}} + t_{\text{фс}} + t_{\text{тр}}, \quad (1)$$

где  $t_{\text{си}}$  – время сбора информации о наличии МТР в подразделениях СМТО;  $t_{\text{фс}}$  – время формирования схемы транспортировки МТР;  $t_{\text{тр}}$  – время транспортировки МТР.

Опыт эксплуатации показывает, что среди составляющих времени  $t_d$  наиболее высокие значения имеет время транспортировки, которое зависит от ряда факторов. Прежде всего это взаимное расположение на местности пунктов отправки и пунктов приема МТР, что определяет пространственную топологию транспортной сети (так называемый пространственный параметр), и время транспортировки, которое определяется расстоянием  $s$  между пунктами доставки и средней скоростью движения  $v$ , которая, в свою очередь, зависит от типа транспортного средства, доставляющего груз. Последние две величины определяют временной параметр транспортной сети.

Анализ существующей СМТО РЭС показывает, что на сегодняшний день задача выбора оптимального маршрута (схемы перевозки) МТР с точки зрения минимума  $t_{\text{тр}}$  решается не в полной мере [6]. Для комплексного учета всех возможных ситуаций и принятия правильного решения по обеспечению





требуемыми ресурсами потребителя необходимо построить модель транспортной сети СМТО, которую назовем пространственно-временной моделью. Для этого решим задачу формализации, то есть осуществим переход от концептуальной модели (разнотипных элементов системы, описанных в терминах предметной области) к ее математическому описанию (формальным однотипным). Очевидно, что наиболее просто это осуществить, используя теорию графов [7–9].

Рассмотрим абстрактный пример. Пусть имеется некая транспортная сеть (рис. 1).

На первом этапе представим узловые пункты транспортной сети (населенные пункты, пункты перегрузки груза и смены транспорта, развилки дорог и т.д.) вершинами (узлами) графа, а пути сообщения между этими пунктами (наземные, воздушные, водные) – его ребрами. В рассматриваемом примере (см. рис. 1) эти пути выделены черными ломаными линиями. Поскольку один и тот же узел в различных случаях может быть как источником ресурса, так и его потребителем, транспортную сеть будем представлять неориентированным графом. Кроме того, исходя из логики функционирования транспортной сети, характерной особенностью данного графа является отсутствие петель (узел не может сам себе доставлять необходимый ресурс).

В итоге получим мультиграф транспортной сети (рис. 2)  $G_n = \langle U, R \rangle$ , где  $U = \{u_i / i = 1, 2, \dots, 5\}$  – множество вершин графа;  $R = \{r_k / k = 1, 2, \dots, 8\}$  – множество ребер графа. В общем случае  $G_n = \langle U, R \rangle$ , где  $U = \{u_i / i = 1, 2, \dots, I\}$ ,  $I$  – количество вершин графа;  $R = \{r_k / k = 1, 2, \dots, K\}$ ,  $K$  – количество ребер графа.

Каждый узел транспортной сети (вершина графа) характеризуется набором определенных действий (например, дозаправка транспортного средства, оформление необходимых документов, перегрузка материальных ресурсов из одного транспортного средства в другое и т.д.), которые необходимо выполнить, чтобы продолжить движение. Все операции, проводимые в узлах транспорт-

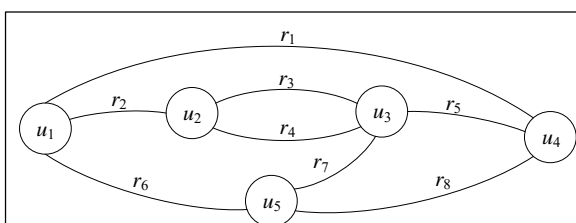


Рис. 2. Неориентированный мультиграф транспортной сети

Fig. 2. A traffic network nonoriented multigraph

ной сети, имеют определенную стоимость и занимают некоторое время. Поэтому каждую вершину графа нагрузим кортежем

$$e_i = [t_i, c_i], \quad (2)$$

где  $i = 1, 2, \dots$ ;  $t_i$  – время выполнения определенных операций (время задержки) в  $i$ -м узле;  $c_i$  – стоимость выполнения этих операций в  $i$ -м узле транспортной сети.

Каждый участок пути транспортной сети (ребро графа) также характеризуется рядом параметров. С точки зрения времени транспортировки МТР выделим протяженность  $s_k$   $k$ -го участка пути ( $r_k$  ребра графа) и качество покрытия  $k$ -го участка, определяющее скорость движения транспортного средства на данном участке (здесь имеется в виду, что в зависимости от качества дороги средняя скорость изменяется). Кроме того, сами транспортные сред-

ства имеют различную среднюю скорость движения  $v(\theta_n)$ , где  $\theta_n$  –  $n$ -й тип транспортного средства,  $\theta_n \in \Theta$ ,  $\Theta = \{\theta_n / n = 1, 2, \dots, N\}$  – множество типов транспортного средства;  $N$  – количество типов транспортных средств транспортной сети, поэтому  $t_{tr}$  также будет различным. Исходя из вышесказанного, для моделирования транспортной сети воспользуемся формализованным представлением транспортных средств (рис. 3), то есть будем использовать общий параметр – среднюю скорость транспортировки определенным типом транспорта  $\theta_n$  МТР по  $k$ -му участку пути –  $v_k(\theta_n)$ . В предложенной классификации транспортных средств (см. рис. 3)  $N = 19$ .

Различие видов и типов транспортных средств определяет и различие в стоимости транспортировки необходимых потребителю ресурсов по  $k$ -му

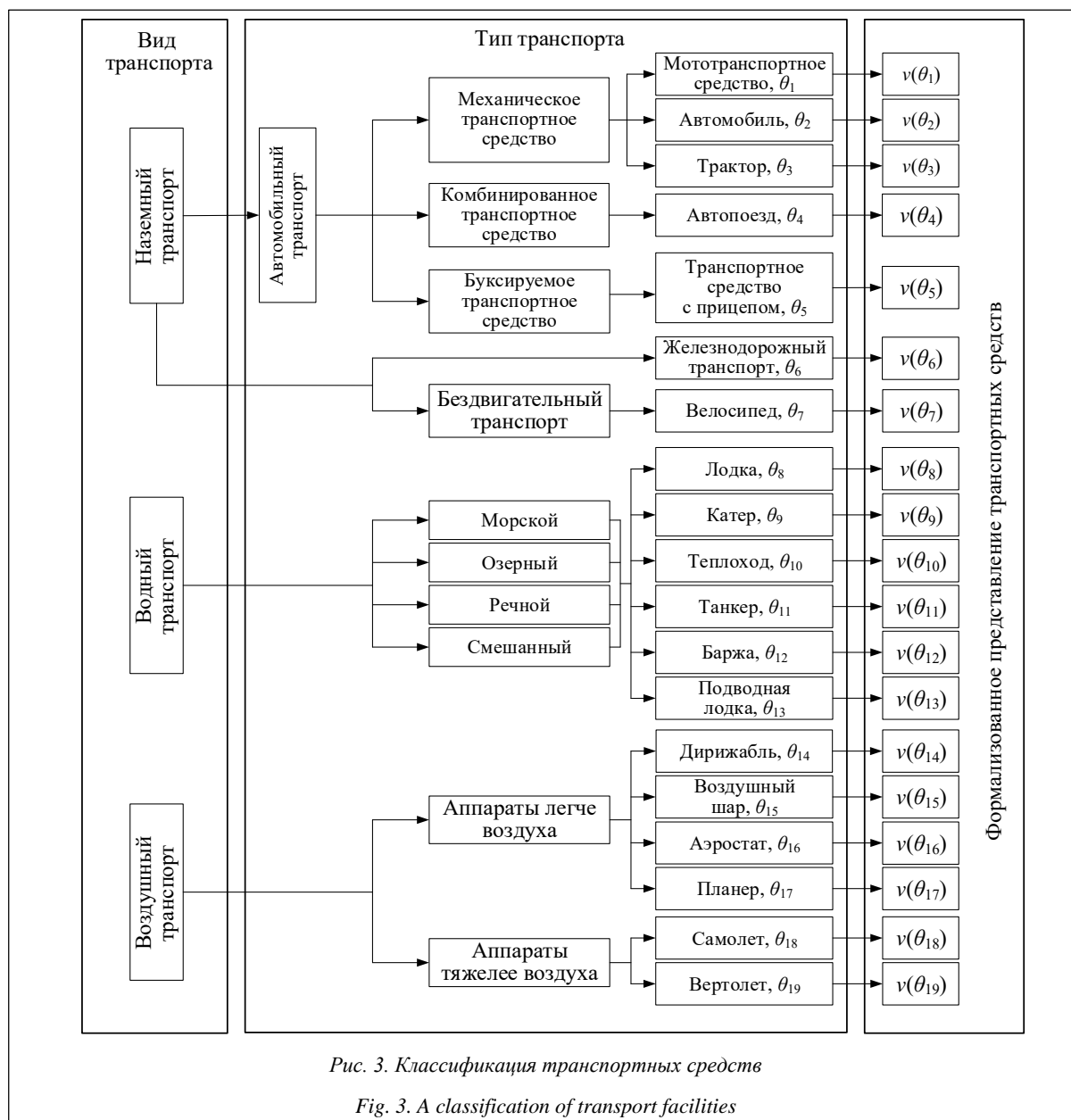


Рис. 3. Классификация транспортных средств

Fig. 3. A classification of transport facilities

участку пути. Данный параметр обозначим как  $c_k(\theta_n)$ .

Также необходимо учитывать время года, время суток, климатические условия (дождь, туман, гололед и т.д.), в которых осуществляется транспортировка МТР, поскольку данные факторы могут приводить к изменению скорости движения транспортного средства. Их влияние учтем коэффициентом скорости  $g_k$ . Его значение может формироваться на основании нормативных документов (правил дорожного движения, руководств по эксплуатации транспортных средств, различных ведомственных правовых документов по безопасности движения и т.д.) и статистических данных по влиянию погоды на скорость потока транспорта.

В результате имеем совокупность параметров (кортеж), характеризующих  $k$ -й участок пути:  $[r_k, s_k, \theta_n, v_k(\theta_n), g_k, c_k(\theta_n)]$ . Данный кортеж интерпретируется следующим образом: по  $k$ -му участку пути протяженностью  $s_k$  движется транспортное средство  $\theta_n$  со скоростью  $v_k(\theta_n)$ . Коэффициент скорости на  $k$ -м участке –  $g_k$ . Стоимость транспортировки МТР по  $k$ -му участку пути транспортным средством  $\theta_n$  составляет  $c_k(\theta_n)$ . Такая характеристика участка пути преобразует его в участок маршрута транспортного средства  $\theta_n$ . Очевидно, что при наличии нескольких типов транспортных средств получим на одном и том же участке пути несколько участков маршрутов, образованных этими транспортными средствами.

Представим данный кортеж в виде веса  $k$ -го ребра графа  $G_n$  (см. рис. 2):  $p = [r_k, s_k, \theta_n, v_k(\theta_n), g_k, c_k(\theta_n)]$ . Совокупность различных весов, характеризующих ребра графа, образует множество весов  $P = \{p_l / l = 1, 2, \dots, L\}$ , где  $L$  – количество весов ребер. При наличии нескольких типов транспортных средств, доставляющих ресурсы по одному и тому же участку пути (ребру), например  $\theta_n$  и  $\theta_m$ , где  $\theta_n, \theta_m \in \Theta$  (см. рис. 3), получим дизъюнкцию различных весов ребра графа, например  $(p_l \vee p_{l+1})$ , где  $p_l = [r_k, s_k, \theta_n, v_k(\theta_n), g_k, c_k(\theta_n)]$  и  $p_{l+1} = [r_k, s_k, \theta_m, v_k(\theta_m), g_k, c_k(\theta_m)]$ ,  $p_l, p_{l+1} \in P$ . То есть с учетом вышеизложенного на  $k$ -м участке пути имеем два участка маршрута транспортных средств –  $\theta_n$  и  $\theta_m$ . Исходя из этого, в общем случае индекс веса ребра графа не будет совпадать с индексом ребра.

Таким образом, нагрузка вершин и ребер графа  $G_n$  преобразует существующие пути транспортной сети в маршруты транспортировки МТР. Другими словами, маршрут – это нагруженный путь, а нагруженный граф  $G_n$  – это граф  $G_m$  маршрутов транспортировки МТР.

При переходе от графа  $G_n$  к графу  $G_m$  необходимо учесть следующее. Поскольку граф путей транспортной сети неориентированный и перевозка материальных средств в различных условиях обстановки возможна как в прямом, так и в обратном направлениях, вес ребра для обоих направлений будет одинаковым.

Абстрагируемся от реальных условий и в произвольном порядке нагрузим граф  $G_n$  (рис. 2) рассматриваемой транспортной сети (рис. 1). В результате получим граф маршрутов  $G_m$  (рис. 4).

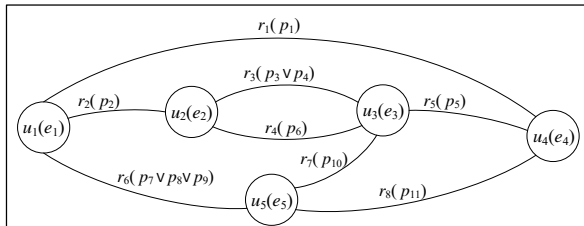


Рис. 4. Граф маршрутов транспортной сети

Fig. 4. A traffic network path graph

В общем виде граф маршрутов транспортной сети опишется выражением

$$G_m = \langle (U, E); (R, P) \rangle,$$

$$U = \{u_1, u_2, \dots, u_i, \dots, u_l\}, E = \{e_1, e_2, \dots, e_i, \dots, e_l\},$$

$$R = \{r_1, r_2, \dots, r_k, \dots, r_k\}, P = \{p_1, p_2, \dots, p_l, \dots, p_l\}, \quad (3)$$

$$e_i = [t_i, c_i], p_l = [r_k, s_k, \theta_n, v_k(\theta_n), g_k, c_k(\theta_n)],$$

$$\theta_n \in \Theta, n = 1, 2, \dots, N.$$

Веса вершин и ребер графа (см. рис. 4) могут выступать в качестве их идентификаторов. Поэтому граф маршрутов опишем матрицей смежности взвешенного графа (весовой матрицей смежности)  $S(G_m) = [s_{ij}]_{l \times l}$  вида

$$s_{ij} = \begin{cases} p_l, & \text{если ребро } r_k \in R, \\ 0, & \text{если ребро } r_k \notin R. \end{cases} \quad (4)$$

В результате весовая матрица смежности  $S(G_m)$  для рассматриваемого графа  $G_m$  будет иметь следующий вид:

$$S(G_m) = \begin{pmatrix} & p_2 & & p_1 & (p_7 \vee p_8 \vee p_9) \\ p_2 & & (p_3 \vee p_4 \vee p_6) & & \\ & (p_3 \vee p_4 \vee p_6) & & p_5 & p_{10} \\ p_1 & & & p_5 & p_{11} \\ (p_7 \vee p_8 \vee p_9) & & p_{10} & p_{11} & \end{pmatrix} \quad (5)$$

Пустые ячейки означают нулевые элементы матрицы. Отсутствие петель в графе  $G_m$  характеризует нулевая главная диагональ матрицы смежности  $S(G_m)$ . Все возможные маршруты между вершинами  $G_m$  могут быть получены с помощью матрицы достижимости [6]:

$$\begin{aligned} D(G_m) &= \sum_{n=1}^{d(G_m)} [S(G_m)]^n = \\ &= S(G_m) + S(G_m)^2 + \dots + S(G_m)^n + \dots \\ &+ S(G_m)^{d(G_m)}, \end{aligned} \quad (6)$$

где  $d(G_m)$  – диаметр графа  $G_m$ .

Под диаметром графа  $G_m$  будем понимать максимальную длину простой цепи между вершинами  $u_i$  и  $u_j$  данного графа. В нашем случае  $d(G_m) = 4$ . Поскольку необходимо получить маршрут от  $i$ -й вершины к  $j$ -й в виде перечисления ребер, а для

$S(G_M)^2 =$				
		$e_1 p_2 e_2 (p_3 \vee p_4 \vee p_6) e_3 \vee$ $\vee e_1 p_1 e_4 p_5 e_3 \vee$ $\vee e_1 (p_7 \vee p_8 \vee p_9) e_5 p_{10} e_3$	$e_1 (p_7 \vee p_8 \vee p_9) e_5 p_{11} e_4$	$e_1 p_1 e_4 p_{11} e_5$
			$e_2 p_2 e_1 p_1 e_4 \vee$ $\vee e_2 (p_3 \vee p_4 \vee p_6) e_3 p_5 e_4$	$e_2 p_2 e_1 (p_7 \vee p_8 \vee p_9) e_5 \vee$ $\vee e_2 (p_3 \vee p_4 \vee p_6) e_3 p_{10} e_5$
$=$	$e_3 (p_3 \vee p_4 \vee p_6) e_2 p_2 e_1 \vee$ $\vee e_3 p_5 e_4 p_1 e_1 \vee$ $\vee e_3 p_{10} e_5 (p_7 \vee p_8 \vee p_9) e_1$		$e_3 p_{10} e_5 p_{11} e_4$	$e_3 p_5 e_4 p_{11} e_5$
	$e_4 p_{11} e_5 (p_7 \vee p_8 \vee p_9) e_1$	$e_4 p_1 e_1 p_2 e_2 \vee$ $\vee e_4 p_5 e_3 (p_3 \vee p_4 \vee p_6) e_2$	$e_4 p_{11} e_5 p_{10} e_3$	$e_4 p_1 e_1 (p_7 \vee p_8 \vee p_9) e_5 \vee$ $\vee e_4 p_5 e_3 p_{10} e_5$
	$e_5 p_{11} e_4 p_1 e_1$	$e_5 (p_7 \vee p_8 \vee p_9) e_1 p_2 e_2 \vee$ $\vee e_5 p_{10} e_3 (p_3 \vee p_4 \vee p_6) e_2$	$e_5 p_{11} e_4 p_5 e_3$	$e_5 (p_7 \vee p_8 \vee p_9) e_1 p_1 e_4 \vee$ $\vee e_5 p_{10} e_3 p_5 e_4$

Рис. 5. Квадрат весовой матрицы смежности

Fig. 5. A weighing adjacency matrix square

нагруженного графа – весов ребер, возведение матрицы смежности  $S(G_M) = [s_{ij}]$  в  $n$ -ю степень будем выполнять по известному правилу перемножения матриц, но при этом операцию умножения будем рассматривать как конкатенацию (приписывание справа к содержимому соответствующего элемента  $i$ -й строки содержимому соответствующего элемента  $j$ -го столбца), то есть стыковку соответствующих весов друг за другом, а операцию сложения полученных в результате умножения слов – как дизъюнкцию. Конкатенация с нулевым элементом даст в итоге нулевой элемент (отсутствие ребра означает обрыв пути). Суммирование матриц (6) также будем выполнять в соответствии с общеизвестным правилом, но по аналогии операцию сложения заменим операцией дизъюнкции. Тогда (6) запишется в виде

$$D(G_M) = \sum_{n=1}^{d(G_M)} [S(G_M)]^n = S(G_M) \vee S(G_M)^2 \vee \dots \vee S(G_M)^n \vee \dots \vee S(G_M)^{d(G_M)}. \quad (7)$$

Для рассматриваемого графа  $G_M$

$$D(G_M) = \sum_{n=1}^4 [S(G_M)]^n = S(G_M) \vee S(G_M)^2 \vee S(G_M)^3 \vee S(G_M)^4. \quad (8)$$

Поскольку в транспортной сети СМТО исключены ситуации, когда промежуточные пункты для одного маршрута транспортировки повторяются, при формировании матрицы достижимости необходимо учитывать только простые маршруты, исключив циклы, то есть повторения вершин графа между начальным  $e_i$  и конечным  $e_j$  узлами маршрута:  $e_i \neq \dots \neq e_q \neq \dots \neq e_j$ .

В соответствии с (8), для примера найдем квадрат весовой матрицы смежности рассматриваемого графа  $S(G_M)^2 = S(G_M) \cdot S(G_M)$ , включив в конкатенацию веса вершин графа (см. рис. 5).

По аналогии, найдя третью  $S(G_M)^3 = S(G_M)^2 \times S(G_M)$  и четвертую  $S(G_M)^4 = S(G_M)^3 \times S(G_M)$  степени матрицы смежности графа  $G_M$ , получим матрицу достижимости  $D(G_M)$ :

$$D(G_M) = \begin{bmatrix} & m_{1,2} & m_{1,3} & m_{1,4} & m_{1,5} \\ m_{2,1} & & m_{2,3} & m_{2,4} & m_{2,5} \\ m_{3,1} & m_{3,2} & & m_{3,4} & m_{3,5} \\ m_{4,1} & m_{4,2} & m_{4,3} & & m_{4,5} \\ m_{5,1} & m_{5,2} & m_{5,3} & m_{5,4} & \end{bmatrix} \quad (9)$$

Содержимое ячеек матрицы (9) представлено системой, которая полностью определяет систему маршрутов транспортной сети СМТО (см. рис. 6).

Таким образом, получена пространственно-временная модель  $\mathbf{H}$  транспортной сети СМТО эксплуатации группировки РЭС, которая представляет собой совокупность графа (8) и матрицы достижимости (рис. 6):  $H = \langle G_M, D(G_M) \rangle$ . В общем виде модель запишется следующим образом:

$$G_M = \langle (U, E); (R, P) \rangle, \\ U = \{u_1, u_2, \dots, u_i, \dots, u_I\}, E = \{e_1, e_2, \dots, e_i, \dots, e_I\}, \\ R = \{r_1, r_2, \dots, r_k, \dots, r_K\}, P = \{p_1, p_2, \dots, p_l, \dots, p_L\}, \\ e_i = [t_i, c_i], p_l = [r_k, s_k, \theta_n, v_k(\theta_n), g_k, c_k(\theta_n)], \\ \theta_n \in \Theta, n = 1, 2, \dots, N.$$

$$D(G_M) = \begin{cases} m_{i,j} = \bigvee_h (e_i p_{x_h} e_{a_h} \dots \vee \dots \vee e_{b_h} p_{y_h} e_j)_h; \\ h = 1, 2, \dots; \forall h: e_i \neq e_{a_h} \neq \dots \neq e_{b_h} \neq e_j; \\ e_i, e_{a_h}, e_{b_h}, e_j \in E; \\ p_{x_h}, p_{y_h} \in P; i = 1, 2, \dots, I; j = 1, 2, \dots, I; i \neq j; \\ m_{i,j} = \bigvee_h (e_i p_{x_h} e_{a_h} \dots \vee \dots \vee e_{b_h} p_{y_h} e_j)_h; \\ h = 1, 2, \dots; \forall h: e_i \neq e_{a_h} \neq \dots \neq e_{b_h} \neq e_j; \\ e_i, e_{a_h}, e_{b_h}, e_j \in E; \\ p_{x_h}, p_{y_h} \in P; j = 1, 2, \dots, I; i = 1, 2, \dots, I; i \neq j. \end{cases} \quad (10)$$

$$\begin{aligned}
D(G_m) = \{ & m_{1,2} = e_1 p_2 e_2 \vee e_1 p_1 e_4 p_5 e_3 (p_3 \vee p_4 \vee p_6) e_2 \vee e_1 (p_7 \vee p_8 \vee p_9) e_5 p_{10} e_3 (p_3 \vee p_4 \vee p_6) e_2 \vee \\
& \vee e_1 (p_7 \vee p_8 \vee p_9) e_5 p_{11} e_4 p_5 e_3 (p_3 \vee p_4 \vee p_6) e_2 \vee e_1 p_1 e_4 p_{11} e_5 p_{10} e_3 (p_3 \vee p_4 \vee p_6) e_2; \\
& m_{1,3} = e_1 p_2 e_2 (p_3 \vee p_4 \vee p_6) e_3 \vee e_1 p_1 e_4 p_5 e_3 \vee e_1 (p_7 \vee p_8 \vee p_9) e_5 p_{10} e_3 \vee \\
& \vee e_1 (p_7 \vee p_8 \vee p_9) e_5 p_{11} e_4 p_5 e_3 \vee e_1 p_1 e_4 p_{11} e_5 p_{10} e_3; \\
& m_{1,4} = e_1 p_1 e_4 \vee e_1 (p_7 \vee p_8 \vee p_9) e_5 p_{11} e_4 \vee e_1 p_2 e_2 (p_3 \vee p_4 \vee p_6) e_3 p_5 e_4 \vee \\
& \vee e_1 (p_7 \vee p_8 \vee p_9) e_5 p_{10} e_3 p_5 e_4 \vee e_1 p_2 e_2 (p_3 \vee p_4 \vee p_6) e_3 p_{10} e_5 p_{11} e_4; \\
& m_{1,5} = e_1 (p_7 \vee p_8 \vee p_9) e_5 \vee e_1 p_1 e_4 p_{11} e_5 \vee e_1 p_2 e_2 (p_3 \vee p_4 \vee p_6) e_3 p_{10} e_5 \vee \\
& \vee e_1 p_1 e_4 p_5 e_3 p_{10} e_5 \vee e_1 p_2 e_2 (p_3 \vee p_4 \vee p_6) e_3 p_5 e_4 p_{11} e_5; \\
& m_{2,3} = e_2 (p_3 \vee p_4 \vee p_6) e_3 \vee e_2 p_2 e_1 p_1 e_4 p_5 e_3 \vee e_2 p_2 e_1 (p_7 \vee p_8 \vee p_9) e_5 p_{10} e_3 \vee \\
& \vee e_2 p_2 e_1 (p_7 \vee p_8 \vee p_9) e_5 p_{11} e_4 p_5 e_3 \vee e_2 p_2 e_1 p_1 e_4 p_{11} e_5 p_{10} e_3; \\
& m_{2,4} = e_2 p_2 e_1 p_1 e_4 \vee e_2 (p_3 \vee p_4 \vee p_6) e_3 p_5 e_4 \vee e_2 p_2 e_1 (p_7 \vee p_8 \vee p_9) e_5 p_{11} e_4 \vee \\
& \vee e_2 (p_3 \vee p_4 \vee p_6) e_3 p_{10} e_5 p_{11} e_4 \vee e_2 p_2 e_1 (p_7 \vee p_8 \vee p_9) e_5 p_{10} e_3 p_5 e_4; \\
& m_{2,5} = e_2 p_2 e_1 (p_7 \vee p_8 \vee p_9) e_5 \vee e_2 (p_3 \vee p_4 \vee p_6) e_3 p_{10} e_5 \vee e_2 p_2 e_1 p_1 e_4 p_{11} e_5 \vee \\
& \vee e_2 (p_3 \vee p_4 \vee p_6) e_3 p_5 e_4 p_{11} e_5 \vee e_2 (p_3 \vee p_4 \vee p_6) e_3 p_5 e_4 p_1 e_1 (p_7 \vee p_8 \vee p_9) e_5 \vee \\
& \vee e_2 p_2 e_1 p_1 e_4 p_5 e_3 p_{10} e_5; \\
& m_{3,4} = e_3 p_5 e_4 \vee e_3 p_{10} e_5 p_{11} e_4 \vee e_3 (p_3 \vee p_4 \vee p_6) e_2 p_2 e_1 p_1 e_4 \vee \\
& \vee e_3 (p_3 \vee p_4 \vee p_6) e_2 p_2 e_1 (p_7 \vee p_8 \vee p_9) e_5 p_{11} e_4; \\
& m_{3,5} = e_3 p_{10} e_5 \vee e_3 p_5 e_4 p_{11} e_5 \vee e_3 (p_3 \vee p_4 \vee p_6) e_2 p_2 e_1 (p_7 \vee p_8 \vee p_9) e_5 \vee \\
& \vee e_3 p_5 e_4 p_1 e_1 (p_7 \vee p_8 \vee p_9) e_5 \vee e_3 (p_3 \vee p_4 \vee p_6) e_2 p_2 e_1 p_1 e_4 p_{11} e_5; \\
& m_{4,5} = e_4 p_{11} e_5 \vee e_4 p_1 e_1 (p_7 \vee p_8 \vee p_9) e_5 \vee e_4 p_5 e_3 p_{10} e_5 \vee e_4 p_5 e_3 (p_3 \vee p_4 \vee p_6) e_2 p_2 e_1 (p_7 \vee p_8 \vee p_9) e_5 \vee \\
& \vee e_4 p_1 e_1 p_2 e_2 (p_3 \vee p_4 \vee p_6) e_3 p_{10} e_5; \\
& m_{2,1} = e_5 p_2 e_1 \vee e_2 (p_3 \vee p_4 \vee p_6) e_3 p_5 e_4 p_1 e_1 \vee e_2 (p_3 \vee p_4 \vee p_6) e_3 p_{10} e_5 (p_7 \vee p_8 \vee p_9) e_1 \vee \\
& \vee e_2 (p_3 \vee p_4 \vee p_6) e_3 p_5 e_4 p_{11} e_5 (p_7 \vee p_8 \vee p_9) e_1; \\
& m_{3,1} = e_5 (p_3 \vee p_4 \vee p_6) e_2 p_2 e_1 \vee e_5 p_5 e_4 p_1 e_1 \vee e_5 p_{10} e_5 (p_7 \vee p_8 \vee p_9) e_1 \vee e_5 p_{10} e_5 p_{11} e_4 p_1 e_1 \vee \\
& \vee e_5 p_5 e_4 p_{11} e_5 (p_7 \vee p_8 \vee p_9) e_1; \\
& m_{3,2} = e_5 (p_3 \vee p_4 \vee p_6) e_2 \vee e_5 p_5 e_4 p_1 e_1 p_2 e_2 \vee e_5 p_{10} e_5 (p_7 \vee p_8 \vee p_9) e_1 p_2 e_2 \vee \\
& \vee e_5 p_{10} e_5 p_{11} e_4 p_1 e_1 p_2 e_2 \vee e_5 p_5 e_4 p_{11} e_5 (p_7 \vee p_8 \vee p_9) e_1 p_2 e_2; \\
& m_{4,1} = e_4 p_1 e_1 \vee e_4 p_{11} e_5 (p_7 \vee p_8 \vee p_9) e_1 \vee e_4 p_5 e_3 (p_3 \vee p_4 \vee p_6) e_2 p_2 e_1 \vee e_4 p_5 e_3 p_{10} e_5 (p_7 \vee p_8 \vee p_9) e_1 \vee \\
& \vee e_4 p_{11} e_5 p_{10} e_3 (p_3 \vee p_4 \vee p_6) e_2 p_2 e_1; \\
& m_{4,2} = e_4 p_1 e_1 p_2 e_2 \vee e_4 p_5 e_3 (p_3 \vee p_4 \vee p_6) e_2 \vee e_4 p_{11} e_5 (p_7 \vee p_8 \vee p_9) e_1 p_2 e_2 \vee \\
& \vee e_4 p_{11} e_5 p_{10} e_3 (p_3 \vee p_4 \vee p_6) e_2 \vee e_4 p_5 e_3 p_{10} e_5 (p_7 \vee p_8 \vee p_9) e_1 p_2 e_2 \vee \\
& \vee e_4 p_1 e_1 (p_7 \vee p_8 \vee p_9) e_5 p_{10} e_3 (p_3 \vee p_4 \vee p_6) e_2; \\
& m_{4,3} = e_4 p_5 e_3 \vee e_4 p_{11} e_5 p_{10} e_3 \vee e_4 p_1 e_1 p_2 e_2 (p_3 \vee p_4 \vee p_6) e_3 \vee e_4 p_1 e_1 (p_7 \vee p_8 \vee p_9) e_5 p_{10} e_3 \vee \\
& \vee e_4 p_{11} e_5 (p_7 \vee p_8 \vee p_9) e_1 p_2 e_2 (p_3 \vee p_4 \vee p_6) e_3; \\
& m_{5,1} = e_5 (p_7 \vee p_8 \vee p_9) e_1 \vee e_5 p_{11} e_4 p_1 e_1 \vee e_5 p_{10} e_3 (p_3 \vee p_4 \vee p_6) e_2 p_2 e_1 \vee \\
& \vee e_5 p_{11} e_4 p_5 e_3 (p_3 \vee p_4 \vee p_6) e_2 p_2 e_1; \\
& m_{5,2} = e_5 (p_7 \vee p_8 \vee p_9) e_1 p_2 e_2 \vee e_5 p_{10} e_3 (p_3 \vee p_4 \vee p_6) e_2 \vee e_5 p_{11} e_4 p_1 e_1 p_2 e_2 \vee \\
& \vee e_5 p_{11} e_4 p_5 e_3 (p_3 \vee p_4 \vee p_6) e_2 \vee e_5 (p_7 \vee p_8 \vee p_9) e_1 p_1 e_4 p_5 e_3 (p_3 \vee p_4 \vee p_6) e_2; \\
& m_{5,3} = e_5 p_{10} e_3 \vee e_5 p_{11} e_4 p_5 e_3 \vee e_5 (p_7 \vee p_8 \vee p_9) e_1 p_2 e_2 (p_3 \vee p_4 \vee p_6) e_3 \vee \\
& \vee e_5 (p_7 \vee p_8 \vee p_9) e_1 p_1 e_4 p_5 e_3 \vee e_5 p_{11} e_4 p_1 e_1 p_2 e_2 (p_3 \vee p_4 \vee p_6) e_3; \\
& m_{5,4} = e_5 p_{11} e_4 \vee e_5 (p_7 \vee p_8 \vee p_9) e_1 p_1 e_4 \vee e_5 p_{10} e_3 p_5 e_4 \vee e_5 p_{10} e_3 (p_3 \vee p_4 \vee p_6) e_2 p_2 e_1 p_1 e_4 \vee \\
& \vee e_5 (p_7 \vee p_8 \vee p_9) e_1 p_2 e_2 (p_3 \vee p_4 \vee p_6) e_3 p_5 e_4.
\end{aligned}$$

Рис. 6. Система маршрутов транспортной сети СМТО

Fig. 6. A transport network route system for a logistic system

Данная модель является основой разработанного программного продукта [10], обеспечивающего выбор оптимального маршрута транспортировки МТР в СМТО эксплуатации группировки РЭС.

#### Литература

1. Игнатьев С.В., Зосиев В.В., Черняк А.В., Малышев А.В. Функциональная структура системы поддержки принятия реше-

ния технического диагностирования радиоэлектронной аппаратуры // Актуальные проблемы вузов ВВС: межвуз. сб. Вып. 21. 2006. С. 87–94.

2. Давыдов П.С. Техническая диагностика радиоэлектронных систем. М.: Радио и связь, 1988. 256 с.

3. Барзилович Е.Ю. Модели технического обслуживания сложных систем. М.: Высш. школа, 1982. 231 с.

4. DSlib (2016). URL: <http://www.dslib.net/sys-analiz/sistema-informacionoj-podderzhki-prinjatija-dlja-tehnicheskogo-obsluzhivaniya-i.html> (дата обращения: 12.02.2017).

5. Дорохов А.Н., Керножицкий В.А., Миронов А.Н., Ше-

стопалова О.Л. Обеспечение надежности сложных технических систем. СПб: Лань, 2016. 352 с.

6. Игнатьев С.В., Тихонов В.Б., Красников А.В., Осипов А.А. Анализ системы материально-технического обеспечения эксплуатации зенитного ракетного вооружения // Вестн. Ярослав. ВВУ ПВО. 2016. Вып. 2. С. 52–57.

7. Оре О. Теория графов. М.: Наука, 1980. 336 с.

8. Горбатов В.А., Горбатов А.В., Горбатова М.В. Дискрет-

ная математика. М.: Астрель, 2003. 447 с.

9. Кристофидес Н. Теория графов. Алгоритмический подход. М.: Мир, 1978. 432 с.

10. Игнатьев С.В., Тихонов В.Б., Осипов А.А. и др. Модель движения материальных потоков системы материально-технического обеспечения радиоэлектронной аппаратуры: Свид. о регистр. прогр. для ЭВМ № 2016610952; Рос. Федерация. Дата регистрации 22.01.2016.

Software & Systems

DOI: 10.15827/0236-235X.119.510-516

Received 13.02.17

2017, vol. 30, no. 3, pp. 510–516

## A SPATIO-TEMPORAL MODEL OF A MATERIAL AND TECHNICAL RESOURCES TRAFFIC NETWORK OF A RADIO-ELECTRONIC FACILITIES GROUP EXPLOITATION

**S.V. Ignatev**<sup>1</sup>, *Dr.Sc. (Engineering), Professor*

**V.B. Tikhonov**<sup>1</sup>, *Ph.D. (Engineering), Associate Professor*

**A.V. Krasnikov**<sup>1</sup>, *Ph.D. (Engineering), Senior Lecturer*

**A.A. Osipov**<sup>1</sup>, *Adjunct, alexalexosipov@mail.ru*

<sup>1</sup> Yaroslavl Higher Military College of Air Defense, Moskovsky Ave., 28, 150001, Yaroslavl, Russian Federation

**Abstract.** Support of complicated radio-electronic facilities (REF) reliability at the established level mostly depends on the operation effectiveness of the current technical exploitation system, with the logistic system (LS) as one of the component part.

The main task of the LS is considered to be well-timed supply of material resources (MR), which are necessary for REF proper exploitation. At the same time, the task of fast and proper decision-making when controlling necessary resources supply becomes relevant due to considerable space diversity and remoteness of special-purpose systems with REF as the basic element from storage and maintenance bases. To perform the task it is necessary to know possible (preferable) routes of MR delivery, taking into consideration emergency situations, traffic jams, roadways heavy maintenance, natural disasters and other factors, as well as the choice of optimal route variants according to the desired time value and/or MR delivery cost. Taking into account the above-mentioned facts, this research suggests a spatio-temporal model of logistics system of the group of radio-electronic facilities transportation network exploitation. This model can help to solve the task of selecting preferable MR delivery routes according to established criteria.

**Keywords:** logistic system, radio-electronic facilities, material resources, spatio-temporal model, transportation network, graph.

## References

1. Ignatev S.V., Zosiev V.V., Tchernyak A.V., Malyshev A.V. Functional structure of decision support system for technical diagnostics of radio electronic equipment. *Aktualnye problemy vuzov VVS: Mezhd. sb. nauch. tr.* [Contemporary Issues of Air Force Universities: Interacademic Collected Papers]. Moscow, Ministry of Defence of the Russian Federation Publ., 2006, iss. 21, pp. 87–94.

2. Davydov P.S. *Tekhnicheskaya diagnostika radioelektronnykh sistem* [Technical Diagnostics of Radio Electronic Systems]. Moscow, Radio i svyaz Publ., 1998, 256 p.

3. Barzilovitch E.Yu. *Modeli tekhnicheskogo obsluzhivaniya slozhnykh sistem* [Models for Complex System Maintenance]. Moscow, Vyssh. Shkola Publ., 1982, 231 p.

4. *DSLlib*. 2016. Available at: <http://www.dslib.net/sys-analiz/sistema-informacionoj-podderzhki-prinjatija-dlja-tehnicheskogo-obsluzhivaniya-i.html> (accessed February 12, 2017).

5. Dorokhov A.N., Kernozhitsky V.A., Mironov A.N., Shestopalova O.L. *Obespechenie nadezhnosti slozhnykh tekhnicheskikh sistem* [Ensuring Reliability of Complex Technical Systems]. 2nd ed., St. Petersburg, Lan Publ., 2016, 352 p.

6. Ignatev S.V., Tikhonov V.B., Krasnikov A.V., Osipov A.A. Analysis of the material support system for anti-aircraft missile weapons operating. *Vestn. Yaroslavskogo VVU PVO* [Bulletin of Yaroslavl Higher Military College of Air Defense]. Yaroslavl, Yaroslavl VVU PVO Publ., 2016, iss. 2.

7. Ore O. *Teoriya grafov* [Theory of Graphs]. Moscow, Nauka Publ., 1980, 336 p.

8. Gorbatov V.A., Gorbatov A.V., Gorbatova M.V. *Diskretnaya matematika* [Discrete Mathematics]. Moscow, Astrel Publ., 2003, 447 p.

9. Kristofides N. *Teoriya grafov. Algoritmitschesky podkhod* [Theory of Graphs. Algorithmic Approach]. Moscow, Mir Publ., 1978, 432 p.

10. Ignatev S.V., Tikhonov V.B., Osipov A.A. *Model dvizheniya materialnykh potokov sistemy materialno-tekhnicheskogo obespecheniya radioelektronnoy apparatury* [A Motion Model of Material Flows of the Material and Technical Support System for Radio Electronic Equipment]. State registration Certificate of the computer program № 2016610952. 2016 (in Russ.).

УДК 004.4

DOI: 10.15827/0236-235X.119.517-523

Дата подачи статьи: 28.02.17

2017. Т. 30. № 3. С. 517–523

## **АВТОМАТИЗАЦИЯ ПРОЦЕССА ФОРМИРОВАНИЯ УЧЕТНОЙ ДОКУМЕНТАЦИИ С ИСПОЛЬЗОВАНИЕМ БАЗЫ ДАННЫХ РАДИОЭЛЕКТРОННЫХ КОМПОНЕНТОВ**

*А.В. Гурьянов, генеральный директор, postmaster@elavt.spb.ru;*

*П.В. Коновалов, к.т.н., инженер, olkesomewhere@gmail.com*

*(ОКБ «Электроавтоматика», ул. Маршала Говорова, 40, г. Санкт-Петербург, 198095, Россия);*

*А.В. Шукалов, к.т.н., доцент, aviation78@mail.ru;*

*И.О. Жаринов, д.т.н., доцент, зав. кафедрой, igor\_rabota@pisem.net;*

*С.А. Леоновец, аспирант, ser2694@ya.ru*

*(Санкт-Петербургский национальный исследовательский университет  
информационных технологий, механики и оптики (Университет ИТМО),*

*Кронверкский просп., 49, г. Санкт-Петербург, 197101, Россия)*

Рассматривается инструментальное ПО, предназначенное для поддержки решения задач автоматизации формирования конструкторской документации, содержащей сведения о цветных и драгоценных металлах, примененных в радиоэлементах, входящих в состав изделия.

Для хранения информации об электрорадиоизделиях и о количестве содержащихся в них ценных металлов используется специально спроектированная БД. Эта база является SQL-совместимой и состоит из четырех взаимосвязанных таблиц. Пользователь может разместить ее как на локальном компьютере, так и в сети. Она содержит необходимую для выполнения расчетных работ информацию и допускает возможность внесения сведений о новых компонентах.

Такую функцию программно реализует процедура, обеспечивающая ручное и автоматическое добавление информации в БД. В ручном режиме элементы заносятся поочередно и производится проверка корректности вводимых данных. В автоматическом режиме информация считывается из файла специального формата XML и элементы добавляются группами с пометкой о наличии данных о содержании в них ценных металлов. Производится проверка добавленных элементов. В отдельном окне программы составляется результирующий проект изделия, содержащий в форме дерева в иерархическом порядке все модули (сборочные единицы), входящие в состав изделия. Для готового проекта выполняется расчет содержания цветных и драгоценных металлов по каждому металлу отдельно и формируются конструкторские документы «расчетные работы». Шаблоны для этих документов должны быть выполнены в соответствии с ГОСТами.

**Ключевые слова:** документация, автоматизация, проектирование, расчетные работы, цветные металлы, драгоценные металлы.

В процессе проектирования бортового приборного оборудования разработчиками используются различные компоненты (сопротивления, конденсаторы, микросхемы и др.), большинство из которых содержат цветные металлы, например рутений, алюминий, медь, бронзу, латунь и др., и/или драгоценные, например золото, серебро, платину, иридий, родий, палладий, осмий. В соответствии с действующим законодательством сведения о наличии ценных металлов в составе изделий подлежат учету и вносятся в состав отдельных видов конструкторских документов (КД) [1–5], подготавливаемых в соответствии с ГОСТ 2.102-68 и протоколом номенклатуры КД, определяющим перечень разрабатываемой документации на соответствующем этапе проектирования:

- расчетные работы (код документа РР1) – расчет драгоценных материалов;
- расчетные работы (код документа РР2) – расчет цветных материалов;
- паспорт (код документа ПС), этикетка (код документа ЭТ), формуляр (код документа ФО);
- руководство по технической эксплуатации (код документа РЭ).

Организационная составляющая работы по учету примененных в изделии ценных металлов

включает в себя документальный контроль за поступлением, движением, инвентаризацией и выбытием драгоценных и цветных металлов, содержащихся в различных сборочных единицах изделий, разрабатываемых на предприятии [6–10]. Таким образом, для каждого изделия должны быть подготовлены документы, оформленные в соответствии с требованиями отраслевых стандартов и содержащие информацию обо всех компонентах, в составе которых присутствуют ценные металлы, а также суммарный расчет общего количества каждого типа драгоценных и цветных металлов в каждой сборочной единице аппаратуры [11–14].

Существуют программные средства, обеспечивающие хранение и ручное заполнение конструкторских документов, например SolidWorks Enterprise PDM от компании SolidWorks. Данная САПР позволяет пользователю заносить данные в специальную форму, а затем, используя готовые шаблоны документов, генерировать их в форматах doc, pdf и др. Однако специализированные программные средства, позволяющие автоматизировать процесс выпуска КД для учета ценных металлов, у отечественных производителей отсутствуют, в связи с чем по сложившейся в авиационной от-

расли практике такие расчетные работы производятся разработчиками вручную.

Для решения задачи автоматизации процесса подготовки таких видов КД может быть использована информационная система 1С, однако предлагаемые на рынке версии продукта не обладают данным функционалом. В связи с этим актуальной является задача разработки нового инструментального ПО, позволяющего в автоматизированном режиме подготавливать КД с кодами РР1, РР2, РС, ЭТ, ФО, РЭ.

Цель данной статьи – представить широкому кругу читателей результаты исследований, посвященных разработке новой программы для инже-

– информация о содержании в компоненте ценных (цветных и драгоценных) металлов.

Для каждого вида металла, содержащегося в компоненте, необходимо указать атрибуты:

- наименование металла;
- тип металла (цветной, драгоценный);
- количество металла, содержащегося в данном компоненте, которое при его отсутствии может быть принято равным нулю.

SQL-совместимая БД состоит из четырех взаимосвязанных таблиц. Таблицы SQL-совместимой БД с информацией о содержании ценных металлов в компонентах могут иметь структуру, представленную на рисунке 1.

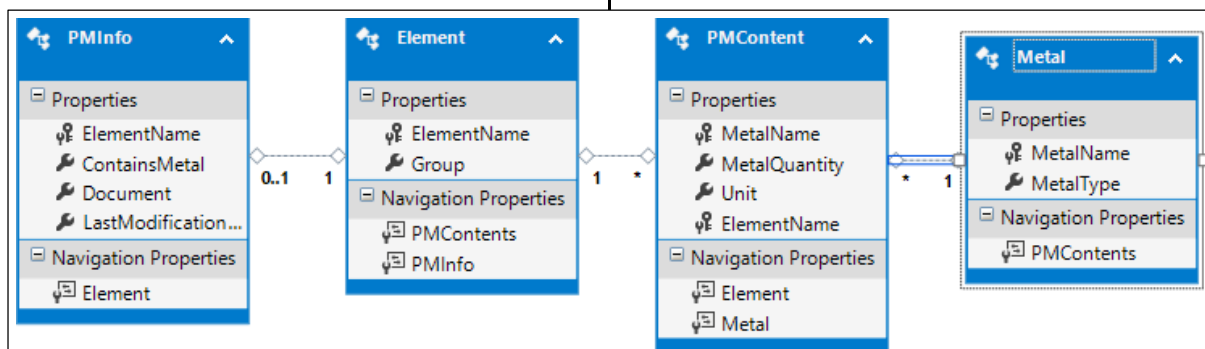


Рис. 1. Структура таблиц SQL-совместимой БД с информацией о содержании ценных металлов в радиоэлектронных компонентах

Fig. 1. The structure of SQL-compatible database tables with information on the content of valuable metals in radio electronic components

нерных расчетов, использующей в качестве хранилища данных о содержании в изделии ценных металлов SQL-совместимую (Structured Query Language) реляционную БД.

#### Организация хранения данных о ценных металлах на основе SQL-совместимой БД

Для каждого радиоэлектронного компонента, входящего в состав БД электрорадиоизделий, применяемых на приборостроительном предприятии, в соответствии с действующими стандартами и нормативно-технической документацией (техническими условиями) указывается следующая информация:

- официальное название компонента (уникально в пределах БД);
- группа, к которой относится данный компонент (сопротивления, конденсаторы, микросхемы и т.д.);
- технический документ (например паспорт, этикетка), в котором заводом-изготовителем дается официальная информация о содержании в компоненте ценных металлов;
- дата внесения изменений о компоненте или первичной регистрации компонента в базе;

Для редактирования записей SQL-совместимой БД используется инструментальная программа, позволяющая как ручной ввод нового компонента, так и внесение большого количества компонентов в автоматическом режиме на основе заранее подготавливаемого списка в формате текстового файла XML (eXtensible Markup Language).

#### Автоматизированное редактирование записей SQL-совместимой БД

Автоматизированное редактирование записей SQL-совместимой БД выполняется разработчиком при помощи программно реализованного инструмента. Основное окно программы редактирования записей SQL-совместимой БД приведено на рисунке 2. Оно позволяет разработчику определять настройки соединения инструментальной ЭВМ, на которой размещено прикладное ПО разработчика, с сервером, на котором находится БД.

БД может быть размещена как на локальном компьютере, так и в сети. После установки соединения прикладного ПО разработчика с БД разработчик может редактировать информацию о компонентах в ручном режиме. Редактирование информации предполагает реализацию функций управления данными, добавления в таблицу новых



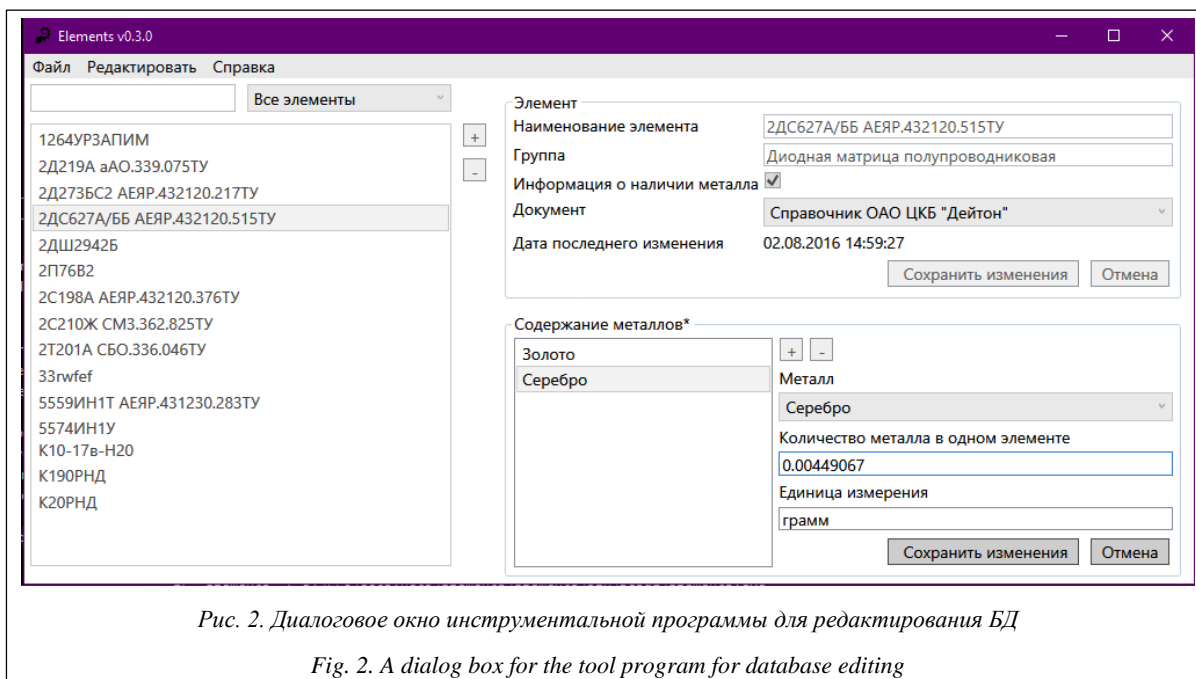


Рис. 2. Диалоговое окно инструментальной программы для редактирования БД

Fig. 2. A dialog box for the tool program for database editing

записей, поиска записей в одной или нескольких таблицах в соответствии с заданными запросами. При внесении изменений производится проверка корректности вводимых сведений. В случае обнаружения несоответствий некорректно заполненные записи помечаются специальным маркером (красной рамкой) и возможность сохранения изменений блокируется.

Для внесения в SQL-совместимую БД информации о ценных металлах, содержащихся в нескольких компонентах, то есть сразу по группе компонентов, инженерные данные представляются структурированно в виде XML-файла, включающего инструкции. При открытии XML-файла с групповой информацией о различных компонентах производится проверка данных на совпадение ком-

понентов, ранее уже занесенных в БД. Данные XML-файла преобразуются к виду, удобному для восприятия разработчиком в диалоговом окне (см. рис. 3). При обнаружении совпадений на экране инструментальной ЭВМ формируется соответствующее информационное сообщение и компоненты, информация о которых уже присутствует в БД, становятся недоступными для дальнейшей работы в автоматическом режиме обновления, так как в этом случае требуется ручная проверка корректности данных. Такая функция реализуется за счет возможности частичной обработки данных XML-файлов.

Остальные компоненты, данные о которых приведены в XML-файле, заносятся в БД группами, для которых указывается либо информация об от-

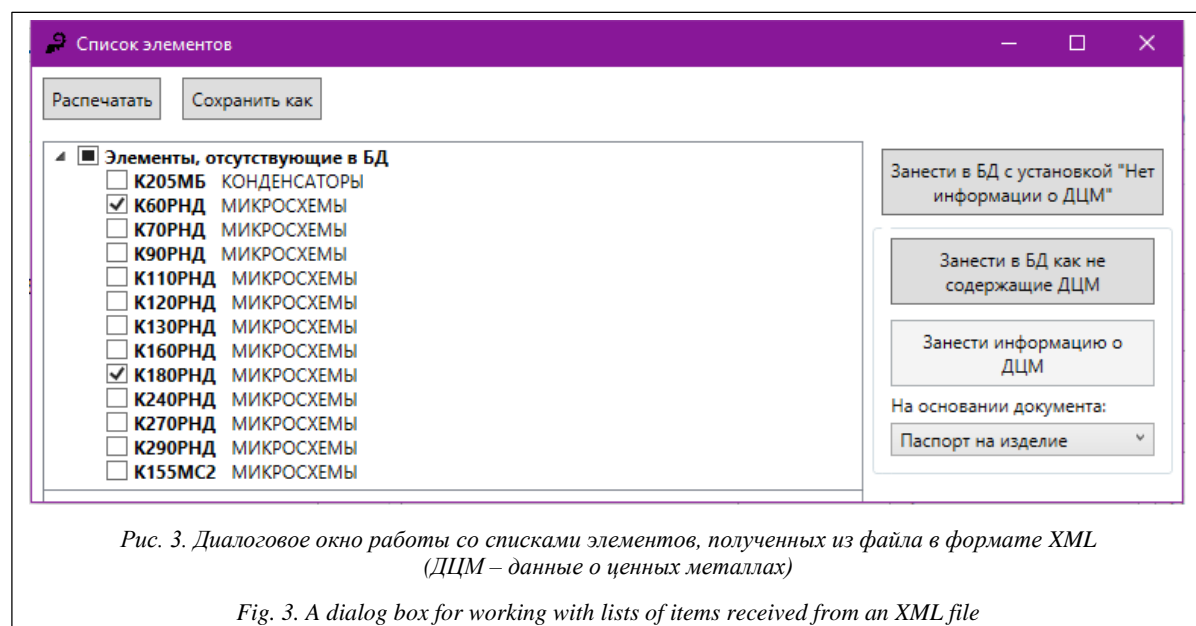


Рис. 3. Диалоговое окно работы со списками элементов, полученных из файла в формате XML (ДЦМ – данные о ценных металлах)

Fig. 3. A dialog box for working with lists of items received from an XML file

сутствии (записи с нулевыми значениями) о содержании ценных металлов, либо документ, на основании которого такая официальная информация получена разработчиком. В основном окне инструментальной программы для редактирования записей БД разработчику также предоставлена возможность выбора компонентов вручную, то есть в таком режиме, когда доступна опция внесения по каждому компоненту всех необходимых сведений, после чего изменения в БД будут сохранены.

### Автоматизированное формирование КД

Расчет содержания ценных металлов производится на основе инженерных данных проекта каж-

дого изделия. Для того чтобы сформировать текстовые КД РР1, РР2, ПС по одному изделию, необходимо получить описание входящих в его состав сборочных единиц – *модуль вычислительный (МВ), модуль напряжений (МН), модуль графический (МГ), модуль коммутации (МК), модуль сопряжения (МС), устройство массовой памяти (УМП), узел межмодульного интерфейса (УМИ), плата коммутации (ПК), фильтр радиопомех (ФРП), корпус К* и т.п. Такие сведения содержатся в КД «Схема деления изделия на составные части», код документа Е1 по единой системе оформления КД.

Для каждой сборочной единицы на соответствующем этапе проектирования (эскизное проек-

Программа расчета содержания металлов

Проект Категории Расчет

Блоки

БГС-945

Модули

Базовые элементы

УМИ104

Плата МВ102

УМП102

Плата МВ101-00

Плата ПК10-6-010

Фильтр ФРП

Фильтр ФРП45

Корпуса

БГС-945

Название элемента БГС-945

ВИДК ВИДК.466514.014

☐ ВИДК задает перечень элементов

☒ ВИДК задает схему деления

☐ элемент покупной

Примечание

подэлементы, входящие в схему деления:

Элемент	ВИДК	ПЭЗ	Количество	Примечание
МН100	ВИДК.436718.016	<input checked="" type="checkbox"/>	2	Изготовление
МВ101-0	ВИДК.467443.060	<input type="checkbox"/>	1	Изготовление
МВ102-0	ВИДК.467443.049	<input type="checkbox"/>	1	
МГ100	ВИДК.468173.034	<input checked="" type="checkbox"/>	6	
МК100	ВИДК.468173.029	<input checked="" type="checkbox"/>	2	
МС102-0	ВИДК.468179.009	<input checked="" type="checkbox"/>	2	
К10-6-01	ВИДК.469543.088	<input type="checkbox"/>	1	

Расчет материалов

Путь к исходным файлам D:\Projects.net3\DeliveryNote\Source\БГС-945

Выберите

Элемент	ВИДК	Файл ПЭЗ	Найден	Игнорировать
МН100	ВИДК.436718.016	D:\Projects.net3\DeliveryNote\Source\БГС-945\ВИДК_436718_016ПЭЗ_02_Плата2.d	<input checked="" type="checkbox"/>	<input type="checkbox"/>
УМИ104	ВИДК.468172.005	D:\Projects.net3\DeliveryNote\Source\БГС-945\ВИДК_468172_005ПЭЗ_00.doc	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Плата МВ101-00	ВИДК.467443.047	D:\Projects.net3\DeliveryNote\Source\БГС-945\ВИДК_467443_047ПЭЗ_03.doc	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Плата МВ102	ВИДК.469535.344	D:\Projects.net3\DeliveryNote\Source\БГС-945\ВИДК_469535_344ПЭЗ_00.doc	<input checked="" type="checkbox"/>	<input type="checkbox"/>
УМП102	ВИДК.467532.010	D:\Projects.net3\DeliveryNote\Source\БГС-945\ВИДК_467532_010ПЭЗ_01.doc	<input checked="" type="checkbox"/>	<input type="checkbox"/>
МГ100	ВИДК.468173.034	D:\Projects.net3\DeliveryNote\Source\БГС-945\ВИДК_468173_034ПЭЗ_00.doc	<input checked="" type="checkbox"/>	<input type="checkbox"/>
МК100	ВИДК.468173.029	D:\Projects.net3\DeliveryNote\Source\БГС-945\ВИДК_468173_029ПЭЗ_00.doc	<input checked="" type="checkbox"/>	<input type="checkbox"/>
МС102-01	ВИДК.468179.009	D:\Projects.net3\DeliveryNote\Source\БГС-945\ВИДК_468179_009ПЭЗ_01.doc	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Плата ПК10-6-010	ВИДК.469535.373	D:\Projects.net3\DeliveryNote\Source\БГС-945\ВИДК_469535_373ПЭЗ_00.doc	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Фильтр ФРП	ВИДК.468833.045	D:\Projects.net3\DeliveryNote\Source\БГС-945\ВИДК_468833_045ПЭЗ_01.doc	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Путь к выходному файлу D:\Projects.net3\DeliveryNote\Source\БГС-945\ВИДК\_466514\_014PP1.doc

Выберите

Рассчитать Отменить

Рис. 4. Диалоговое окно пользовательского интерфейса программы для расчета содержания ценных металлов

Fig. 4. A user interface dialog box of the program for calculating the content of valuable metals

тирование, технический проект, техническое предложение, рабочее конструкторское проектирование) разрабатывается, как правило, в Microsoft Word, текстовый КД с кодом ПЭЗ – перечень элементов, в котором последовательно перечисляются используемые в составе сборочной единицы радиоэлектронные компоненты. Таким образом, в соответствии с иерархическим уровнем возрастания технической сложности изделия [15, 16] «сборочная единица → модуль → блок → система → комплекс» на основании данных расчетов по каждой сборочной единице формируется расчет сведений о суммарном содержании ценных металлов в изделии. В процессе расчетов данные о содержании ценных металлов суммируются раздельно по каждому типу металла.

На рисунке 4 показаны диалоговое окно программы расчета сведений о содержании ценных металлов в изделии и оформление КД РР1, РР2, ПС, ЭТ, ФО, РЭ по шаблонам, разработанным по ГОСТ 2.608-78, ГОСТ 2.601-2006, ГОСТ 2.106-96 и др. на предприятии для текстового процессора Microsoft Word версии 2013. В качестве изделия, сведения о котором подготавливаются в диалоговом окне (рис. 4), произвольно выбрана бортовая графическая система БГС-945, в состав которой входят различные сборочные единицы. Структура БГС-945 соответствует изделию, разработанному в соответствии с принципами интегрированной модульной авионики [17–19].

### Выводы

Разработанные инструментальные программные средства позволяют в автоматизированном режиме решить задачу подготовки в текстовом процессоре Microsoft Word КД с кодами РР1, РР2, ПС, ЭТ, ФО, РЭ, содержащих сведения о драгоценных и цветных металлах, присутствующих в элементной базе бортовой радиоэлектронной аппаратуры. Результатом применения инструментального программного средства являются сокращение временных затрат на подготовку отдельных видов документации (до 80 %) и снижение процента ошибок итоговых расчетных работ. Методика оценки эффекта снижения трудоемкости от автоматизации процедур разработки КД сложного изделия приведена в [13]. Конструкторский документ ПС объемом 11 страниц разрабатывался квалифицированным инженером примерно за 32 часа. С использованием разработанного программного средства удалось сделать это за 9 часов. Фрагмент документа, автоматически генерируемого программой, представлен на рисунке (см. [http://www.swsys.ru/uploaded/image/2017\\_3/2017-3-dop/1.jpg](http://www.swsys.ru/uploaded/image/2017_3/2017-3-dop/1.jpg)).

Сопутствующим результатом являются БД компонентов и инструментарий сопровождения базы, разработанные таким образом, чтобы дальнейшее расширение функциональных возможностей ПО и числа элементов в БД производилось без измене-

ния основной программы вычислений. Организованная по описанным принципам БД электрорадиоизделий является общей для подготовки на предприятии автоматизированным способом различных видов КД.

Программное средство разработано и функционирует на базе инструментальной ЭВМ с характеристиками: ASUS K56CB-X0391H, процессор Intel(R) Core(TM) i5-3337U, 4 ядра, тактовая частота 1,8 ГГц, оперативная память 6 Gb под управлением операционной системы Windows 8.1.

### Литература

1. Авдеева М., Чиркин А. Перевод бумажной документации в электронный вид // САПР и графика. 2004. № 1. С. 70–72.
2. Садовников Д., Ноздрин А., Ширяев Н. Система управления технической и проектно-конструкторской документацией // САПР и графика. 2002. № 5. С. 74–77.
3. Кукаренко Е., Молочко Д. Управление потоками знаний в техническом документообороте предприятия // САПР и графика. 2001. № 10. С. 35–37.
4. Бычков И., Ващук Ю. Конструкторская спецификация – информационная основа управления предприятием // САПР и графика. 2001. № 9. С. 90–95.
5. Riley Kathryn. Language and communication: essential concepts for user interface and documentation design. IEEE transactions on professional communication, 2002, vol. 45, no. 1, p. 54.
6. Гатчин Ю.А., Жаринов И.О., Жаринов О.О. Архитектура программного обеспечения автоматизированного рабочего места разработчика бортового авиационного оборудования // Науч.-технич. вестн. информ. технологий, механики и оптики. 2012. № 2. С. 140–141.
7. Bogatyrev V.A., Parshutina S.A. Redundant distribution of requests through the network by transferring them over multiple paths. Communications in Computer and Information Science, 2016, vol. 601, pp. 199–207.
8. Парамонов П.П., Гатчин Ю.А., Жаринов И.О., Жаринов О.О., Дейко М.С. Принципы построения отраслевой системы автоматизированного проектирования в авиационном приборостроении // Науч.-технич. вестн. информ. технологий, механики и оптики. 2012. № 6. С. 111–117.
9. Bogatyrev V.A., Bogatyrev A.V. Functional reliability of a realtime redundant computational process in cluster architecture systems. Automatic Control and Computer Sciences, 2015, vol. 49, no. 1, pp. 46–56.
10. Благоданов С.А., Уткин С.Б., Батова С.В., Коновалов П.В. Опыт применения технологии эмуляции процессов при разработке компонентов программного обеспечения авиационных систем // Программная инженерия. 2015. № 8. С. 18–25.
11. Kostishin M.O., Shukalov A.V., Zharinov I.O., Zharinov O.O., Ershov A.N. Colorimetric features of design and production of aircraft display systems in the product lifecycle. Indian Jour. of Sc. and Tech. 2016, vol. 9, no. 29, Art. 99451.
12. Bogatyrev V.A., Parshutina S.A. Efficiency of redundant multipath transmission of requests through the net-work to destination servers. Communications in Computer and Information Science. 2016, vol. 678, pp. 290–301.
13. Жаринов И.О., Жаринов О.О., Шек-Иовсепянц Р.А., Суслов В.Д. Оценка снижения трудоемкости подготовки конструкторской документации с использованием CALS-технологий в приборостроении // Науч.-технич. вестн. информ. технологий, механики и оптики. 2012. № 4. С. 151–153.
14. Парамонов П.П., Гатчин Ю.А., Жаринов И.О., Жаринов О.О., Шек-Иовсепянц Р.А. Организация планирования деятельности приборостроительного предприятия при проведении научно-исследовательских и опытно-конструкторских работ в авиационной отрасли // Науч.-технич. вестн. информ. технологий, механики и оптики. 2012. № 5. С. 153–155.
15. Bogatyrev V.A. Fault Tolerance of Clusters Configurations with Direct Connection of Storage Devices. Automatic Control and Computer Sciences, 2011, vol. 45, no. 6, pp. 330–337.

16. Bogatyrev V.A., Parshutina S.A., Poptcova N.A., Bogatyrev A.V. Efficiency of redundant service with de-struction of expired and irrelevant request copies in real-time clusters. *Communications in Computer and Information Science*, 2016, vol. 678, pp. 337–348.

17. Гатчин Ю.А., Жаринов И.О. Основы проектирования вычислительных систем интегрированной модульной авионики: монография. М.: Машиностроение, 2010. 224 с.

18. Богатырев В.А., Богатырев С.В. Объединение резервированных серверов в кластеры высоконадежной компьютерной системы // *Информационные технологии*. 2009. № 6. С. 41–47.

19. Богатырев В.А. Комбинаторно-вероятностная оценка надежности и отказоустойчивости кластерных систем // *Приборы и системы. Управление, контроль, диагностика*. 2006. № 6. С. 21–26.

Software &amp; Systems

DOI: 10.15827/0236-235X.119.517-523

Received 28.02.17

2017, vol. 30, no. 3, pp. 517–523

### AUTOMATED GENERATION OF ACCOUNTING DOCUMENTS USING RADIOELECTRONIC COMPONENTS DATABASE

A.V. Guryanov<sup>1</sup>, Director General, postmaster@elavt.spb.ru

P.V. Kononov<sup>1</sup>, Ph.D. (Engineering), Engineer, olkesomewhere@gmail.com

A.V. Shukalov<sup>2</sup>, Ph.D. (Engineering), Associate Professor, aviation78@mail.ru

I.O. Zharinov<sup>2</sup>, Dr.Sc. (Engineering), Associate Professor, Head of Chair, igor\_rabota@pisem.net

S.A. Leonovets<sup>2</sup>, Postgraduate Student, ser2694@ya.ru

<sup>1</sup> Design Bureau “Electroavtomatika”, Marshala Govorova Str. 40, St. Petersburg, 198095, Russian Federation

<sup>2</sup> The National Research University of Information Technologies, Mechanics and Optics, Kronverksky Ave. 49, St. Petersburg, 197101, Russian Federation

**Abstract.** The paper considers a tool software intended to support solving tasks of automation of forming of design documentation containing data on non-ferrous and precious metals in radioelements, which are a part of a product. In order to store information on electroradio products and on the amount of valuable metals in them, there is a specially designed database. This base is SQL-compatible and consists of four interconnected tables. The user can place it both on the local computer, and in a network. It contains information that is necessary for calculation works and allows introduction of data on new components. Such function is realized programmatically by the procedure providing manual and automatic addition of information in the database. The manual mode allows bringing elements serially and checks correctness of the entered data. In the automatic mode, information is read out from the XML file, and elements are added by groups, which are marked about the content of valuable metals in them. There goes a check of the added elements. A separate program window shows the resultant project of a product. It keeps all modules (assembly units), which are a part of a product, in a tree form in a hierarchical order.

Calculation of content of non-ferrous and precious metals for each metal separately is carried out for the finished project. There also should be design documents “settlement works”. Templates for these documents are in accordance with state standard 2.608-78, 2.601-2006, 2.106-96.

**Keywords:** design documentation, automation, text document, generation.

### References

1. Avdeeva M., Chirkin A. Conversion of paper documents into electronic format. *SAPR i grafika* [CAD and Graphics]. 2004, no. 1, pp. 70–72 (in Russ.).
2. Sadovnikov D., Nozdrin A., Shiryaev N. A control system for technical and design documentation. *SAPR i grafika* [CAD and Graphics]. 2002, no. 5, pp. 74–77 (in Russ.).
3. Kukarenko E., Molochko D. Managing knowledge flows in enterprise technical document workflow. *SAPR i grafika* [CAD and Graphics]. 2001, no. 10, pp. 35–37 (in Russ.).
4. Bychkov I., Vashchuk Yu. Design specification is the information basis of enterprise management. *SAPR i grafika* [CAD and Graphics]. 2001, no. 9, pp. 90–95 (in Russ.).
5. Riley Kathryn. Language and communication: essential concepts for user interface and documentation design. *IEEE Trans. on Professional Communication*. 2002, vol. 45, no. 1, p. 54.
6. Gatchin Yu.A., Zharinov I.O., Zharinov O.O. Software architecture for the automated workplace of the onboard aviation equipment developer. *Nauch.-tekhnich. vestn. inform. tekhnology, mekhaniki i optiki* [Scientific and Technical Journal of Information Technologies, Mechanics and Optics]. 2012, no. 2 (78), pp. 140–141 (in Russ.).
7. Bogatyrev V.A., Parshutina S.A. Redundant distribution of requests through the network by transferring them over multiple paths. *Communications in Computer and Information Science*. 2016, vol. 601, pp. 199–207.
8. Paramonov P.P., Gatchin Yu.A., Zharinov I.O., Zharinov O.O., Deyko M.S. Principles of branch system creation for the automated design in aviation instrumentation. *Nauch.-tekhnich. vestn. inform. tekhnology, mekhaniki i optiki* [Scientific and Technical Journal of Information Technologies, Mechanics and Optics]. 2012, no. 6 (82), pp. 111–117 (in Russ.).
9. Bogatyrev V.A., Bogatyrev A.V. Functional reliability of a realtime redundant computational process in cluster architecture systems. *Automatic Control and Computer Sciences*. 2015, vol. 49, no. 1, pp. 46–56.

10. Blagonravov S.A., Utkin S.B., Batova S.V., Konovalov P.V. Using emulation technics in the development of avionics software components. *Programmnaya inzheneriya* [Software Engineering]. 2015, no. 8, pp. 18–25 (in Russ.).
11. Kostishin M.O., Shukalov A.V., Zharinov I.O., Zharinov O.O., Ershov A.N. Colorimetric features of design and production of aircraft display systems in the product lifecycle. *Indian Jour. of Science and Technology*. 2016, vol. 9, no. 29, art. 99451.
12. Bogatyrev V.A., Parshutina S.A. Efficiency of redundant multipath transmission of requests through the network to destination servers. *Communications in Computer and Information Science*. 2016, vol. 678, pp. 290–301.
13. Zharinov I.O., Zharinov O.O., Shek-Iovsepyants R.A., Suslov V.D. Robustness drop estimation of design documentation preparation by calcs-technologies instrument. *Nauch.-tekhnich. vestn. inform. tekhnology, mekhaniki i optiki* [Scientific and Technical Journal of Information Technologies, Mechanics and Optics]. 2012, no. 4 (80), pp. 151–153 (in Russ.).
14. Paramonov P.P., Gatchin Yu.A., Zharinov I.O., Zharinov O.O., Shek-Iovsepyants R.A. Organization and planning of the instrument-making enterprise activities at carrying out of scientific-research and experimental-design works in aviation. *Nauch.-tekhnich. vestn. inform. tekhnology, mekhaniki i optiki* [Scientific and Technical Journal of Information Technologies, Mechanics and Optics]. 2012, no. 6 (82), pp. 152–154 (in Russ.).
15. Bogatyrev V.A. Fault Tolerance of Clusters Configurations with Direct Connection of Storage Devices. *Automatic Control and Computer Sciences*. 2011, vol. 45, no. 6, pp. 330–337.
16. Bogatyrev V.A., Parshutina S.A., Poptcova N.A., Bogatyrev A.V. Efficiency of redundant service with destruction of expired and irrelevant request copies in real-time clusters. *Communications in Computer and Information Science*. 2016, vol. 678, pp. 337–348.
17. Gatchin Yu.A., Zharinov I.O. *Osnovy proektirovaniya vychislitelnykh sistem integrirovannoy modulnoy avioniki* [Fundamentals of computer systems design for integrated modular avionics]. Monograph. Moscow, Mashinostroenie Publ., 2010, 224 p.
18. Bogatyrev V.A., Bogatyrev S.V. Association reservation servers in clusters highly reliable computersystem. *Informatsionnye tekhnologii* [Information Technologies]. 2009, no. 6, pp. 41–47 (in Russ.).
19. Bogatyrev V.A. The combinatorial stochastic method of reliability evaluation and fault tolerance for networks with cluster architecture. *Pribory i sistemy. Upravlenie, kontrol, diagnostika* [Instruments and Systems: Monitoring, Control, and Diagnostics]. 2006, no. 6, pp. 21–26 (in Russ.).

## Примеры библиографического описания статьи

1. Гурьянов А.В., Коновалов П.В., Шукалов А.В., Жаринов И.О., Леоновец С.А. Автоматизация процесса формирования учетной документации с использованием базы данных радиоэлектронных компонентов // Программные продукты и системы. 2017. Т. 30. № 3. С. 517–523. DOI: 10.15827/0236-235X.119.517-523.
2. Guryanov A.V., Konovalov P.V., Shukalov A.V., Zharinov I.O., Leonovets S.A. Automated generation of accounting documents using radioelectronic components database. *Programmnye produkty i sistemy* [Software & Systems]. 2017, vol. 30, no. 3, pp. 517–523 (in Russ.). DOI: 10.15827/0236-235X.119.517-523.

УДК 004.42

DOI: 10.15827/0236-235X.119.524-528

Дата подачи статьи: 19.05.17

2017. Т. 30. № 3. С. 524–528

## **ОПТИМИЗАЦИЯ ЗАДАЧИ ОБ ОПРЕДЕЛЕНИИ КОНФЛИКТОВ С ОПАСНЫМИ ЗОНАМИ ДВИЖЕНИЯ ЛЕТАТЕЛЬНЫХ АППАРАТОВ ДЛЯ ВЫПОЛНЕНИЯ НА INTEL XEON PHI**

*А.А. Рыбаков, к.ф.-м.н., ведущий научный сотрудник, rybakov@jscc.ru  
(Межведомственный суперкомпьютерный центр РАН – филиал ФНЦ НИИСИ РАН,  
Ленинский просп., 32а, г. Москва, 119334, Россия)*

Во время полета летательный аппарат генерирует вихревой спутный след. Данный след может представлять опасность для других участников воздушного движения, особенно в зонах, в которых одновременно находится большое количество летательных аппаратов. Поэтому задача моделирования вихревых следов и определения конфликтов с ними является крайне важной и востребованной.

В данной статье вихревой след рассматривается как совокупность опасных зон отдельных отрезков траектории движения. Опасная зона отдельного отрезка траектории представлена пучком сфер – множеством сфер с центрами на данном отрезке. Далее рассматривается прямой математический метод определения пересечения прямой и пучка сфер. Прямая представляет мгновенную траекторию движения собственного летательного аппарата, тогда как множество пучков сфер формирует полную опасную зону полета.

Так как количество отрезков траекторий в опасной зоне достаточно велико, целесообразно использование ускорителей для осуществления вычислений. В статье рассматриваются применение Intel Xeon Phi для данной задачи и использование специального приема ручной оптимизации исходного кода для того, чтобы добиться применения векторизации кода. Такой подход позволяет добиться пятикратного ускорения решения задачи при обходе около  $10^7$  отрезков движения внутри опасной зоны.

**Ключевые слова:** вихревой след, опасная зона движения летательного аппарата, определение конфликтов, Intel Xeon Phi, векторизация.

Во время движения летательный аппарат (ЛА) создает вихревые возмущения, которые остаются вдоль траектории его полета. Совокупность этих возмущений формирует турбулентный (вихревой) след ЛА [1], который со временем эволюционирует и в конечном итоге разрушается [2]. Пока вихревой след существует, он может представлять опасность для других участников воздушного движения при попадании их в данный след [3]. Особенно это критично в местах скопления ЛА, где на небольшом участке пространства может находиться сразу большое количество вихревых следов (аэродромы) [4]. Поэтому необходимо осуществлять постоянный мониторинг вихревых следов вплоть до их разрушения и оценивать опасность их влияния на собственный ЛА [5]. В данной статье рассматривается задача определения потенциальных конфликтов с множеством вихревых следов ЛА. Приводится алгоритм решения данной задачи, а также рассматривается его реализация на вычислителе Intel Xeon Phi.

### **Постановка задачи**

Траекторию движения ЛА зададим набором точек в пространстве, последовательно соединенных отрезками. Будем считать, что в данных точках известны характеристики ЛА, в том числе и интенсивность создаваемых им вихревых возмущений, а также то, что опасная зона одной конкретной точки представляет собой шар с центром в этой точке. Точку в трехмерном пространстве зададим радиус-вектором, направленным из начала координат в эту

точку, а сферу – ее центром и радиусом следующим образом:  $S(\bar{C}, R) = \left\{ \bar{P} \mid |\bar{P} - \bar{C}|^2 = R^2 \right\}$ .

Примем, что опасная зона отрезка является объединением опасных зон всех его точек, где радиус опасной зоны точки отрезка меняется линейно от одного его конца до другого. Для задания опасной зоны отрезка введем понятие пучка сфер, опирающегося на две различные сферы. Пусть заданы две сферы:  $S_0(\bar{C}_0, R_0)$  и  $S_1(\bar{C}_1, R_1)$ .

Тогда пучком сфер, опирающимся на них, назовем такое множество сфер, центры и радиусы которых определяются следующим образом:

$$\begin{cases} \bar{C}(\alpha) = \bar{C}_0 + \alpha(\bar{C}_1 - \bar{C}_0) = \bar{C}_0 + \alpha\Delta\bar{C}, \\ R(\alpha) = R_0 + \alpha(R_1 - R_0) = R_0 + \alpha\Delta R, \end{cases}$$

где параметр  $\alpha$  изменяется от 0 до 1. Нетрудно заметить, что опасная зона отрезка представляет собой область, ограниченную опасными зонами его концов, являющимися шарами, и множеством общих внешних касательных данных шаров. Кроме того, отметим, что опасная зона отрезка является выпуклым телом. Опасную зону всей траектории движения ЛА будем определять как объединение опасных зон всех отрезков его движения

На рисунке 1 показан пример траектории движения ЛА, закрашенными кругами обозначены опасные зоны точек, в которых известны характеристики ЛА, также очерчена опасная зона всей траектории движения.

Итак, в задаче обнаружения возможных конфликтов с опасными зонами траекторий движения

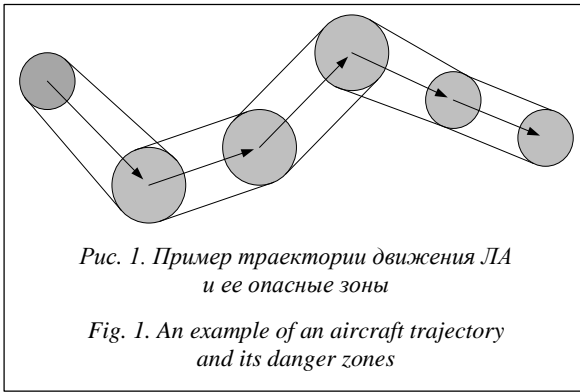


Рис. 1. Пример траектории движения ЛА и ее опасные зоны

Fig. 1. An example of an aircraft trajectory and its danger zones

в местах скопления ЛА задаются множество траекторий движения различных ЛА, а также положение и вектор мгновенной скорости собственного ЛА, для которого нужно определить конфликт. Положение и вектор скорости собственного ЛА определяют луч, состоящий из множества точек  $\vec{P}(t) = \vec{P}_0 + t\vec{V}$  при  $t \geq 0$ , для которого нужно определить, пересекает ли он опасную зону траектории движения хотя бы одного ЛА. В случае конфликта требуется найти точку первого пересечения для определения уровня угрозы, то есть времени, оставшегося до пересечения опасной зоны.

### Решение

Для решения поставленной задачи необходимо решить задачу пересечения прямой с пучком сфер. Сначала рассмотрим пересечение прямой и отдельной сферы. Без ограничения общности будем считать, что собственный ЛА находится в начале координат (этого всегда можно добиться с помощью параллельного переноса). Пересечение прямой со сферой находится из следующей системы уравнений:

$$\begin{cases} \vec{P} = t\vec{V}, \\ |\vec{P} - \vec{C}|^2 = R^2. \end{cases}$$

Данная система уравнений преобразуется в квадратное уравнение относительно  $t$ , дискриминант которого равен  $4((\vec{C} \cdot \vec{V})^2 - |\vec{V}|^2 (|\vec{C}|^2 - R^2))$ .

Так как значение дискриминанта может быть вычислено непосредственно, дальнейшее нахождение точек пересечения не представляет труда (если он неотрицательный).

Перейдем к поиску точек пересечения прямой и пучка сфер. Для каждой конкретной сферы из пучка можно выписать условие наличия точек пересечения:

$$(\vec{C}(\alpha) \cdot \vec{V})^2 - |\vec{V}|^2 (|\vec{C}(\alpha)|^2 - R(\alpha)^2) \geq 0.$$

Подставив конкретные выражения для центра и радиуса сферы и выполнив необходимые преобразования, получим квадратное неравенство относительно параметра  $\alpha$ :  $A_2\alpha^2 + 2A_1\alpha + A_0 \geq 0$ , где

$$\begin{cases} A_2 = (\vec{\Delta C} \cdot \vec{V})^2 + |\vec{V}|^2 (\Delta R^2 - |\vec{\Delta C}|^2), \\ A_1 = (\vec{C}_0 \cdot \vec{V})(\vec{\Delta C} \cdot \vec{V}) + |\vec{V}|^2 (R_0 \Delta R - (\vec{C}_0 \cdot \vec{\Delta C})), \\ A_0 = (\vec{C}_0 \cdot \vec{V})^2 + |\vec{V}|^2 (R_0^2 - |\vec{C}_0|^2). \end{cases}$$

Полученное неравенство можно решить относительно  $\alpha$ , так как все коэффициенты вычисляются непосредственно. Нас интересует решение данного неравенства только на отрезке  $[0, 1]$ , оно представляет собой либо пустое множество, либо отрезок.

Если на отрезке  $[0, 1]$  приведенное неравенство не имеет решения, прямая не пересекает пучок сфер. Допустим, на отрезке  $[0, 1]$  данное неравенство имеет решение  $[\alpha_0, \alpha_1] \subseteq [0, 1]$ .

Для произвольного значения  $\alpha$  из данного отрезка корни уравнения

$$|\vec{V}|^2 t^2 - 2(\vec{C}(\alpha) \cdot \vec{V})t + (|\vec{C}(\alpha)|^2 - R(\alpha)^2) = 0$$

существуют и выражаются следующим образом:

$$t_{1,2}(\alpha) = \frac{(\vec{C}(\alpha) \cdot \vec{V}) \pm \sqrt{A_2\alpha^2 + 2A_1\alpha + A_0}}{|\vec{V}|^2}.$$

Искомые точки пересечения прямой с пучком сфер соответствуют минимальному и максимальному значениям параметра  $t(\alpha)$ , достигаемым на отрезке  $[\alpha_0, \alpha_1]$ . Максимальное и минимальное значения  $t(\alpha)$  могут достигаться либо на концах отрезка, либо в точках локального экстремума. Для точки локального экстремума должно быть выполнено соотношение  $t'_{1,2}(\alpha) = 0$  или в явном виде

$$(\vec{\Delta C} \cdot \vec{V}) \pm \frac{A_2\alpha + A_1}{\sqrt{A_2\alpha^2 + 2A_1\alpha + A_0}} = 0,$$

$$\begin{cases} q(A_2\alpha^2 + 2A_1\alpha + A_0) = (A_2\alpha + A_1)^2, \\ q = (\vec{\Delta C} \cdot \vec{V})^2. \end{cases}$$

После преобразования получим следующее квадратное уравнение относительно  $\alpha$ :

$$A_2(q - A_2)\alpha^2 + 2A_1(q - A_2)\alpha + (qA_0 - A_1^2) = 0,$$

решая которое, найдем потенциальные точки локальных экстремумов  $\alpha_1, \alpha_2$ .

Данные точки нужно учитывать только в случае их попадания в отрезок  $[\alpha_0, \alpha_1]$ . В общем случае значения находятся по четырем значениям  $\alpha$ :

$$\begin{cases} t_1 = \min \{t_1(\alpha_0), t_1(\alpha_1), t_1(\alpha_1), t_1(\alpha_2)\}, \\ t_2 = \max \{t_2(\alpha_0), t_2(\alpha_1), t_2(\alpha_1), t_2(\alpha_2)\}. \end{cases}$$

По найденным значениям непосредственно находятся точки пересечения прямой с пучком сфер. Для задачи, рассматриваемой в статье, необходимо получить ответ о пересечении с пучком сфер не на всей прямой, а на луче, соответствующем неотрицательным  $t$ . Конечно, возможен случай, когда собственный ЛА уже находится в опасной зоне и пересечение с данной опасной зоной



произошло при  $t < 0$ , но данный случай легко сводится к общему.

### Особенности решения

Рассмотрим подробнее решение представленного ранее квадратного неравенства  $A_2\alpha^2 + 2A_1\alpha + A_0 \geq 0$ .

Утверждалось, что решением данного неравенства на отрезке  $[0, 1]$  является либо пустое множество, либо отрезок. Единственным потенциально возможным вариантом, когда это не так, является вариант, при котором  $A_2 > 0$  и оба значения  $\alpha$ , при которых неравенство обращается в равенство, находятся внутри отрезка  $[0, 1]$ . Это означает, что прямая пересекает сферы  $S_0$  и  $S_1$ , однако не пересекает хотя бы одну другую сферу из пучка. Но это невозможно в силу выпуклости пучка сфер, а значит, этот случай исключен.

Теперь рассмотрим подробнее случай  $A_2 < 0$ :

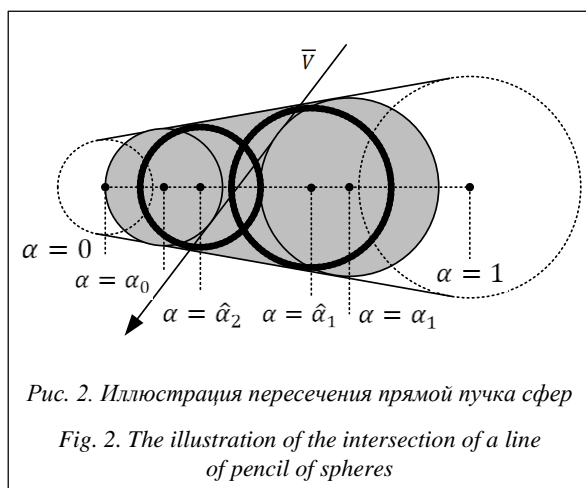
$$(\overline{\Delta C} \cdot \overline{V})^2 + |\overline{V}|^2 (\Delta R^2 - |\overline{\Delta C}|^2) < 0.$$

Раскрыв скалярное произведение векторов и выполнив необходимые преобразования, получим следующее условие на угол между скоростью собственного ЛА и отрезком, содержащим центры сфер пучка:  $|\sin(\overline{\Delta C}, \overline{V})| > \frac{|\Delta R|}{|\overline{\Delta C}|}$ , где  $|\Delta R|/|\overline{\Delta C}|$

представляет собой синус угла раствора пучка сфер. Заметим, что данный угол раствора всегда очень мал, так как характеристики ЛА меняются медленно во время движения и  $|\Delta R|$  близко к нулю. Таким образом, случай  $A_2 < 0$  является наиболее частым и данное условие выполняется в подавляющем большинстве случаев (рис. 2).

### Программная реализация и оптимизация

Опишем программную реализацию функции, которая по набору пучков сфер выдает координаты точек пересечения с ними. Для простоты будем



считать, что задана только одна траектория движения ЛА. Входными данными функции являются четыре массива: три массива с координатами точек, последовательно соединенных отрезками  $(x, y, z)$ , и один массив с радиусами опасных зон этих точек  $(r)$ . Также на вход подается вектор скорости собственного ЛА. Тогда алгоритм поиска точек пересечения с одним пучком сфер можно записать в виде следующей последовательности шагов.

1. Вычислить коэффициенты  $A_2, A_1, A_0$ , фигурирующие в квадратном неравенстве.

2. Если  $A_2 > 0$ , решить квадратное неравенство на отрезке  $[0, 1]$  и перейти к п. 5, иначе перейти к п. 3.

3. Если  $A_2 < 0$ , решить квадратное неравенство на отрезке  $[0, 1]$  и перейти к п. 5, иначе перейти к п. 4.

4. Так как  $A_2 = 0$ , решить линейное неравенство на отрезке  $[0, 1]$  и перейти к п. 5.

5. Если решение неравенства из пп. 2–4 на отрезке  $[0, 1]$  пусто, перейти к следующему пучку сфер, иначе перейти к п. 6.

6. На отрезке  $[\alpha_0, \alpha_1]$ , найденном в пп. 2–4, найти точки, в которых значения  $t_{1,2}$  достигают своих экстремумов.

7. По значениям  $\alpha_0, \alpha_1, \alpha_1, \alpha_2$  в общем случае найти моменты времени пересечения прямой с пучком сфер.

Данный алгоритм был реализован на языке C, а исполняемые коды опробованы на ускорителях Intel Xeon Phi 7110X [6], которые входят в состав находящегося в МСЦ РАН вычислительного комплекса МВС-10П [7]. Однако полученные результаты выявили неэффективное использование возможностей ускорителя. Основной проблемой стало наличие в алгоритме довольно сложного управления. В частности, разбор различных случаев при решении квадратного неравенства и при нахождении точек экстремумов  $\alpha_1, \alpha_2$ . Такое сложное управление привело к невозможности эффективного применения векторизации в цикле обработки пучков сфер [8].

Так как оптимизирующий компилятор [9] из состава Intel Parallel Studio, который использовался для компиляции кода, не смог подготовить эффективный исполняемый код, для устранения недостатков в эффективности было принято решение модифицировать код таким образом, чтобы удалить из цикла маловероятные ветви исполнения [10]. Как было показано выше, в подавляющем большинстве случаев при решении квадратного неравенства выполняется условие  $A_2 < 0$ . Однако даже в этом случае решение квадратного неравенства на отрезке  $[0, 1]$  почти всегда представляет собой пустое множество. Переписав условие отсутствия пересечений с пучком сфер, в данном случае получим выражение, которое истинно с вероятностью, близкой к единице:



$$(A_2 < 0) \wedge (m > 0) \wedge (A_1^2 - A_2 A_0 < m^2),$$

где  $m = \max(A_1 + A_2, -A_1)$ . Если обозначить это вероятное условие через *cond*, можно переписать основную цикл обработки пучков сфер в следующем виде:

```
for (<цикл по всем пучкам сфер>)
{
    <вычисления>

    if (cond)
    {
        continue; // prob. ~100%
    }
    else
    {
        <маловероятные вычисления>
    }
}
```

Данный цикл можно расщепить на два цикла, используя дополнительный массив, содержащий значения условий *cond*. При этом получим следующий результат:

```
for (<цикл по всем пучкам сфер>)
{
    <вычисления>
    tmp[i] = cond;
}

for (<цикл по всем пучкам сфер>)
{
    if (!tmp[i])
    {
        <маловероятные вычисления>
    }
}
```

В оптимизированном варианте основная часть вычислений приходится на первый цикл, при этом он не содержит сложного управления, а значит, подходит для векторизации. Второй цикл, конечно, неэффективно исполняется на ускорителе, но его

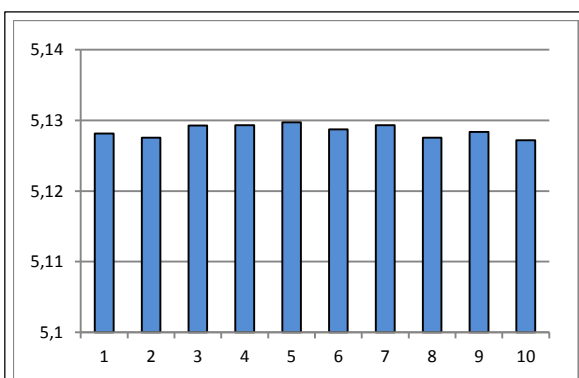


Рис. 3. Результаты ускорения оптимизированного варианта определения конфликта с опасными зонами

Fig. 3. The results of the acceleration of the optimized version of dangerous zones conflict definition

время исполнения настолько мало, что им можно пренебречь.

Были проведены сравнительные запуски с моделированием случайных траекторий движения ЛА для неоптимизированного и оптимизированного вариантов и получены стабильные результаты, которые демонстрируют ускорение вычислений за счет векторизации более чем в 5 раз по сравнению с неоптимизированным вариантом.

На рисунке 3 представлены результаты запусков. По горизонтальной оси отмечены 10 запусков, каждый из которых состоял из обработки  $10^7$  пучков сфер. По вертикали указано ускорение, полученное при использовании векторизованного варианта по сравнению с исходным.

### Заключение

При решении реальных практических задач с применением современных аппаратных вычислительных ресурсов сложно ожидать эффективного исполнения программы даже при использовании высококачественного компилятора, способного проводить оптимизацию кода. Зачастую задача требует вмешательства исследователя, способного определить особенности алгоритма и должным образом использовать их. Часто небольшие изменения логики вычислений, не влияющие на результат, но использующие особенности алгоритма, способны привести к кратному повышению производительности.

### Литература

1. Аубакиров Т.О., Желанников А.И., Иванов П.Е., Ништ М.И. Спутные следы и их воздействие на летательные аппараты. Моделирование на ЭВМ. Алматы: Гылым, 1999. 280 с.
2. Вышинский В.В., Судаков Г.Г. Вихревой след самолета в турбулентной атмосфере // Тр. ЦАГИ. 2006. Вып. 2667. 155 с.
3. Бабкин В.И., Белоцерковский А.С., Турчак Л.И., Баранов Н.А., Замятин А.И., Каневский М.И., Морозов В.В., Паскунов И.В., Чижов Н.Ю. Системы обеспечения вихревой безопасности полетов летательных аппаратов. М.: Наука, 2008. 373 с.
4. Бурлуцкий С.Г. Вопросы обеспечения вихревой безопасности аэропортов // Системный анализ и логистика. 2014. № 11. С. 37–40.
5. Кудрявцев Н.С. Оценка эффективности систем обеспечения вихревой безопасности полетов // Естественные и математические науки в современном мире. 2016. № 11. С. 31–37.
6. Jeffers J., Reinders J. Intel Xeon Phi coprocessor high performance programming. Morgan Kaufmann Publ., 2013, 432 p.
7. Описание интерфейса пользователя, предназначенного для работы с интеловской гибридной архитектурой суперЭВМ... URL: <http://www.jscc.ru/informat/MVS-10PInter.pdf> (дата обращения: 15.05.2017).
8. Шабанов Б.М., Телегин П.Н., Аладышев О.С. Особенности использования многоядерных процессоров // Программные продукты и системы. 2008. № 2. С. 7–9.
9. Aho A.V., Lam M.S., Sethi R., Ullman J.D. Compilers: principles, techniques, and tools (2nd Ed.). Pearson Education Inc. Publ., 2006, 1038 p.
10. Allen R., Kennedy K. Optimizing compilers for modern architectures. Morgan Kaufmann Publ., 2001, 790 p.

# OPTIMIZATION OF THE PROBLEM OF CONFLICT DETECTION WITH DANGEROUS AIRCRAFT MOVEMENT AREAS TO EXECUTE ON INTEL XEON PHI

A.A. Rybakov<sup>1</sup>, Ph.D. (Physics and Mathematics), Leading Researcher, rybakov@jscc.ru

<sup>1</sup> Joint Supercomputer Center of RAS, Lenin Ave. 32a, Moscow, 119991, Russian Federation

**Abstract.** An aircraft generates a wake vortex during its flight. This wake might be dangerous for other air traffic participants, especially in zones with a large number of aircrafts. So, the problem of vortex wake modeling and conflicts detection is very important and in demand.

The article considers a vortex wake as a set of dangerous zones in separate traffic segments. A dangerous zone of separate traffic segment is a pencil of spheres – a set of spheres with centers on this segment. Next, we consider the direct mathematical method of detecting an intersection of a line and a pencil of spheres. The line is an immediate trajectory of own aircraft, while a pencil of spheres forms the whole dangerous flight zone.

Since the number of traffic segments in a dangerous zone is large enough, it is appropriate to use accelerators in such calculations. The paper considers using Intel Xeon Phi coprocessors for this task and applying some manual sourcecode optimization to enforce execution code vectorization. This approach leads to five-time acceleration of calculation for 10<sup>7</sup> traffic segments in a dangerous zone.

**Keywords:** vortex wake, aircraft trajectory dangerous zone, conflicts detection, Intel Xeon Phi, vectorization.

## References

1. Aubakirov T.O., Zhelannikov A.I., Ivanov P.E., Nisht M.I. *Sputnye sledy i ikh vozdeystvie na letatelnye apparaty. Modelirovanie na EVM* [Companion Tracks and Their Impact on an Aircraft. Computer Simulation]. Almaty, 1999, 280 p.
2. Vyshinsky V.V., Sudakov G.G. Aircraft vortex trail in turbulent atmosphere. *Trudy TsAGI* [TsAGI Science Jour.]. 2006, iss. 2667, 155 p. (in Russ.).
3. Babkin V.I., Belotserkovsky A.S., Turchak L.I., Baranov N.A., Zamyatin A.I., Kanevsky M.I., Morozov V.V., Pasekunov I.V., Chizhov N.Yu. *Sistemy obespecheniya vikhrevoy bezopasnosti poletov letatelnykh apparatov* [Vortex Safety Systems for Aircrafts]. Moscow, Nauka Publ., 2008, 373 p.
4. Burlutsky S.G. Vortex security for airports. *Sistemny analiz i logistika* [Int. Jour. of System Analysis and Logistics Sciences]. 2014, iss. 11, pp. 37–40 (in Russ.).
5. Kudryavtsev N.S. Evaluation of vortex safety systems effectiveness. *Estestvennye i matematicheskie nauki v sovremen-nom mire* [Natural and Mathematical Sciences in the Modern World]. 2016, no. 11 (46), pp. 31–37 (in Russ.).
6. Jeffers J., Reinders J. *Intel Xeon Phi Coprocessor High Performance Programming*. Morgan Kaufmann Publ., 2013, 432 p.
7. *Opisanie interfeysa polzovatelya, prednaznachennogo dlya raboty s intelovskoy gibridnoy arkhitekturoy superEVM...* [A Description of a User Interface Designed to Work with the Intel Hybrid Supercomputer (SC), where Intel Xeon Phi Coprocessors are Used Together with Intel Xeon Processors]. Available at: <http://www.jscc.ru/informat/MVS-10PInter.pdf> (accessed May 15, 2017).
8. Shabanov B.M., Telegin P.N., Aladyshev O.S. Features of using multi-core processors. *Programmnye produkty i sistemy* [Software & Systems]. 2008, no. 2, pp. 7–9 (in Russ.).
9. Aho A.V., Lam M.S., Sethi R., Ullman J.D. *Compilers: Principles, Techniques, and Tools*. 2nd ed., Pearson Education Inc. Publ., 2006, 1038 p.
10. Allen R., Kennedy K. *Optimizing Compilers for Modern Architectures*. Morgan Kaufmann Publ., 2001, 790 p.

## Примеры библиографического описания статьи

1. Рыбаков А.А. Оптимизация задачи об определении конфликтов с опасными зонами движения летательных аппаратов для выполнения на Intel Xeon Phi // Программные продукты и системы. 2017. Т. 30. № 3. С. 524–528. DOI: 10.15827/0236-235X.119.524-528.
2. Rybakov A.A. Optimization of the problem of conflict detection with dangerous aircraft movement areas to execute on Intel Xeon Phi. *Programmnye produkty i sistemy* [Software & Systems]. 2017, vol. 30, no. 3, pp. 524–528 (in Russ.). DOI: 10.15827/0236-235X.119.524-528.

УДК 681.518.22

DOI: 10.15827/0236-235X.119.529-536

Дата подачи статьи: 06.04.17

2017. Т. 30. № 3. С. 529–536

## **АЛГОРИТМЫ ИНФОРМАЦИОННОЙ ОЦЕНКИ СОВМЕСТИМОСТИ СРЕДСТВ МОБИЛЬНОЙ СВЯЗИ И СТАНЦИЙ ВОЗДУШНОЙ РАДИОНАВИГАЦИОННОЙ СЛУЖБЫ**

*О.Ю. Вавулов, аспирант, ovavulov@mail.ru*

*(Российский научно-технический центр информации по стандартизации, метрологии и оценке соответствия, Нахимовский просп., 31, корп. 2, г. Москва, 117418, Россия);*

*А.В. Сухов, д.т.н., профессор, avs57@mail.ru*

*(Московский авиационный институт (национальный исследовательский университет), Волоколамское шоссе, 4, г. Москва, 125993, Россия);*

*В.Н. Решетников, д.ф.-м.н., профессор, зав. кафедрой, rvp@mail.ru*

*(Московский авиационный институт (национальный исследовательский университет), Волоколамское шоссе, 4, г. Москва, 125993, Россия);*

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

В современном информационном обществе количество пользователей, нуждающихся в качественной, быстрой и непрерывной мобильной связи, неуклонно растет. В связи с этим усиленно разрабатываются и внедряются новые технологии и стандарты качества в этой области (LTE, 4G). При этом важно учитывать также и возможность электромагнитной совместимости со средствами других служб связи, особенно отвечающих за безопасность, например, воздушной радионавигационной службы.

В настоящей работе рассмотрена разработка алгоритма информационной оценки состояния комплекса средств воздушной радионавигационной службы в условиях деструктивного помехового воздействия, оказываемого системами мобильной связи. Математически состояние комплекса средств воздушной радионавигационной службы представлено в информационном пространстве, основанном на информационной мере – энтропии покрытия, которая показывает, насколько реальные характеристики рассматриваемой системы соответствуют своим нормативным значениям.

Предложенный метод опробован для оценки помехового воздействия на реальные российские станции воздушной радионавигационной службы, размещенные в приграничном регионе, приведен пример применения алгоритма для оценки совместимости комплекса средств воздушной радионавигационной службы и реальной сети мобильной связи.

Предложен алгоритм координации с комплексом средств воздушной радионавигационной службы нового частотного присвоения для единичной базовой станции. Указанный алгоритм основан на расчете напряженности поля суммарной помехи в месте размещения станции воздушной радионавигационной службы от всех станций сухопутной подвижной службы и на сравнении рассчитанного значения с предельно допустимым, заданным для конкретного типа станции воздушной радионавигационной службы.

**Ключевые слова:** электромагнитная совместимость, информационная оценка, воздушная радионавигационная служба, мобильная связь, энтропия покрытия, координация.

Международный союз электросвязи (далее – МСЭ) (глобальная организация, иницилирующая и координирующая действия мирового сообщества в области развития телекоммуникаций) в целях обеспечения эффективной эксплуатации средств электросвязи разработал Регламент радиосвязи. Этот документ призван путем распределения различным службам различных частотных диапазонов свести к минимуму возможность их взаимного помехового воздействия. Однако, согласно ему, в ряде случаев один и тот же частотный интервал оказывается распределенным одновременно как сухопутной подвижной службе (СПС), так и воздушной радионавигационной службе (ВРНС).

Вследствие означенных причин встает вопрос о точных методах оценки состояния комплекса средств ВРНС в условиях помехового воздействия от средств СПС, а также при необходимости о координации новых частотных присвоений для станций указанных служб. В настоящее время удовле-

творительных методик и алгоритмов в этой области не существует.

Основным показателем, по которому можно оценить защищенность станции ВРНС, является напряженность поля суммарной помехи, создаваемой средствами СПС в месте нахождения защищаемой станции. Множественная помеха рассчитывается по принципу суперпозиции электромагнитных полей, то есть результирующая мощность является суммой мощностей полей помех, создаваемых в этом месте каждой из станций СПС в отдельности:

$$P_{\text{sum}} = \sum_{i=1}^N P_i, \quad (1)$$

где  $P_i$  – мощность поля помехи, создаваемой в месте размещения станции ВРНС  $i$ -й станцией СПС (Вт);  $P_{\text{sum}}$  – результирующая мощность поля помехи от всех учитываемых станций СПС в месте размещения станции ВРНС (Вт);  $N$  – общее число учитываемых станций СПС.

Запишем принцип суперпозиции (1) для напряженностей поля, выраженных в дБмкВ/м [1]:

$$E_{sum} = 10 \times \lg \left( \sum_{i=1}^N 10^{0,1E_i} \right), \quad (2)$$

где  $E_{sum}$ ,  $E_i$  – соответственно результирующая и создаваемая  $i$ -й станцией СПС напряженности поля, выраженные в дБмкВ/м.

В качестве критерия защиты станций ВРНС [2] принимаются задаваемые разработчиками для каждого типа станций предельно допустимые значения поля суммарной помехи в месте размещения станции, приведенные в Рекомендации МСЭ-R М.1830 «Технические характеристики и критерии защиты систем воздушной радионавигационной службы в полосе частот 645–862 МГц».

Для расчета напряженности поля помехи  $E_i$ , создаваемой одиночной  $i$ -й станцией СПС, необходимо иметь следующие исходные данные:

- расстояние ( $r$ ) между станциями ВРНС и СПС;
- эквивалентная изотропно излучаемая мощность (ЭИИМ)  $P_{MS}$  передатчиков средств СПС: базовых станций (БС) и абонентских терминалов (АТ) ( $P_{BS}$ ,  $P_{ST}$ );
- высота подвеса приемной антенны станции ВРНС ( $h_{ARNS}$ );
- высота подвеса передающей антенны БС в конкретном секторе ( $h_{BS}$ );
- эффективная высота антенны БС по азимуту от БС на станцию ВРНС ( $h_{eff}$ );
- характеристика диаграммы направленности (ДН) антенны БС.

Для оценки эффективности координации средств СПС со станциями ВРНС в полной мере необходимо учитывать не только совокупную мощность поля помех, но и требования, установленные соглашениями между администрациями связи граничащих государств по использованию определенных частотных диапазонов. В указанных соглашениях в основном рассмотрены приведенные выше параметры в виде заданных на их допустимые значения ограничений, определенных с учетом расстояния от средств СПС до государственной границы.

Для комплексной обобщенной оценки результатов координации можем применить информационный подход [3], основанный на использовании информационной меры – энтропии покрытия [4]. Энтропия покрытия  $H_n$  в обобщенной форме характеризует степень соответствия/несоответствия реальных характеристик системы их нормативным значениям [5–7], учитывая возможность их изменения в сторону как уменьшения, так и увеличения. Для описания состояния станции ВРНС в предметной области в условиях множественной помехи от средств СПС в информационной области введем вектор реальных показателей  $R_p$ , компонентами которого будут являться действительные значения

вышеприведенных характеристик. Аналогично вводится вектор нормативных показателей  $R_n$ , составленный из множества требуемых значений.

Тогда в качестве элемента покрытия множества нормативных характеристик множеством реальных введем следующий вектор [8]:

$$\Pi(t) = \begin{pmatrix} \Pi_1 \\ \Pi_2 \end{pmatrix} = \begin{pmatrix} \| [R_p \setminus (R_p \cap R_n(t))] \cup (R_p \cap R_n(t)) \| / \| R_p \cap R_n(t) \| \\ \| R_n(t) \| / (\| R_n(t) \| - r \| R_n(t) \setminus (R_p \cap R_n(t)) \|) \end{pmatrix}, \quad (3)$$

где  $\setminus$  – операция разности множеств;  $\| \cdot \|$  – первая норма;  $r$  – коэффициент приоритета элемента, для которого рассчитывается относительная энтропия покрытия,  $r \in (0; 1)$ , обычно принимается равным в диапазоне от 0,9 до 0,999.

С учетом (3) энтропия покрытия  $H_n$  (бит) в обобщенном смысле определяется выражением

$$H_n = (\log_2 \Pi_1 + i \log_2 \Pi_2), \quad (4)$$

где  $i$  – мнимая единица;  $\Pi_1$ ,  $\Pi_2$  – компоненты вектора элемента покрытия.

Очевидно, что энтропия покрытия – это комплексная величина, при этом ее действительная часть характеризует запас по значениям отдельных показателей по сравнению с нормативными, а мнимая – повышенные значения других характеристик. Условие  $H_n = 0$  соответствует либо идеальному случаю, когда реальные показатели совпадают с требуемыми, либо состоянию компромисса, при котором избыток по одним показателям сбалансирован пониженными значениями других.

В качестве демонстрации применения информационного подхода к анализу электромагнитной совместимости средств ВРНС со средствами СПС с использованием информационной меры – энтропии покрытия – была проведена информационная оценка характеристик БС сети мобильной связи одного из сопредельных государств, развертываемой в непосредственной близости от границы с Россией, на соответствие условиям, приведенным в соглашении между администрациями связи РФ и сопредельного государства по совместному использованию определенного частотного диапазона. Рассматриваемое соглашение призвано в том числе обеспечить защиту средств ВРНС от помех в соответствии с действующими критериями (ограничение на мощность помехи на входе приемника радиозлектронного средства (РЭС) или ограничение на напряженность электромагнитного поля помехи в районе размещения приемной антенны РЭС ВРНС). Однако при координации размещения БС могут возникать ситуации, требующие компромиссных решений, учитывающих особые пожелания администрации связи сопредельного государства. Естественно, что учет этих пожеланий не должен приводить к невыполнению критериев уровня помех для действующих РЭС ВРНС. Превышение ограничений соглашения не обязательно приводит к нарушениям этих критериев, но может привести

Таблица 1

## Зависимость значения плотности развертывания от местоположения БС

Table 1

## Dependence of a deployment density on a base station location

Расстояние до границы РФ	Координаты БС		
	Приграничный район № 1	Приграничный район № 2	Приграничный район № 3
Менее 10 км	Требуется координация	Требуется координация	Требуется координация
10–15 км	Требуется координация	Требуется координация	1 БС на 100 км <sup>2</sup>
15–30 км	1 БС на 100 км <sup>2</sup>	1 БС на 100 км <sup>2</sup>	1 БС на 100 км <sup>2</sup>
30–60 км	10 БС на 100 км <sup>2</sup>	1 БС на 100 км <sup>2</sup>	10 БС на 100 км <sup>2</sup>
60–100 км	100 БС на 100 км <sup>2</sup>	100 БС на 100 км <sup>2</sup>	100 БС на 100 км <sup>2</sup>

к ограничениям по развертыванию новых средств ВРНС. Поэтому необходим инструмент, в полной мере оценивающий как запас, так и недостаток устойчивости к помехам ВРНС. Такая оценка ситуации может быть проведена в информационном пространстве.

Оценка проводилась для набора потенциально опасных БС, размещенных менее чем в 100 км от границы РФ. При этом для всех приведенных БС излучение сигнала происходит на центральной частоте 826 МГц в полосе шириной 5 МГц с ЭИИМ передатчиков 316 Вт.

Исходя из заданных в соглашении ограничений информационная оценка электромагнитной совместимости проводилась по следующим показателям:

1)  $h_{eff}$ , м – эффективная высота антенны БС по азимуту от БС на границу РФ (значения данного показателя не должны превышать нормативного  $(h_{eff})_{норм} = 60$  м для всех БС, находящихся не более чем в 60 км от границы);

2)  $d$ , шт. на 100 км<sup>2</sup> – плотность развертывания БС (нормативное значение по этому показателю  $d_{норм}$  выбирается в зависимости от конкретных условий размещения БС из таблицы 1);

3)  $E$ , дБмкВ/м – прогнозируемая средняя напряженность электрического поля, создаваемого конкретной БС при передаче на границе РФ (нормативное значение по этому показателю  $E_{норм}$  выбирается в зависимости от местоположения БС: в приграничном районе № 1 – 53 дБмкВ/м, в приграничном районе № 2 – 57 дБмкВ/м, в приграничном районе № 3 – 58 дБмкВ/м).

Эффективная высота антенны БС  $h_{eff}$  рассчитывалась программно [9] как сумма высоты подвеса антенны БС над поверхностью земли в конкретном месте ее размещения и высоты рельефа окружающей БС местности по азимуту на границу РФ, которая вычисляется на основании значений, взятых из БД SRTM3 (Shuttle Radar Topography Mission) [10], глобальной матрицы высот земной поверхности и усредненных на отрезке от 3 до 15 км удаления от передатчика БС. Расчет прогнозируемой средней напряженности поля, создаваемого отдельной БС на границе РФ, проводился в соответствии с Рекомендацией МСЭ-R P.1546-5 для 10 % времени и 50 % местоположений.

В таблице 2 приведены результаты сопоставления требуемых значений показателей в сокращенном варианте для отдельных станций сети с реальными значениями, предложенными государством, на территории которого развернута сеть. Общий размер исследуемой сети – 871 станция. Реальные значения показателей, превышающие нормативные значения, выделены жирным шрифтом.

В условиях рассматриваемой задачи для  $i$ -й БС расчет составляющих вектора элемента покрытия множества нормативных значений  $j$ -го показателя множеством реальных (3) будем осуществлять следующим образом:

$$\Pi_{ji} = K_n \frac{|(R_j)_i - (R_{норм})_j|}{(R_{норм})_j} + 1, \quad (5)$$

где  $(R_j)_i$  – реальное значение  $j$ -го показателя для  $i$ -й БС;  $(R_{норм})_j$  – нормативное значение  $j$ -го показателя;  $K_n$  – коэффициент нормировки, который выбирается из соотношения  $\frac{(R_{max})_j}{(R_{норм})_j} \times K_n = 10$ , где

$(R_{max})_i$  – максимальное значение  $j$ -го показателя.

Значение итоговой энтропии покрытия:

$$H_n = \sum_{j=1}^3 \log_2 \sum_{i=1}^N \begin{cases} \Pi_{ji}, (R_j)_i \leq (R_{норм})_j \\ 0, (R_j)_i > (R_{норм})_j \end{cases} + \\ + i \sum_{j=1}^3 \log_2 \sum_{i=1}^N \begin{cases} \Pi_{ji}, (R_j)_i > (R_{норм})_j \\ 0, (R_j)_i \leq (R_{норм})_j \end{cases}. \quad (6)$$

Видно, что, согласно (6), действительная часть энтропии покрытия характеризует запас по устойчивости к деструктивным факторам отдельных показателей по сравнению с нормативными, а мнимая часть – недостаточность запаса устойчивости. Нулевое значение энтропии покрытия будет соответствовать случаю, когда достигается значение состояния полного компромисса.

Наглядно алгоритм информационной оценки деструктивного влияния конкретной мобильной сети на комплекс средств ВРНС представлен на рисунке 1.

Информационная оценка по приведенному выше алгоритму была выполнена для трех ситуаций:

Таблица 2

## Значения показателей для БС одной сети

Table 2

## The real metric values for one-network base stations

Условное обозначение БС	Показатель № 1 $h_{eff}$ , м		Показатель № 2 $d$ , шт. на 100 км <sup>2</sup>		Показатель № 3 $E$ , дБмкВ/м	
	Реальный	Нормативный	Реальный	Нормативный	Реальный	Нормативный
052С	45,00	60	33	100	0,46	58
052С	39,00	60	33	100	14,64	58
229	28,30	60	33	100	-9,14	58
229	28,30	60	12	100	23,84	53
229	28,30	60	12	100	11,49	53
22АМ	52,23	60	12	100	31,31	53
23А	34,00	60	2	1	19,99	58
23В	58,20	60	2	1	26,99	58
23В	<b>66,80</b>	60	2	1	21,71	58
23В	<b>63,00</b>	60	2	1	10,98	53
23D	<b>79,57</b>	60	2	1	-1,37	53
23D	<b>79,57</b>	60	3	100	-10,00	58
23D	<b>72,70</b>	60	3	100	14,28	58
24А	22,03	60	3	100	1,14	58
24А	22,03	60	12	100	18,19	53
24А	22,03	60	12	100	31,14	53
25А	<b>97,28</b>	60	12	100	9,31	53
25А	<b>106,50</b>	60	2	100	-8,45	58
25А	<b>94,00</b>	60	2	100	7,74	58
...						

– исходные предложения администрации связи сопредельного государства;

– полное исключение станций СПС сопредельного государства, не удовлетворяющих условиям соглашения;

– ситуация компромисса, учитывающая предложения администрации связи сопредельного государства и не допускающая запрещенного по критерию защиты уровня помех на РЭС ВРНС.

Результаты оценки этих ситуаций в значениях энтропии покрытия приведены в таблице 3. Необходимо отметить, что все три ситуации рассмотрены с условием обязательного выполнения требований по допустимому уровню помех для действующих РЭС ВРНС.

Таблица 3

## Результаты информационной оценки

Table 3

## The results of information evaluation

Ситуация	Значение действительной части (запас), бит	Значение мнимой части (превышения), бит
Предложения администрации связи	32,10	15,74
Координация с комплексом средств ВРНС	31,79	0
Компромисс (соглашение)	31,58	2,60

Приведенные результаты показывают следующее.

- Предложения администрации связи сопредельного государства не в полной мере соответствуют условиям соглашения, обеспечивающим защиту РЭС ВРНС от вредного действия помех. Величина мнимой составляющей энтропии покрытия очень высока. Но в отдельных районах размещения РЭС СПС имеются достаточные ресурсы по запасу устойчивости к помехам, что показывает действительная составляющая энтропии покрытия.

- При полном выполнении условий соглашения защита РЭС ВРНС обеспечивается в полном соответствии с критериями по допустимым уровням помех, при этом мнимая составляющая энтропии покрытия принимает нулевое значение. Несколько снижается запас устойчивости к вредному воздействию помех, действительная составляющая энтропии покрытия меньше, чем в первой ситуации.

- При компромиссном решении значения действительной и мнимой составляющих энтропии покрытия снижаются по сравнению с первой ситуацией. Действительная составляющая энтропии покрытия снижается незначительно, а мнимая составляющая – довольно значительно, с 15,74 до 2,6 бит. Целесообразно установить для мнимой составляющей допустимое пороговое значение, которое сле-

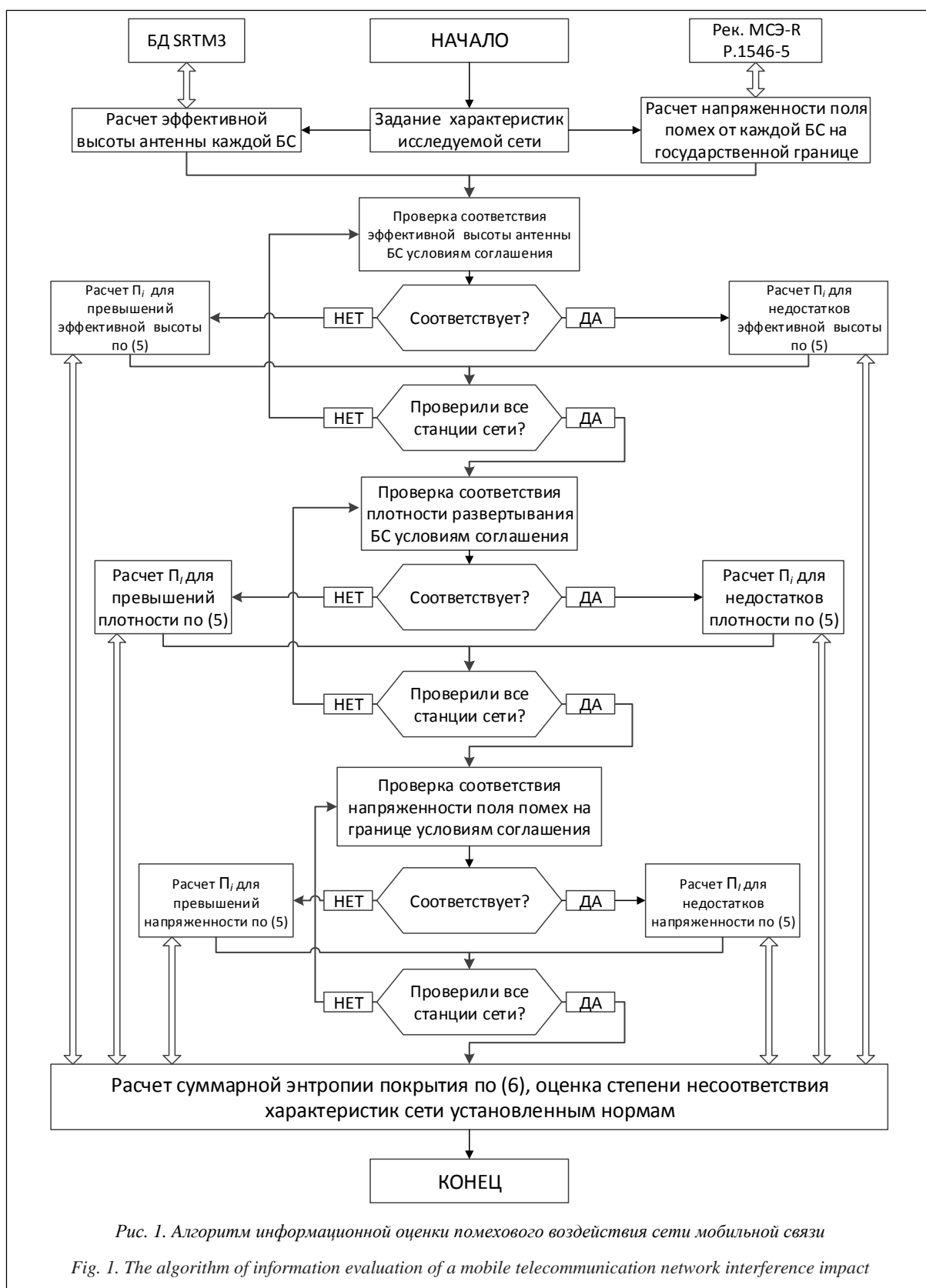


Рис. 1. Алгоритм информационной оценки помехового воздействия сети мобильной связи

Fig. 1. The algorithm of information evaluation of a mobile telecommunication network interference impact

дует учитывать при проведении координации БС сопредельного государства.

Наличие мнимой составляющей характеризует возможные ограничения на размещение новых

средств ВРНС в определенных приграничных районах, а достаточно большая величина действительной составляющей энтропии покрытия определяет довольно высокие потенциальные возможности

обеих администраций связи по размещению новых РЭС.

Следует также отметить, что при оптимизации сетей и при проведении координации энтропия покрытия обеспечивает возможность применения математических методов, основанных на ее использовании. Примеры решения оптимизационных задач с использованием энтропии покрытия приведены в работах [3, 5–9].

Наряду с приведенным выше анализом комплексного помехового воздействия не менее актуальной задачей является оценка вклада излучения, создаваемого единичной БС СПС. Решение этого вопроса позволяет адекватно реагировать на изменение помеховой обстановки в приграничном регионе, то есть в зависимости от характеристик проектируемой станции СПС, указанных в запросе на координацию частотного присвоения для нее, обоснованно удовлетворять или отклонять указанный запрос. При этом необходимо учитывать как ограничения, заданные в действующем соглашении между запрашивающей и затронутой администрациями связи, так и прогнозируемое среднее значение напряженности поля помех от станций СПС в месте размещения станции ВРНС (с учетом помехового вклада ранее скоординированных станций).

При расчетах напряженности поля помехи принимаются следующие допущения:

- средства СПС используют режим частотного разделения каналов (FDD – frequency division duplex), который подразумевает, что БС и АТ работают в разных частотных диапазонах;
- антенны АТ являются всенаправленными;
- эффективная высота антенн АТ равна 1,5 м;
- БС одной сети могут одновременно работать на одной частоте;
- АТ, расположенные в различных секторах антенны БС, могут одновременно работать на одной частоте;
- суммарная ЭИИМ всех АТ, работающих в одном секторе антенны БС, соответствует максимальной ЭИИМ одного АТ;
- в целях обеспечения безопасности полетов воздушного судна для расчетов напряженности поля помех, создаваемых средствами СПС в месте размещения воздушной станции ВРНС, используется уравнение распространения радиоволн в свободном пространстве, приведенное в Рекомендации МСЭ-Р Р.525-2 «Расчет ослабления в свободном пространстве»;
- для расчета напряженности поля помех от станций СПС в месте размещения наземной стан-

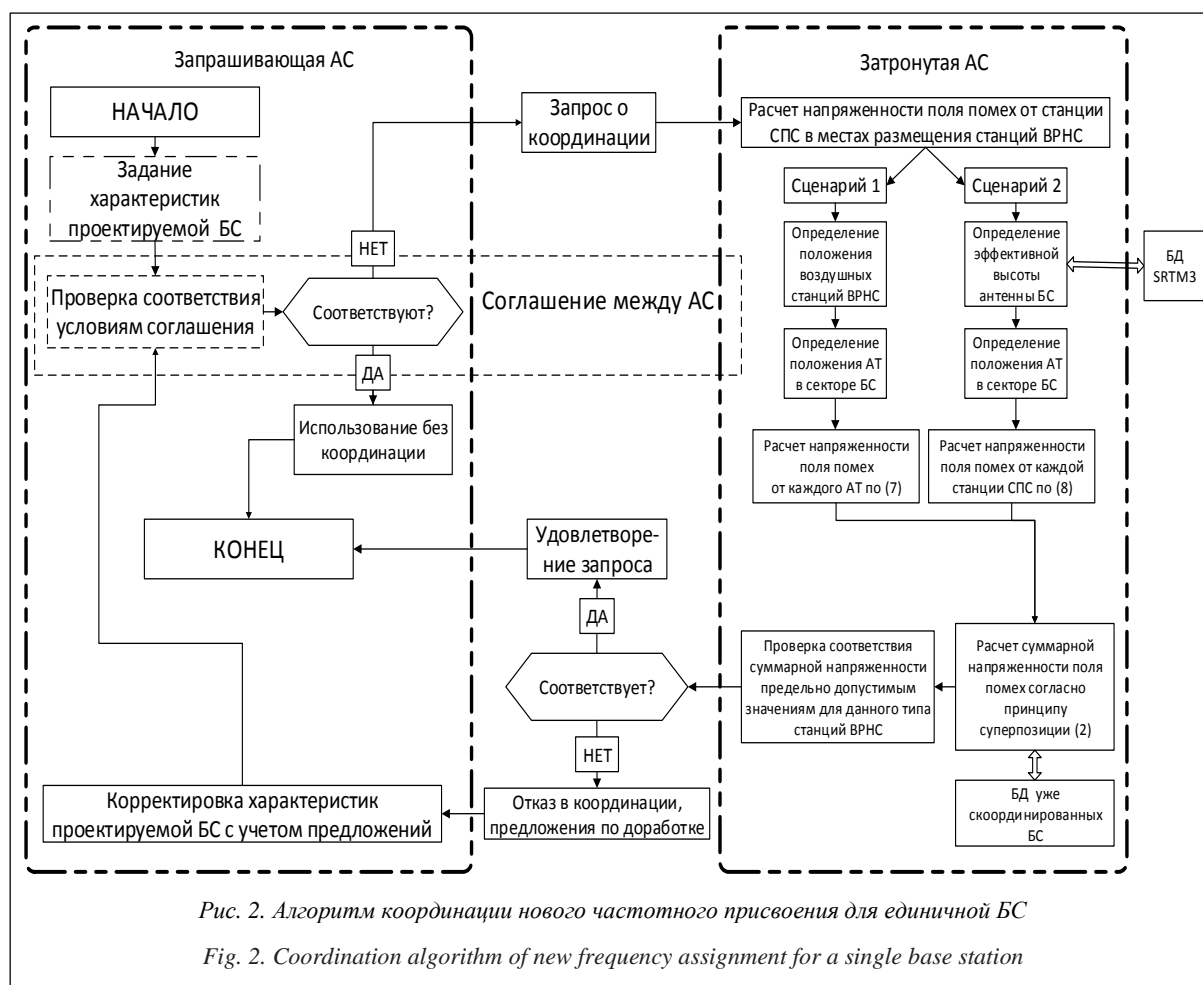


Рис. 2. Алгоритм координации нового частотного присвоения для единичной БС

Fig. 2. Coordination algorithm of new frequency assignment for a single base station



ции ВРНС используется Рекомендация МСЭ-R P.1546-5 для 10 % времени и 50 % местоположений.

Возможны два характерных сценария помеховой обстановки:

- сценарий 1: помеха от средств СПС на воздушные станции ВРНС;
- сценарий 2: помеха от средств СПС на наземные станции ВРНС.

В случае реализации 1-го сценария алгоритм для проведения расчетов заключается в следующем:

- определяется место размещения воздушной станции ВРНС путем размещения на минимальном удалении от границы в пределах зоны обслуживания соответствующей наземной станции ВРНС;

- определяется положение АТ (работающего с максимальной мощностью) в каждом секторе БС путем размещения АТ на минимальном удалении от воздушной станции ВРНС в пределах зоны обслуживания рассматриваемого сектора БС;

- рассчитывается напряженность поля помехи от каждого АТ в месте размещения воздушной станции ВРНС по формуле, приведенной в Рекомендации МСЭ-R P.525-2:

$$E_i = P_i - 20 \lg r_i + 74,8 + G_{ARNS}, \quad (7)$$

где  $E_i$  – напряженность поля помехи, создаваемой  $i$ -й станцией СПС в месте размещения станции ВРНС (дБмкВ/м);  $P_i$  – максимальная ЭИИМ  $i$ -й станции СПС (дБВт);  $r_i$  – расстояние от  $i$ -й станции СПС до станции ВРНС (км);  $G_{ARNS}$  – коэффициент усиления антенны станции ВРНС (дБи);

- по закону суперпозиции электромагнитных полей (2) рассчитывается суммарная помеха от всех АТ в месте размещения воздушной станции ВРНС;

- суммарная помеха от всех АТ в месте размещения воздушной станции ВРНС сравнивается с предельно допустимым уровнем суммарных помех, и делается вывод о возможности совмещения с рассматриваемой воздушной станцией ВРНС.

При реализации сценария 1 со стороны БС воздействия на бортовые станции не происходит. Это объясняется тем, что ввиду высокой мощности передатчиков БС нет возможности их совместного использования с воздушными станциями ВРНС. Рабочие частоты указанных станций должны находиться на разных неперекрывающихся диапазонах радиочастот.

В случае реализации сценария 2 учитывается помеховое воздействие от всех типов средств СПС, расчеты ведутся по следующему алгоритму:

- по координатам БС и высоте подвеса антенны БС в конкретном секторе вычисляется эффективная высота антенны БС по азимуту на наземную станцию ВРНС программным способом с использованием матрицы высот земной поверхности (данные SRTM3);

- определяется положение АТ (работающего с максимальной мощностью) в каждом секторе БС путем размещения АТ на минимальном удалении от станции ВРНС в пределах зоны обслуживания рассматриваемого сектора БС;

- рассчитывается напряженность поля помехи от каждой БС и от каждого АТ в месте размещения наземной станции ВРНС по формуле, приведенной в Рекомендации МСЭ-R P.525-2:

$$E_i = P_i - L_i + 20 \lg f + 167,2 + G_{ARNS}, \quad (8)$$

где  $E_i$ ,  $P_i$ ,  $G_{ARNS}$  – аналоги величин, приведенных в (7);  $f$  – частота излучения станции СПС (ГГц);  $L_i$  – потери при распространении помехи от станции СПС до станции ВРНС (дБ), рассчитываются в соответствии с Рекомендацией МСЭ-R P.1546-5;

- по закону суперпозиции электромагнитных полей (2) рассчитывается суммарная помеха от всех средств СПС в месте размещения наземной станции ВРНС;

- полученная суммарная помеха в месте размещения наземной станции ВРНС сравнивается с предельно допустимым уровнем суммарной помехи и делается вывод о возможности совмещения с рассматриваемой наземной станцией ВРНС.

Наглядно алгоритм координации нового частотного присвоения для единичной БС СПС и комплекса средств ВРНС показан на рисунке 2.

Таким образом, в настоящей работе представлен алгоритм комплексной информационной оценки состояния комплекса средств ВРНС в условиях деструктивного помехового воздействия средств СПС, позволяющий математически определить степень неудовлетворенности параметрами рассматриваемой сети мобильной связи с помощью введения информационной меры – энтропии покрытия. Кроме того, предложена удовлетворительная методика (с приведением соответствующего алгоритма) координации отдельных станций СПС.

### Литература

1. Пудовкин А.П., Панасюк Ю.Н., Иванков А.А. Основы теории антенн. Тамбов: Изд-во ТГТУ, 2011. С. 7.
2. Юрков Н.К., Андреев П.Г., Жумабаева А.С. Проблема обеспечения электромагнитной совместимости радиоэлектронных средств // Надежность и качество: тр. Междунар. симпози. Пенза. 2015. Т. 1. С. 201–203.
3. Сухов А.В., Новиков О.П., Прокопенко В.С. Динамика информационных потоков в системе управления сложным техническим комплексом в информационном пространстве, основанном на энтропии покрытия // Межотраслевая информационная служба. 2012. № 1. С. 16–21.
4. Ломакин М.И., Коровайцев А.А., Сухов А.В. Информационно-энтропийный подход к оценке метрологического ресурса средств измерений // Измерительная техника. 2014. № 12. С. 14–17.
5. Сухов А.В., Мячин А.В. Методы и технологии разработки управленческих решений при создании сложных технических комплексов. М.: Изд-во ВА РВЧН им. Петра Великого, 2008. С. 53–54.
6. Calculation of effective antenna heights using the SRTM3 Terrain Database. URL: <http://www.itu.int/SRTM3/> (дата обращения: 04.04.2017).

7. Farr T.G., Rosen P.A., Caro E., Crippen R., Duren R., Hensley S., Kobrick M., Paller M., Rodriguez E., Roth L., Seal D., Shaffer S., Shimada J., Umland J., Werner M., Oskin M., Burbank D.,

Alsdorf D. The shuttle radar topography mission. *Reviews of Geophysics*, 2007, vol. 45, is. 2. URL: <http://onlinelibrary.wiley.com/doi/10.1029/2005RG000183/epdf> (дата обращения: 04.04.2017).

Software & Systems

DOI: 10.15827/0236-235X.119.529-536

Received 06.04.17

2017, vol. 30, no. 3, pp. 529–536

## ALGORITHMS OF INFORMATION ESTIMATION OF COMPATIBILITY OF MOBILE TELECOMMUNICATION DEVICES AND AERONAUTICAL RADIONAVIGATION SERVICE STATIONS

**O.Yu. Vavulov**<sup>1</sup>, Postgraduate Student, [ovavulov@mail.ru](mailto:ovavulov@mail.ru)

**A.V. Sukhov**<sup>2</sup>, Ph.D. (Engineering), Professor, [avs57@mail.ru](mailto:avs57@mail.ru)

**V.N. Reshetnikov**<sup>2,3</sup>, Dr.Sc. (Physics and Mathematics), Professor, Head of Chair, [rvn@mail.ru](mailto:rvn@mail.ru)

<sup>1</sup> Russian Scientific-Technical Information Centre for Standardization, Metrology And Conformity Assessment, Nakhimovsky Av. 31-2, Moscow, 117418, Russian Federation

<sup>2</sup> Moscow Aviation Institute (National Research University), Volokolamskoe Highway 4, Moscow, 125993, Russian Federation

<sup>3</sup> Center of Visualization and Satellite Information Technologies SRISA, Nakhimovsky Ave. 36/1, Moscow, 117218, Russian Federation

**Abstract.** Every member in modern information society requires high-quality, high-speed and widespread mobile telecommunication. The number of such users increases steadily. Therefore, new technologies are being rapidly developed and the newest quality standards (LTE, 4G) are being implemented in this area. At the same time, it is also important to take into account the possibility of electromagnetic compatibility with other telecommunication service devices, especially safety services, for instance, aeronautical radionavigation service.

the present article considered The issue of information estimation algorithm development for the state of the technical complex of aeronautical radionavigation service in conditions of destructive interference impact from mobile telecommunication systems. The state of technical complex of aeronautical radionavigation service is defined in the mathematical terms in information space is set by means of the information measure – covering entropy that indicates how real characteristics of the considered systems correspond to their normative values. The suggested method has been applied for estimation of interference impact on the real Russian stations of aeronautical radionavigation service, located in the border region, the example of algorithm application is presented for compatibility estimation of the technical complex of aeronautical radionavigation service with the real mobile telecommunication network.

With this coordination algorithm of the technical complex of aeronautical radionavigation service with new frequency assignment for a single base station is suggested. This algorithm is based on calculation of intensity of the aggregate interference field strength produced by all land mobile stations in the area of location of aeronautical radionavigation service station and comparison of the calculated value with the permissible value for specific type of aeronautical radionavigation service station.

**Keywords:** electromagnetic compatibility, information estimation, aeronautical radionavigation service, mobile telecommunication, covering entropy.

## References

1. Pudovkin A.P., Panasyuk Yu.N., Ivankov A.A. *Osnovy teorii antenn* [Fundamentals of Antenna Theory]. Tambov, TSTU Publ., 2011, p. 7 (in Russ.).
2. Yurkov N.K., Andreev P.G., Zhumabaeva A.S. The problem of ensuring electromagnetic compatibility of radio electronic means. *Nadezhnost i kachestvo: tr. Mezhdunar. simpoz.* [Proc. of the Int. Symp. Reliability and Quality]. Penza, 2015, vol. 1, pp. 201–203 (in Russ.).
3. Sukhov A.V., Novikov O.P., Prokopenko V.S. Dynamics of information streams in a control system of a difficult technical complex in the information field based on covering entropy. *Mezhotraslevaya informatsionnaya sluzhba* [Interindustry Information Service]. 2012, no. 1, pp. 16–21 (in Russ.).
4. Lomakin M.I., Korovaytsev A.A., Sukhov A.V. Information-entropy approach to assessing resource metrological measuring instruments. *Izmeritelnaya tekhnika* [Measuring equipment]. 2014, no. 12, pp. 14–17 (in Russ.).
5. Sukhov A.V., Myachin A.V. *Metody i tekhnologii vyrabotki upravlencheskikh resheny pri sozdaniy slozhnykh tekhnicheskikh kompleksov* [Methods and technologies of development of administrative decisions when creating complex technical systems]. Moscow, Military Academy RVSN them. Peter the Great Publ., 2008, pp. 53–54 (in Russ.).
6. *Calculation of Effective antenna heights using the SRTM3 Terrain Database*. Available at: <http://www.itu.int/SRTM3/> (accessed April 4, 2017).
7. Farr T.G., Rosen P.A., Caro E., Crippen R., Duren R., Hensley S., Kobrick M., Paller M., Rodriguez E., Roth L., Seal D., Shaffer S., Shimada J., Umland J., Werner M., Oskin M., Burbank D., Alsdorf D. *The Shuttle Radar Topography Mission. Reviews of Geophysics*, 2007, vol. 45, iss. 2. Available at: <http://onlinelibrary.wiley.com/doi/10.1029/2005RG000183/epdf> (accessed April 4, 2017).

УДК 004.422.8

DOI: 10.15827/0236-235X.119.537-545

Дата подачи статьи: 14.12.16

2017. Т. 30. № 3. С. 537–545

## **КОМПЛЕКС ПРОГРАММ ДЛЯ ОПРЕДЕЛЕНИЯ ПАРАМЕТРОВ ЭЛЕКТРИЧЕСКИХ ДУГ ТРЕХФАЗНОГО ПЕРЕМЕННОГО ТОКА, ГОРЯЩИХ НА ГОРИЗОНТАЛЬНУЮ ПОВЕРХНОСТЬ**

И.М. Ячиков, д.т.н., профессор, jachikov@mail.ru;

Е.М. Костылева, аспирант, ezaretskaya@yandex.ru

(Магнитогорский государственный технический университет им. Г.И. Носова,  
просп. Ленина, 38, г. Магнитогорск, 455000, Россия)

В металлургической промышленности эксплуатируется множество агрегатов, в которых в качестве источников тепловой энергии используются электрические дуги, горящие на горизонтальную тепловоспринимающую поверхность. Знание электромагнитных сил, действующих на дуги, их положение и поведение важны при конструировании новых печей и автоматизации их работы.

Обзор литературы показывает, что многие вопросы, связанные с электрическими дугами трехфазного переменного тока, освещены недостаточно полно, поскольку на промышленных или опытных установках изучение большинства проблем технически сложно, дорого и часто невыполнимо по самым разным причинам.

Возможным направлением исследования является создание адекватных математических моделей, их реализация в виде прикладного ПО и дальнейшее компьютерное моделирование.

В данной работе показаны математическое модели и разработанный комплекс программ, включающий в себя модули для определения напряженности магнитного поля, электромагнитных сил, формы дуг и теплового потока вблизи трех дуг трехфазного переменного тока, горящих на горизонтальную поверхность.

Результатом работы созданного программного комплекса являются расчет и визуализация векторных полей – напряженности магнитного поля и среднеинтегральных электромагнитных сил, а также визуализация средней за период формы осей столбов электрических дуг и распределение тепловых потоков по горизонтальной тепловоспринимающей поверхности.

**Ключевые слова:** трехфазный переменный ток, напряженность магнитного поля, электромагнитные силы, форма оси столба дуги, тепловой поток, математическая модель, программное обеспечение.

В настоящее время существует множество агрегатов, в которых в качестве источников тепловой энергии используются трехфазные электрические дуги, горящие на горизонтальную поверхность. Это дуговые печи для выплавки металлов и сплавов, рудотермические, рудовосстановительные печи и другие. Знание ряда параметров электрических дуг, таких как магнитные поля, электромагнитные силы, действующие на дуги, положение и динамика поведения дуг, а также тепловые потоки излучения вблизи горящих дуг, является важным при эксплуатации существующих, конструировании новых печей и автоматизации их работы. На основе литературного анализа установлено, что многие вопросы, связанные с электрическими дугами трехфазного переменного тока, горящими на горизонтальную поверхность, освещены недостаточно полно. На промышленных или опытных установках исследование этих и других подобных проблем технически сложно, дорого и часто невозможно по самым разным причинам. Одним из возможных методов их изучения являются создание адекватных математических моделей, их реализация в виде ПО и дальнейшее исследование с использованием компьютерных моделей.

Для электрических дуг трехфазного переменного тока наиболее практически значимыми являются такие параметры, как напряженность магнитного поля в районе горения дуг, электромагнитные силы, действующие на дуги, форма дуг и радиаци-

онные тепловые потоки, падающие на горизонтальную поверхность.

В работах [1, 2] рассмотрена математическая модель для определения результирующей мгновенной напряженности магнитного поля в произвольной точке пространства  $A$  от токов, проходящих через три параллельных электрода. Находятся векторы напряженности магнитных полей, вызываемые токами в  $j$ -м электроде, а затем определяется результирующая сумма векторов с использованием принципа суперпозиции магнитных полей.

В выбранной полярной системе координат относительно центра распада электродов центры первого, второго и третьего электродов имеют координаты  $C_1(r_p, 0)$ ,  $C_2(r_p, 2\pi/3)$  и  $C_3(r_p, 4\pi/3)$  соответственно. Выбранная полярная система координат и совмещенная с ней декартова система координат  $Ox$ , а также схематическое расположение токов и точки  $A$  показаны на рисунке 1.

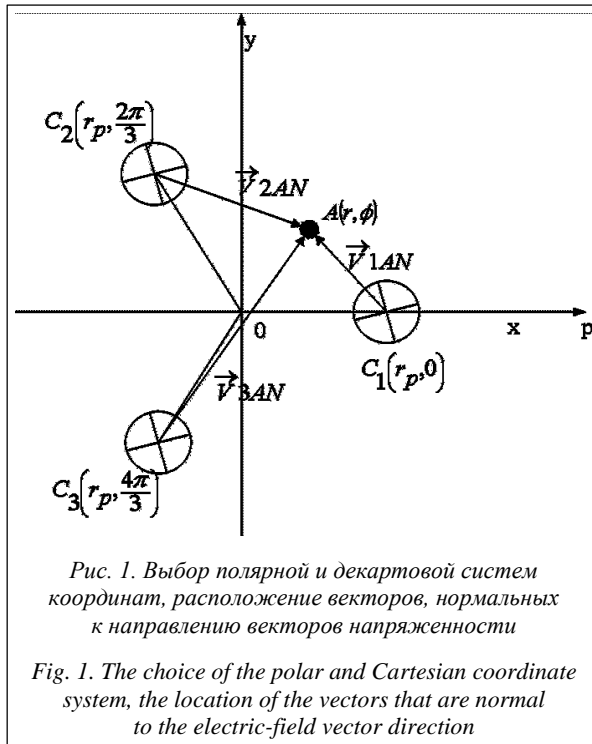
Напряженность магнитного поля  $\vec{H}_A$  в произвольной точке  $A(r, \varphi)$  определяется суммой трех векторов, координаты которых определяются по формуле

$$\vec{H}_{Aj} = \vec{H}_{Aj} \left( (-y_A + y_{Cj}) \cdot d, (x_A - x_{Cj}) \cdot d \right), \quad (1)$$

где  $d = \frac{i_0 \cdot \sin(2\pi\tau + \varphi_j)}{2 \cdot \pi \cdot \left( (-y_A + y_{Cj})^2 + (x_A - x_{Cj})^2 \right)}$ ;  $r_p$  – радиус

окружности, на которой расположены токи;  $\tau$  –

время;  $i_0$  – амплитуда тока;  $\varphi_j = 2\pi(j-1)/3$ ,  $j = 1, 2, 3$ ;  $\nu = 50$  Гц – частота тока;  $T = 1/\nu$  – период;  $\vec{H}_{Aj}$  – напряженность магнитного поля в точке  $A$  от тока  $j$ -го электрода;  $A(x_a, y_a)$  – координаты точки  $A$ ;  $C_j(x_{cj}, y_{cj})$  – координаты точек  $C_j$ .



Результирующая напряженность магнитного поля в точке  $A$  от действия трех токов, проходящих через графитированные электроды

$$\vec{H}_A = \vec{H}_A(x_{H1} + x_{H2} + x_{H3}, y_{H1} + y_{H2} + y_{H3}), \quad (2)$$

где  $x_{Hj}$ ,  $y_{Hj}$  – координаты  $x$  и  $y$  вектора напряженности магнитного поля от действия  $j$ -го тока.

На основе математической модели в среде разработки и моделирования MathCad создан программный модуль, позволяющий определять магнитные поля вблизи трех параллельно горящих дуг трехфазного переменного тока. В качестве входных параметров задаются диаметр распада электродов, амплитудное значение тока дуг и полярные координаты точки, в которой ищем магнитное поле. Программа позволяет строить графики изменения напряженности магнитного поля в зависимости от безразмерного расстояния до центра распада электродов в различные моменты времени или при разных значениях угловой координаты, нормированные векторные поля напряженности магнитного поля вблизи трех дуг трехфазного переменного тока, годографы векторов напряженности магнитного поля. В качестве примера были построены нормированные векторные поля напряженности магнитного поля в различные моменты времени. Так, для моментов времени  $\tau = n \cdot T/2$ , где  $n = 0, 1, 2, \dots$ ,

магнитное поле вблизи первого электрода отсутствует. В последующие моменты времени оно постепенно увеличивается, принимает максимальное значение при  $\tau = T/4 + n \cdot T/2$ , после чего постепенно уменьшается и к моменту времени  $\tau = T/2 + n \cdot T/2$  пропадает. Аналогично ведут себя напряженности магнитных полей вблизи второго и третьего электродов, возле которых поле отсутствует в моменты времени  $\tau = T/6 + n \cdot T/2$  и  $\tau = T/3 + n \cdot T/2$  и максимально  $\tau = 5T/12 + n \cdot T/2$  и  $\tau = 7T/12 + n \cdot T/2$  соответственно. Установлено, что вблизи трех электрических дуг, горящих на горизонтальную поверхность, имеет место бегущее магнитное поле. На рисунке 2 приведен пример работы программы, показаны нормированные векторные поля напряженности магнитного поля вблизи трех параллельно горящих дуг трехфазного переменного тока в разные моменты времени, а также годографы вектора напряженности магнитного поля внутри и вне диаметра распада электродов.

Для электрометаллургической теории и практики важными являются методы расчета локальной электромагнитной силы, действующей на элемент дуги, и среднеинтегральной силы, действующей на всю дугу. При постоянных токах для локальной силы, действующей на элемент дуги от тока, протекающего через жидкий металл, в [3] получено выражение

$$\frac{dF}{dl} = f_m(z) = \frac{i_1 i_0 \mu_0}{4\pi(\Delta + z)} \frac{l}{\sqrt{l^2 + (\Delta + z)^2}},$$

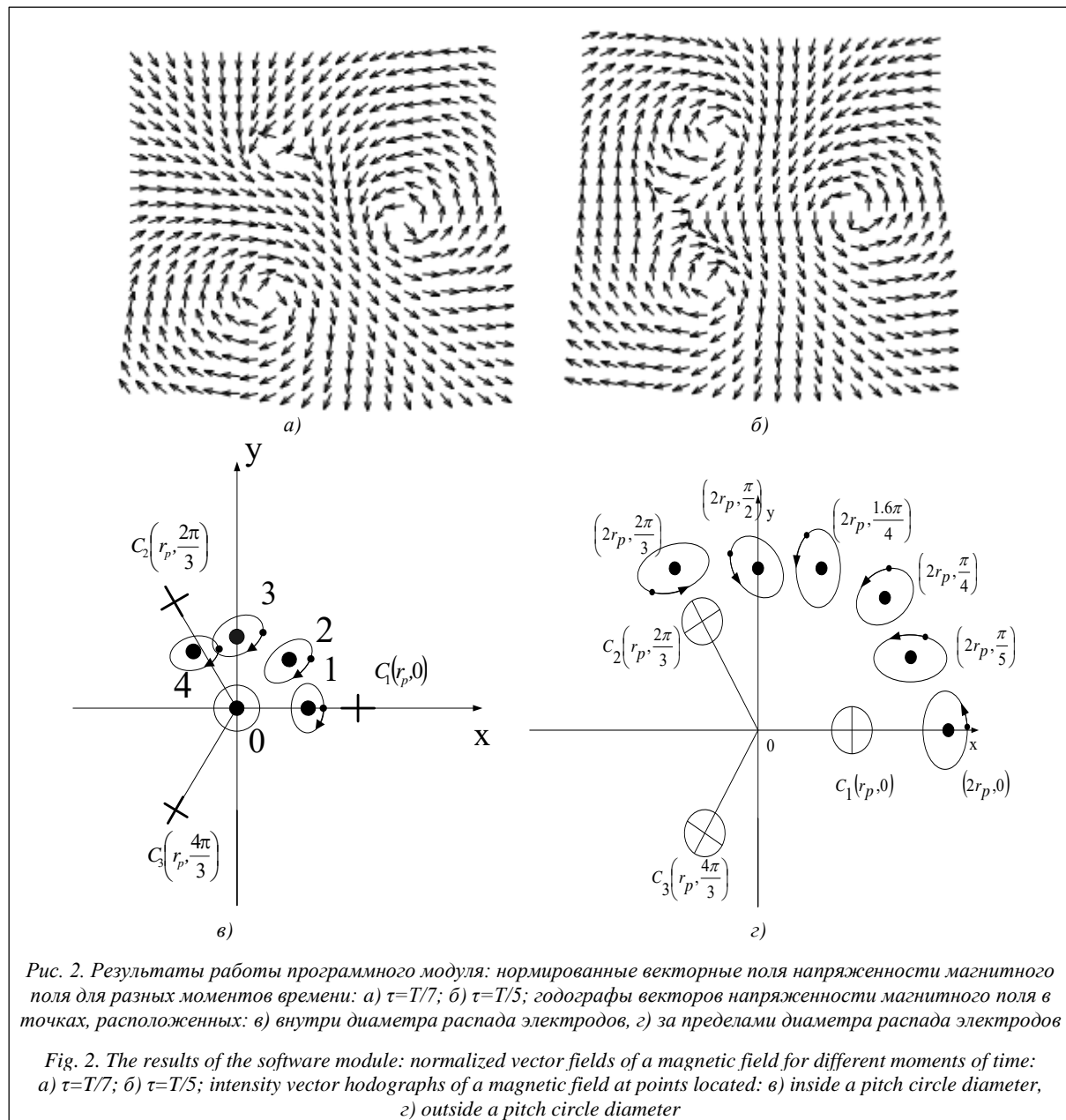
а для среднеинтегральной электромагнитной силы [4, 5] зависимость

$$F = \frac{\mu_0 i_1 i_0}{4\pi} \ln \left[ \frac{(\sqrt{l^2 + \Delta^2} + l)(\Delta + l_d)}{\Delta(\sqrt{(l_d + \Delta)^2 + l^2} + l)} \right], \quad (3)$$

где  $\Delta = 3,2-3,6$  см – глубина электрической оси токопроводящего слоя от поверхности жидкого металла [6];  $i_0$  – ток дуги;  $i_1$  – ток, протекающий через жидкий металл со стороны другой дуги;  $l_d$  – длина дуги;  $l$  – длина горизонтального отрезка, по которому протекает ток  $i_1$ ;  $z$  – расстояние от точки привязки дуги на горизонтальной поверхности до рассматриваемого элемента дуги.

В работе [4] получены выражения для электромагнитной силы, действующей на ток дуги  $i_1 = i_A$  со стороны трех фазных токов  $i_{AB}$ ,  $i_{BC}$ ,  $i_{CA}$ , протекающих через металл. Установлено, что модуль равнодействующей силы пропорционален  $\sqrt{3}I^2/4$ . С учетом действующего значения линейного тока  $I_{дл} = \sqrt{3}I/\sqrt{2}$  получено локальное среднее значение электромагнитной силы, действующей на единицу длины дуги:

$$f_m(z) = \frac{\mu_0 I_{дл}^2}{8\sqrt{3}\pi} \frac{1}{(\Delta + z)} \frac{l}{\sqrt{l^2 + (\Delta + z)^2}}, \quad (4)$$



и среднеинтегральное значение силы, действующей на дугу:

$$F_m = \frac{\mu_0 I_{\text{дл}}^2}{8\sqrt{3}\pi} \ln \left[ \frac{(\sqrt{l^2 + \Delta^2} + l)(\Delta + l_{\text{д}})}{\Delta(\sqrt{(l_{\text{д}} + \Delta)^2 + l^2} + l)} \right]. \quad (5)$$

В работе [7] рассмотрено взаимодействие двух вертикальных дуг постоянного тока, получена локальная электромагнитная сила, действующая на единицу длины дуги, на ток столба дуги  $i_0$  со стороны тока  $i_1$ , протекающего через дугу (длиной  $l_{\text{д}}$ ) и графитированный электрод (длиной  $l_{\text{эд}}$ ):

$$\frac{dF}{dl} = f_{\text{д}} = \frac{i_1 i_0 \mu_0}{4\pi l} \left( \frac{z}{\sqrt{l^2 + z^2}} + \frac{l_{\text{д}} + l_{\text{эд}} - z}{\sqrt{l^2 + (l_{\text{д}} + l_{\text{эд}} - z)^2}} \right),$$

и среднеинтегральная сила, действующая на дугу:

$$F_{\text{д}} = \frac{\mu_0 i_1 i_0}{4\pi l} \int_0^{l_{\text{д}}} \left( \frac{z}{\sqrt{l^2 + z^2}} + \frac{l_{\text{д}} + l_{\text{эд}} - z}{\sqrt{l^2 + (l_{\text{д}} + l_{\text{эд}} - z)^2}} \right) dz.$$

Среднюю результирующую силу взаимодействия одной из дуг в трехфазной дуговой печи переменного тока с двумя другими дугами с учетом действующего значения линейного тока  $I_{\text{дл}} = I / \sqrt{2}$  и локальную электромагнитную силу, действующую на единицу длины дуги [4, 7], можно определить следующим образом:

$$F_0 = \frac{\sqrt{3}\mu_0 I_{\text{дл}}^2}{4\pi l} \left( \sqrt{l^2 + l_{\text{д}}^2} - l - \sqrt{l^2 + l_{\text{эд}}^2} + \sqrt{l^2 + (l_{\text{д}} + l_{\text{эд}})^2} \right), \quad (6)$$

$$f_{\text{д}}(z) = \frac{\sqrt{3}\mu_0 I_{\text{дл}}^2}{4\pi l} \left( \frac{z}{\sqrt{l^2 + z^2}} + \frac{l_{\text{д}} + l_{\text{эд}} - z}{\sqrt{l^2 + (l_{\text{д}} + l_{\text{эд}} - z)^2}} \right). \quad (7)$$

На основе разработанных математических моделей в среде разработки и моделирования MatLab создана компьютерная программа «Расчет электромагнитных сил, действующих на дуги в трехфазной дуговой печи», позволяющая рассчитывать основные электромагнитные силы (Свид. о гос. регистр. № 2016618499), ее главное окно показано на рисунке 3. Исходными данными для программы являются расстояние между дугами, длина дуги, длина электрода, действующее значение тока. Все они проходят проверку на корректность ввода [8]. Далее рассчитываются значения среднеинтегральных

сил от токов, протекающих по жидкому металлу, и от токов, протекающих через параллельные дуги и электроды. Значения этих сил одинаковы для каждой из дуг и различаются только направлениями. Далее при помощи методов аналитической геометрии визуализируются расположение электродов, среднеинтегральных электромагнитных сил и их результирующее направление. Также в программе имеется возможность построения графиков локальных и среднеинтегральных электромагнитных сил. При нажатии на кнопку «Графики» на экране появляется рабочее окно для вывода графиков (рис. 4).

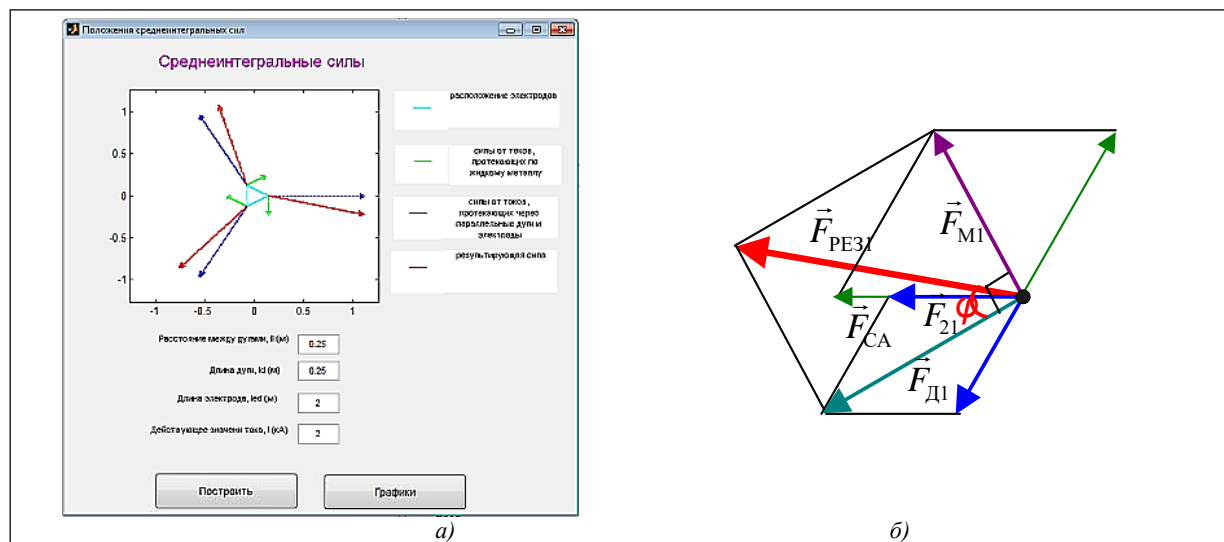


Рис. 3. Программа «Расчет электромагнитных сил, действующих на дуги в трехфазной дуговой печи»: а) главное окно, б) результаты моделирования основных векторов среднеинтегральных электромагнитных сил, действующих на дугу

Fig. 3. A program “Calculation of electromagnetic forces affecting arcs in a three-phase arc furnace”: a) the main window, б) simulation results of the main vectors of average integral electromagnetic forces affecting the arc

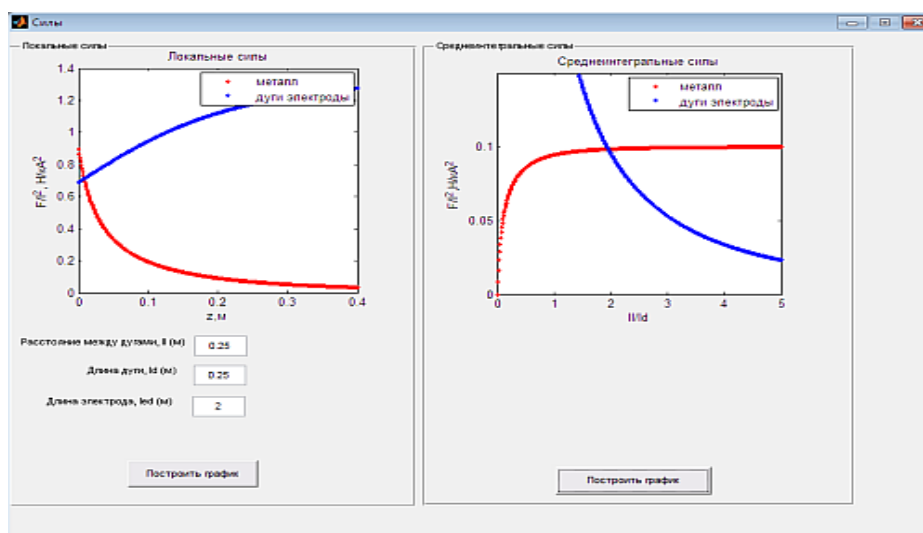


Рис. 4. Рабочее окно программы «Расчет электромагнитных сил, действующих на дуги в трехфазной дуговой печи» для вывода графиков локальных и среднеинтегральных электромагнитных сил

Fig. 4. The working window of the program “Calculation of electromagnetic forces affecting arcs in a three-phase arc furnace” for the charts of local and average integral electromagnetic forces

Строится график локальных сил, представляющий собой зависимость  $F / I_{\text{дл}}^2$  – значения локальной силы, деленной на квадрат тока от координаты точки  $z$ , и график среднеинтегральных сил  $F / I_{\text{дл}}^2$  от безразмерной координаты точки  $l/l_{\text{дл}}$ .

Рассмотрим математическую модель формы параллельно горящих на горизонтальную поверхность электрических дуг постоянного или переменного тока при их электромагнитном взаимодействии под действием силы  $\vec{F}_d$ . Для определения формы двух дуг постоянного тока при их электромагнитном взаимодействии использовалась математическая модель, предложенная в работе [9]. Форма оси первой дуги  $y(z)$  (рис. 5), через которую протекает ток  $i_1$ , при электромагнитном взаимодействии с дугой, по которой протекает ток  $i_2$ , описывается следующим интегро-дифференциальным уравнением:

$$\begin{aligned}
& \frac{i_2}{i_1} \int_0^l \frac{\left( \frac{dy(\xi)}{d\xi} (\xi - z) - (y(\xi) - y(z)) \right)}{\left( \left( |y(\xi)| + y(z) \right)^2 + (\xi - z)^2 \right)^{3/2}} d\xi - \\
& - \int_0^l \frac{\left( \frac{dy(\zeta)}{d\zeta} (\xi - z) - (y(\zeta) - y(z)) \right)}{\left( \left( y(\zeta) + y(z) \right)^2 + (\xi - z)^2 + y_0^2 \right)^{3/2}} d\zeta - \\
& - \frac{\mu_0 \beta(z)}{4\pi \left( 1 + (dy/dz)^2 \right)^{3/2}} \left| \frac{d^2 y}{dz^2} \right| = 0, \tag{8}
\end{aligned}$$

где  $\beta(z)$  – функция, определяющая параметры дуги;  
 $y_0$  – среднее геометрическое расстояние площади сечения дуги в ее цилиндрической части от самой себя.

Уравнение (8) относится к классу интегро-дифференциальных уравнений. Свойства таких уравнений и методы их решения рассмотрены в работах [10–16]. В отдельных частных случаях найдены аналитические решения, но в общем случае они мо-

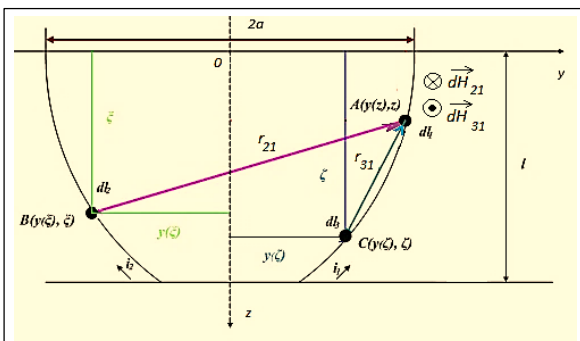


Рис. 5. Основные размеры, используемые в математической модели взаимодействия дуг

*Fig. 5. The basic dimensions in an arc interaction mathematical model*

гут быть решены только приближенно. Численное решение получается в результате итерационного процесса, точные оценки погрешности получены не для всех случаев. При численных расчетах в качестве критерия для завершения итерационного процесса применяется малое различие результатов, полученных на последовательных шагах процесса. Задача сводится к нахождению функции, которая с приемлемой точностью удовлетворяет уравнению (8). Решение затрудняется тем, что оно нелинейное, так как искомая функция входит в знаменатель правой части и в дифференциальном уравнении имеется модуль.

Для решения подобных интегро-дифференциальных уравнений обычно используются приближенные итерационные алгоритмы, например метод Пикара [12]. Однако его численная реализация показала, что для описываемой задачи полученный итерационный процесс является расходящимся.

Для нахождения приближенного решения этого уравнения предложены и реализованы три алгоритма [17]. По *первому алгоритму* предполагалось, что по дугам текут равные постоянные токи  $i_1 = i_2$  (дуги зеркально симметричны относительно оси  $Oz$ ), а неизвестная функция положения оси первой дуги имеет форму параболы  $y(z) = a_1 z^2 + b_1 z + c$ .

Применяя начальные условия  $y(0) = a$ ,  $\left. \frac{dy}{dz} \right|_{z=0} = \alpha$ ,

где  $\alpha$  – тангенс угла наклона оси дуги в ее начальном сечении, получено  $c = a$ ,  $b_1 = \alpha$ . Решая нелинейное уравнение (8) при разных  $z$ , находились неизвестный коэффициент  $a_1$ , который затем усреднялся.

По *второму алгоритму* также рассматривался случай одинаковых токов  $i_1 = i_2$ , при этом искомая функция имела вид полинома  $n$ -й степени  $y(z) = a_0 + a_1z + a_2z^2 + a_3z^3 + \dots + a_nz^n$ . Неизвестные коэффициенты  $a_2, a_3, \dots, a_n$  находились как минимум функции нескольких переменных:  $\varphi_1(a_2, a_3, \dots) = \int_0^l \Phi^2 dz = \min$  или  $\varphi_2(a_2, a_3, \dots) = \int_0^l |\Phi| dz = \min$ , где  $\Phi$  – левая часть уравнения (8).

По *третьему алгоритму* рассматривалось нахождение формы осей двух дуг при разных точках, проходящих через них,  $i_1 \neq i_2$ . В этом случае для нахождения двух неизвестных функций решалась система двух дифференциальных уравнений (8), записанных для каждой дуги, с граничными условиями:

$$y_1(0) = a, \left. \frac{dy_1}{dz} \right|_{z=0} = \alpha_1, y_2(0) = -a, \left. \frac{dy_2}{dz} \right|_{z=0} = \alpha_2.$$

Для получения приближенного решения предполагалось, что искомые функции формы осей столбов дуг имеют вид  $y_j(z) = a_{0j}z + a_{1j}z + a_{2j}z^2 + a_{3j}z^3 + \dots + a_{nj}z^n$ , где  $j = 1, 2$  – номер дуги. Из начальных условий, принимая  $\alpha_1 = \alpha_2 = 0$ , имеем  $a_{11} = a_{12} = 0$ ,  $a_{01} = a_{02} = 0$ . Искомые коэффициенты



многочленов  $a_{2j} = a_{3j} = \dots = a_{nj}$  находились как минимум функции нескольких переменных:

$$\varphi_3 = \int_0^l (\Phi_1^2 + \Phi_2^2) dz = \min, \text{ где } \Phi_1, \Phi_2 - \text{левые части}$$

уравнения (8), записанные для первой и второй дуг соответственно.

Уравнение равновесия элемента столба дуги расширено и применено для нахождения формы осей двух или трех дуг постоянного или трехфазного переменного тока и может быть записано как

$$k_1 d\vec{F}_{1-2} + d\vec{F}_{1-1} + d\vec{F}_{1-q} = 0, \quad (9)$$

где  $k_1$  – коэффициент, учитывающий вид электромагнитного взаимодействия;  $d\vec{F}_{1-2}$  – сила взаимодействия дуги с токами, протекающими по другим дугам;  $\vec{F}_{1-1}$  – сила взаимодействия дуги с собственным током;  $d\vec{F}_{1-q}$  – центростремительная сила, выпрямляющая дугу. Значение коэффициента  $k_1$  и характер электромагнитного взаимодействия зависят от рода тока и количества дуг [17].

На основе разработанных математических моделей в среде разработки и моделирования MatLab создан программный модуль «Взаимодействие дуг», позволяющий в зависимости от количества дуг, рода и значения протекающего по ним тока, расстояния между дугами, физических свойств плазмообразующего газа рассчитывать и визуализировать форму дуг (Свид. о гос. регистр. № 2016618499); главное окно программы показано на рисунке 6. Исходными данными являются расстояние между дугами, длина дуги, токи первой и второй дуг, степень полинома, при помощи которого аппроксимируется форма дуги, род тока, участвующего во взаимодействии дуг. Далее с учетом физических свойств плазмообразующего газа и

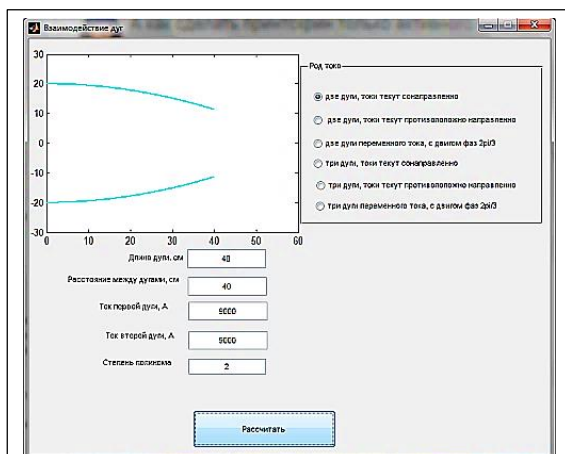


Рис. 6. Рабочее окно программы «Взаимодействие дуг» для расчета и построения формы осей столбов дуг

Fig. 6. The working window of the program "Arc interaction" for calculating and constructing shapes of arc axes

свойств окружающей среды находятся коэффициенты для построения формы оси столба дуги. Программа строит график формы осей столбов дуг  $y(z)$  в сантиметрах.

В дуговой сталеплавильной печи (ДСП) трансформация электрической энергии в тепловую происходит в разрядном промежутке. За счет электромагнитного взаимодействия электрических токов, протекающих по графитированным электродам и в поверхностных слоях расплава с током дуги, происходит ее выдувание от центра печи. Средний угол наклона столба дуги  $\theta$  по отношению к нормали зеркала ванны колеблется для ДСП различной мощности и емкости в пределах  $45^\circ$ – $65^\circ$  (см., например, [18, 19]).

В работе [20] была получена модель теплового потока на произвольно расположенную площадку  $dS$ , находящуюся на поверхности ванны от электрической дуги, расположенной под углом к ней:

$$q_{\text{плз}}(r, A) = \frac{K_3 \cdot P \cdot \sqrt{1-A^2}}{2\pi^2 \cdot l_d \cdot r} \left[ \frac{-B + A \cdot \operatorname{arctg}\left(\frac{A}{B}\right)}{B^3} + \frac{B \cdot r(l_d A + r) - A(r^2 + 2A \cdot r \cdot l_d + l_d^2) \operatorname{arctg}\left(\frac{l_d + rA}{rB}\right)}{B^3(r^2 + 2A \cdot r \cdot l_d + l_d^2)} \right], \quad (10)$$

где  $A = \cos\psi \cdot \sin\theta$ ;  $B = \sqrt{A^2 - 1}$ ;  $r$  – расстояние от начала координат до площадки  $dS$ ;  $\psi$  – угол, под которым расположена площадка  $dS$  к началу координат;  $\theta$  – угол между направлением излучения и нормалью к поверхности ванны;  $K_3$  – поправочный экспериментальный коэффициент, учитывающий долю мощности дуги, идущую на излучение;  $P$  – полная мощность излучения дуги;  $l_d$  – длина дуги (рис. 7).

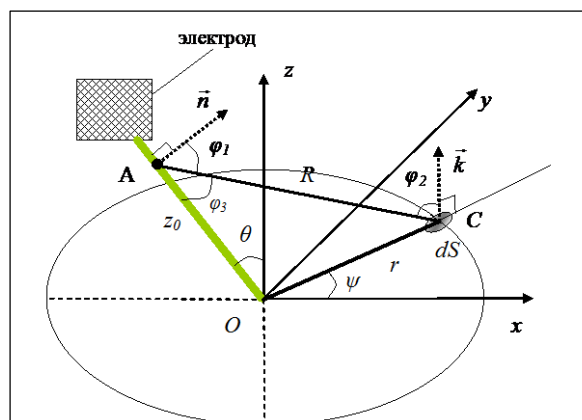


Рис. 7. К расчету плотности теплового излучения дуги на площадку  $dS$

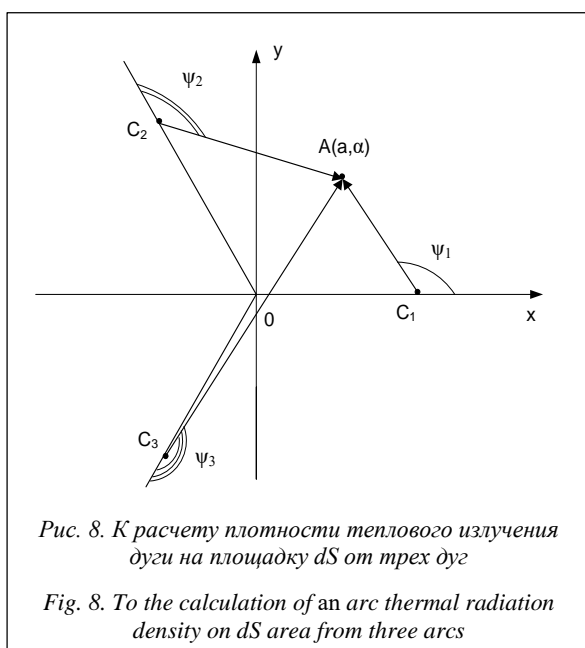
Fig. 7. To the calculation of an arc thermal radiation density on  $dS$  area



Тепловой поток от трех наклонных горящих электрических дуг длиной  $l_d$  на произвольно расположенную горизонтальную площадку находится по принципу суперпозиции как сумма тепловых потоков, падающих в данную точку  $A$  (рис. 8). Тепловой поток, падающий на элементарную площадку, определяется как

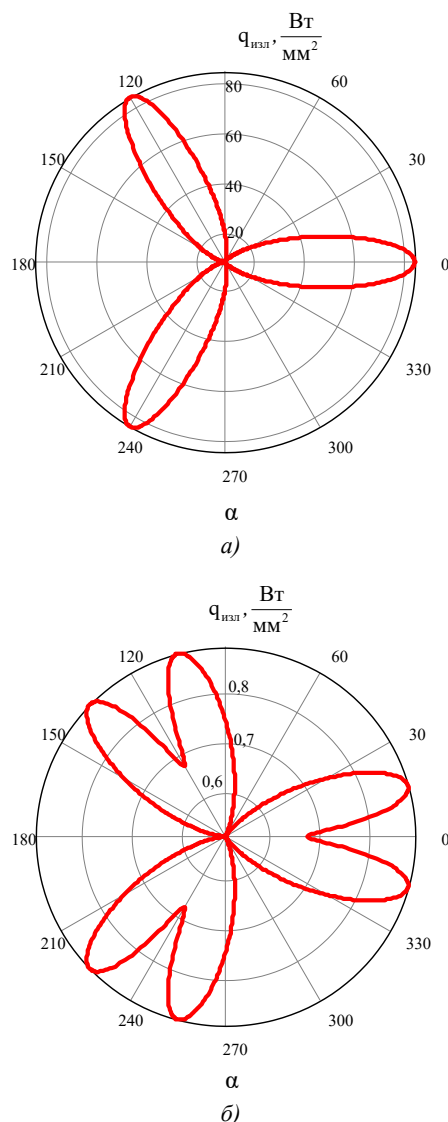
$$q_{\text{изл}}(a, \alpha, \theta) = \sum_{j=1}^3 q_{j \text{ изл}}(r_j, \psi_j, \theta), \quad (11)$$

где  $q_{j \text{ изл}}(r_j, \psi_j, \theta)$  – тепловой поток, падающий на горизонтальную поверхность от  $j$ -й дуги;  $r_j$  – расстояние от  $j$ -й дуги до площадки  $dS$ ;  $\psi_j$  – угол, под которым расположена площадка  $dS$  к  $j$ -й дуге;  $\theta$  – угол между направлением излучения и нормалью к поверхности ванны [21].



С использованием созданной математической модели (10)–(11) разработан компьютерный модуль, позволяющий рассчитывать и визуализировать тепловые потоки, падающие на горизонтальную поверхность, вблизи горения трех наклонных электрических дуг. В качестве входных параметров задаются диаметр распада электродов, действующее значение тока, координаты точек, в которых ищется и визуализируется тепловой поток. На рисунке 9 изображено распределение теплового потока по азимутальной координате при разном расстоянии от центра распада электродов в трехфазной дуговой печи ДСП-180 (ток дуги – 50 кА,  $l_d = 0,5$  м, диаметр распада электродов – 1,2 м), построенное при помощи разработанного модуля.

Таким образом, на основе оригинальных математических моделей разработан комплекс компьютерных программ для определения основных параметров электрических дуг трехфазного переменного тока. Комплекс программ позволяет рассчитать напряженность магнитных полей, опре-



делить и визуализировать среднеинтегральные электромагнитные силы, действующие на дуги, и локальные силы, действующие на элементы дуги: от токов, протекающих через жидкий металл, и от токов, проходящих через параллельно горящие дуги и графитированные электроды. Также специальные модули позволяют рассчитать и визуализировать формы осей столбов дуг в зависимости от их количества и рода тока и для определения и визуализации теплового потока от трех горящих наклонных электрических дуг по азимутальной координате. Структурная схема комплекса программ представлена на рисунке 10. Разработанный комплекс компьютерных программ может использоваться при конструировании многоэлектродных



дуговых или плазменных печей для определения положения электрических дуг в системах автоматизированного регулирования их тепловой работы, для оценки стойкости футеровки вблизи горящих дуг при разных технологических режимах, для определения воздействия электромагнитных полей на приборы и человека.

#### Литература

1. Леушин А.И. Дуга горения. М.: Металлургия, 1973. 240 с.
2. Ячиков И.М., Зарецкая Е.М. Анализ поведения магнитного поля вблизи электродов дуговых печей посредством математического моделирования // Изв. вузов: Черная металлургия. 2011. № 1. С. 18–20.
3. Миронов Ю.М. Электрическая дуга в электротехнологических установках: монография. Чебоксары: Изд-во Чуваш. ун-та, 2013. 290 с.
4. Yachikov I.M., Kostyleva E.M. Electromagnetic forces on the arc in a three-phase arc furnace. Steel in Translation. 2015. vol. 45, no. 7, pp. 467–472. DOI: 10.3103/S0967091215070141.
5. Егоров А.В. Расчет мощности и параметров электроплавильных печей. М.: Изд-во МИСиС, 2000. 272 с.
6. Егоров А.В. Электроплавильные печи черной металлургии. М.: Металлургия, 1985. 280 с.
7. Никольский Л.Е., Смоляренко В.Д., Кузнецов Л.Н. Тепловая работа дуговых сталеплавильных печей. М.: Металлургия, 1981. 320 с.
8. Лазарев Ю. Моделирование процессов и систем в MATLAB. СПб: Питер, 2005. 512 с.
9. Бортничук Н.И., Крутянский М.М. Плазменно-дуговые плавильные печи. М.: Энергоиздат, 1981. 120 с.
10. Азбелев Н.В., Максимов В.П., Рахматуллина Л.Ф. Элементы современной теории функционально-дифференциальных уравнений. Методы и приложения. Ижевск: Изд-во ИКИ, 2002. 304 с.
11. Дмитриев С.С., Кузнецов Е.Б. Численное решение систем интегро-дифференциально-алгебраических уравнений с запаздывающим аргументом // Журн. вычислит. матем. и матем. физики. 2008. № 3. С. 430–444.
12. Вержбицкий В.М. Численные методы. Математический анализ и обыкновенные дифференциальные уравнения. М.: Высш.шк., 2001. 383 с.
13. Федоренко Р.П. Введение в вычислительную физику. Долгопрудный: Интеллект, 2008. 504 с.
14. Зализняк В.Е. Основы научных вычислений. Введение в численные методы для физиков. М.: Editorial URSS, 2002. 296 с.
15. Бахвалов Н.С., Жидков Н.П., Кобельков Г.М. Численные методы. М.: Бином. Лаборатория знаний, 2003. 632 с.
16. Лёзин И.А. Решение систем полиномиальных уравнений на ЭВМ // Программные продукты и системы. 2012. № 3. С. 22–25.
17. Ячиков И.М., Костылева Е.М. Математическое моделирование формы дуг при их электромагнитном взаимодействии. Сообщ. 2: Форма электрических дуг постоянного или трехфазного переменного тока, горящих между электродами и токоподводящей поверхностью // Изв. вузов: Черная металлургия. 2014. № 5. С. 56–61.
18. Макаров А.Н. Теплообмен в дуговых сталеплавильных печах. Тверь: Изд-во ТГТУ, 1998. 184 с.
19. Миронов Ю.М., Михадаров Д.Г. Влияние межэлектродного промежутка на отклонение столба дуги // Вестн. ЧГУ. 2009. № 2. С. 139–142.
20. Yachikov I.M., Zaretskaya E.M. Radiant heat flux from an electric arc inclined to the plane of the receiving surface. Steel in Translation, 2011, vol. 41, no. 7, pp. 550–554. DOI: 10.3103/S0967091211070151.
21. Макаров А.Н., Свенчанский А.Д. Оптимальные тепловые режимы дуговых сталеплавильных печей. М.: Энергоатомиздат, 1992. 96 с.

#### A SOFTWARE SUITE TO DETERMINE THE PARAMETERS OF THREE-PHASE ALTERNATING CURRENT ELECTRIC ARCS BURNING IN A HORIZONTAL SURFACE

I.M. Yachikov<sup>1</sup>, Dr.Sc. (Engineering), Professor, yachikov@mail.ru  
E.M. Kostyleva<sup>1</sup>, Postgraduate Student, ezaretskaya@yandex.ru

<sup>1</sup> G.I. Nosov Magnitogorsk State Technical University, Lenin Ave. 38, Magnitogorsk, 455000, Russian Federation

**Abstract.** The metallurgical industry uses a lot of equipment with electrical arcs as the sources of thermal energy. They burn in the direction of the horizontal heat-absorbing surface. Knowing electromagnetic forces affecting the arcs, their location and behavior is significant when designing new furnaces and automating their operation.

The literature review shows that many issues connected with electrical arcs of three-phase alternating current are underserved due to technical complexity, high cost and often impossible on industrial or pilot plants.

A possible method of research is the development of appropriate mathematical models, their implementation as applied software and further computer modeling.

The paper shows mathematical models and developed software, which includes modules for determining the magnetic field intensity, electromagnetic forces, arc and heat flow forms, near three arcs of three-phase alternating current burning in the direction of a horizontal surface.

The result of the created software complex is calculation and representation of vector fields including magnetic field strength and average-integral electromagnetic forces, as well as visualization of the average form of electric arc column axis by the period and the heat flux distribution on a horizontal heat-absorbing surface.

**Keywords:** three-phase alternating current, magnetic field, electromagnetic forces, form of arc column axis, heat flux, mathematical model, software.

### References

1. Leushin A.I. *Duga gorennya* [The Burning Arc]. Moscow, Metallurgiya Publ., 1973, 240 p.
2. Yachikov I.M., Zaretskaya E.M. The analysis of a magnetic field behavior near electric arc furnace electrodes using mathematical modeling. *Izvestiya vysshikh uchebnykh zavedeniy. Chernaya metallurgiya* [Izvestiya. Ferrous Metallurgy]. 2011, no. 1, pp. 18–20 (in Russ.).
3. Mironov Yu.M. *Elektricheskaya duga v elektrotehnologicheskikh ustanovkakh* [An Electric Arc in Electrotechnological Installations]. Monograph. Cheboksary, Chuvash. un-t Publ., 2013, 290 p.
4. Yachikov I.M., Kostyleva E.M. Electromagnetic forces on the arc in a three-phase arc furnace. *Steel in Translation*. 2015, vol. 45, no. 7, pp. 467–472.
5. Egorov A.V. *Raschet moshchnosti i parametrov elektroplavilnykh pechey* [The Calculation of Power and Parameters of Electric Smelting Furnaces]. Moscow, MISiS Publ., 2000, 272 p.
6. Egorov A.V. *Elektroplavilnye pechi chernoy metallurgii* [Electric Melting Furnace for Ferrous Metallurgy]. Moscow, Metallurgiya Publ., 1985, 280 p.
7. Nikolsky L.E., Smolyarenko V.D., Kuznetsov L.N. *Teplovaya rabota dugovykh staleplavilnykh pechey* [Thermal Performance of Electric Arc Furnaces]. Moscow, Metallurgiya Publ., 1981, 320 p.
8. Lazarev Yu. *Modelirovanie protsessov i sistem v MATLAB* [Modeling Processes and Systems in MATLAB]. St. Petersburg, Piter Publ., 2005, 512 p.
9. Bortnichuk N.I., Krutyansky M.M. *Plazmenno-dugovye plavilnye pechi* [Plasma-arcs Melting Furnace]. Moscow, Energoizdat Publ., 1981, 120 p.
10. Azbelev N.V., Maksimov V.P., Rakhmatullina L.F. *Elementy sovremennoy teorii funktsionalno-differentsialnykh uravneniy. Metody i prilozheniya* [Elements of the Modern Theory of Functional Differential Equations. Methods and Applications]. Moscow, IKI Publ., 2002, 304 p.
11. Dmitriev S.S., Kuznetsov E.B. Numerical solution to systems of delay integrodifferential algebraic equations. *Zhurn. vychislitel'noy matematiki i matematicheskoy fiziki* [Computational Mathematics and Mathematical Physics]. 2008, vol. 48, no. 3, pp. 406–419.
12. Verzhbitsky V.M. *Chislennyye metody. Matematichesky analiz i obyknovennyye differentsialnye uravneniya* [Numerical Methods. Mathematical Analysis and Ordinary Differential Equations]. Moscow, Vyssh. shk. Publ., 2001, 383 p.
13. Fedorenko R.P. *Vvedenie v vychislitel'nuyu fiziku* [An Introduction to Computational Physics]. 2nd ed., Intellect Publ., 2008, 504 p.
14. Zaliznyak V.E. *Osnovy nauchnykh vychisleniy. Vvedenie v chislennyye metody dlya fizikov* [Fundamentals of Scientific Computing. Introduction to Numerical Methods for Physicists]. URSS Publ., 2002, 296 p.
15. Bakhvalov N.S., Zhidkov N.P., Kobelkov G.M. *Chislennyye metody* [Numerical Methods]. Moscow, Binom. Laboratoriya znany Publ., 2003, 632 p.
16. Lezin I.A. Solving simultaneous polynomial equations using a computer. *Programmnye produkty i sistemy* [Software & Systems]. Tver, 2012, no. 3, pp. 22–25 (in Russ.).
17. Yachikov I.M., Kostyleva E.M. The mathematical modeling of arcs' form during their electromagnetic coupling. *Message 2. Izvestiya vysshikh uchebnykh zavedeniy. Chernaya metallurgiya* [Izvestiya. Ferrous Metallurgy]. 2014, vol. 57, no. 5, pp. 56–61.
18. Makarov A.N. *Teploobmen v dugovykh staleplavilnykh pechakh* [Heat Exchange in Arc Steel-Melting Furnaces]. Tver, TSTU Publ., 1998, 184 p.
19. Mironov Yu.M., Mikhadarov D.G. The influence of the interelectrode gap on the deviation of an arc column. *Vestn. ChGU* [Cherepovets State Univ. Bulletin]. 2009, no. 2, pp. 139–142 (in Russ.).
20. Yachikov I.M., Zaretskaya E.M. Radiant heat flux from an electric arc inclined to the plane of the receiving surface. *Steel in Translation*. 2011, vol. 41, no. 7, pp. 550–554.
21. Makarov A.N., Svenchansky A.D. *Optimalnye teplovye rezhimy dugovykh staleplavilnykh pechey* [Optimum Thermal Conditions of Electric Arc Furnaces]. Moscow, Energoatomizdat Publ., 1992, 96 p.

УДК 537.9: 004.942

DOI: 10.15827/0236-235X.119.546-552

Дата подачи статьи: 08.02.17

2017. Т. 30. № 3. С. 546–552

## **ПОСТРОЕНИЕ И ИССЛЕДОВАНИЕ ЛИНЕЙНЫХ РЕГРЕССИОННЫХ МОДЕЛЕЙ ЭЛЕКТРИЧЕСКОЙ ЕМКОСТИ МИКРОСЛОЕВ МАГНИТНЫХ НАНОЖИДКОСТЕЙ**

*Т.Ф. Морозова, к.ф.-м.н., доцент, morozova2701005@mail.ru;*

*М.С. Демин, к.ф.-м.н., доцент, demin\_ms@mail.ru;*

*А.С. Морозов, магистрант, timss1993@yandex.ru*

*(Северо-Кавказский федеральный университет, ул. Пушкина, 1, г. Ставрополь, 355009, Россия)*

В работе представлены результаты синтеза и анализа линейных регрессионных моделей, описывающих экспериментальные исследования электрической емкости ячейки с микрослоем магнитной наножидкости, проведенные при изменении входных параметров (факторов): поляризующего напряжения (0, ..., 20 В), объемной концентрации дисперсной фазы (2,3, ..., 14,4 %), межэлектродного расстояния (100, ..., 200 мкм), температуры слоя (293, ..., 368 К).

Приведены двухмерные и трехмерные графические зависимости электрической емкости микрослоя магнитной наножидкости в зависимости от входных параметров. Анализ зависимостей показал наличие максимума электрической емкости при температуре ~340 К в исследуемом концентрационном диапазоне. При изменении концентрации дисперсной фазы от 2,3 % до 14,4 % и приложении поляризующего напряжения также имеются характерные максимумы электрической емкости, которые при повышении температуры становятся более выраженными. С увеличением концентрации дисперсной фазы возрастает величина поляризующего напряжения, при котором обнаруживается максимум электрической емкости.

Для получения математической модели, связывающей электрическую емкость ячейки с магнитной наножидкостью и ее параметрами (факторами), использовалось ПО для статистического анализа IBM SPSS Statistics. Получены регрессионные модели, учитывающие главные факторы и эффекты от их взаимодействия. Произведен анализ полученных моделей, оценены их погрешности (ошибки аппроксимации).

При оценке погрешности рассматривались минимальная, максимальная и средняя погрешности, на основе этих данных делались выводы о целесообразности применения модели. Наилучшую аппроксимацию экспериментальных данных дала модель, учитывающая эффекты от взаимодействия факторов.

**Ключевые слова:** магнитная наножидкость, концентрация, микрослой, поляризация, электрическое поле, регрессионная модель, аппроксимация.

Объектом исследования являлись *магнитные наножидкости* (МНЖ), представляющие собой высокодисперсные коллоиды ферромагнетиков; их электрофизические параметры (такие как электрическая емкость) изменяются при изменении концентрации дисперсной фазы, температуры и напряженности электрического и магнитного полей. Актуальной является задача получения математической модели, которая могла бы описать изменения величины электрической емкости МНЖ при различных воздействиях. Для идентификации сложных объектов, к которым можно отнести и МНЖ, широко используются экспериментально-статистические методы, которые позволяют установить зависимости между входными параметрами (факторами) и выходными (показателями функционирования объекта) в виде уравнений регрессии [1–7]. Свойства объекта исследования возможно описать различными моделями, при этом они должны удовлетворять требованиям адекватности, содержательности и быть максимально простыми.

В работах [7, 8] проведена статистическая обработка экспериментальных исследований микрослоев МНЖ, которая заключалась в получении математической модели, описывающей взаимосвязь между величиной электрической емкости ячейки с микрослоем МНЖ и входными параметрами: объ-

емной концентрацией дисперсной фазы ( $\phi$ , %), межэлектродным расстоянием ( $d$ , мм), величиной приложенного поляризующего напряжения ( $U_p$ , В) и температурной средой ( $t$ , °С). Представлены и проанализированы результаты экспериментальных исследований микрослоев МНЖ [7, 9] с более чем 2 000 проведенных опытов при изменении указанных параметров ячейки. Характерные зависимости проведенных экспериментальных исследований представлены на рисунках 1 и 2, которые показывают, что подача постоянного поляризующего напряжения на микрослой МНЖ приводит к нелинейному изменению электроемкости с ярко выраженным максимумом. Диапазон входных параметров составляет: а) для прикладываемого поляризующего напряжения  $U_p$  от 0 до 20 В; б) для межэлектродного расстояния от 0,1 до 0,2 мм; в) для объемной концентрации дисперсной фазы  $\phi$  от 2,3 до 14,4 %; г) для температуры МНЖ от 20 до 95° С. Микрослой МНЖ типа «магнетит в керосине» создавался между двумя электродами, в качестве которых применялись стеклянные пластины с проводящей поверхностью  $\text{In}_2\text{O}_3\text{-SnO}_2$  толщиной до 0,4 мкм. Толщина микрослоя (0,1, ..., 0,2) мм задавалась введением фторопластовых пленок, размеры электродов (40×50) мм<sup>2</sup>.

В работе [10] выдвинуто предположение о том, что поляризация в МНЖ обусловлена преимуще-

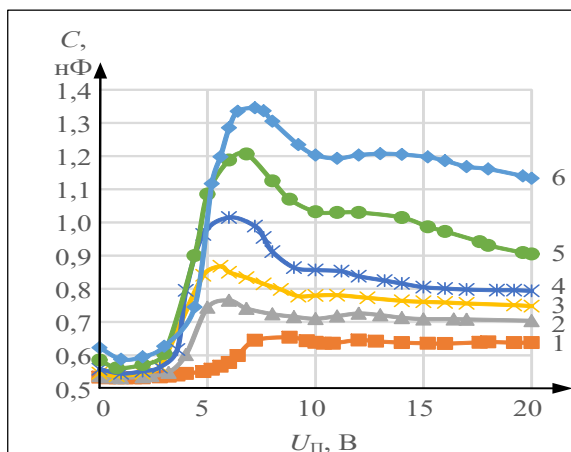


Рис. 1. Характерное изменение емкости  $C = f(U_{\text{п}})$  микрослоя МНЖ  $d = 100$  мкм с концентрацией дисперсной фазы 8,3 %: 1 – 293 К, 2 – 318 К, 3 – 328 К, 4 – 338 К, 5 – 358 К, 6 – 368 К

Fig. 1. Typical capacitance change  $C = f(U_{\text{п}})$  of 100- $\mu\text{m}$ -thick MNF layer with a disperse phase concentration of 8,3 %. MNF temperature is as follows: 1 – 293 K, 2 – 318 K, 3 – 328 K, 4 – 338 K, 5 – 358 K, 6 – 368 K

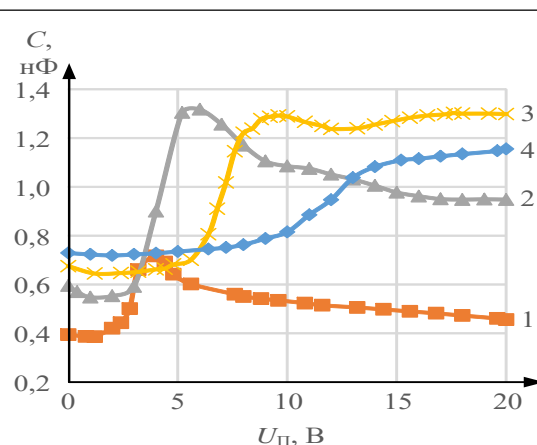


Рис. 2. Характерные зависимости емкости  $C = f(U_{\text{п}})$  из концентрационного ряда МНЖ «магнетит в керосине» при  $T = 368$  К,  $d = 100$  мкм: 1 – 2,3 %, 2 – 6,3 %, 3 – 10,3 %, 4 – 14,4 %

Fig. 2. Typical capacitance dependences  $C = f(U_{\text{п}})$  of 100- $\mu\text{m}$ -thick MNF layer with MNF temperature of 368 K with the following solid concentrations: 1 – 2,3 %, 2 – 6,3 %, 3 – 10,3 %, 4 – 14,4 %

ственно двумя механизмами поляризации. Аналогичные МНЖ исследовались в работе [11], в которой экспериментально полученные частотные зависимости емкости и сопротивления ячейки с МЖ с высокой точностью описываются двухкомпонентной моделью жидкости: с «быстрыми» и «медленными» носителями заряда и соответственно двумя механизмами поляризации.

Анализ приведенных зависимостей показал наличие максимума электрической емкости при температуре 338 К в исследуемом концентрационном диапазоне. При воздействии поляризующего напряжения  $\sim 5$  В и температуре  $\sim 340$  К наблюдается ярко выраженный максимум электрической емкости, дальнейшее повышение температуры приводит к монотонному увеличению электрической емкости вплоть до 368 К (рис. 1, 3).

При изменении концентрации дисперсной фазы от 2,3 % до 14,4 % и приложении поляризующего напряжения также имеются характерные максимумы электрической емкости, при повышении температуры максимум становится более выраженным (рис. 1, 4). При увеличении концентрации дисперсной фазы возрастает величина поляризующего напряжения, при котором достигается максимум электрической емкости (рис. 2, 4).

Для получения математической модели, связывающей электрическую емкость ячейки с МНЖ и ее параметрами (концентрация дисперсной фазы, межэлектродное расстояние, прикладываемое к электродам напряжение и температура микрослоя МНЖ), использовалось ПО для статистического анализа, реализующее необходимые базовые функции для его проведения IBM SPSS Statistics [12–17].

Были получены регрессионные модели, учитывающие главные факторы и эффекты взаимодействия факторов. Произведен анализ полученных моделей, оценены их погрешности (ошибки аппроксимации). При оценке погрешности рассматривались минимальная, максимальная и средняя погрешности, на основе этих данных делались выводы о целесообразности применения модели.

Для линейной регрессионной модели [8], полученной в программе регрессионного анализа LRSVD, произведен расчет ее погрешностей, из которого следует, что линейная аппроксимация зависимости  $C = f(\varphi; d; U_{\text{п}}; T)$  при описании экспериментальных данных дает среднюю ошибку в 34,7 % и для отдельных опытов достигает 100 %, что не может считаться удовлетворительным.

С использованием IBM SPSS была также получена линейная модель  $C = f(\varphi; d; U_{\text{п}}; T)$ , которая позволила снизить среднюю ошибку до 17,978 % и максимальную до 78,849 %:

$$C = -135,920 + 24,013 \cdot \varphi - 3501,599 \cdot d + 8,636 \cdot U + 2,829 \cdot T. \quad (1)$$

Несмотря на значительное повышение точности модели, ее применение не является целесообразным, следовательно, необходимо усложнять модель для повышения ее адекватности, поэтому было принято решение учесть в модели эффекты взаимодействия факторов.

Для построения моделей в IBM SPSS предварительно температура из градусов Цельсия была переведена в единицы термодинамической температуры Кельвина. Далее произведено кодирование факторов (варьирование кодированных факторов

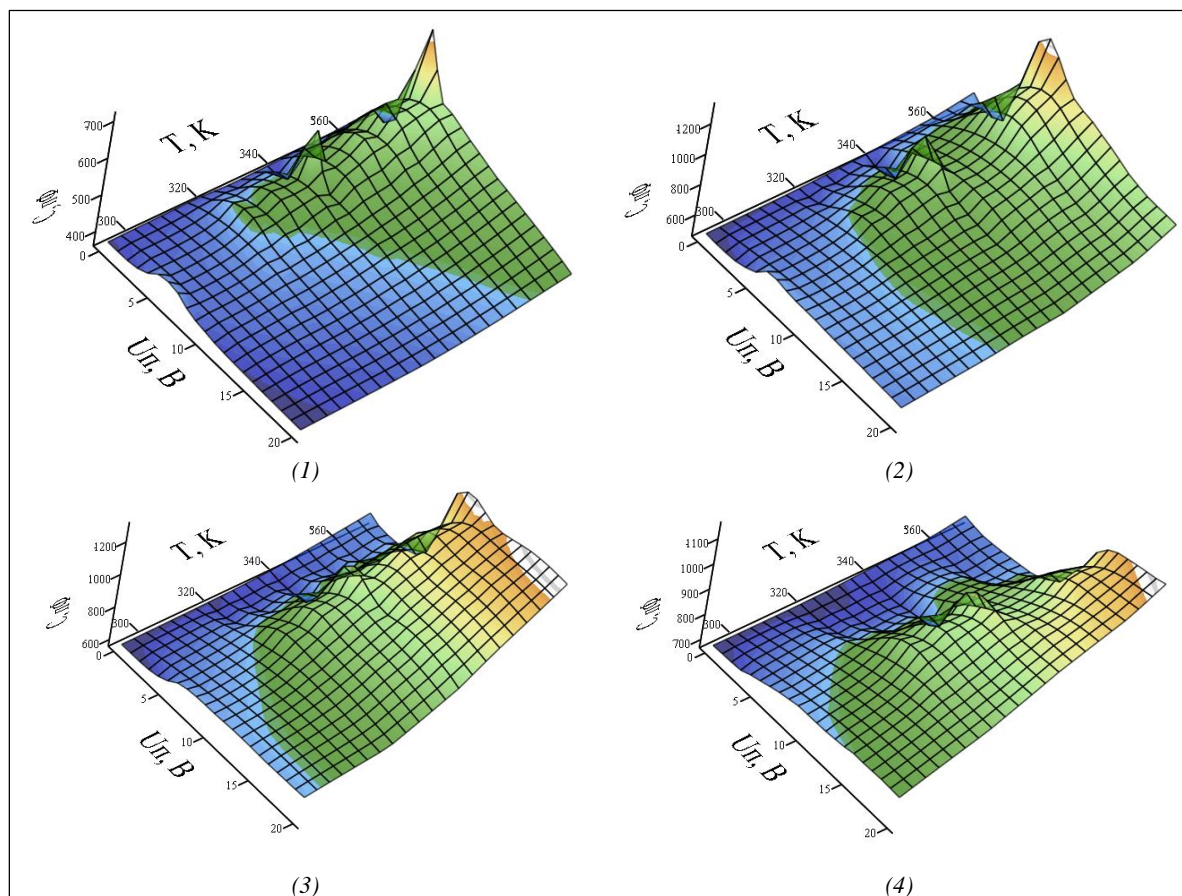


Рис. 3. Зависимости емкости  $C = f(T; U_{\pi})$  при  $d = 100$  мкм и  $\varphi$ : (1) 2,3 %, (2) 6,3 %, (3) 10,3 %, (4) 14,4 %

Fig. 3. Dependences  $C = f(T; U_{\pi})$  of 100- $\mu$ m-thick MNF layer with the following solid concentrations  $\varphi$ : (1) 2,3 %, (2) 6,3 %, (3) 10,3 %, (4) 14,4 %

производилось на уровнях от  $-1$  до  $+1$ ). Данные об электрической емкости микрослоя МНЖ при соответствующих значениях факторов (объемной концентрации дисперсной фазы  $\varphi[-1; +1]$ , межэлектродным расстоянием  $d[-1; +1]$ , величиной приложенного поляризирующего напряжения  $U_{\pi}[-1; +1]$  и температурной средой  $T[-1; +1]$ ) передавались в IBM SPSS. В программе были получены расчетные оценки коэффициентов регрессионной модели и их стандартные ошибки на 95-процентном доверительном интервале. Построение линейной модели производилось с учетом эффектов взаимодействия факторов. По полученным оценкам проводилась проверка по критерию Стьюдента для уровня доверия 0,95 с целью определения значимых коэффициентов регрессии, незначимые коэффициенты регрессии отбрасывались (за исключением постоянной составляющей  $E0$ ), а значимые заново пересчитывались (табл. 1). В результате была получена следующая модель:

$$C = -777,021 + 67,143 \cdot \varphi + 2190,186 \cdot d + 4,513 \cdot T - 0,161 \cdot \varphi \cdot T - 13,863 \cdot d \cdot T - 11,486 \cdot U \cdot \varphi + 37,375 \cdot \varphi \cdot d \cdot U + 0,044 \cdot \varphi \cdot T \cdot U - 0,151 \cdot \varphi \cdot d \cdot T \cdot U, \quad (2)$$

которая позволила более чем на 20 % снизить среднюю и максимальную ошибки по сравнению с моделью (1) до 13,027 % и 61,464 % соответственно.

Таблица 1

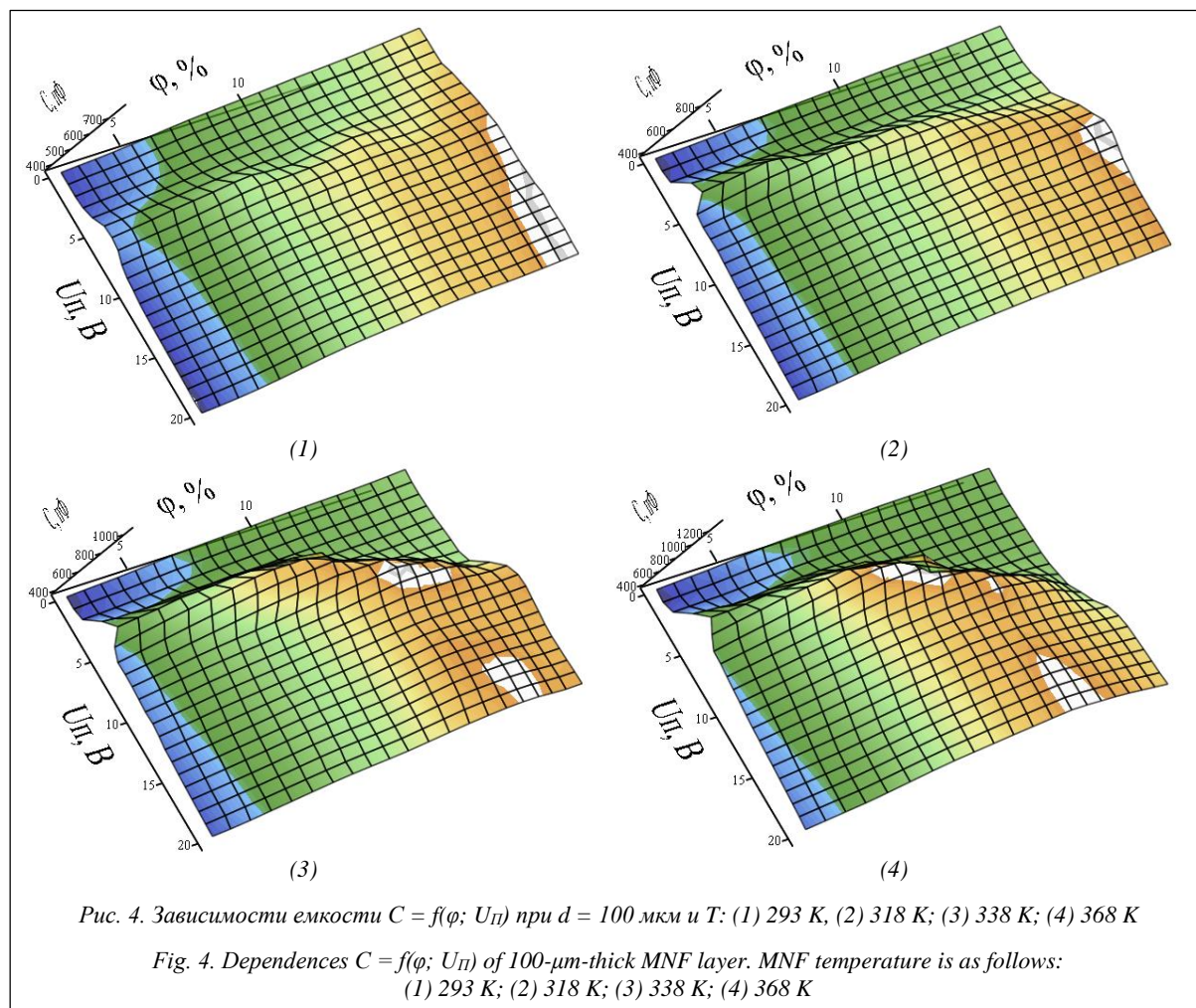
Оценки коэффициентов регрессионной модели (2)

Table 1

Estimates of (2) regression model coefficients

Коэффициент уравнения регрессии	Фактор или эффект взаимодействия факторов	Оценка	Стандартная ошибка	t-критерий
$E0$	—	-777,021	157,454	4,93
$fl$	$\varphi$	67,143	10,093	6,65
$d1$	$d$	2190,186	944,090	2,32
$t1$	$T$	4,513	0,469	9,63
$fltl$	$\varphi \cdot T$	-0,161	0,030	5,36
$d1tl$	$d \cdot T$	-13,863	2,811	4,93
$ulfl$	$U \cdot \varphi$	-11,488	1,494	7,69
$fldlul$	$\varphi \cdot d \cdot U$	37,375	9,287	4,02
$fltlul$	$\varphi \cdot T \cdot U$	0,044	0,004	9,80
$fldl1ul$	$\varphi \cdot d \cdot T \cdot U$	-0,151	0,028	5,48





Учет эффектов взаимодействия факторов позволил уменьшить погрешности модели, однако для практического применения достигнутых результатов недостаточно. Поскольку ранее [10, 11] отмечалось наличие в МНЖ по крайней мере двух механизмов поляризации, целесообразным представляется рассмотрение модели для двух диапазонов. Поэтому на следующем этапе диапазон изменения поляризующего напряжения был разделен на два участка: (0...10) В и (10...20) В, для которых были синтезированы линейные модели и модели ЛУЭВФ (линейные с учетом эффектов взаимодействия факторов).

Для участка изменения  $U_{\Pi}$  (0...10) В получены оценки параметров моделей (табл. 2) с учетом их значимости следующих видов.

Линейная модель:

$$C = 11,599 + 16,308 \cdot \varphi - 3237,896 \cdot d + 23,704 \cdot U + 2,238 \cdot T. \quad (3a)$$

Модель ЛУЭВФ:

$$C = 72,251 + 82,538 \cdot \varphi - 330,937 \cdot U + 1,618 \cdot T - 309,084 \cdot \varphi \cdot d - 0,203 \cdot \varphi \cdot T - 6,338 \cdot d \cdot T + 4,308 \cdot U \cdot \varphi + 1255,383 \cdot U \cdot d + 1,055 \cdot U \cdot T + 0,908 \cdot \varphi \cdot d \cdot T - 3,789 \cdot d \cdot T \cdot U - 0,077 \cdot \varphi \cdot d \cdot T \cdot U. \quad (4a)$$

Оценки коэффициентов регрессионных моделей (3a) и (4a)

Estimates of (3a) and (4a) regression model coefficients

Коэффициент уравнения регрессии	Фактор или эффект взаимодействия факторов	Оценка	Стандартная ошибка	t-критерий
Для модели (3a)				
$E0$	—	11,599	46,403	0,25
$fl$	$\varphi$	16,308	0,823	19,83
$dl$	$d$	-3237,896	65,249	49,62
$ul$	$U$	23,704	1,103	21,49
$tl$	$T$	2,238	0,133	16,80
Для модели (4a)				
$E0$	—	72,251	108,509	0,67
$fl$	$\varphi$	82,538	23,536	3,51
$ul$	$U$	-330,937	32,847	10,08
$tl$	$T$	1,618	0,316	5,11
$fldl$	$\varphi \cdot d$	-309,084	139,993	2,21
$fltl$	$\varphi \cdot T$	-0,203	0,069	2,94
$dltl$	$d \cdot T$	-6,338	0,692	9,17
$ulfl$	$U \cdot \varphi$	4,308	0,725	5,94
$uldl$	$U \cdot d$	1255,383	203,411	6,17
$ultl$	$U \cdot T$	1,055	0,098	10,82
$fldltl$	$\varphi \cdot d \cdot T$	0,908	0,422	2,15
$fldltlul$	$\varphi \cdot d \cdot T \cdot U$	-0,077	0,014	5,66
$dltlul$	$d \cdot T \cdot U$	-3,789	0,621	6,10

Оценки коэффициентов регрессионных моделей (36) и (46)

Таблица 3

Estimates of (4a) and (46) regression model coefficients

Table 3

Коэффициент уравнения регрессии	Фактор или эффект взаимодействия факторов	Оценка	Стандартная ошибка	t-критерий
Для модели (36)				
$E0$	–	-334,736	51,387	6,51
$fl$	$\varphi$	33,540	0,872	38,48
$dl$	$d$	-3849,494	68,244	56,41
$ul$	$U$	-1,335	1,074	1,24
$tl$	$T$	3,764	0,142	26,52
Для модели (46)				
$E0$	–	-2940,356	620,990	4,73
$fl$	$\varphi$	145,865	43,027	3,39
$dl$	$d$	9949,851	3326,599	2,99
$ul$	$U$	156,873	43,393	3,62
$tl$	$T$	11,167	1,844	6,05
$fldl$	$\varphi \cdot d$	-267,769	60,315	4,44
$fltl$	$\varphi \cdot T$	-0,268	0,126	2,12
$dltl$	$d \cdot T$	-35,766	9,762	3,66
$ulfl$	$U \cdot \varphi$	-18,103	3,338	5,42
$uldl$	$U \cdot d$	-544,553	239,504	2,27
$ultl$	$U \cdot T$	-0,523	0,128	4,08
$fldlul$	$\varphi \cdot d \cdot U$	58,313	12,353	4,72
$fltlul$	$\varphi \cdot T \cdot U$	0,058	0,010	5,97
$fldltlul$	$\varphi \cdot d \cdot T \cdot U$	-0,181	0,034	5,27
$dltlul$	$d \cdot T \cdot U$	1,786	0,701	2,55

Сравнение погрешностей моделей

Таблица 4

Comparison of model errors

Table 4

Модель	Погрешность модели	Фактическое значение, %	Изменение погрешности по отношению к линейной (1), %
Линейная (1)	Минимальная	0,021	–
	Максимальная	78,849	–
	Средняя	17,978	–
Линейная (1) от 0 до 10 В	Минимальная	0,021	–
	Максимальная	78,849	–
	Средняя	19,864	–
Линейная (1) от 10 до 20 В	Минимальная	0,024	–
	Максимальная	72,917	–
	Средняя	15,528	–
ЛУЭВФ (2)	Минимальная	0,02	-4,76
	Максимальная	61,464	-22,05
	Средняя	13,027	-27,54
ЛУЭВФ (2) от 0 до 10 В	Минимальная	0,034	61,90
	Максимальная	60,232	-23,61
	Средняя	14,3	-28,01
ЛУЭВФ (2) от 10 до 20 В	Минимальная	0,02	-16,67
	Максимальная	61,464	-15,71
	Средняя	11,478	-26,08
Линейная на интервале от 0 до 10 В (3а)	Минимальная	0,005	-76,19
	Максимальная	74,687	-5,28
	Средняя	15,784	-20,54
ЛУЭВФ на интервале от 0 до 10 В (4а)	Минимальная	0,00032	-98,48
	Максимальная	72,425	-8,15
	Средняя	10,817	-45,54
Линейная на интервале от 10 до 20 В (3б)	Минимальная	0,025	4,17
	Максимальная	79,078	8,45
	Средняя	15,071	-2,94
ЛУЭВФ на интервале от 10 до 20 В (4б)	Минимальная	0,028	16,67
	Максимальная	55,378	-24,05
	Средняя	8,695	-44,00

Для участка изменения  $U_{\text{п}}$  (10...20) В получены оценки параметров моделей (табл. 3) с учетом их значимости следующих видов.

Линейная модель:

$$C = -334,736 + 33,54 \cdot \varphi - 3849,494 \cdot d - 1,335 \cdot U + 3,764 \cdot T. \quad (36)$$



Модель ЛУЭВФ:

$$C = -2940,356 + 145,865 \cdot \varphi + 9949,851 \cdot d + \\ + 156,873 \cdot U + 11,167 \cdot T - 267,769 \cdot \varphi \cdot d - \\ - 0,268 \cdot \varphi \cdot T - 35,766 \cdot d \cdot T - 18,103 \cdot U \cdot \varphi - \\ - 544,553 \cdot U \cdot d - 0,523 \cdot U \cdot T + 58,313 \cdot \varphi \cdot d \cdot U + \\ + 0,058 \cdot \varphi \cdot T \cdot U + 1,786 \cdot d \cdot T \cdot U - 0,181 \cdot \varphi \cdot d \cdot T \cdot U. \quad (46)$$

Данные о полученных моделях и сравнение их с линейной (1) моделью сведен в таблицу 4, где в столбце 4 приводится изменение погрешности моделей по отношению к линейной модели (1).

Анализ таблицы 4 позволил сделать следующие выводы:

- линейные модели (1) и (3) во всем диапазоне изменения поляризующего напряжения и при делении его на два интервала не позволили снизить среднюю ошибку аппроксимации менее чем на 15 %;

- модель ЛУЭВФ (2) позволила уменьшить среднюю ошибку аппроксимации на ~26 % по отношению к модели (1);

- наилучшие результаты были получены по модели ЛУЭВФ (4) на интервалах изменения поляризующего напряжения (0...10 В) и (10...20 В), где среднюю ошибку аппроксимации удалось снизить до 10,817 % и 8,695 % соответственно, а сравнение с линейной моделью (1) показало ее уменьшение на ~45 %.

Таким образом, наилучшую аппроксимацию экспериментальных данных дает модель (4), учитывающая эффекты взаимодействия факторов с разбиением на два интервала.

Из проведенного анализа следует, что для более точного описания зависимости электрической емкости ячейки с МНЖ от толщины микрослоя, температуры среды, величины поляризующего напряжения и объемной концентрации дисперсной фазы необходимо применение нелинейных регрессионных моделей.

#### Литература

1. Круг Г.К., Сосулин Ю.А., Фатуев В.А. Планирование эксперимента в задачах идентификации и экстраполяции. М.: Наука, 1977. 208 с.

2. Дрейпер Н., Смит Г. Прикладной регрессионный анализ. Множественная регрессия. М.: Диалектика, 2007. 912 с.

3. Фёрстер Э., Рёнц Б. Методы корреляционного и регрессионного анализа. М.: Финансы и статистика, 1981. 302 с.

4. Гайдышев И.П. Анализ и обработка данных: специальный справочник. СПб: Питер, 2001. 752 с.

5. Кендалл М., Стьюарт А. Теория распределений. М.: Наука, 1966. 566 с.

6. Кендалл М., Стьюарт А. Статистические выводы и связи. М.: Наука, 1973. 899 с.

7. Кожевников В.М., Морозова Т.Ф., Филиппов С.А. Статистическая обработка результатов исследования электрофизических свойств тонких слоев магнитной жидкости // Сб. науч. тр. СтГУ: Естественная наука. 1999. Вып. 2. С. 104–107.

8. Морозова Т.Ф., Демин М.С., Морозов А.С. Регрессионный анализ экспериментальных данных электрофизических свойств тонких слоев магнитной жидкости // Наука в современном информационном обществе: матер. IX Междунар. науч.-практич. конф. США, Северный Чарльстон. Изд-во CreateSpace, 2016. С. 136–139.

9. Kozhevnikov V.M., Morozova T.F. Dielectric permittivity of a magnetic fluid stratum in electric and magnetic fields. *Magnetohydrodynamics*, 2001, vol. 37, no. 4, pp. 383–388.

10. Морозова Т.Ф., Демин М.С. Анализ взаимосвязи процессов поляризации с микроструктурированием в слое магнитной жидкости // Журн. технич. физики. 2017. Т. 87. № 2. С. 286–293.

11. Кожевников В.М., Ларионов Ю.А., Демин М.С. Электрокинетические параметры магнитоэлектрического коллоида в нестационарных режимах при воздействии электрического и магнитного полей // Вестн. Сев.-Кавказ. гос. технич. ун-та. 2007. № 1. С. 56–61.

12. Бююль А., Цефель П. SPSS: искусство обработки информации. Анализ статистических данных и восстановление скрытых закономерностей. СПб: ДиаСофтЮп, 2005. 608 с.

13. Пользовательские таблицы 24 IBM SPSS. URL: [ftp://public.dhe.ibm.com/software/analytics/spss/documentation/statistics/24.0/ru/client/Manuals/IBM\\_SPSS\\_Custom\\_Tables.pdf](ftp://public.dhe.ibm.com/software/analytics/spss/documentation/statistics/24.0/ru/client/Manuals/IBM_SPSS_Custom_Tables.pdf) (дата обращения: 01.10.2016).

14. IBM SPSS Statistics Base 24. URL: [ftp://public.dhe.ibm.com/software/analytics/spss/documentation/statistics/24.0/ru/client/Manuals/IBM\\_SPSS\\_Statistics\\_Base.pdf](ftp://public.dhe.ibm.com/software/analytics/spss/documentation/statistics/24.0/ru/client/Manuals/IBM_SPSS_Statistics_Base.pdf) (дата обращения: 01.10.2016).

15. IBM SPSS Statistics 24: Руководство пользователя. URL: [ftp://public.dhe.ibm.com/software/analytics/spss/documentation/statistics/24.0/ru/client/Manuals/IBM\\_SPSS\\_Statistics\\_Core\\_System\\_User\\_Guide.pdf](ftp://public.dhe.ibm.com/software/analytics/spss/documentation/statistics/24.0/ru/client/Manuals/IBM_SPSS_Statistics_Core_System_User_Guide.pdf) (дата обращения: 01.10.2016).

16. Elliott A.C., Woodward W.A. IBM SPSS by example: a practical guide to statistical data analysis. SAGE Publ., Inc., 368 p.

17. Meyers L.S., Gamst G.C., Guarino A.J. Performing data analysis using IBM SPSS. Wiley Publ., 736 p.

#### DEVELOPMENT AND INVESTIGATION OF LINEAR REGRESSION MODELS OF CAPACITANCE OF MAGNETIC NANOFLUIDS MICROLAYERS

**T.F. Morozova**<sup>1</sup>, Ph.D. (Physics and Mathematics), Associate Professor, [morozova2701005@mail.ru](mailto:morozova2701005@mail.ru)

**M.S. Demin**<sup>1</sup>, Ph.D. (Physics and Mathematics), Associate Professor, [demin\\_ms@mail.ru](mailto:demin_ms@mail.ru)

**A.S. Morozov**<sup>1</sup>, Graduate Student, [timss1993@yandex.ru](mailto:timss1993@yandex.ru)

<sup>1</sup> North-Caucasus Federal University, Pushkin St. 1, Stavropol, 355009, Russian Federation

**Abstract.** The paper presents the results of the analysis and synthesis of linear regression models, which describe experimental studies of electric capacity of a cell with a magnetic nanofluid (MNF) microlayer, when changing the following input

parameters (factors): a polarizing voltage (0...20 V), a dispersed phase volume concentration (2,3...14,4 %), an interelectrode distance (100...200  $\mu\text{m}$ ), layer temperature (293...368 K).

The article shows two- and three-dimensional MNF microlayer capacitance curves depending on input parameters. Dependence analysis showed high electrical capacity at ~340 K within the studied concentration range. While changing a dispersed phase concentration from 2,3 % to 14,4 % and using a polarizing voltage, there is also characteristic electrical capacitance maximum which becomes more evident at higher temperatures. With increasing dispersed phase concentration, the magnitude of polarizing voltage increases when the electrical capacitance maximum is detected.

To get a mathematical model linking the cell capacitance with MNF and its parameters (factors), the authors used IBM SPSS Statistics (the software for statistical analysis). The obtained regression models take into account the main factors and the effects of their interaction. The paper shows the analysis of the obtained models and estimates their errors (approximation errors).

While evaluating errors, the authors considered the minimum, maximum and average errors. Based on these data they made the conclusions on the model appropriateness. The best approximation of the experimental data is in the model that takes into account the effects of factor interaction.

**Keywords:** magnetic nanofluids, concentration, microlayer, polarization, electric field, regression model, approximation.

### References

1. Krug G.K., Sosulin Yu.A., Fatuev V.A. *Planirovanie eksperimenta v zadachakh identifikatsii i ekstrapolyatsii* [Planning an Experiment in Identification and Extrapolation Problems]. Moscow, Nauka Publ., 1977, 208 p.
2. Dreyper N., Smit G. *Prikladnoy regressionny analiz. Mnozhestvennaya regressiya* [Applied Regression Analysis. Multiple Regression]. 3-e izd. Moscow, Dialektika Publ., 2007, 912 p.
3. Forster E., Ronz B. *Metody korrelyatsionnogo i regressionnogo analiza* [Methods of Correlation and Regression Analysis]. Moscow, Finansy i statistika Publ., 1981, 302 p.
4. Gaydyshev I.P. *Analiz i obrabotka dannykh: spetsialny spravochnik* [Data Analysis and Processing: a Special Guide]. St. Petersburg, Piter Publ., 2001, 752 p.
5. Kendall M., Stuart A. *The Advanced Theory of Statistics*. Charles Griffin Publ., London, 1943 (Russ.ed.: Moscow, Nauka Publ., 1966, 566 p.).
6. Kendall M., Stuart A. *The Advanced Theory of Statistics. Vol. 2: Inference and Relationship*. Hafner Publ. Company, 1961 (Russ.ed.: Moscow, Nauka Publ., 1973, 899 p.).
7. Kozhevnikov V.M., Morozova T.F., Filippov S.A. Statistical processing of the results of the investigation on electrophysical properties of magnetic fluid thin layers. *Sb. nauch. tr. StGTU, ser. "Estestvennonauchnaya"* [Proc. Stavropol State Technical Univ.]. 1999, iss. 2, pp. 104–107 (in Russ.).
8. Morozova T.F., Demin M.S., Morozov A.S. Regression analysis of experimental data of electrophysical properties of thin layers of a magnetic fluid. *Nauka v sovremennom informatsionnom obshchestve: mater. IX Mezhdunar. nauch.-praktich. konf.* [Science in the Modern Information Society: Proc. 9th Int. Science and Practice Conf.]. CreateSpace Publ., 2016, pp. 136–139 (in Russ.).
9. Kozhevnikov V.M., Morozova T.F. Dielectric permittivity of a magnetic fluid stratum in electric and magnetic fields. *Magnetohydrodynamics*. 2001, vol. 37, no. 4, pp. 383–388.
10. Morozova T.F., Demin M.S. Analysis of the interrelation between the processes of polarization and microstructuring in a magnetic fluid layer. *Technical Physics*. 2017, vol. 62, iss. 2, pp. 314–321.
11. Kozhevnikov V.M., Larionov Yu.A., Demin M.S. Electrokinetic parameters of a magnetodielectric colloid in nonstationary regimes under the influence of electric and magnetic fields. *Vestn. Sev.-Kavkaz. gos. tekhnich. un-ta* [The Bulletin of North-Caucasus State Tech. Univ.]. 2007, no. 1, pp. 56–61 (in Russ.).
12. Byuyul A., Tsefel P. *SPSS: iskusstvo obrabotki informatsii. Analiz statisticheskikh dannykh i vosstanovlenie skrytykh zakonomernostey* [SPSS: the Art of Information Processing. Analysis of Statistical Data and Restoration of Hidden Patterns]. St. Petersburg, DiaSoftYup Publ., 2005, 608 p.
13. *Polzovatel'skie tablitsy 24 IBM SPSS* [24 IBM SPSS User Tables]. Available at: [ftp://public.dhe.ibm.com/software/analytics/spss/documentation/statistics/24.0/ru/client/Manuals/IBM\\_SPSS\\_Custom\\_Tables.pdf](ftp://public.dhe.ibm.com/software/analytics/spss/documentation/statistics/24.0/ru/client/Manuals/IBM_SPSS_Custom_Tables.pdf) (accessed October 1, 2016).
14. *IBM SPSS Statistics Base 24*. Available at: [ftp://public.dhe.ibm.com/software/analytics/spss/documentation/statistics/24.0/ru/client/Manuals/IBM\\_SPSS\\_Statistics\\_Base.pdf](ftp://public.dhe.ibm.com/software/analytics/spss/documentation/statistics/24.0/ru/client/Manuals/IBM_SPSS_Statistics_Base.pdf) (accessed October 1, 2016).
15. *IBM SPSS Statistics 24: Rukovodstvo polzovatelya* [IBM SPSS Statistics 24: User Guide]. Available at: [ftp://public.dhe.ibm.com/software/analytics/spss/documentation/statistics/24.0/ru/client/Manuals/IBM\\_SPSS\\_Statistics\\_Core\\_System\\_User\\_Guide.pdf](ftp://public.dhe.ibm.com/software/analytics/spss/documentation/statistics/24.0/ru/client/Manuals/IBM_SPSS_Statistics_Core_System_User_Guide.pdf) (accessed October 1, 2016).
16. Elliott A.C., Woodward W.A. *IBM SPSS by Example: A Practical Guide to Statistical Data Analysis*. 2015, 2nd ed., SAGE Publ., 368 p.
17. Meyers L.S., Gamst G.C., Guarino A.J. *Performing Data Analysis Using IBM SPSS*. Wiley Publ., 736 p.

# СОДЕРЖАНИЕ

<b>Аряшев С.И., Бобков С.Г., Зубковский П.С., Морев С.А., Рогаткин Б.Ю.</b> Высокопроизводительный микропроцессор 1890BM118 с архитектурой КОМДИВ для создания доверенных систем.....	345
<b>Гиацинтов А.М., Мамросенко К.А.</b> Воспроизведение видеоданных высокой четкости в виртуальной трехмерной среде имитационно-тренажерных систем.....	353
<b>Масальский Н.В.</b> Моделирование распределения потенциала в двухзатворном полевом нанотранзисторе со структурой кремний на изоляторе с асимметричным затвором .....	359
<b>Мадера А.Г., Решетников В.Н.</b> Многофункциональный программный комплекс теплового проектирования электронных систем: требования к архитектуре и функциональным возможностям моделирования.....	367
<b>Генов А.А., Русаков К.Д., Хиль С.Ш.</b> Идентификация состояния сложной технической системы в условиях неопределенности измерительной информации.....	373
<b>Галащенко В.А., Костюхин К.А.</b> Проблемы отладки многопроцессных систем.....	378
<b>Афанаскин И.В., Ялов П.В., Гиацинтов А.М., Родителей А.В.</b> Решение задач оптимизации при суперэлементном моделировании разработки нефтяных месторождений.....	384
<b>Годунов А.Н., Солдатов В.А., Хоменков И.И.</b> Реализация каналов спецификации ARINC 653 в операционной системе реального времени Багет 3 .....	392
<b>Слинкин Д.И.</b> Анализ современных методов тестирования и верификации проектов сверхбольших интегральных схем.....	401
<b>Байков Н.Д., Годунов А.Н.</b> Сравнение производительности отечественных и импортных микропроцессоров .....	409
<b>Ульянов С.В., Решетников А.Г., Мамаева А.А.</b> Гибридные когнитивные нечеткие системы управления автономным роботом на основе нейроинтерфейса и технологии мягких вычислений .....	420
<b>Ровбо М.А., Овсянникова Е.Е., Чумаченко А.А.</b> Обзор средств имитационного моделирования коллективов роботов с элементами социальной организации .....	425
<b>Ульянов С.В., Решетников А.Г., Кошелев К.В.</b> Разработка системы стереозрения для мобильного робота .....	435
<b>Щербаков М.В., Задиран К.С., Голубев А.В., Аль-Гунаид Моххамед Амин</b> Фреймворк для анализа и прогнозирования временных рядов при разработке компонент проактивных систем поддержки принятия решений .....	439
<b>Бакаев М.А.</b> Современные тенденции в автоматизированной оценке юзабилити и поведенческие факторы в алгоритмах поисковых систем .....	447

<b>Кораблин Ю.П., Кочергин А.С., Шипов А.А.</b> Верификация моделей систем на базе эквивалентной характеристики формул LTL.....	456
<b>Соколов А.В., Батура Т.В.</b> Разработка системы разрешения анафоры на основе методов машинного обучения.....	461
<b>Котов С.Л., Демирский А.А.</b> Информационно-аналитическая система оценивания трудозатрат и стоимости создания программных средств.....	469
<b>Амелькин С.А., Иванова О.С.</b> Расчет оптимальной зависимости цены информационного ресурса при его реализации на нескольких рынках.....	474
<b>Осочкин А.А., Фомин В.В., Флегонтов А.В.</b> Метод частотно-морфологической классификации текстов .....	478
<b>Быкова В.В., Монгуш Ч.М.</b> Алгоритмы концептуального моделирования и классификации текстов в корпусе тувинского языка .....	487
<b>Канюков С.И., Коновалов А.В., Муйземнек О.Ю.</b> Концепция получения intersubъективных ответов на нечеткие запросы к базе данных САПР технологических процессовковки .....	496
<b>Гурьянов А.В., Шукалов А.В., Жаринов И.О., Костишин М.О., Леоновец С.А.</b> Автоматизация процессов подготовки программной документации на изделия радиоэлектронной промышленности .....	504
<b>Игнатъев С.В., Тихонов В.Б., Красников А.В., Осипов А.А.</b> Пространственно-временная модель транспортной сети системы материально-технического обеспечения эксплуатации группировки радиоэлектронных средств .....	510
<b>Гурьянов А.В., Коновалов П.В., Шукалов А.В., Жаринов И.О., Леоновец С.А.</b> Автоматизация процесса формирования учетной документации с использованием базы данных радиоэлектронных компонентов .....	517
<b>Рыбаков А.А.</b> Оптимизация задачи об определении конфликтов с опасными зонами движения летательных аппаратов для выполнения на Intel Xeon Phi .....	524
<b>Вавулов О.Ю., Сухов А.В., Решетников В.Н.</b> Алгоритмы информационной оценки совместимости средств мобильной связи и станций воздушной радионавигационной службы .....	529
<b>Ячиков И.М., Костылева Е.М.</b> Комплекс программ для определения параметров электрических дуг трехфазного переменного тока, горящих на горизонтальную поверхность .....	537
<b>Морозова Т.Ф., Демин М.С., Морозов А.С.</b> Построение и исследование линейных регрессионных моделей электрической емкости микрослоев магнитных наножидкостей .....	546

## C O N T E N T

<b>Aryashev S.I., Bobkov S.G., Zubkovsky P.S., Morev S.A., Rogatkin B.Yu.</b> High-performance microprocessor 1890BM118 for trusted computing systems .....	345
<b>Giatsintov A.M., Mamrosenko K.A.</b> High resolution video playback in virtual 3d environment in training simulation systems .....	353
<b>Masalsky N.V.</b> Potential distribution simulation for a dual-gate field silicon on insulator nanotransistor with an asymmetric gate .....	359
<b>Madera A.G., Reshetnikov V.N.</b> A software complex for electronic system thermal design: requirements for architecture and functional possibilities of modeling .....	367
<b>Genov A.A., Rusakov K.D., Hill S.Sh.</b> Identification of a complex technical system functional state under conditions of measurement data ambiguity .....	373
<b>Galatenko V.A., Kostyukhin K.A.</b> Multiprocess system debugging problems .....	378
<b>Afanaskin I.V., Yalov P.V., Giatsintov A.M., Roditelev A.V.</b> Optimization problems solution based on superelement modeling of oil-field development .....	384
<b>Godunov A.N., Soldatov V.A., Homenkov I.I.</b> Channels implementation of ARINC 653 specification in RTOS Baget 3 .....	392
<b>Slinkin D.I.</b> Analysis of modern methods for vlsi project testing and verification .....	401
<b>Baykov N.D., Godunov A.N.</b> Comparison of russian and foreign microprocessor performance .....	409
<b>Ulyanov S.V., Reshetnikov A.G., Mamaeva A.A.</b> Hybrid cognitive fuzzy control systems for an autonomous robot based on neurointerface and soft computing .....	420
<b>Rovbo M.A., Ovsyannikova E.E., Chumachenko A.A.</b> Review of simulation modeling tools for robot groups with social organization elements .....	425
<b>Ulyanov S.V., Reshetnikov A.G., Koshelev K.V.</b> Development of a pattern recognition system for a mobile robot .....	435
<b>Shcherbakov M.V., Zadiran K.S., Golubev A.V., Al-Gunaid Mohammed Amin</b> A framework for analysis and forecasting of time series in the development of proactive decision support system components .....	439
<b>Bakaev M.A.</b> Modern trends in automated usability evaluation and behavioral factors in search engine algorithms .....	447
<b>Korablin Yu.P., Kochergin A.S., Shipov A.A.</b> System model verification based on equational characteristics of LTL formulas .....	456
<b>Sokolov A.V., Batura T.V.</b> Anaphor resolution system development based on machine learning methods .....	461
<b>Kotov S.L., Demirsky A.A.</b> An enterprise information system for estimating labor and software development costs .....	469

<b>Amelkin S.A., Ivanova O.S.</b> Calculation of the optimal dependency of information resource price during its marketing at several markets.....	474
<b>Osochkin A.A., Fomin V.V., Flegontov A.V.</b> Method of frequency-morphological classification of texts .....	478
<b>Bykova V.V., Mongush Ch.M.</b> Algorithms of conceptual modeling and text classification in the tuvan language corpus .....	487
<b>Kanyukov S.I., Kononov A.V., Muizemnek O.Yu.</b> A concept of intersubjective response receiving to fuzzy queries to forging computer-aided process planning database.....	496
<b>Guryanov A.V., Shukalov A.V., Zharinov I.O., Kostishin M.O., Leonovets S.A.</b> Automatization of software documentation preparation processes for radio-electronics industry devices .....	504
<b>Ignatev S.V., Tikhonov V.B., Krasnikov A.V., Osipov A.A.</b> A spatio-temporal model of a material and technical resources traffic network of a radio-electronic facilities group exploitation.....	510
<b>Guryanov A.V., Kononov P.V., Shukalov A.V., Zharinov I.O., Leonovets S.A.</b> Automated generation of accounting documents using radioelectronic components database .....	517
<b>Rybakov A.A.</b> Optimization of the problem of conflict detection with dangerous aircraft movement areas to execute on Intel Xeon Phi.....	524
<b>Vavulov O.Yu., Sukhov A.V., Reshetnikov V.N.</b> Algorithms of information estimation of compatibility of mobile telecommunication devices and aeronautical radionavigation service stations.....	529
<b>Yachikov I.M., Kostyleva E.M.</b> A software suite to determine the parameters of three-phase alternating current electric arcs burning in a horizontal surface .....	537
<b>Morozova T.F., Demin M.S., Morozov A.S.</b> Development and investigation of linear regression models of capacitance of magnetic nanofluids microlayers .....	546

## *Вниманию подписчиков*

Международный журнал  
«Программные продукты и системы»  
распространяется по подписке  
в странах СНГ и за их пределами.  
В Каталоге подписных изданий  
Агентства «Роспечать» индекс журнала  
70799

*Телефон (482-2) 39-91-49*

*Факс (482-2) 39-91-00*

[www.swsys.ru](http://www.swsys.ru)

[red@cps.tver.ru](mailto:red@cps.tver.ru)