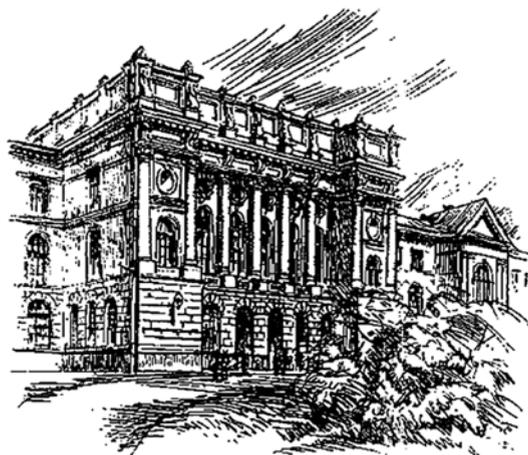


МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ



# НАУЧНО-ТЕХНИЧЕСКИЕ ВЕДОМОСТИ

САНКТ-ПЕТЕРБУРГСКОГО ГОСУДАРСТВЕННОГО  
ПОЛИТЕХНИЧЕСКОГО УНИВЕРСИТЕТА

---

---

Информатика. Телекоммуникации.  
Управление

---

---

**3(174) 2013**

Издательство Политехнического университета  
Санкт-Петербург  
2013

# НАУЧНО-ТЕХНИЧЕСКИЕ ВЕДОМОСТИ САНКТ-ПЕТЕРБУРГСКОГО ГОСУДАРСТВЕННОГО ПОЛИТЕХНИЧЕСКОГО УНИВЕРСИТЕТА

## РЕДАКЦИОННЫЙ СОВЕТ

*Васильев Ю.С.*, академик РАН (председатель); *Алферов Ж.И.*, академик РАН;  
*Костюк В.В.*, академик РАН; *Лопота В.А.*, чл.-кор. РАН;  
*Окрепилов В.В.*, академик РАН; *Рудской А.И.*, чл.-кор. РАН;  
*Патон Б.Е.*, академик НАН Украины и РАН; *Федоров М.П.*, академик РАН;  
*Фортов В.Е.*, академик РАН.

## РЕДАКЦИОННАЯ КОЛЛЕГИЯ

*Васильев Ю.С.*, академик РАН (главный редактор); *Арсеньев Д.Г.*, д-р техн. наук, профессор;  
*Бабкин А.В.*, д-р экон. наук, профессор (зам. гл. редактора);  
*Боронин В.Н.*, д-р техн. наук, профессор; *Глухов В.В.*, д-р экон. наук, профессор;  
*Дегтярева Р.В.*, д-р ист. наук, профессор; *Иванов А.В.*, д-р техн. наук, профессор;  
*Иванов В.К.*, д-р физ.-мат. наук, профессор; *Козловский В.В.*, д-р физ.-мат. наук, профессор;  
*Рудской А.И.*, чл.-кор. РАН (зам. гл. редактора); *Юсупов Р.М.*, чл.-кор. РАН.

## ИНФОРМАТИКА. ТЕЛЕКОММУНИКАЦИИ. УПРАВЛЕНИЕ

### РЕДАКЦИОННЫЙ СОВЕТ ЖУРНАЛА

*Юсупов Р.М.*, чл.-кор. РАН – председатель;  
*Абрамов С.М.*, чл.-кор. РАН;  
*Воеводин В.В.*, чл.-кор. РАН;  
*Заборовский В.С.*, д-р техн. наук, профессор;  
*Козлов В.Н.*, д-р техн. наук, профессор;  
*Фотиади А.Э.*, д-р физ.-мат. наук, профессор;  
*Черноруцкий И.Г.*, д-р техн. наук, профессор.

### РЕДАКЦИОННАЯ КОЛЛЕГИЯ ЖУРНАЛА

*Юсупов Р.М.*, чл.-кор. РАН – председатель;  
*Арсеньев Д.Г.*, д-р техн. наук, профессор – зам. председателя;  
*Бабкин А.В.*, д-р экон. наук, профессор – зам. председателя;  
*Антонов В.И.*, д-р техн. наук, профессор;  
*Голландцев Ю.А.*, д-р техн. наук, профессор;  
*Карпов Ю.Г.*, д-р техн. наук, профессор;  
*Коротков А.С.*, д-р техн. наук, профессор;  
*Макаров С.Б.*, д-р техн. наук, профессор;  
*Устинов С.М.*, д-р техн. наук, профессор;  
*Цикин И.А.*, д-р техн. наук, профессор;  
*Шкодыврев В.П.*, д-р техн. наук, профессор;  
*Клавдиев В.Е.*, канд. техн. наук, доцент.

*Журнал с 1995 года издается под научно-методическим руководством Российской академии наук. С 2008 года выпускается в составе сериального периодического издания «Научно-технические ведомости СПбГПУ» ISSN 1994-2354.*

Журнал с 2002 года входит в Перечень ведущих рецензируемых научных журналов и изданий, в которых должны быть опубликованы основные результаты диссертаций на соискание ученой степени доктора и кандидата наук.

Сведения о публикациях представлены в Реферативном журнале ВИНИТИ РАН, в международной справочной системе «Ulrich`s Periodical Directory».

Журнал зарегистрирован Федеральной службой по надзору в сфере информационных технологий и массовых коммуникаций (Роскомнадзор). Свидетельство о регистрации ПИ № ФС77-51457 от 19.10.2012 г.

Подписной индекс **47517** в каталоге «Газеты. Журналы» Агентства «Роспечать».

Журнал включен в базу данных «Российский индекс научного цитирования» (РИНЦ), размещенную на платформе Национальной электронной библиотеки на сайте <http://www.elibrary.ru>

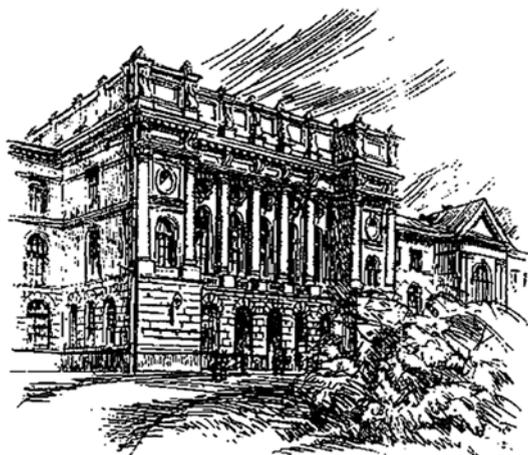
При перепечатке материалов ссылка на журнал обязательна.

Точка зрения редакции может не совпадать с мнением авторов статей.

Адрес редакции и издательства: Россия, 195251, Санкт-Петербург, ул. Политехническая, д. 29.  
Тел. редакции (812) 552-62-16.

© Санкт-Петербургский государственный политехнический университет, 2013

THE MINISTRY OF EDUCATION AND SCIENCE OF THE RUSSIAN FEDERATION



ST. PETERSBURG STATE  
POLYTECHNICAL UNIVERSITY  
**JOURNAL**

---

---

Computer Science.  
Telecommunications and Control Systems

---

---

**3(174) 2013**

Polytechnical University Publishing House  
Saint Petersburg  
2013

# ST. PETERSBURG STATE POLYTECHNICAL UNIVERSITY JOURNAL

## EDITORIAL COUNCIL

*Y.S. Vasiliev* – full member of the Russian Academy of Sciences, President of the St. Petersburg State Polytechnical University, editor-in-chief; *Zh.I. Alferov* – full member of the Russian Academy of Sciences; *V.V. Kostiuk* – full member of the Russian Academy of Sciences; *V.V. Lopota* – corresponding member of the Russian Academy of Sciences; *V.V. Okrepilov* – full member of the Russian Academy of Sciences; *B.E. Paton* – full member of the Russian Academy of Sciences and the National Academy of Sciences of Ukraine; *A.I. Rudskoy* – corresponding member of the Russian Academy of Sciences; *M.P. Fedorov* – full member of the Russian Academy of Sciences; *V.E. Fortov* – full member of the Russian Academy of Sciences.

## EDITORIAL BOARD

*Y.S. Vasiliev* – full member of the Russian Academy of Sciences, President of the St. Petersburg State Polytechnical University, editor-in-chief; *D.G. Arseniev* – Dr.Sc.(tech.), prof.; *A. V. Babkin* – Dr.Sc. (econ.), prof., deputy editor-in-chief; *V.N. Boronin* – Dr.Sc.(tech.), prof.; *V.V. Glukhov* – Dr.Sc. (econ.), prof.; *R. V. Degtyareva* – Dr.Sc. (history), prof.; *A. V. Ivanov* – Dr.Sc.(tech.); *V.K. Ivanov* – Dr.Sc.(phys.-math.), prof.; *V.V. Kozlovsky* – Dr.Sc.(phys.-math.), prof.; *A.I. Rudskoy* – corresponding member of the Russian Academy of Sciences, deputy editor-in-chief; *R.M. Yusupov* – corresponding member of the Russian Academy of Sciences.

# COMPUTER SCIENCE. TELECOMMUNICATIONS AND CONTROL SYSTEMS

## JOURNAL EDITORIAL COUNCIL

*R.M. Yusupov* – corresponding member of the Russian Academy of Sciences, head of the editorial council; *S.M. Abramov* – corresponding member of the Russian Academy of Sciences; *V.V. Voevodin* – corresponding member of the Russian Academy of Sciences; *V.S. Zaborovsky* – Dr.Sc.(tech.), prof.; *V.N. Kozlov* – Dr.Sc.(tech.), prof.; *A.E. Fotiadi* – Dr.Sc.(phys.-math.), prof.; *I.G. Chernorutsky* – Dr.Sc.(tech.), prof.

## JOURNAL EDITORIAL BOARD

*R.M. Yusupov* – corresponding member of the Russian Academy of Sciences, head of the editorial board;  
*D.G. Arseniev* – Dr.Sc.(tech.), prof., deputy head of the editorial board;  
*A. V. Babkin* – Dr.Sc. (econ.), prof., deputy head of the editorial board;  
*V.I. Antonov* – Dr.Sc.(tech.), prof.;  
*Y.A. Gollandtsev* – Dr.Sc.(tech.), prof.;  
*Y.G. Karpov* – Dr.Sc.(tech.), prof.;  
*A.S. Korotkov* – Dr.Sc.(tech.), prof.;  
*S.B. Makarov* – Dr.Sc.(tech.), prof.;  
*S.M. Ustinov* – Dr.Sc.(tech.), prof.;  
*I.A. Tsikin* – Dr.Sc.(tech.), prof.;  
*V.P. Shkodyrev* – Dr.Sc.(tech.), prof.;  
*V.Ye. Klavdiev* – Candidate of Technical Sciences, associate prof.

*The journal is published under scientific and methodical guidance of the Russian Academy of Sciences since 1995. The journal is published since 2008 as part of the periodical edition «Nauchno-tekhnicheskie vedomosti SPbGPU» (ISSN 1994-2354).*

The journal is included in the List of Leading Peer-Reviewed Scientific Journals and other editions to publish major findings of PhD theses for the research degrees of Doctor of Sciences and Candidate of Sciences.

The publications are presented in the VINITI RAS Abstract Journal and Ulrich's Periodical Directory International Database.

The journal is registered with the Federal Service for Supervision in the Sphere of Telecom, Information Technologies and Mass Communications (ROSKOMNADZOR). Certificate ПИ № ФС77-51457 issued Oct. 19, 2012.

Subscription index **47517** in the «Journals and Magazines» catalogue, Rospechat agency.

The journal is on the Russian Science Citation Index (RSCI) database

© Scientific Electronic Library (<http://elibrary.ru/>).

No part of this publication may be reproduced without clear reference to the source.

The views of the authors can contradict the views of the Editorial Board.

The address: 195251 Polytekhnicheskaya Str. 29, St. Petersburg, Russia.



# Содержание

## Проблемы передачи и обработки информации

<b>Бендерская Е.Н., Толстов А.А.</b> Тенденции развития средств аппаратной поддержки нейровычислений .....	9
--	---

## Радиотехника, антенны, СВЧ-устройства

<b>Сухов И.А., Акимов В.П.</b> Метод повышения разрешающей способности пеленгатора с кольцевой антенной решеткой при использовании алгоритма MUSIC.....	19
<b>Алексеев М.А., Сороцкий В.А.</b> Уменьшение искажений при генерировании радиосигналов ключевыми методами.....	26

## Системный анализ и управление

<b>Волкова В.Н., Черненко Л.В., Магер В.Е.</b> Классификация моделей в системном анализе.....	33
<b>Ланцев Е.А., Доррер М.Г.</b> Получение агентной имитационной модели из дискретно-событийного описания бизнес-процесса .....	44

## Математическое моделирование: методы, алгоритмы, технологии

<b>Ястребов М.Ю.</b> Поиск Фибоначчи — обоснование, система счисления и хеширование.....	53
<b>Шубников В.Г., Беляев С.Ю.</b> Подавление шума и оценка различий в изображениях.....	58

## Управление в социальных и экономических системах

<b>Антонова Е.И., Васильев И.А.</b> Проблема организации грузовых работ на железной дороге контейнерного терминала .....	67
--	----

## Информационные и телекоммуникационные технологии в образовании

<b>Каширин И.Ю., Медведев Р.Е.</b> Онтологическое накопление учебных знаний на основе сервиса информационных сетей .....	75
<b>Иванченко Д.А., Хмельков И.А., Райчук Д.Ю., Митрофанов А.М., Самочадин А.В., Рогов П.А.</b> Применение подходов BYOD для построения стратегии информатизации высшего учебного заведения .....	85

## Конференция Microsoft

<b>Кириленко Я.А., Григорьев С.В., Авдюхин Д.А.</b> Разработка синтаксических анализаторов в проектах по автоматизированному реинжинирингу информационных систем .....	94
<b>Дробинцев П.Д., Никифоров И.В., Котляров В.П.</b> Методика проектирования тестов сложных программных комплексов на основе структурированных UCM моделей.....	99
<b>Колосов А.С., Богоявленский Ю.А.</b> Параллельный алгоритм построения графа ИКТ-инфраструктуры интернет-провайдера.....	105

<b>Никифоров И.В., Дробинцев П.Д., Котляров В.П.</b> <i>Интегральные критерии проверки требований к программному обеспечению</i> .....	111
<b>Тютин Б.В., Веселов А.О., Котляров В.П.</b> <i>Масштабирование выполнения тестового набора при автоматизированном тестировании</i> .....	118
<b>Тутыгин В.С., Пелконен И.А., Кириленко К.С.</b> <i>Способ и программная реализация прецизионного измерения времени задержки радиоимпульса</i> .....	123
<b>Боинов Н.В, Александрова О.В., Котлярова Л.П.</b> <i>Модернизация методики разработки мобильных приложений для платформы WINDOWS PHONE 8</i> .....	129
<b>Заславский М.М., Кринкин К.В., Рябиков Э.М., Черкалова В.В., Цамутали А.Б.</b> <i>Инструментарий надежного хранения данных</i> .....	134
<b>Жигулин А.Ю., Сафонов В.О.</b> <i>Проектирование видеоконтента на сложные поверхности с помощью нескольких устройств вывода</i> .....	139
<b>Батюков А.М., Ампилова Н.Б.</b> <i>Модифицированный алгоритм моделирования агрегации, ограниченной диффузией</i> .....	143
<b>Никифоров И.В., Котляров В.П., Дробинцев П.Д.</b> <i>Ограничения на многопоточные конструкции и временные задержки языка UCM</i> .....	148



# Contents

## Information Transfer and Processing

<b>Benderskaya E.N., Tolstov A.A.</b> <i>Trends of hardware implementation of neural networks</i> .....	9
---	---

## Radio Engineering, Aerials, SHF-devices

<b>Sukhov I.A., Akimov V.P.</b> <i>Resolution improvement of MUSIC algorithm for the direction-finder based on a circular antenna array</i> .....	19
<b>Alekseev M.A., Sorotsky V.A.</b> <i>Distortions reduction for switch-mode generated radio signals</i> .....	26

## System Analysis and Control

<b>Volkova V.N., Chernenkaya L.V., Mager V.E.</b> <i>Classification of models in system analysis</i> .....	33
<b>Lantsev Y.A., Dorrer M.G.</b> <i>Creating agent-based model from the business process discrete-event model</i> .....	44

## Mathematical Modelling: Methods, Algorithms, Technologies

<b>Yastrebov M.Yu.</b> <i>The Fibonacci search – the substantiation, the number system and the hashing</i> .....	53
<b>Shubnikov V.G., Belyaev S.Yu.</b> <i>Image difference estimation and noise removing</i> .....	58

## Management in Social and Economic Systems

<b>Antonova E.I., Vasilev I.A.</b> <i>The problem of cargo operation management on container terminal railway</i> .....	67
---	----

## Information and Telecommunication Technologies in Education

<b>Kashirin I.Yu., Medvedev R.E.</b> <i>Ontological accumulation of educational knowledge based on the service information networks</i> .....	75
<b>Ivanchenko D.A., Khmelkov I.A., Raychuk D.Yu., Mitrofanov A.M., Samochadin A.V., Rogov P.A.</b> <i>The use of BYOD approaches for implementation of university IT-strategy</i> .....	85

## Microsoft Conference

<b>Kirilenko I.A., Grigoriev S.V., Avdiukhin D.A.</b> <i>Syntax analyzers development in automated reengineering of informational system</i> .....	94
<b>Drobintsev P.D., Nikiforov I.V., Kotliarov V.P.</b> <i>Formal models structurization based technique of complex software projects testing</i> .....	99
<b>Kolosov A.S., Bogoyavlensky Yu.A.</b> <i>A parallel algorithm for constructing a graph of a local Internet service provider's ICT-infrastructure</i> .....	105
<b>Nikiforov I.V., Drobintsev P.D., Kotliarov V.P.</b> <i>Integrated criteria of software requirement checking</i> ..	111
<b>Tyutin B.V., Veselov A.O., Kotlyarov V.P.</b> <i>Test suite execution scaling for automated testing</i> .....	118

<b>Tutygin V.S., Pelkonen I.A., Kirilenko K.S.</b> <i>Method and software for precision measurement of time delay echo-signals of radars</i> .....	123
<b>Voinov N.V., Aleksandrova O.V., Kotlyarova L.P.</b> <i>Modernization of the methodic of WINDOWS PHONE 8 mobile applications development</i> .....	129
<b>Zaslavskiy M.M., Krinkin K.V., Ryabikov E.M., Cherkalova V.V., Tsamutali A.B.</b> <i>Toolset for secure data storage</i> .....	134
<b>Zhigulin A.Yu., Safonov V.O.</b> <i>Video content projecting on complex surfaces using multiple output devices</i> .....	139
<b>Batyukov A.M., Ampilova N.B.</b> <i>A modified diffusion-limited aggregation modelling algorithm</i> .....	143
<b>Nikiforov I.V., Kotliarov V.P., Drobintsev P.D.</b> <i>Restrictions on concurrent construction and time delays of UCM language</i> .....	148

УДК 004.8.032.26, 681.513.8

*Е.Н. Бендерская, А.А. Толстов*

## **ТЕНДЕНЦИИ РАЗВИТИЯ СРЕДСТВ АППАРАТНОЙ ПОДДЕРЖКИ НЕЙРОВЫЧИСЛЕНИЙ**

*E.N. Benderskaya, A.A. Tolstov*

### **TRENDS OF HARDWARE IMPLEMENTATION OF NEURAL NETWORKS**

---

Рассмотрены вопросы аппаратной реализации биоинспирированных нейронных сетей. Представлены результаты сравнительного анализа существующих нейроморфных систем. Определены этапы разработки хаотической нейронной сети на базе графического ускорителя.

**НЕЙРОННАЯ СЕТЬ. БИОИНСПИРИРОВАННЫЕ ВЫЧИСЛЕНИЯ. НЕЙРОМОРФНЫЕ СИСТЕМЫ. АППАРАТНАЯ ПОДДЕРЖКА.**

Hardware implementations of bio-inspired neural networks are under consideration. A comparative analysis of neuromorphic systems with other approaches is provided. The stages of chaotic neural network hardware implementation based on graphics processing units are proposed.

**NEURAL NETWORK. BIO-INSPIRED COMPUTING. NEUROMORPHIC SYSTEMS. HARDWARE IMPLEMENTATION.**

---

В данной статье сравниваются основные характеристики современных разработок, посвященных нейровычислениям и нейроморфным системам. Нейроморфными системами принято считать такие системы, которые являются аппаратно-программной реализацией моделей биоинспирированных нейронных сетей, а нейроморфингом — сам процесс создания таких систем [1, 2]. Особое внимание направлено на анализ аппаратных средств, адаптированных под биологически инспирированные нейронные сети. Необходимость такого аналитического обзора современных нейровычислительных проектов вызвана поиском решений аппаратной поддержки хаотических нейронных сетей с нелинейной динамикой [3]. Такие нейронные сети (НС) являются многообещающей технологией в решении широкого

спектра задач. Как и импульсные нейронные сети, хаотические НС являются биологически инспирированными. В выводах по результатам работы выделены наиболее перспективные направления разработки средств аппаратной поддержки нейронных сетей.

#### **Существующие подходы к реализации нейровычислений на аппаратном уровне**

В настоящее время нейровычисления находятся на той стадии развития, когда над ними работают не только известные научные школы, но и ведущие ИТ-корпорации. Среди основных используемых технических средств можно выделить следующие:

1) традиционные ПК для целей мелко-масштабных исследований, а также для работы достаточно простых нейронных сетей.

Моделирование крупномасштабных систем на ПК является проблемой вследствие невысокой производительности такого рода устройств;

2) специализированные устройства для параллельных высокопроизводительных вычислений. Сюда входят современные графические карты (GPU), а также семейство программируемых логических интегральных схем (FPGA). По сравнению с обычными ПК такие устройства предлагают ряд преимуществ. Во-первых, это более высокие показатели пиковой производительности (десятки и более раз, если сравнивать устройства одного поколения). Во-вторых, — параллельная архитектура, которая является адекватной нейронным сетям;

3) кластеры из ПК. Решают проблему низкой производительности отдельных машин. Применяются и для крупномасштабного моделирования нейронных сетей;

4) суперкомпьютеры. Часто для моделирования используют суперкомпьютеры. Это оправдано в силу того, что именно такие устройства позволяют достичь максимальной производительности вычислительной системы, доступной на данный момент. Главные недостатки такой модели — огромная стоимость суперкомпьютеров, огромные затраты электроэнергии;

5) устройства различных типов, специализированные для целей нейровычислений. Единого стандарта или применяемого подхода в этой области нет. Два основных преимущества специализированных средств — это низкое энергопотребление и относительно невысокая стоимость устройств.

Разработки нейроморфных систем ведутся с целью по возможности полно повторить эффективный биологический прототип — мозг, который являет собой пример мощнейшей вычислительной системы, имеющей весьма компактный размер и низкое энергопотребление. Поэтому основной тенденцией в разработке аппаратной поддержки нейровычислений стала ориентация на биоинспирированные НС и создание нейроморфных систем.

#### **Проекты нейроморфинга**

По результатам анализа различных ис-

точников информации по существующим проектам нейроморфинга (НМ) выделены проекты, представленные в табл. 1. Для рассмотрения выбраны наиболее масштабные проекты, которые выполняются в известных крупных компаниях со значительным объемом финансирования под руководством признанных ученых и ведущих научных школ [4].

Для оценки рассматриваемых проектов и выявления подходящих подходов для аппаратной реализации хаотических НС проанализированы как задачи проектов и математическая база, используемая для моделирования работы отдельных нейронов и их связей (синапсов), так и используемые технологии разработки нейроустройств. Краткие данные по проектам сведены в табл. 2. Также в ней отмечен тип используемой структуры при реализации аппаратного решения — является ли структура универсальной или специфической, учитывающей особенности той биологической подсистемы, которая взята за основу как прототип нейроморфной системы. Ввиду того, что НС представляют собой параллельное решение любой задачи, то даже специфические структуры являются, как правило, регулярными (типа решетка), и отличаются составом базовых обрабатываемых элементов и алгоритмами их взаимодействия.

#### **Базовые логические элементы нейроустройств**

В качестве базового элемента во всех рассмотренных проектах используются модели импульсных нейронов. Наиболее простая модель такого нейрона — модель интегрирования и возбуждения — Leaky Integrate-and-Fire (I&F), предложенная в 1907 г. и являющая собой пример максимально упрощенного биологического нейрона. Такая модель хорошо подходит для теоретических исследований, но не является полноценной моделью с точки зрения того, что она не генерирует импульс сама по себе. Данная модель применяется в проектах DARPA SyNAPSE и SpiNNaker [5].

Простейшая модель нейрона, который является «полноценным» — это модель ква-

Таблица 1

Основные проекты нейроморфинга

Название проекта	Ведущий исследователь	Цели проекта
Blue Brain Project, 2005 – настоящее время	Университет EPFL (Швейцария), Г. Маркрам	Максимально правдоподобно воспроизвести работу мозга млекопитающих с дальнейшими практическими применениями в медицине. Одно из основных направлений — изучение заболеваний нервной системы. Применение суперкомпьютера IBM Blue Gene для моделирования
FACETS, BrainScaleS, 2005–2010 BrainScaleS 2011 – настоящее время	Консорциум европейских научных школ под началом Гейдельбергского университета (Германия), К. Мейер	Разработка новых базовых технологий, реализация нейросистем в виде аппаратных средств, разработка архитектуры систем, биологические исследования
Neurogrid, 2006 – настоящее время	Группа Brains-In-Silicon, Стенфордский университет, К. Боэн	Повторить достижение Blue Brain Project, моделирующего работу мозга на суперкомпьютере, но сделать это на специализированном устройстве собственной разработки, с значительно меньшей стоимостью, размером и энергопотреблением по сравнению с суперкомпьютером IBM Blue Gene. На изготовленном устройстве проводить моделирование различных функций мозга
SyNAPSE, 2006–2018	DARPA совместно с IBM Labs, HRL и рядом научных школ США, Д. Модха, Н. Сриниваса	Моделирование мозга млекопитающих животных и человека, создание нейроморфных устройств, разработка архитектуры нейронных систем. Проект развивается по нескольким направлениям параллельно и относительно независимо
SpiNNaker, 2007–2014	Ряд вузов Соединенного Королевства под руководством Манчестерского университета, С. Фербер	Построение распределенной вычислительной системы, ориентированной под работу нейронных сетей различных типов, изучение особенностей работы мозга млекопитающих на примере задачи построения такой системы и ее эксплуатации
MoNETA / Cog Ex Machina, 2008 – настоящее время	Бостонский университет совместно с компанией Hewlett-Packard (HP), М. Версаче, Г. Снейдер	Роботостроение, разработка новых аппаратных архитектур, моделирование

дратичного нейрона (Quadratic Integrate-and-fire (QI&F)), он генерирует импульсы, работая в режиме интегратора [6]. Модель нейрона, подобного квадратичному, применяется в проекте Neurogrid [7].

Е. Ижикевичем подробно исследованы различные модели нейронов и в 2003 г. предложена простая модель выбора – Simple Model of Choice (SMC), позволяющая при определенной настрой-

Таблица 2

Характеристики проектов нейроморфинга

Проект	Нейрон	Синапс	Аппаратная реализация	Тип структуры
FACETS	Гибридная модель (Adaptive Exponential Integrate-and-Fire)	Модель синаптической пластичности, разработанная на основе импульсной временной пластичности синапсов (STDP)	1. Моделирование биологической нейросети на суперкомпьютере 2. Нейровычислительный модуль на основе специфичной цифро-аналоговой архитектуры с распределенной системой памяти и гибкой настройкой параметров синапсов. Общее количество нейронов на такой плате равняется 200 тыс., синапсов – 50 млн. Устройство обладает высокими характеристиками производительности: биологическая нейронная сеть, реализованная в таком устройстве, моделируется в 1000 раз быстрее своего реального прототипа, что открывает исследователям огромные возможности [9]	1. Универсальная 2. Специфическая, регулярная
SpiNNaker	Использование простейших моделей абстрактных нейронов – модель LI&F, модель Ижикевича	STDP-модель	Специфический суперкомпьютер, адаптированный к моделированию крупномасштабных НС. Построение ведется на основе процессоров ARM, плотной упаковкой нескольких процессоров в большие ядра [5]	Специфическая, регулярная
SyNAPSE	Использование простейших моделей абстрактных нейронов: модель LI&F, модель Ижикевича	Берстовый вариант STDP	1. Суперкомпьютер для моделирования крупномасштабной НС (IBM). 2. Вычислительное устройство на основе цифро-аналогового подхода (IBM), состоящее из 256 нейронов [10]. 3. Разработка устройств на основе мемристоров (HRL) [11]	1. Универсальная. 2. Специфическая, регулярная. 3. Специфическая, регулярная
Neurogrid	Различные типы моделей с богатой динамикой, включающей в себя берстовый режим	STDP-модель	Наиболее крупный реализованный цифро-аналоговый модуль среди прочих проектов (1 млн нейронов). Синапсы реализованы при помощи FPGA и модуля памяти, а плата с нейронами состоит из аналоговых микросхем [7]. Применение аналоговой модели является решением проблемы энергопотребления – устройство на 1 млн нейронов потребляет 3,5 Вт электроэнергии	Специфическая, регулярная

Cog Ex Machina	Широкий спектр различных типов моделей – обусловленность цифровым подходом, позволяющим безболезненно модифицировать данный компонент	Широкий спектр моделей	Поддержка различных архитектур: 1. Суперкомпьютер, состоящий из большого числа GPU. 2. Мемристорный подход, технологически обеспечиваемый со стороны компании HP [6]	1. Специфическая, регулярная. 2. Специфическая, регулярная
----------------	---	------------------------	--	---

ке имитировать работу большого числа известных биологических нейронов, проявляя свойства возбуждения и берстовой активности [8], что делает эту модель удобной для построения крупномасштабных нейроморфных систем. Так, например, нейросистема, состоящая из 100 миллиардов таких нейронов, смоделирована на кластере Veowolf в 2005 г. [8]. Данная модель также применяется в проектах DARPA SyNAPSE и SpiNNaker [5].

Модель, предложенная Е. Ижикевичем, является примером гибридной модели нейрона. Гибридные модели образуют собой класс нейронных моделей, отличающихся друг от друга функцией генерации импульса [8]. Одной из таких моделей является разработанная в 2005 г. и используемая в рамках проекта FACETS модель адаптивно-интегрирования-возбуждения – Adaptive Exponential Integrate-and-Fire (AEI&F) [9].

Наиболее важная модель биологического нейрона – это, безусловно, модель нейрона Ходжкина–Хаксли (НХХ), параметры которой получены в 1952 г. В отличие от перечисленных выше моделей, НХХ – это реальный прототип биологического нейрона. Параметры такой модели отвечают за реальные физические характеристики – токи, протекающие в клетке, мембранную емкость, проводимости, типы ионов и т. д. Платой за реалистичность воспроизводимых процессов является вычислительная сложность НХХ [8].

Рассмотрение используемых в проектах типов нейронов будет неполным без проекта Blue Brain Project. Напрямую он не относится к проектам нейроморфинга, т. к. его основной целью является моделирование, а не создание нейроустройств. Г. Маркрам, руководитель данного проекта, подчеркивает ориентированность на моделирование максимально приближенных к реальности нейронов, для чего использование простых моделей (CMS, LI&F и др.) является невозможным. Акцент сделан на то, что на данный момент еще не известно, какие свойства и процессы внутри нейронов являются основными в функционировании мозга, а какие являются побочными эффектами эволюции, не имеющими отношения к деятельности нейросистемы. Проект Blue Brain призван изучать мозг, моделируя его деятельность, и только после прояснения ответа на вопрос, какие механизмы в нейроне являются основными, – только тогда можно будет прибегнуть к вычислительной оптимизации, сокращающей возможности нейрона. В противовес данному подходу С. Фёрбер, руководитель проекта SpiNNaker, утверждает, что в его проекте намеренно использованы простейшие модели, а акцент сделан на противоположный Г. Маркраму подход [5]. Путем построения распределенной высокопроизводительной системы, нацеленной на нейровычисления, исследователи смогут понять основополагающие принципы ра-

боты мозга, синхронизации его составных частей, решая эти же задачи в своем проекте. Таким образом, реализуется обратное моделирование мозга.

Проект Neurogrid стоит несколько особняком от перечисленных выше проектов. Предполагается, что могут использоваться различные модели импульсных нейронов. Известны факты использования нейронов, подобных квадратичному импульсному нейрону, а также нейрону, обеспечивающему более сложную динамику, в частности, берстовый режим работы [8].

### Подходы к аппаратной реализации НС

В рассматриваемых проектах используются или создаются различные аппаратные средства. Их можно разделить на классы по используемым подходам:

1. Цифро-аналоговый подход. Как правило, аналоговые микросхемы используются для реализации физических нейронов, а синапсы являются цифровым дополнением к модели. У аналогового подхода есть минусы: создание таких схем в любом случае получается «под заказ», они не являются универсальными, в отличие от цифровых компонентов. Подобный подход применяется в проекте BrainScaleS/FACETS. Еще два типичных представителя данного типа устройств — проекты SyNAPSE (IBM) и Neurogrid (см. табл. 2).

2. Использование мемристоров. В 2008 г. компания HP разработала физическое устройство мемристор. Модель мемристора была открыта в середине XX в. Л. Чуа, но считалось, что такое устройство невозможно реализовать в виде физической схемы. Основные преимущества мемристора следующие: он имеет нелинейное сопротивление, которое можно регулировать электрическим сигналом (аналогия с биологическим нейроном); обладает свойством памяти и имеет очень маленький размер, следовательно, мемристормы могут быть плотно упакованы в запоминающий модуль; материалы, из которых изготовлен мемристор, позволяют встраивать его в существующие микросхемы [5]. Работа команды под руководством М. Версаче в

проекте Бостонского университета нацелена на сотрудничество с HP для разработки и создания мемристорных устройств. Мемристорным разработкам посвящена также часть проекта SyNAPSE, осуществляемая компанией HRL, которая в 2011 г. представила мемристорный массив [11]. В течение нескольких лет ожидается внедрение мемристоров в существующие вычислительные архитектуры. Это позволит значительно повысить производительность систем, что откроет широкие возможности для моделирования нейронных сетей.

3. Создание суперкомпьютеров, ориентированных на нейровычисления. Разработкой этого направления занимается проект SpiNNaker (Spiking Neural Network Architecture). Архитектура такого проекта ориентирована на энергоэффективность, построение из традиционных компонентов (используются ARM-процессоры), распределенность системы. Как уже было отмечено, архитектура вычислительной системы (а не типы нейронов) в этом проекте является главной его частью. Специализированный суперкомпьютер также используется в проекте Бостонского университета. Программный продукт Cog, разрабатываемый командой, нацелен на поддержку различных аппаратных архитектур. Помимо упомянутых выше мемристоров, используется также и суперкомпьютер, состоящий из нескольких сотен GPU. Цифровой подход проекта позволяет использовать различные модели нейронов и синапсов, обеспечивая большую гибкость.

### Российские проекты и исследования в области нейроморфинга

Среди российских разработок стоит отметить следующие:

- Разработка НТЦ «Модуль» нейропроцессора Л1879ВМ1. Данный процессор, построенный на основе модели RISC, содержит в себе поддержку инструкций VLIW (Very Large Instruction Word) и SIMD (Single Instruction Multiple Data), а отличительной его чертой является поддержка разрядно-зависимых вычислений (количество разрядов варьируется от 1 до 64).

- Разработка компонента мемристора исследователями ООО «ТАСО», являющегося высокотехнологичным предприятием, созданным Тюменским ГУ [12]. Как было отмечено ранее, первооткрывателями устройства мемристора стала компания HP, а само устройство является многообещающим как с точки зрения создания высокопроизводительных и при этом компактных машин, так и с позиции применимости в качестве базового элемента аппаратной реализации нейронной сети. Российские исследователи стали пятой в мире научной группой, создавшей мемристор.

- Разработка нейрогибридных систем в НИЦ «Курчатовский институт». Идея нейрогибридной системы состоит в использовании слоя клеток мозга, помещенных в питательную среду и выращиваемых в ней, для управления роботами. Акцент, сделанный авторами проекта, заключается в следующем: исследователи на данный момент имеют далеко не полную картину деталей функционирования мозга, поэтому построение вычислительных средств на базе компьютеров и современных технологий НС может не дать необходимых результатов. При использовании слоя живых клеток мозга исследователи получают возможность как оперативно наблюдать за происходящими в клетках явлениями (при помощи микроскопии и инструментов молекулярной биологии), так и использовать для управления роботом вычислительную структуру естественного происхождения [13].

Отдельно от рассмотренных подходов стоят разработки нейроморфных средств для реализации искусственных подсистем мозга (мозжечка, гиппокампа и др.), а также связанных с ним периферических систем. На основе таких нейроморфных средств разрабатываются различные системы управления роботами [2], однако для поддержки аппаратной реализации отдельных структур НС такие специализированные разработки не предназначены.

#### **Успехи, нерешенные задачи и перспективы нейроморфинга**

Общие выводы по рассмотренным проектам, учитывая объемы финансирования и

вовлеченность в дело крупных компаний, а также рассмотренные выше пути достижения целей, следующие:

- тема создания нейроморфных систем крайне актуальна на данный момент [2];
- аппаратные устройства, разработанные в рамках проектов, и научные знания, полученные исследователями, находят применение в разных областях [6];

не существует единого подхода к решению любой из поставленных в проектах задач. Более того, ведущие ученые не имеют на данный момент согласованных предположений о правильности того или иного пути к достижению целей в исследованиях мозга в целом и нейроморфинге в частности.

Все рассмотренные проекты добились определенных успехов в своей деятельности. Так, в проектах Neurogrid, SyNAPSE (IBM), FACETS сконструированы и произведены модели аппаратных цифроаналоговых устройств, моделирующих работу импульсных биологически инспирированных нейронных сетей. Дальнейшие направления деятельности в этих проектах видятся в увеличении масштабов реализуемых ими нейронных сетей и сложности базовых элементов.

В проекте Бостонского университета и ветви проекта SyNAPSE, реализуемой HRL, в ближайшие годы ожидается появление аппаратных устройств на основе мемристоров, которые позволят совершить качественный скачок в производительности вычислительных систем. Направления развития проектов нейроморфных систем:

- Увеличение вычислительной мощности.
- Усложнение математической модели базовых элементов по мере появления новых знаний и увеличения производительности устройств.
- Расширение математической базы для поддержки различных классов нейронных сетей в одном устройстве.
- Уменьшение аппаратных устройств в размерах. Этому способствуют архитектуры, отличные от фон Неймана, а также разработка высокоскоростных наноэлементов памяти.

Проблема крупных проектов заключается в том, что в них решается задача в глобальном контексте. При этом остается свободной ниша по решению задачи практического применения нейроморфных систем, которая может быть решена широким спектром технологий нейронных сетей, среди которых присутствуют не только импульсные нейронные сети, но и осцилляторно-хаотические нейронные сети и их разновидности. Отсюда следует необходимость проведения собственных разработок по поддержке аппаратной реализации хаотических НС [14].

### **Предлагаемая реализация хаотической НС на основе технологии Nvidia CUDA**

Современные видеоускорители от компаний Nvidia и AMD отлично подходят для выполнения ресурсоемких задач, для которых требуется параллельное исполнение. Вполне естественным является применение данных устройств к нейронным сетям. Как уже было сказано выше, использование массивно-параллельных архитектур вроде видеокарт или FPGA является одним из используемых подходов при выборе аппаратной базы для реализации НС. Проведем анализ возможности реализации осцилляторной хаотической НС, решающей задачу кластеризации [14].

Алгоритм функционирования нейронной сети предполагает следующие стадии:

1. Настройка весов нейронной сети. В процессе вычисления весовых коэффициентов необходимым знанием является вычисление количества ближайших соседей каждой входной точки, для чего применяется триангуляция Делоне. Количество нейронов равно количеству входных точек, а связи в такой сети – каждый с каждым.

2. Динамика нейронной сети (непосредственный процесс функционирования) и интерпретация этой динамики при помощи метода фазовой синхронизации.

В качестве аппаратной базы выбрана видеокарта с поддержкой технологии Nvidia CUDA. Приведем краткое описание процессов работы нейронной сети с учетом выбранной аппаратной поддержки.

1. Определение для каждого нейрона

количества ближайших соседей будет выполнено при помощи построения триангуляции Делоне методом Форчуна. Данный метод имеет для сети из  $N$  нейронов асимптотическую сложность  $O(N \log N)$ , является по своей сути последовательным и не требует реализации на GPU.

2. Вычисление и установка весовых коэффициентов НС – достаточно простая задача, которая является дружественной параллельным архитектурам и удобно реализуется при помощи GPU. Асимптотическая сложность операции  $O(N^2)$ .

3. Динамика нейронной сети. Аналогично п. 2 – является подходящей частью для использования GPU и имеет такую же асимптотическую сложность –  $O(N^2)$ .

4. Интерпретация динамики на всем периоде наблюдения. Данная операция предполагает сравнение выхода нейрона со всеми остальными нейронами и применение к полученным значениям ряда параметров. При получении новой порции данных от нейронов сложность такой операции также будет составлять  $O(N^2)$  и реализовываться с применением GPU.

По предварительным оценкам, ввиду параллельного характера алгоритмов, производительность комбинированной системы, состоящей из современного GPU + CPU, по сравнению с использованием обычной CPU-реализации будет выше в 10 и более раз.

Анализ современных проектов по аппаратной поддержке нейровычислений позволяет сделать вывод о том, что наибольшее распространение получают проекты, ориентированные на реализацию биологически ориентированных НС, включающих в свои модели учет все больших деталей и особенностей функционирования биологического прототипа – нейрона. В связи с этим возрастает и сложность разрабатываемых решений, а во многих случаях уменьшается универсальность, ввиду отсутствия единого подхода. Для аппаратной реализации хаотических НС в настоящее время более подходящим подходом является использование технологии графических ускорителей, что подтверждается и предварительной проработкой необходимых для этого этапов.

СПИСОК ЛИТЕРАТУРЫ

1. **Boahen, K.** Neuromorphic Microchips [Text] / K. Boahen // Scientific American. –2005. –Vol. 292. –№ 5. –P. 56–63.
2. **Станкевич, Л.А.** Нейроморфные средства для управления роботами [Текст] / Л.А. Станкевич, В.Г. Капралов, И.В. Клочков // Сб. науч. тр. XIII Всерос. науч.-техн. конф. Нейроинформатика 2011. –М., 2011. –Ч. 3. –С. 125–132.
3. **Benderskaya, E.N.** Nonlinear Trends in Modern Artificial Intelligence: A New Perspective [Text] / E.N. Benderskaya // Beyond AI: Interdisciplinary Aspects of Artificial Intelligence. Topics in Intelligent Engineering and Informatics. –Springer, 2013. –Vol. 4. –P. 113–124.
4. **Hunkins, J.** Neuroscience Expert Dr. Henry Markram on the IBM «Cat Brain» Simulation: «IBM’s claim is a HOAX» [Электронный ресурс] / J. Hunkins // Режим доступа: <http://technology-report.com/2009/11/neuroscience-expert-dr-henry-markram-on-the-ibm-cat-brain-simulation-ibms-claim-is-a-hoax/> (Дата обращения 01.03.2013)
5. **Furber, S.** Neural systems engineering. Review [Text] / S. Furber, S. Temple // J. of the Royal Society Interface. –2007. –Vol. 4. –№ 13. –P. 193–206.
6. **Snider, G.** From Synapses to Circuitry: Using Memristive Memory to Explore the Electronic Brain [Text] / G. Snider [et al.] // IEEE Computer. –Vol. 44. –Iss. 2. –P. 21–28.
7. **Choudhary, S.** Silicon Neurons that Compute [Text] / S. Choudhary, S. Sloan, K. Boahen [et al.] // International Conf. on Artificial Neural Networks, LNCS. –Springer, Heidelberg, 2012. –Vol. 7552. –P. 121–128.
8. **Izhikevich, E.M.** Hybrid Spiking Models [Text] / E.M. Izhikevich // Phil. Trans. R. Soc. A. –2010. –Vol. 368. –№ 1930. –P. 5061–5070.
9. FACETS Summary Flyer [Электронный ресурс] / Fast Analog Computing with Emergent Transient States web site // Режим доступа: [http://facets.kip.uni-heidelberg.de/images/4/48/Public--FACETS\\_15879\\_Summary-flyer.pdf](http://facets.kip.uni-heidelberg.de/images/4/48/Public--FACETS_15879_Summary-flyer.pdf) (Дата обращения 01.03.2013)
10. **Kim, K.W.** A Functional Hybrid Memristor Crossbar-Array/CMOS System for Data Storage and Neuromorphic Applications [Text] / K.W. Kim, S. Gaba, W. Lu [et al.] // Nano Letters. –2012. –Vol. 12. –P. 389–395.
11. **Imam, N.** Implementation of olfactory bulb glomerular-layer computations in a digital neurosynaptic core [Text] / N. Imam, T.A. Clenald, D. Modha [et al.] // Frontiers in Neuroscience. –2012. –Vol. 6. –Article 83. –13 p.
12. Мемристор – шаг к искусственному интеллекту. Новые разработки ученых ТюмГУ [Электронный ресурс] / Веб-сайт Тюменского государственного университета // Режим доступа: <http://utmn.ru/news/6468> (Дата обращения 01.03.2013)
13. **Анохин, К.В.** Современные подходы к моделированию активности культур нейронов in vitro [Текст] / К.В. Анохин, М.С. Бурцев, В.А. Ильин [и др.] // Математическая биология и биоинформатика. –2012. –Т. 7. –№ 2. –С. 372–397.
14. **Benderskaya, E.N.** Multidisciplinary Trends in Modern Artificial Intelligence: Turing’s Way [Text] / E.N. Benderskaya, S.V. Zhukova // AIECM – Turing 2013, Book Chapters: Artificial Intelligence, Evolutionary Computation and Metaheuristics. – Springer, 2013. –P. 320–343.

REFERENCES

1. **Boahen K.** Neuromorphic Microchips / Scientific American. – 2005. – Vol. 292. – № 5. – P. 56–63.
2. **Stankevich L.A., Kapralov V.G., Klochkov I.V.** Neiromorfnye sredstva dlia upravleniia robotami / Sb. nauch. trudov XIII Vseros. nauch.-tekhn. konf. Neuroinformatika 2011. – Moscow, 2011. – Ch. 3. – S. 125–132. (rus)
3. **Benderskaya E.N.** Nonlinear Trends in Modern Artificial Intelligence: A New Perspective / Beyond AI: Interdisciplinary Aspects of Artificial Intelligence. Topics in Intelligent Engineering and Informatics. – Springer, 2013. – Vol. 4. – P. 113–124.
4. **Hunkins J.** Neuroscience Expert Dr. Henry Markram on the IBM «Cat Brain» Simulation: IBM’s claim is a HOAX. Available <http://technology-report.com/2009/11/neuroscience-expert-dr-henry-markram-on-the-ibm-cat-brain-simulation-ibms-claim-is-a-hoax/> (Accessed 01.03.2013)
5. **Furber S., Temple S.** Neural systems engineering. Review / J. of the Royal Society Interface. – 2007. –Vol. 4. –№ 13 – P. 193–206.
6. **Snider G. et al.** From Synapses to Circuitry: Using Memristive Memory to Explore the Electronic Brain / IEEE Computer. –Vol. 44. –Iss. 2. –P. 21–28.
7. **Choudhary S., Sloan S, Boahen K. et al.** Silicon Neurons that Compute / International Conf. on Artificial Neural Networks, LNCS. – Springer, Heidelberg, 2012. – Vol. 7552. – P. 121–128.
8. **Izhikevich E.M.** Hybrid Spiking Models / Phil. Trans. R. Soc. A. – 2010 – Vol. 368. – № 1930. –P. 5061–5070.

9. FACETS Summary Flyer; Fast Analog Computing with Emergent Transient States web site. Available [http://facets.kip.uni-heidelberg.de/images/4/48/Public--FACETS\\_15879\\_Summary-flyer.pdf](http://facets.kip.uni-heidelberg.de/images/4/48/Public--FACETS_15879_Summary-flyer.pdf) (Accessed 01.03.2013)

10. **Kim K.W., Gaba S., Lu W. et al.** A Functional Hybrid Memristor Crossbar-Array/CMOS System for Data Storage and Neuromorphic Applications; *Nano Letters*. – 2012. – Vol. 12. – P. 389–395.

11. **Imam N., Clenald T.A., Modha D. et al.** Implementation of olfactory bulb glomerular-layer computations in a digital neurosynaptic core / *Frontiers in Neuroscience*. – 2012. – Vol. 6. – Article 83. – 13 p.

12. Memristor – шаг к искусственному

интеллекту. *Novye razrabotki uchenykh TiumGU; Veb-sait Tiumenskogo gosudarstvennogo universiteta*. Available <http://utmn.ru/news/6468> (Accessed 01.03.2013) (rus)

13. **Anokhin K.V., Burtsev M.S., Il'in V.A. i dr.** *Sovremennye podkhody k modelirovaniu aktivnosti kul'tur neironov in vitro / Matematicheskaja biologija i bioinformatika*. – 2012. – Т. 7. – № 2. – S. 372–397. (rus)

14. **Benderskaya E.N., Zhukova S.V.** *Multi-disciplinary Trends in Modern Artificial Intelligence: Turing's Way / AIECM – Turing 2013; Book Chapters: Artificial Intelligence, Evolutionary Computation and Metaheuristics*. – Springer, 2013. – P. 320–343.

---

**БЕНДЕРСКАЯ Елена Николаевна** – доцент кафедры компьютерных систем и программных технологий Санкт-Петербургского государственного политехнического университета, кандидат технических наук.

195251, Россия, Санкт-Петербург, ул. Политехническая, д. 21.

Тел. (812) 297-42-18, e-mail: Helen.bend@gmail.com

**BENDERSKAYA, Elena N.** *St. Petersburg State Polytechnical University.*

195251, Politekhnikeskaya Str. 21, St. Petersburg, Russia.

E-mail: Helen.bend@gmail.com

**ТОЛСТОВ Александр Александрович** – аспирант кафедры компьютерных систем и программных технологий Санкт-Петербургского государственного политехнического университета.

195251, Россия, Санкт-Петербург, ул. Политехническая, д. 21.

Тел. (812) 297-42-18, e-mail: gm.alex@gmail.com

**TOLSTOV, Aleksandr A.** *St. Petersburg State Polytechnical University.*

195251, Politekhnikeskaya Str. 21, St. Petersburg, Russia.

E-mail: gm.alex@gmail.com

УДК 621.396.663

*И.А. Сухов, В.П. Акимов*

## **МЕТОД ПОВЫШЕНИЯ РАЗРЕШАЮЩЕЙ СПОСОБНОСТИ ПЕЛЕНГАТОРА С КОЛЬЦЕВОЙ АНТЕННОЙ РЕШЕТКОЙ ПРИ ИСПОЛЬЗОВАНИИ АЛГОРИТМА MUSIC**

*I.A. Sukhov, V.P. Akimov*

### **RESOLUTION IMPROVEMENT OF MUSIC ALGORITHM FOR THE DIRECTION-FINDER BASED ON A CIRCULAR ANTENNA ARRAY**

Предложен метод увеличения разрешающей способности алгоритма MUSIC для кольцевых антенных решеток, основанный на формировании нескольких подрешеток из элементов кольцевой антенной решетки. Данный метод имеет повышенную разрешающую способность при меньших вычислительных затратах.

**КЛАССИФИКАЦИЯ МНОЖЕСТВА СИГНАЛОВ. КОЛЬЦЕВАЯ АНТЕННАЯ РЕШЕТКА. РАДИОПЕЛЕНГАЦИЯ. АЛГОРИТМ ПЕЛЕНГАЦИИ. ФОРМИРОВАНИЕ ЛУЧЕЙ.**

Method of the MUSIC algorithm improvement resolution for the direction-finder is suggested. The method is based on forming number of subarrays from elements of the circular antenna array. It gives the possibility to improve the direction-finder resolution.

**MULTIPLE SIGNAL CLASSIFICATION. CIRCULAR ANTENNA ARRAY. DIRECTION-FINDING. DIRECTION-FINDING ALGORITHM. BEAMFORMING.**

Одним из наиболее известных сверхразрешающих алгоритмов, используемых для пеленгации в сложной электромагнитной обстановке, является алгоритм MUSIC (Multiple Signal Classification) [1].

В данной статье предлагается метод увеличения разрешающей способности алгоритма MUSIC. Улучшение разрешающих характеристик алгоритма пеленгации достигается за счет формирования направленных подрешеток из элементов кольцевой антенной решетки.

Пусть на  $N$ -элементную кольцевую антенную решетку, состоящую из точечных антенных элементов, падают электромаг-

нитные волны от источников радиоизлучения, находящихся в дальней зоне излучения. На всех элементах антенной решетки одновременно измеряется амплитудно-фазовое распределение поля от источников радиоизлучения. Измеренные данные удобно записать в виде  $N$ -мерного вектора  $\mathbf{X} = [x_1(t_k), x_2(t_k), \dots, x_N(t_k)]^T, k = 1, \dots, K$ . Задачей является определение количества приходящих сигналов и азимутов источников радиоизлучения по  $K$  временным выборкам (отсчетам) измеренного вектора в моменты времени  $t_k, k = 1, \dots, K$ .

Сигнал, принимаемый  $n$ -м антенным элементом  $x_n(t_k)$ , может быть представлен

в следующем виде:

$$\mathbf{X}(t_k) = \sum_{m=1}^M \mathbf{a}(\varphi_m) s_m(t_k) + \mathbf{N}(t_k), k = 1, \dots, K, (1)$$

где приняты следующие обозначения:

$$\begin{aligned} \mathbf{a}(\varphi_m) &= [F_1(\varphi_m) \exp(-j\psi_1(\varphi_m)), \\ &F_2(\varphi_m) \exp(-j\psi_2(\varphi_m)), \dots, \\ &\dots, F_N(\varphi_m) \exp(-j\psi_N(\varphi_m))]^T \end{aligned}$$

– матрица-столбец, определяющая амплитуды и фазы сигналов на элементах антенной решетки при условии, что сигнал приходит с направления  $\varphi_m$ ;

$F_n(\varphi_m)$  – амплитуда принимаемого сигнала  $n$ -м антенным элементом при условии, что сигнал приходит с направления  $\varphi_m$ ;

$s_m(t_k)$  – сигнал от  $m$ -го источника радиозлучения в момент времени  $t_k$ ;

$\mathbf{N} = [n_1(t_k), n_2(t_k), \dots, n_N(t_k)]^T, k = 1, \dots, K$  – матрица размером  $N \times K$ , представляющая аддитивный гауссовский шум в приемных трактах;

$\Phi \in \{\varphi_1, \varphi_2, \dots, \varphi_M\}$  – направления на источники сигналов;

$e^{j\psi_n(\varphi)}$  – фаза сигнала, приходящего с азимута  $\varphi$ , на  $n$ -м антенном элементе;

$\Gamma$  – символ транспонирования.

Выражение (1) можно записать в более компактной форме:

$$\mathbf{X} = \mathbf{A}(\Phi) \cdot \mathbf{S} + \mathbf{N}, (2)$$

где  $\mathbf{X}$  и  $\mathbf{N}$  – матрицы размером  $N \times K$ ,  $\mathbf{S}$  – матрица размером  $M \times K$ .

$$\mathbf{X} = [x_1(t_k), x_2(t_k), \dots, x_N(t_k)]^T, k = 1, \dots, K;$$

$$\mathbf{A}(\Phi) = [\mathbf{a}(\varphi_1), \mathbf{a}(\varphi_2), \dots, \mathbf{a}(\varphi_M)];$$

$$\mathbf{S} = [s_1(t_k), s_2(t_k), \dots, s_M]^T, k = 1, \dots, K;$$

$$\mathbf{N} = [n_1(t_k), n_2(t_k), \dots, n_N(t_k)]^T, k = 1, \dots, K.$$

Для кольцевой антенной решетки функция  $\psi_n(\varphi_m)$  выглядит следующим образом:

$$\psi_n(\varphi_m) = \pi d \cos\left(\frac{2\pi(n-1)}{N} - \varphi_m\right), (3)$$

где  $d$  – диаметр антенной решетки в длинах волн.

Корреляционная матрица принятых сигналов  $\mathbf{R}$  выглядит следующим образом:

$$\begin{aligned} \mathbf{R} &= \mathbf{X} \cdot \mathbf{X}^H = \\ &= (\mathbf{A}(\Phi) \cdot \mathbf{S} + \mathbf{N})(\mathbf{A}(\Phi) \cdot \mathbf{S} + \mathbf{N})^H = \end{aligned} (4)$$

$$= \mathbf{A}(\Phi) \cdot \mathbf{S} \cdot \mathbf{S}^H \cdot \mathbf{A}(\Phi)^H + \mathbf{N} \cdot \mathbf{N}^H,$$

где  $\mathbf{H}$  – символ комплексного сопряжения и транспонирования.

Предположим, что шумы на антенных элементах не коррелированы между собой, тогда:

$$n_i(t)n_j(t) = \begin{cases} \sigma^2, & i = j \\ 0, & i \neq j \end{cases}, (5)$$

где  $\sigma^2$  – мощность шума на антенных элементах.

$$\mathbf{R} = \mathbf{A}(\Phi) \cdot \mathbf{S} \cdot \mathbf{S}^H \cdot \mathbf{A}(\Phi)^H + \sigma^2 \mathbf{I}, (6)$$

где  $\mathbf{I}$  – единичная матрица.

Корреляционная матрица может быть разложена на собственные векторы  $\mathbf{v}_n = [v_{1,n}, v_{2,n}, \dots, v_{N,n}]^T, n = 1, \dots, N$  и собственные числа  $\Lambda = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_M, \lambda_{M+1}, \dots, \lambda_N)$ . Собственные числа сортируются по убыванию и, используя критерий Акайке [2], пространство корреляционной матрицы делится на сигнальное и шумовое подпространства. Корреляционная матрица в базисе из собственных векторов выглядит следующим образом:

$$\mathbf{R} = \mathbf{Q}_S \cdot \mathbf{\Lambda}_S \cdot \mathbf{Q}_S^H + \mathbf{Q}_N \cdot \mathbf{\Lambda}_N \cdot \mathbf{Q}_N^H, (7)$$

где  $\mathbf{\Lambda}_S = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_M)$  – собственные числа сигнального подпространства;

$\mathbf{\Lambda}_N = \text{diag}(\lambda_{M+1}, \lambda_{M+2}, \dots, \lambda_N)$  – собственные числа шумового подпространства;

$\mathbf{Q}_S = [\mathbf{v}_1, \mathbf{v}_1, \dots, \mathbf{v}_M]$  – матрица собственных векторов-столбцов сигнального подпространства;

$\mathbf{Q}_N = [\mathbf{v}_{M+1}, \mathbf{v}_{M+2}, \dots, \mathbf{v}_N]$  – матрица собственных векторов-столбцов шумового подпространства.

Сверхразрешающий алгоритм пеленгации MUSIC основан на ортогональности  $\mathbf{a}(\Phi)^H$  (в направлениях на источники сигналов) шумовому подпространству  $\mathbf{Q}_N$  корреляционной матрицы  $\mathbf{R}$ :

$$\mathbf{a}(\Phi)^H \cdot \mathbf{Q}_N = 0. (8)$$

В алгоритме MUSIC используется квадрат произведения  $\mathbf{a}(\Phi)$  на матрицу шумового подпространства:

$$\begin{aligned} |\mathbf{a}(\Phi)^H \mathbf{Q}_N|^2 &= (\mathbf{a}(\Phi)^H \mathbf{Q}_N) \times \\ &\times (\mathbf{a}(\Phi)^H \mathbf{Q}_N)^H = \mathbf{a}(\Phi)^H \mathbf{Q}_N \mathbf{Q}_N^H \mathbf{a}(\Phi). \end{aligned} (9)$$

Пеленгационная характеристика алгоритма MUSIC выглядит следующим образом:

$$P_{\text{MUSIC}}(\varphi) = \frac{1}{\mathbf{a}(\varphi)^H \mathbf{Q}_N \mathbf{Q}_N^H \mathbf{a}(\varphi)}. \quad (10)$$

Если угол  $\varphi$  совпадает с одним из азимутов прихода сигнала  $\Phi$ , то знаменатель функции  $P_{\text{MUSIC}}(\varphi)$  стремится к нулю, а сама функция имеет максимум в этом направлении.

Данный алгоритм обеспечивает высокую разрешающую способность и высокую точность пеленгации при воздействии нескольких сигналов на антенную решетку. В данной работе предлагается способ увеличения разрешающей способности. Для этого процесс оценки пеленга необходимо разделить на две стадии. На первой стадии проводится предварительная оценка пеленгов любым алгоритмом пеленгации (например, неадаптивный формирователь луча [3], метод Кейпона [4], MUSIC). На второй стадии с помощью данных, полученных с антенных элементов решетки, происходит уточнение оцененных пеленгов согласно структурной схеме на рис. 1.

Антенная решетка условно разбивается на  $B$  подрешеток по  $L$  элементов, на основе которых формируется  $B$  лучей (диаграмм направленности) в направлениях  $\Theta \in \{\theta_1, \theta_2, \dots, \theta_B\}$ . Направления, в которых формируются лучи, выбираются близкими к пеленгам, полученным на первой стадии. Затем проводится процесс пеленгации классическим алгоритмом MUSIC, используя данные от подрешеток.

Формирование лучей может быть реализовано различными диаграммообразующими

методами. В рамках данной работы выбран классический метод формирования луча [3], основанный на синфазном сложении сигналов, принимаемых антенной решеткой с заданного направления. Согласно этому методу вектор измеренных данных  $\mathbf{X}$  необходимо умножить на вектор весовых коэффициентов  $\mathbf{W}$ , который в свою очередь совпадает с вектором  $\mathbf{a}(\Theta)$ , определяющим амплитуды и фазы на антенных элементах в случае прихода сигнала с направления  $\Theta$ .

Так как антенная подрешетка может быть составлена из любых элементов антенной решетки, то вектор  $\mathbf{a}_b(\varphi_m)$ , определяющий амплитуды и фазы на элементах антенной подрешетки, состоит из соответствующих элементов  $\mathbf{a}(\varphi_m)$ . В свою очередь весовой вектор подрешетки  $\mathbf{W}_b$  может быть представлен следующим образом:

$$\mathbf{W}_b = \mathbf{a}_b(\varphi_m), b = 1, \dots, B. \quad (11)$$

Выходные данные от подрешеток  $\mathbf{Y} = [y_1(t_k), y_2(t_k), \dots, y_B(t_k)]^T, k = 1, \dots, K$  определяются произведением весового вектора соответствующей подрешетки на вектор принятых данных соответствующими антенными элементами подрешетки  $\mathbf{X}_b$ :

$$y_b(t_k) = \mathbf{W}_b^H \cdot \mathbf{X}_b, b = 1, \dots, B. \quad (12)$$

Аналогично могут быть получены данные на выходе от подрешеток в результате воздействия  $M$  сигналов на антенную решетку:

$$\mathbf{Y} = \mathbf{W}_b^H \cdot \mathbf{A}_b(\Theta) \cdot \mathbf{S} + \mathbf{W}_b^H \cdot \mathbf{N}_b, b = 1, \dots, B. \quad (13)$$

Корреляционная матрица для данных на выходе подрешеток имеет следующий вид:

$$\mathbf{R}_B = \mathbf{Y} \cdot \mathbf{Y}^H. \quad (14)$$

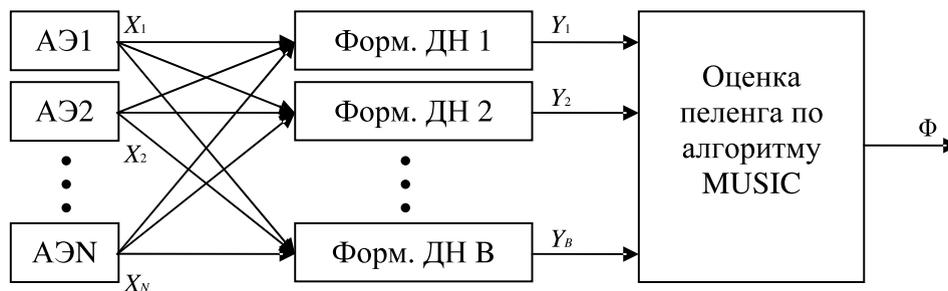


Рис. 1. Структурная схема алгоритма пеленгации (стадия 2)

Корреляционная матрица  $\mathbf{R}_B$  размером  $B \times B$  может быть разложена на собственные векторы  $\mathbf{v}_b = [v_{1,b}, v_{2,b}, \dots, v_{B,b}]^T, b = 1, \dots, B$  и собственные числа, принадлежащие сигнальному подпространству  $\Lambda_{SB} = \text{diag}(\lambda_{b_1}, \lambda_{b_2}, \dots, \lambda_{b_M})$  и шумовому подпространству  $\Lambda_{NB} = \text{diag}(\lambda_{b_1}, \lambda_{b_2}, \dots, \lambda_{b_{B-M}})$ :

$$\mathbf{R}_B = \mathbf{Q}_{SB} \cdot \Lambda_{SB} \cdot \mathbf{Q}_{SB}^H + \mathbf{Q}_{NB} \cdot \Lambda_{NB} \cdot \mathbf{Q}_{NB}^H \quad (15)$$

где  $\mathbf{Q}_{SB}$  и  $\mathbf{Q}_{NB}$  – матрицы, состоящие из собственных векторов-столбцов сигнального и шумового подпространств корреляционной матрицы  $\mathbf{R}_B$  соответственно.

Для обработки данных от подрешеток по алгоритму MUSIC необходимо использовать вектор  $\mathbf{a}_b(\varphi_m)$ , умноженный на весовой вектор  $\mathbf{W}_B$ , и шумовую матрицу  $\mathbf{Q}_{NB}$ , состоящую из собственных векторов шумового подпространства корреляционной матрицы  $\mathbf{R}_B$ . Пеленгационная характеристика алгоритма MUSIC выглядит следующим образом:

$$PB_{\text{MUSIC}}(\varphi) = \frac{1}{(\mathbf{W}_B \cdot \mathbf{a}_B(\varphi))^H \cdot \mathbf{Q}_{NB} \cdot \mathbf{Q}_{NB}^H \cdot (\mathbf{W}_B \cdot \mathbf{a}_B(\varphi))} \quad (16)$$

Для определения пеленга необходимо найти  $M$  значений углов  $\varphi$ , при которых пеленгационная характеристика  $PB_{\text{MUSIC}}(\varphi)$  имеет наибольшие локальные максимумы. Эти углы и будут оценками пеленгов приходящих сигналов. Количество приходящих на антенную решетку сигналов  $M$  можно оценить по критерию Акайке или по критерию минимальной длины описания [5].

При применении описанного метода имеется несколько степеней свободы. Количество антенных элементов, используе-

мых для объединения в подрешетку, может быть от одного до  $N$ , при этом для объединения в подрешетку могут использоваться любые антенные элементы. Количество подрешеток  $B$  ничем не ограничено, но увеличение количества подрешеток приводит к значительному увеличению вычислительных ресурсов, т. к.  $B$  определяет порядок корреляционной матрицы подрешеток  $\mathbf{R}_B$ . Направления главных максимумов диаграмм направленностей подрешеток  $\Theta$  могут быть выбраны любыми, но наилучший эффект достигается в случае, если сформированные лучи направлены в область предположительных углов прихода сигнала.

На рис. 2 представлен график зависимости пеленгационных характеристик  $P_{\text{MUSIC}}$  и  $PB_{\text{MUSIC}}$  в случае прихода двух сигналов с направлений  $95^\circ$  и  $110^\circ$  на кольцевую антенную решетку из ненаправленных антенных элементов. Для расчета пеленгационных характеристик использовалась девятиэлементная антенная решетка  $N = 9$ , количество отсчетов данных для оценки пеленга  $K = 10$ , соотношение сигнал/шум (С/Ш) 10 дБ, диаметр антенной решетки равен длине волны принимаемых сигналов. Для расчета  $PB_{\text{MUSIC}}$  использовались шесть подрешеток из девяти элементов, формирующих диаграммы в направлениях  $70^\circ, 80^\circ, 90^\circ, 100^\circ, 110^\circ, 120^\circ$ .

Из графика видно, что пеленгационная характеристика классического алгоритма пеленгации MUSIC имеет один максимум, в то время как пеленгационная характеристика предлагаемого алгоритма имеет максимумы в направлениях, близких к истинным пеленгам.

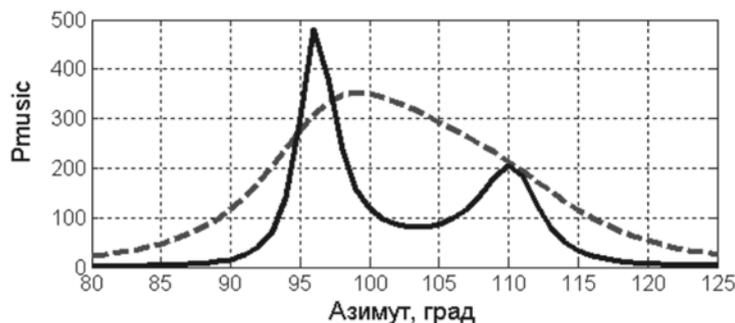


Рис. 2. Пеленгационные характеристики  $P_{\text{MUSIC}}$  (---) и  $PB_{\text{MUSIC}}$  (—)

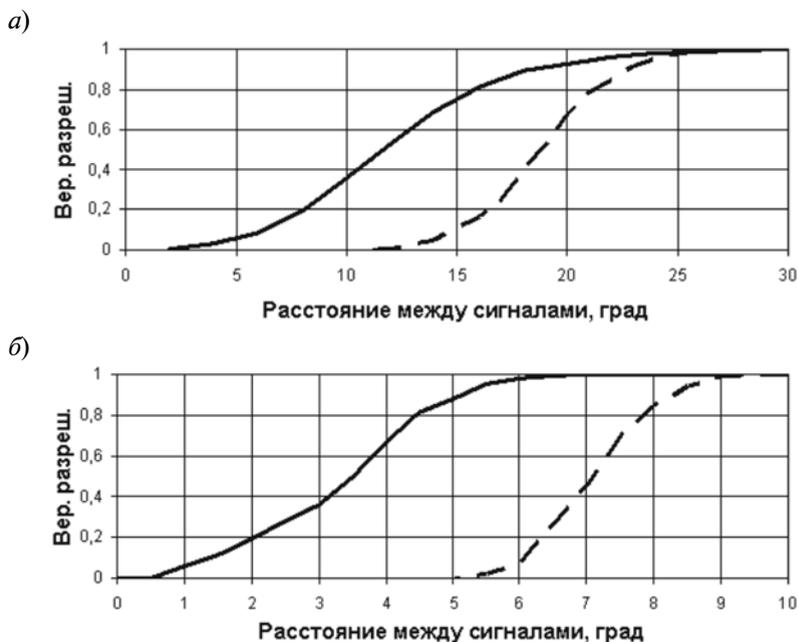


Рис. 3. График зависимости вероятности разрешения двух сигналов алгоритмом MUSIC (---) и предлагаемым алгоритмом пеленгации (—) при  $C/Ш = 10$  дБ,  $K = 10$  (а) и при  $C/Ш = 20$  дБ,  $K = 100$  (б)

Проведено моделирование прихода двух сигналов с целью сравнения разрешающей способности классического алгоритма пеленгации MUSIC и алгоритма MUSIC на основе формирования лучей. Моделирование проводилось в аналогичных условиях, сформированы три диаграммы в направлениях  $90^\circ$ ,  $105^\circ$  и  $120^\circ$ . Один из пеленгуемых сигналов расположен на азимуте  $110^\circ$ , положение второго изменялось от  $80^\circ$  до  $108^\circ$ . Для каждого положения сигнала эксперимент повторялся 1000 раз для оценки вероятности разрешения сигналов. Результаты эксперимента представлены на рис. 3 а.

Аналогично проведено моделирование прихода двух сигналов при соотношении сигнал/шум 20 дБ, числе отсчетов  $K = 100$ , диаметром  $d$ , равном длине волны пеленгуемых сигналов. Азимут одного сигнала равен  $60^\circ$ , азимут второго сигнала менялся от  $50^\circ$  до  $59,5^\circ$ . Предлагаемым алгоритмом сформировано три луча из всех антенных элементов в направлениях  $50^\circ$ ,  $60^\circ$  и  $70^\circ$ . В результате проведенного эксперимента получен график зависимости вероятности разрешения двух сигналов от углового расстояния между ними (рис. 3 б).

Из графика видно, что предлагаемый алгоритм пеленгации обеспечивает более высокую вероятность разрешения двух сигналов.

Проведенные исследования алгоритма пеленгации MUSIC на основе формирования лучей показали, что наилучшая разрешающая способность достигается при формировании подрешеток из всех антенных элементов  $L = N$ .

Увеличение количества подрешеток  $B$  не приводит к значительному увеличению разрешающей способности. Количество подрешеток (лучей) определяет порядок корреляционной матрицы. Увеличение количества подрешеток значительно увеличивает вычислительные затраты. Во многих случаях для разрешения двух сигналов достаточным является формирование трех лучей в области предположительного нахождения пеленгуемых сигналов. Предлагаемый алгоритм также может быть применен и для антенных решеток, состоящих из направленных антенных элементов, при этом амплитудные диаграммы направленности могут быть учтены функцией  $F_n$ . Взаимное влияние между антенными эле-

ментами может быть учтено, т. к. матрица столбец  $\mathbf{a}(\varphi)$  может быть измерена при калибровке антенной системы и использована при применении предлагаемого алгоритма.

Предлагаемый алгоритм не требует значительных вычислительных затрат, т. к. для повышения разрешающей способности может использоваться меньше лучей, чем количество антенных элементов, что приводит к понижению размера корреляционной матрицы. Также снижаются вычислительные затраты при поиске локальных максимумов пеленгационной характеристики, т. к. не требуется производить поиск по всему азимуту, достаточно провести поиск в секторе углов, в которых были сформированы лучи. Для дополнительного уменьшения вычислительных затрат первую стадию оценки пеленгов

можно провести алгоритмом формирования луча [3], не требующего разложения корреляционной матрицы по собственным векторам и собственным числам. Пеленгационная функция этого алгоритма выглядит следующим образом:

$$P(\varphi) = \mathbf{a}(\varphi)^H \cdot \mathbf{R} \cdot \mathbf{a}(\varphi). \quad (17)$$

В данной статье предложен метод, который позволяет увеличить вероятность разрешения близкорасположенных источников излучения и может быть применен в качестве второй стадии к любому алгоритму оценки пеленгов. Предложенный метод не требует значительных вычислительных затрат, т. к. порядок корреляционной матрицы на второй стадии оценки пеленга может быть меньше, чем количество элементов в антенной решетке.

#### СПИСОК ЛИТЕРАТУРЫ

1. Schmidt, R. Multiple emitter location and signal parameter estimation [Text] / R. Schmidt // Proc. RADC Spectrum Estimation Workshop. —1979. —P. 243–258.
2. Akaike, H. A new look at statistical model identification [Text] / H. Akaike // IEEE Trans. Automat. Control. —1974. —№ 19. —P. 716–723.
3. Krim, H. Two decades of array signal processing research: the parametric approach [Text] / H. Krim, M. Viberg // IEEE Signal Process. Mag.

—1999. —№ 13 (4). —P. 67–94.

4. Кейпон, Дж. Обработка данных большой сейсмической группы способом многомерного максимального правдоподобия [Текст] / Р. Дж. Гринфилд, Р. Дж. Колкер // ТИИЭР. —1967. —Т. 55. —№ 22. —С. 66–82.

5. Ратынский, М.В. Адаптация и сверхразрешение в антенных решетках [Текст] / М.В. Ратынский. —М.: Радио и связь, 2003. — 200 с.

#### REFERENCES

1. Schmidt R. Multiple emitter location and signal parameter estimation / Proc. RADC Spectrum Estimation Workshop. —1979. —P. 243–258.
2. Akaike H. A new look at statistical model identification / IEEE Trans. Automat. Control. —1974. — № 19. —P. 716–723.
3. Krim H. Two decades of array signal processing research: the parametric approach / IEEE Signal

Process. Mag. —1999. —№ 13 (4). —P. 67–94.

4. Keipon Dzh. Obrabotka dannykh bol'shoi seismicheskoi gruppy sposobom mnogomernogo maksimal'nogo pravdopodobii / TIIEP. —1967. —Т. 55. —№ 22. — S. 66–82. (rus)

5. Ratynskii M.V. Adaptatsiia i sverkhrazreshenie v antenykh reshetkakh. —Moscow: Radio i sviaz', 2003. — 200 s. (rus)

**СУХОВ Игорь Александрович** — аспирант кафедры радиотехники Санкт-Петербургского государственного политехнического университета.

195251, Россия, Санкт-Петербург, ул. Политехническая, д. 29.

E-mail: sia23@mail.ru

**SUKHOV, Igor A.** — St. Petersburg State Polytechnical University.

195251, Politekhnikeskaya Str. 29, St. Petersburg, Russia.

E-mail: sia23@mail.ru

**АКИМОВ Валерий Петрович** – профессор кафедры радиотехники Санкт-Петербургского государственного политехнического университета, доктор физико-математических наук.

195251, Россия, Санкт-Петербург, ул. Политехническая, д. 29.

E-mail: [valeri\\_akimov@mail.ru](mailto:valeri_akimov@mail.ru)

**AKIMOV, Valerij P.** – *St. Petersburg State Polytechnical University.*

195251, Politekhnikeskaya Str. 29, St. Petersburg, Russia.

E-mail: [valeri\\_akimov@mail.ru](mailto:valeri_akimov@mail.ru)

УДК 621.37

*М.А. Алексеев, В.А. Сороцкий*

## **УМЕНЬШЕНИЕ ИСКАЖЕНИЙ ПРИ ГЕНЕРИРОВАНИИ РАДИОСИГНАЛОВ КЛЮЧЕВЫМИ МЕТОДАМИ**

*M.A. Alekseev, V.A. Sorotsky*

### **DISTORTIONS REDUCTION FOR SWITCH-MODE GENERATED RADIO SIGNALS**

Рассмотрен перспективный метод генерирования радиосигналов с заданными характеристиками в ключевых усилителях мощности. На основе математической модели радиосигнала проведена оценка погрешности, обусловленной отклонением огибающей и фазы высокочастотного заполнения от их эталонных значений. Метод может использоваться в радиопередающих устройствах, применяемых в системах радиосвязи, радионавигации и радиовещания.

**РАДИОСИГНАЛ. РАДИОПЕРЕДАЮЩЕЕ УСТРОЙСТВО. КЛЮЧЕВОЙ РЕЖИМ. ИСКАЖЕНИЕ. МОДЕЛЬ.**

The paper gives an overview of promising switch-mode method for generation of radio signals with specified characteristics. Based on the mathematical model of radio signal an error assessment for the envelope and phase deviation is proposed. The method can be used in communication, navigation or broadcasting radio transmitters.

**RADIO SIGNAL. RADIO TRANSMITTER. SWITCH MODE. DISTORTIONS. MODEL.**

Благодаря своей высокой энергетической эффективности ключевые генераторы нашли широкое применение в качестве усилителей мощности радиопередающих устройств (РПДУ), используемых в системах радиосвязи и радионавигации. Сравнительно недавно область применения ключевого режима расширилась в сторону радиовещания, создав предпосылки для разработки высокоэффективных РПДУ диапазонов длинных и средних волн [1, 2].

В перечисленных выше разнообразных применениях РПДУ должны не только обеспечивать высокие энергетические показатели, но и осуществлять формирование радиосигналов с заданными требованиями к поведению огибающей и фазы высокочастотного заполнения. Вместе с тем хорошо известно, что реализация последнего условия в ключевом режиме работы представляет собой нетривиальную задачу.

Цель настоящего исследования – оцен-

ка достижимой точности формирования огибающей и фазы высокочастотного заполнения радиосигналов при использовании комбинированного метода, предусматривающего динамическое изменение на отдельных интервалах времени количества генераторных ячеек, задействованных в формировании сигнала, и длительностей генерируемых ими выходных импульсов [3].

#### **Математическая модель радиосигнала**

Рассмотрим в общем виде напряжение (рис. 1), которое может быть сформировано с помощью набора генераторных ячеек (ГЯ), образующих усилитель мощности РПДУ. Здесь напряжения  $U_1, U_2, \dots, U_N$  – амплитуды ступеней,  $\alpha_{11}, \alpha_{12}, \alpha_{21}, \alpha_{22}, \dots, \alpha_{N1}, \alpha_{N2}$  – угловые координаты ступеней, выраженные в радианах. При этом условимся, что первый индекс параметра  $\alpha$  обозначает номер ступени, а второй индекс, принима-

ющий значения 1 или 2, обозначает, соответственно, момент начала или окончания этой ступени.

Для того чтобы вычислить ток, протекающий в антенне радиопередатчика под воздействием знакопеременной последовательности импульсов напряжения  $u_n$ , показанной на рис. 1, воспользуемся методом наложения. На основе данного метода ток, протекающий в антенне РПДУ, может быть представлен в виде суперпозиции токов  $i_n(t)$ , обусловленных воздействием знакопеременной последовательности прямоугольных импульсов с амплитудами  $U_n$  и угловыми координатами  $\alpha_{n1}, \alpha_{n2}$  ( $n = 1, 2, \dots, N$ ):

$$i_A(t) = \sum_{n=1}^N i_n(t). \quad (1)$$

Каждый из импульсов напряжения  $u_n$  удобно представить в виде суперпозиции двух ступеней напряжения, сдвинутых во времени друг относительно друга и имеющих противоположную полярность:

$$u_n(t) = (-1)^{n+1} U_n \left[ 1 \left( t - \frac{\alpha_{n1}}{\omega} \right) - 1 \left( t - \frac{\alpha_{n2}}{\omega} \right) \right], \quad (2)$$

где  $1(t)$  – единичная функция;  $\omega$  – круговая частота генерируемых колебаний.

С учетом (1), (2) ток в антенне РПДУ окончательно может быть представлен в виде

$$i_A(t) = \sum_{n=1}^N (-1)^{n+1} \left[ i_n \left( t - \frac{\alpha_{n1}}{\omega} \right) - i_n'' \left( t - \frac{\alpha_{n2}}{\omega} \right) \right], \quad (3)$$

где ток  $i_n(t)$  обусловлен первым слагаемым, стоящим в прямоугольных скобках в (2) (т. е. фронтом импульса напряжения), а ток  $i_n''(t)$  – соответственно вторым слагаемым в этом выражении, определяющим спад импульса напряжения  $u_n(t)$ .

Для вычисления токов  $i_n(t)$  и  $i_n''(t)$  ( $n = 1, 2, \dots, N$ ) воспользуемся математическим аппаратом операционного исчисления [4]. Изображение преобразованного по Лапласу скачка напряжения положительной полярности амплитудой  $U_n$ , произошедшего в момент времени  $t_{n1} = \alpha_{n1}/\omega$ , имеет вид:

$$u_n^{(+)}(p) = \frac{U_n}{p} e^{-pt_{n1}}. \quad (4)$$

По аналогии с (4) запишем изображение преобразованного по Лапласу скачка напряжения отрицательной полярности

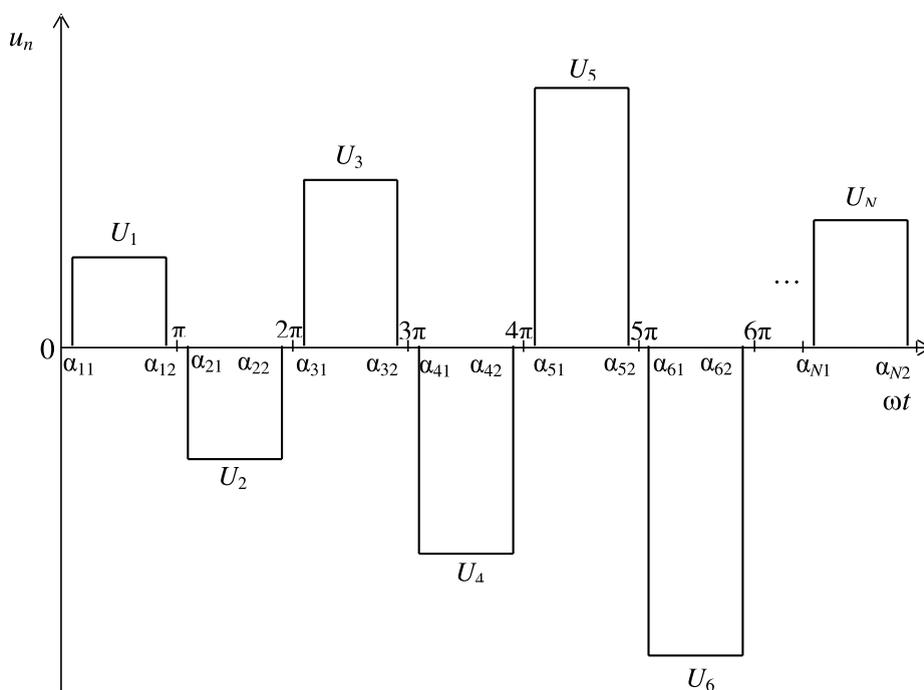


Рис. 1. Выходное напряжение усилителя мощности

амплитудой  $U_n$ , произошедшего в момент времени  $t_{n2} = \alpha_{n2}/\omega$ , имеющего вид:

$$u_n^{(-)}(p) = -\frac{U_n}{p} e^{-pt_{n2}}. \quad (5)$$

Суммируя слагаемые (4) и (5) для всех значений  $n$  ( $n = 1, 2, \dots, N$ ), получим выражение в операторной форме для напряжения, действующего на входе антенной системы:

$$u(p) = \sum_{n=1}^N (-1)^{n+1} \frac{U_n}{p} [e^{-pt_{n1}} - e^{-pt_{n2}}]. \quad (6)$$

Для определения изображения преобразованного по Лапласу тока антенны помимо напряжения  $\tilde{u}(p)$  необходимо также записать в операторной форме сопротивление антенны. Полагая, что эквивалентная схема замещения антенны, включая согласующую цепь, может быть представлена в виде последовательного контура с элементами  $L$ ,  $C$  и  $r$ , находим:

$$\begin{aligned} Z(p) &= pL + \frac{1}{pC} + r = \\ &= \frac{L}{p} \left[ p + \frac{r}{2L} (1 + \sqrt{1 - 4Q^2}) \right] \times \\ &\times \left[ p + \frac{r}{2L} (1 - \sqrt{1 - 4Q^2}) \right], \end{aligned} \quad (7)$$

где  $Q = \frac{\sqrt{L/C}}{r}$  – добротность антенного контура.

С учетом (6) и (7) изображение тока, протекающего через антенну, может быть представлено в следующем виде:

$$\begin{aligned} i_A(p) &= \frac{u(p)}{Z(p)} = \\ &= \frac{\sum_{n=1}^N (-1)^{n+1} \frac{U_n}{p} [e^{-pt_{n1}} - e^{-pt_{n2}}]}{\frac{L}{p} \left[ p + \frac{r}{2L} (1 + \sqrt{1 - 4Q^2}) \right] \left[ p + \frac{r}{2L} (1 - \sqrt{1 - 4Q^2}) \right]}. \end{aligned} \quad (8)$$

Из последнего выражения нетрудно найти оригинал тока в антенне:

$$\begin{aligned} \tilde{i}_A(\omega t) &= 2 \frac{e^{-\frac{\omega t}{2Q}}}{\sqrt{4Q^2 - 1}} \sum_{n=1}^N (-1)^{n+1} \tilde{u}_n \times \\ &\times \left\{ e^{\frac{\omega t_{n1}}{2Q}} \sin \left[ \frac{\sqrt{4Q^2 - 1}}{2Q} \omega(t - t_{n1}) \right] - \right. \end{aligned} \quad (9)$$

$$\left. - e^{\frac{\omega t_{n2}}{2Q}} \sin \left[ \frac{\sqrt{4Q^2 - 1}}{2Q} \omega(t - t_{n2}) \right] \right\}.$$

Для удобства вычислений ток и напряжение в (9) представлены в нормированном виде:

$$\tilde{i}_A(\omega t) = \frac{i_A(\omega t)}{[i_A(\omega t)]_{\text{имп max}}}, \quad \tilde{u}_n = \frac{U_n}{[i_A(\omega t)]_{\text{имп max}} r},$$

где  $[i_A(\omega t)]_{\text{имп max}}$  – максимальное значение тока в антенне РПДУ.

Полученные математические соотношения позволяют определить основные характеристики генерируемого в РПДУ сигнала и оценить степень отклонения этих характеристик от эталонных значений. В качестве примера рассмотрим особенности формирования радиосигналов с характеристиками, приведенными в [5, 6].

### Регулирование характеристик огибающей радиосигнала

Для определения амплитуд полуволн тока антенны необходимо предварительно вычислить временные положения экстремумов на каждом полупериоде:

$$\begin{aligned} \frac{d\tilde{i}_A(t)}{dt} &= \sum_{n=1}^N (-1)^{n+1} \tilde{u}_n \times \\ &\times \left\{ -e^{\frac{\omega t_{n1}}{2Q}} \sin \left[ \frac{\sqrt{4Q^2 - 1}}{2Q} \omega(t - t_{n1}) \right] + \right. \\ &+ e^{\frac{\omega t_{n2}}{2Q}} \sin \left[ \frac{\sqrt{4Q^2 - 1}}{2Q} \omega(t - t_{n2}) \right] \left. \right\} + \\ &+ \sum_{n=1}^N (-1)^{n+1} \tilde{u}_n \sqrt{4Q^2 - 1} \times \\ &\times \left\{ e^{\frac{\omega t_{n1}}{2Q}} \cos \left[ \frac{\sqrt{4Q^2 - 1}}{2Q} \omega(t - t_{n1}) \right] - \right. \\ &\left. - e^{\frac{\omega t_{n2}}{2Q}} \cos \left[ \frac{\sqrt{4Q^2 - 1}}{2Q} \omega(t - t_{n2}) \right] \right\} = 0. \end{aligned} \quad (10)$$

Решением этого уравнения является набор точек

$$(x_{\max})_n = \frac{1}{b} \left[ n\pi - \arctg \frac{\sum_{n=1}^N (-1)^{n+1} \varepsilon_{n2}}{\sum_{n=1}^N (-1)^{n+1} \varepsilon_{n1}} \right], \quad (11)$$

$$n = 1, 2, \dots, N,$$

где  $\varepsilon_{n1} = -\mu_{n1} \cos b\alpha_{n1} + 2Qb\mu_{n1} \sin b\alpha_{n1} + \mu_{n2} \times \cos b\alpha_{n2} + 2Qb\mu_{n2} \sin b\alpha_{n2}$ ;

$\varepsilon_{n2} = 2Qb\mu_{n1} \cos b\alpha_{n1} + \mu_{n1} \sin b\alpha_{n1} - \mu_{n2} \sin b\alpha_{n2} - 2Qb\mu_{n2} \sin b\alpha_{n2}$ ;

$b = \frac{\sqrt{4Q^2 - 1}}{2Q}$ ,  $\mu_{n1} = \tilde{u}_n e^{\frac{\omega t_{n1}}{2Q}}$ ,  $\mu_{n2} = \tilde{u}_n e^{\frac{\omega t_{n2}}{2Q}}$ ,  $x = \omega t$ .

Для того чтобы найти амплитуды полу- волн  $(\tilde{i}_{A\max})_n$ , ( $n = 1, 2, \dots, N$ ) тока антен- ны, полученные значения набора времен- ных координат  $(x_{\max})_n = (\omega t_{\max})_n$  необходимо подставить в (9).

В процессе исследования рассмотре- ны три возможных случая регулирования ширины ступеней напряжения, приклады- ваемого ко входу антенной системы: сим- метричное регулирование, когда сужение импульсов напряжения происходит одина- ково с обеих сторон, и два случая несимме- тричного регулирования. В первом из них варьируется положение фронта импульсов напряжения, а положение спада остается фиксированным. Во втором случае несим- метричного регулирования все происходит наоборот: положение фронта фиксирова- но, а временное положение спада варьиру- ется.

Анализируя в целом полученные резуль- таты, стоит отметить следующее. Регулиро- вание ширины ступеней напряжения, при- кладываемого ко входу антенной системы, позволяет достаточно точно подстраивать амплитуду полуволн тока генерируемых радиосигналов по отношению к эталонным значениям. При этом наименее эффектив- ным с точки зрения регулирования ампли- туд полуволн тока антенны оказался второй способ: несимметричное регулирование с варьированием положения фронта импуль- сов напряжения. Дело в том, что заметное сужение ширины импульсов напряжения неизбежно приведет к снижению мощности генерируемого радионавигационного сиг- нала. С учетом этого обстоятельства следует

отдать предпочтение либо симметричному регулированию ширины импульсов напря- жения, либо случаю несимметричного ре- гулирования с неизменным временным по- ложением фронтов импульсов напряжения. Окончательный выбор конкретного метода регулирования может быть сделан с учетом результатов исследования влияния угловых координат импульсов напряжения на харак- теристики высокочастотного заполнения радиоимпульсов, рассмотренных ниже.

### Регулирование характеристик высокочастотного заполнения

Для определения моментов перехода тока антенны через ноль снова обратимся к соотношению (9). Приравняв его нулю и выполнив необходимые математические преобразования, с учетом введенных выше обозначений получим уравнение для оты- скания корней  $\omega t_{Zn}$  ( $n = 1, 2, \dots, N$ ):

$$\sum_{n=1}^N (-1)^{n+1} (\mu_{n1} \sin bx \cos b\alpha_{n1} - \mu_{n1} \cos bx \sin b\alpha_{n1} - \mu_{n2} \sin bx \cos b\alpha_{n2} + \mu_{n2} \cos bx \sin b\alpha_{n2}) = 0. \quad (12)$$

Воспользовавшись известными прави- лами преобразования тригонометрических выражений, окончательно придем к сле- дующему уравнению:

$$\sqrt{\left[ \sum_{n=1}^N (-1)^{n+1} \vartheta_{n1} \right]^2 + \left[ \sum_{n=1}^N (-1)^{n+1} \vartheta_{n2} \right]^2} \times \sin \left( bx + \arctg \frac{\sum_{n=1}^N (-1)^{n+1} \vartheta_{n2}}{\sum_{n=1}^N (-1)^{n+1} \vartheta_{n1}} \right) = 0, \quad (13)$$

где  $\vartheta_{n1} = \mu_{n1} \cos b\alpha_{n1} - \mu_{n2} \cos b\alpha_{n2}$ ,  $\vartheta_{n2} = \mu_{n2} \times \sin b\alpha_{n2} - \mu_{n1} \sin b\alpha_{n1}$ .

Решением (13) является набор точек  $\omega t_{Zn}$  ( $n = 1, 2, \dots, N$ ), в которых ток антен- ны изменяет свою полярность:

$$x_{0n} = \omega t_{Zn} = \frac{1}{b} \left( n\pi - \arctg \frac{\sum_{n=1}^N (-1)^{n+1} \vartheta_{n2}}{\sum_{n=1}^N (-1)^{n+1} \vartheta_{n1}} \right). \quad (14)$$

Семейство зависимостей, иллюстри-

$|n\pi - \omega t_{Zn}|_{\max}$ , град

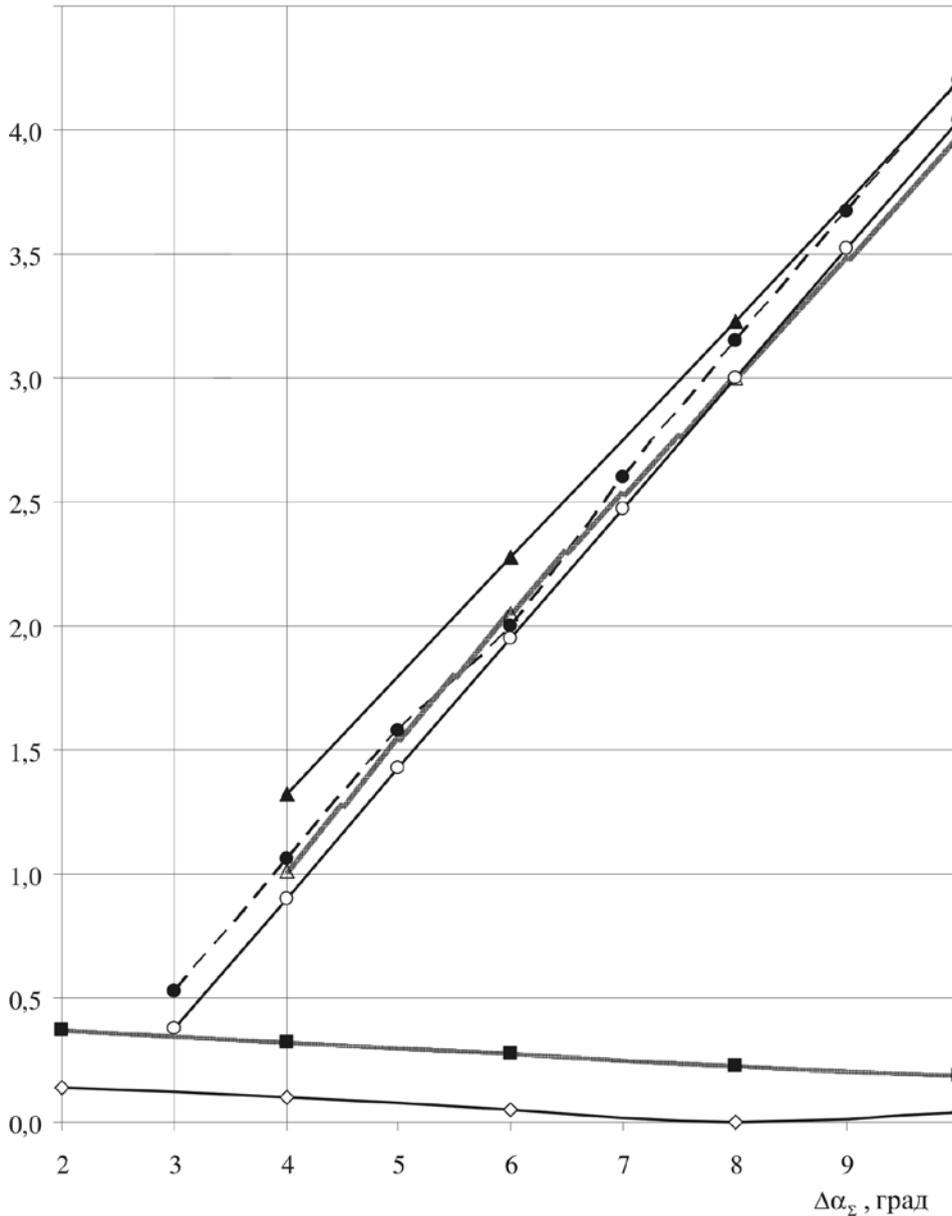


Рис. 2. Зависимость модуля максимального отклонения точек перехода через ноль в радиоимпульсе от величины суммарного сужения импульсов напряжения на выходе усилителя мощности РПДУ

(—■—) 1; (—●—) 2; (—▲—) 3; (—◇—) 4; (—▴—) 5; (—○—) 6

рующее характерное поведение значений точек перехода тока антенны (9) через ноль и построенное с использованием (14), представлено на рис. 2. По вертикальной оси на этом графике отложены максимальные значения модуля  $|n\pi - \omega t_{Zn}|_{\max}$  ( $n = 1,$

$2, \dots, N$ ), реализуемые при заданном суммарном сужении импульсов напряжения  $\Delta\alpha_\Sigma = [\alpha_{n1} - (n - 1)\pi] + (n\pi - \alpha_{n2})$ .

На рис. 2 представлены следующие зависимости:

1 – максимальное отклонение точки перехода через ноль в генерируемом радиосигнале от номинального значения при симметричном<sup>1</sup> регулировании параметров  $\alpha_{n1}$  и  $\alpha_{n2}$  ( $n = 1, 2, \dots, N$ );

2 – максимальное отклонение точки перехода через ноль в радиосигнале от номинального значения при несимметричном регулировании параметров  $\alpha_{n1}$  и  $\alpha_{n2}$  ( $n = 1, 2, \dots, N$ ), при этом регулируется положение переднего фронта импульса напряжения на выходе ключевого генератора ( $\alpha_{n1} = \text{var}$ ), а положение заднего фронта фиксировано и удовлетворяет условию  $(n\pi - \alpha_{n2}) = 1^\circ$ , ( $n = 1, 2, \dots, N$ );

3 – максимальное отклонение точки перехода через ноль в навигационном сигнале от номинального значения при несимметричном регулировании параметров  $\alpha_{n1}$  и  $\alpha_{n2}$  ( $n = 1, 2, \dots, N$ ), при этом регулируется положение заднего фронта импульса напряжения на выходе ключевого генератора ( $\alpha_{n2} = \text{var}$ ), а положение переднего фронта фиксировано и удовлетворяет условию  $[\alpha_{n1} - (n - 1)\pi] = 1^\circ$ , ( $n = 1, 2, \dots, N$ );

4 – максимальное отклонение фазы тока антенны в эталонной точке ( $t = 30$  мкс) при симметричном регулировании параметров  $\alpha_{n1}$  и  $\alpha_{n2}$  ( $n = 1, 2, \dots, N$ );

5 – максимальное отклонение фазы тока антенны в эталонной точке ( $t = 30$  мкс) при несимметричном регулировании параметров  $\alpha_{n1}$  и  $\alpha_{n2}$  ( $n = 1, 2, \dots, N$ ), при этом регулируется положение переднего фронта импульса напряжения на выходе ключевого генератора ( $\alpha_{n1} = \text{var}$ ), а положение заднего фронта фиксировано и удовлетворяет условию  $(n\pi - \alpha_{n2}) = 1^\circ$ , ( $n = 1, 2, \dots, N$ );

<sup>1</sup> Под симметричным регулированием понимается случай, когда сужение импульса напряжения в ключевом генераторе происходит одинаково с обеих сторон.

6 – максимальное отклонение фазы тока антенны в эталонной точке ( $t = 30$  мкс) при несимметричном регулировании параметров  $\alpha_{n1}$  и  $\alpha_{n2}$  ( $n = 1, 2, \dots, N$ ), при этом регулируется положение заднего фронта импульса напряжения на выходе ключевого генератора ( $\alpha_{n2} = \text{var}$ ), а положение переднего фронта фиксировано и удовлетворяет условию  $[\alpha_{n1} - (n - 1)\pi] = 1^\circ$ , ( $n = 1, 2, \dots, N$ ).

Анализ зависимостей на рис. 2 позволяет установить ряд важных закономерностей. Во-первых, и модуль максимального значения фазового сдвига точки перехода тока антенны через ноль в радиосигнале, и модуль максимального отклонения фазы тока антенны в эталонной точке ( $t = 30$  мкс) принимают минимальные значения при симметричном регулировании параметров  $\alpha_{n1}$  и  $\alpha_{n2}$  ( $n = 1, 2, \dots, N$ ). В частности, при симметричной форме импульсов напряжения (см. рис. 1) максимальное отклонение фазы тока антенны в эталонной точке ( $t = 30$  мкс) не превышает  $0,14^\circ$ . Во-вторых, регулирование угловых координат  $\alpha_{n1}$  и  $\alpha_{n2}$  ( $n = 1, 2, \dots, N$ ) импульсов напряжения, формируемых в усилителе мощности РПДУ, позволяет осуществить подстройку значений амплитуд полуволн тока антенны без сколь-либо значительного изменения положения точек перехода через ноль тока антенны.

Полученные в настоящей работе математические соотношения, описывающие поведение тока в антенне РПДУ, а также результаты сравнительного анализа различных способов регулирования характеристик огибающей и фазы высокочастотного заполнения генерируемых радиосигналов позволяют осуществить рациональный выбор параметров ступеней выходного напряжения ключевого усилителя мощности РПДУ исходя из допустимой погрешности относительно эталонных значений.

#### СПИСОК ЛИТЕРАТУРЫ

1. 25 kW–2,000 kW Digital /Analog Medium Wave Transmitters [Электронный ресурс] / Режим доступа: <http://www.nautel.com>
2. DX AM Transmitter Family: 10, 15 and 200-2000 kW Solid State AM Transmitters [Элек-

тронный ресурс] / Режим доступа <http://harris-broadcast.com>

3. Сороцкий, В.А. Методы формирования сигналов в радиопередающих устройствах перспективных навигационных систем [Текст]

/ В.А. Сороцкий, В.М. Царев // Научно-технические ведомости СПбГПУ, Информатика. Телекоммуникации. Управление. –СПб.: Изд-во Политехнического ун-та, 2013. –№ 1 (164). – С. 25–32.

4. Деч, Г. Руководство к практическому применению преобразования Лапласа и

Z-преобразования [Текст] / Г. Деч; Пер. с нем. –М.: Наука, Главная редакция Физматлит, 1971. – 288 с.

5. ГОСТ Р 53168-2008. Система радионавигации «Чайка». Сигналы передающих станций. Технические требования. –М.: Стандартинформ, 2009. – 21 с.

#### REFERENCES

1. 25 kW–2,000 kW Digital/Analog Medium Wave Transmitters Available <http://www.nautel.com>

2. DX AM Transmitter Family: 10, 15 and 200-2000 kW Solid State AM Transmitters. Available <http://harrisbroadcast.com>

3. Sorotskii V.A., Tsarev V.M. Metody formirovaniia signalov v radioperedaiushchikh ustroistvakh perspektivnykh navigatsionnykh sistem [Methods of generation signals in transmitters of perspective navigation systems] / Nauchno-tekhnicheskie vedomosti SPbGPU. Informatika. Telekommunikatsii. Upravlenie [St. Petersburg State

Polytechnical University Journal. Computer Science. Telecommunications and Control Systems]. –St. Petersburg: Izd-vo Politekhnikheskogo un-ta, 2013. –№ 1 (164). –S. 25–32. (rus)

4. Dech G. Rukovodstvo k prakticheskomu primeneniiu preobrazovaniia Laplasa i Z-preobrazovaniia; Per. s nem. – Moscow: Nauka, Glavnaia redaktsiia Fizmatlit, 1971. – 288 s. (rus)

5. GOST R 53168-2008. Sistema radionavigatsii «Chaika». Signaly peredaiushchikh stantsii. Tekhnicheskie trebovaniia. – Moscow: Standartinform, 2009. – 21 s. (rus)

---

**СОРОЦКИЙ Владимир Александрович** – *заведующий кафедрой радиотехники и телекоммуникаций Санкт-Петербургского государственного политехнического университета, профессор, доктор технических наук.*

195251, Санкт-Петербург, ул. Политехническая, д. 29.

E-mail: sorotsky@mail.spbstu.ru

**SOROTSKY, Vladimir A.** *St. Petersburg State Polytechnical University.*

195251, Politekhnikheskaya Str. 29, St. Petersburg, Russia.

E-mail: sorotsky@mail.spbstu.ru

**АЛЕКСЕЕВ Михаил Александрович** – *студент кафедры радиотехники и телекоммуникаций Санкт-Петербургского государственного политехнического университета.*

195251, Санкт-Петербург, ул. Политехническая, д. 29.

**ALEKSEEV, Mikhail A.** *St. Petersburg State Polytechnical University.*

195251, Politekhnikheskaya Str. 29, St. Petersburg, Russia.

УДК 510.67

*В.Н. Волкова, Л.В. Черненкокая, В.Е. Магер*

## **КЛАССИФИКАЦИЯ МОДЕЛЕЙ В СИСТЕМНОМ АНАЛИЗЕ**

*V.N. Volkova, L.V. Chernenkaya, V.E. Mager*

### **CLASSIFICATION OF MODELS IN SYSTEM ANALYSIS**

---

Приведен краткий анализ существующих подходов к классификации моделей. Предложена концепция классификации моделей по методам моделирования. Дана характеристика направлений и методов моделирования, включенных в структуру предлагаемой классификации.

СИСТЕМА. МОДЕЛЬ. КЛАССИФИКАЦИЯ. ПРИЗНАКИ КЛАССИФИКАЦИИ. МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ. КЛАССИФИКАЦИЯ ПО МЕТОДАМ МОДЕЛИРОВАНИЯ.

Brief analysis of existing approaches on models' classification provided. The concept for classification of models is offered. Characteristics of orientations and methods of modelling, included in the proposed structure of classification, are submitted.

SYSTEM. MODEL. CLASSIFICATION. CRITERIA OF CLASSIFICATION. MATHEMATICAL MODELLING. CLASSIFICATION ACCORDING TO METHODS OF MODELLING.

---

Системы, процессы, устройства, явления могут быть представлены различными видами моделей. Необходимость в моделировании чаще всего обусловлена задачей прогнозирования поведения сложных систем, специфика которых не позволяет проводить их натурные исследования или испытания. Выбор методов моделирования является важной и сложной задачей. От грамотного выбора зависят сложность модели, ее адекватность, трудозатраты, связанные с ее реализацией. В данной статье мы ставим перед собой задачу упорядочения имеющейся в монографиях и учебных изданиях информации об укрупненных классах моделей, применяемых в теории и практике системного анализа, для упрощения процедуры выбора адекватной модели, максимально соответствующей конкретной задаче проектирования сложного техни-

ческого или социально-экономического комплекса.

Для выбора моделей разработаны их классификации. Классификацией в общем понимании называется разбиение объектов на классы по наиболее существенным признакам. Цель классификации:

ограничить выбор подходов к отображению системы;

сопоставить выделенным классам методы системного анализа;

сформировать общие рекомендации по выбору методов для соответствующего класса систем.

Обратимся к терминологии. Согласно определению Карла Людвиг фон Берталанфи (1901–1972), основоположника теории систем, «система — это совокупность элементов, находящихся в определенных отношениях друг с другом и со средой». При этом

система, в принципе, может быть одновременно охарактеризована несколькими признаками, т. е. ей может быть найдено место одновременно в разных классификациях, каждая из которых может оказаться полезной при выборе методов моделирования.

В свою очередь, модель также представляет собой некоторую систему, исследование которой служит средством для получения информации об объекте исследования (другой системе). В своей книге «Основы системного анализа» Ф.П. Тарасенко пишет: «Будучи системой, модель обладает всеми ... общесистемными свойствами. Отличительная особенность моделей от других систем — в их предназначенности отображать моделируемый оригинал, заменять его в определенном отношении, т. е. содержать и представлять информацию об оригинале. Модель является, следовательно, *формой существования знаний* об объекте». Таким образом, *модель* (фр. *module*, от лат. *modulus* — «мера, аналог, образец») — это описание системы (математическое, вербальное и т. д.), отображающее определенную группу компонентов системы, их свойств и отношений, существенных для управления системой.

Выбор типа модели зависит от целей моделирования, а также от объема и характера исходной информации о рассматриваемом объекте и возможностей исследователя. Для одного объекта (или исследуемой системы) могут быть построены несколько моделей, если решаются разные задачи, приводящие к разным целям моделирования. Выбор конкретного вида модели и ее дальнейшая реализация зависят от квалификации исследователя, но также связаны с его личностными предпочтениями, отражающими сферу научных интересов, глубину понимания и опыт моделирования. Для одного и того же объекта разные исследователи могут построить разные модели, даже если поставленная задача моделирования однотипна.

С другой стороны, для разных объектов могут быть построены модели одинакового вида, вне зависимости от личности исследователя или его предпочтений, которые он хочет воплотить в ходе моделирования. От-

сюда вытекает двойственный характер цели моделирования: цель объективна, поскольку определяется задачей исследования, но субъективна, т. к. ее корректирует «под себя» конкретный исследователь в зависимости от имеющего опыта и индивидуальной мотивации.

Связь между объектом и его моделью носит опосредованный характер, поскольку между ними располагается системное описание объекта. Моделирование всегда предполагает принятие допущений той или иной степени важности, поэтому уже на этапе формирования системного описания могут быть «отсечены» важные несущественные свойства объекта. Состав этих свойств определяется субъективной стороной процесса моделирования, например, они могут находиться вне интересов конкретного исследователя. Однако логично считать, что если системное описание объекта позволяет однозначно восстановить объект, то модель, построенная на базе такого системного описания, можно назвать системной моделью объекта.

В любой задаче моделирования существует некоторый порог сложности, переступить который можно только ценой отказа от требований точности решений. Например, при компьютерной реализации решения многоэкстремальных задач точные методы их решения могут оказаться экономически невыгодными по сравнению с более простыми приближенными методами. Эффект, полученный от уточнения решения, не окупит дополнительных затрат на его отыскание. К этому следует добавить стоимость получения результатов моделирования, включающую стоимость разработки модели и цену получения требуемой информации. Таким образом, необходимо определить оптимум между выполнением требований к адекватности модели, ее точности, целесообразности, обеспечением качества построенной модели, с одной стороны, и затратами, необходимыми для достижения этих требований, с другой.

Оценка и обеспечение качества модели может быть выполнена на уже построенной модели. Поэтому *качество модели* должно быть «заложено» в нее еще на этапе выбора



типа модели и базироваться на опыте исследователя, о котором сказано выше. Для структурированной аккумуляции такого опыта исследователь должен быть подготовлен к оценке параметров (показателей) качества, которыми он будет руководствоваться в будущем.

Важнейшим показателем качества модели, определяющим недопустимость «отсечения» некоторых свойств объекта, которые могут попасть в разряд несущественных, мы считаем *системную информативность модели*. В этом случае модель можно символически отобразить как носитель информации об оригинале (объекте):

$$J_{\text{def}} \equiv \langle Q, N, L, IS \rangle,$$

где  $Q$  – оригинал (объект, источник информации);  $N$  – субъект («наблюдатель» по Эшби), т. е. лицо, которому потребовалась информация об оригинале для достижения целей (исследования, принятия решения и др.);  $L$  – язык для исследования гносеологических аспектов отношения «модель – оригинал»;  $IS$  – инфраструктура, обеспечивающая моделирование, включающая технологии и условия моделирования [3].

Совокупность технологий в инфраструктуре представлена как

$$\text{TECH} = \{\text{METH}, \text{MEANS}, \text{ALG}, \dots\},$$

где METH – методы; MEANS – средства; ALG – алгоритмы, выбранные для реализации модели.

Условия существования модели определяются совокупностью факторов, влияющих на ее создание и функционирование, которые разделяются на внешние и внутренние:

$$\text{COND} = \{\varphi_{\text{ex}}, \varphi_{\text{in}}\}.$$

Приведенное выше символическое представление модели в формализованном виде не учитывает функционирования системы и ее взаимоотношения со средой. Чтобы учесть это взаимодействие, модель можно представить в виде множества величин, описывающих процесс функционирования реальной системы и образующих в общем случае следующие подмножества:

совокупность  $x_i \in X$ ,  $i = 1 \dots n_x$  – входных

воздействий  $X$  на  $S$ ;

совокупность воздействий внешней среды  $v_l \in V$ ,  $l = 1 \dots n_v$ ;

совокупность внутренних (собственных) параметров системы  $h_k \in H$ ,  $k = 1 \dots n_h$ ;

совокупность выходных характеристик системы  $y_j \in Y$ ,  $j = 1 \dots n_y$ .

Рассмотрим методы классификации моделей.

Исходно все модели делились на две группы: физические и абстрактные. *Физические* модели эквивалентны или подобны оригиналу, но имеют другую физическую природу. Их делят на натуральные, квазинатуральные, масштабные, аналоговые. *Физические* модели широко применяются при проектировании технических устройств и систем.

Абстрактное представление системы заключается в создании умозрительных образов или моделей материальных систем, которые подразделяются на описательные (логические) и символические (математические). *Логические* модели есть результат дедуктивного или индуктивного представления материальных систем. Это системы понятий и определений (совокупность представлений) о структуре, об основных закономерностях состояний и о динамике материальных систем.

*Символические* модели представляют собой формализацию логических и включают:

статические математические модели – описание материальных систем средствами математического аппарата (уравнениями состояния);

динамические математические модели, которые можно рассматривать как математическую формализацию процессов материальных (или абстрактных) систем.

К средствам абстрактного описания систем относятся также языки химических формул, схем, чертежей, карт, диаграмм и т. п.

В литературе описаны и другие классификации. Подходы к классификации определяются выбираемыми признаками. При этом признаки классификации также интерпретируют неодинаково. Так, С.В. Микони и В.А. Ходаковский [13] предлагают

следующий вариант классификации:

1) по степени абстрагирования (содержательные, формальные, формализованные);

2) по детальности отражения свойств объекта (концептуальные, конструктивные);

3) по форме представления (знаковые, графические, табличные или матричные);

4) по реализации (физические, компьютерные);

5) по степени определенности отношений между переменными (детерминированные, недетерминированные);

6) по структуре областей определения и значений функций (непрерывные, дискретные).

Другие авторы предлагают разнообразные признаки классификации по характеру моделей и моделируемых объектов; по сферам приложения (в технике, физических науках, кибернетике и т. д.); по способу отображения (эвристические, натурные и математические); по целям исследования; по особенностям представления (простые и сложные, однородные и неоднородные, открытые и закрытые, статические и динамические, вероятностные и детерминированные и т. п.); по методам моделирования и др. С учетом такого многообразия признаков можно разработать многоаспектную классификацию, однако создать универсальную классификацию – задача практически невыполнимая. Например, Ф.П. Тарасенко [17] считает, что все огромное разнообразие моделей достаточно разделить всего на три типа: модель состава (перечень существенных частей системы), модель структуры системы (перечень существенных связей между частями системы) и модель черного ящика (перечень существенных связей системы с окружающей средой). Любые модели являются либо одной из них, либо их целевой комбинацией.

Широчайшее распространение получили математические модели, которые представляют собой формализованное отображение системы с помощью абстрактного языка, с помощью математических соотношений, отражающих структуру или процесс функционирования системы. Для составления математических моделей используют любые

математические средства: алгебраическое, дифференциальное, интегральное исчисления, теорию множеств, теорию алгоритмов и т. д.

Математические модели первоначально классифицировали следующим образом: аналитические и численные, а аналитические модели, в свою очередь, на детерминированные и вероятностные.

К математическим моделям относят также программные комплексы (пакеты программ), а также формально-логические модели, созданные на формальном языке. Если изменения в системе носят случайный характер и данных о состоянии системы недостаточно для предсказания ее поведения в последующие моменты времени при случайных воздействиях, то систему не удастся отобразить детерминированными моделями; в таких случаях применяются стохастические и другие типы моделей.

Одним из важнейших признаков для классификации моделей систем является классификация по методам моделирования, призванная обеспечить оптимальный выбор типа модели, совпадающего со спецификой исследуемого объекта (системы), условиями поведения системы, задачами моделирования. Предлагаемая классификация моделей по методам моделирования показана на рисунке.

Среди известных методов особо выделяются методы математического моделирования. Однако формализованное отображение системы с помощью абстрактного языка не сводится только к детерминированным и вероятностным моделям. К математическим можно отнести теоретико-множественные модели, модели математической логики, теории графов. Если в основу классификации положить не математические методы, а методы формализованного представления систем, то к этому классу относятся также модели математической логики (обычно кратко называемые лингвистическими) и семиотические модели.

В поисках термина для выделения класса моделей, объектов и процессов, которые не могут быть сразу отображены аналитическими и другими формализованными моделями, предлагалось классифицировать



модели по степени абстрагирования, выделяя наряду с формальными и формализованными моделями *словесные* [14]. Однако словесное, точнее – вербальное (в терминах предметной области) отображение задачи моделирования обычно считают постановкой задачи, а не моделью, а в качестве этапа процесса моделирования рассматривают процесс перевода *вербального* описания в *формальное* (или формализованное).

Кроме формальных и формализованных моделей, основанных на математических методах и методах формализованного представления систем (МФПС), можно выделить особый класс моделей, основанных на методах активизации интуиции специалистов (МАИС). В этот класс моделей можно отнести модели выработки коллективных решений (например, в форме сценариев, которые можно считать словесными или вербальными моделями), модели структуризации, организации сложных экспертиз, морфологические модели.

Самостоятельными разделами теории моделирования можно также считать специальные методы или подходы, сочетающие средства МАИС и МФПС – *ситуационное, структурно-лингвистическое, когнитивное, логико-лингвистическое, лингвокомбинаторное* моделирование, а также моделирование, основанное на *информационном подходе к анализу систем*, моделирование путем организации процесса *постепенной формализации с переключением* МАИС и МФПС, обеспечивающее переход от менее формализованной модели к более формализованной.

На рисунке приведены укрупненные разделы классификации моделей по признаку «методы моделирования». Здесь математические модели классифицированы по методам формализованного представления систем (МФПС). Автор классификации – Ф.Е. Темников [1] – выделил следующие обобщенные группы (классы) методов:

- *аналитические* (методы классической математики, включая интегро-дифференциальное исчисление, методы поиска экстремумов функций, вариационное исчисление и т. п.; методы математического программирования – линейного, нелинейного,

целочисленного, динамического, широко применяющиеся в теории оптимизации; первые работы по теории игр и т. п.);

- *статистические* (включающие и теоретические разделы математики – теорию вероятностей, математическую статистику, и направления прикладной математики, использующие стохастические представления – теорию массового обслуживания, методы статистических испытаний (основанные на методе Монте-Карло), методы выдвижения и проверки статистических гипотез А. Вальда и другие методы статистического моделирования);

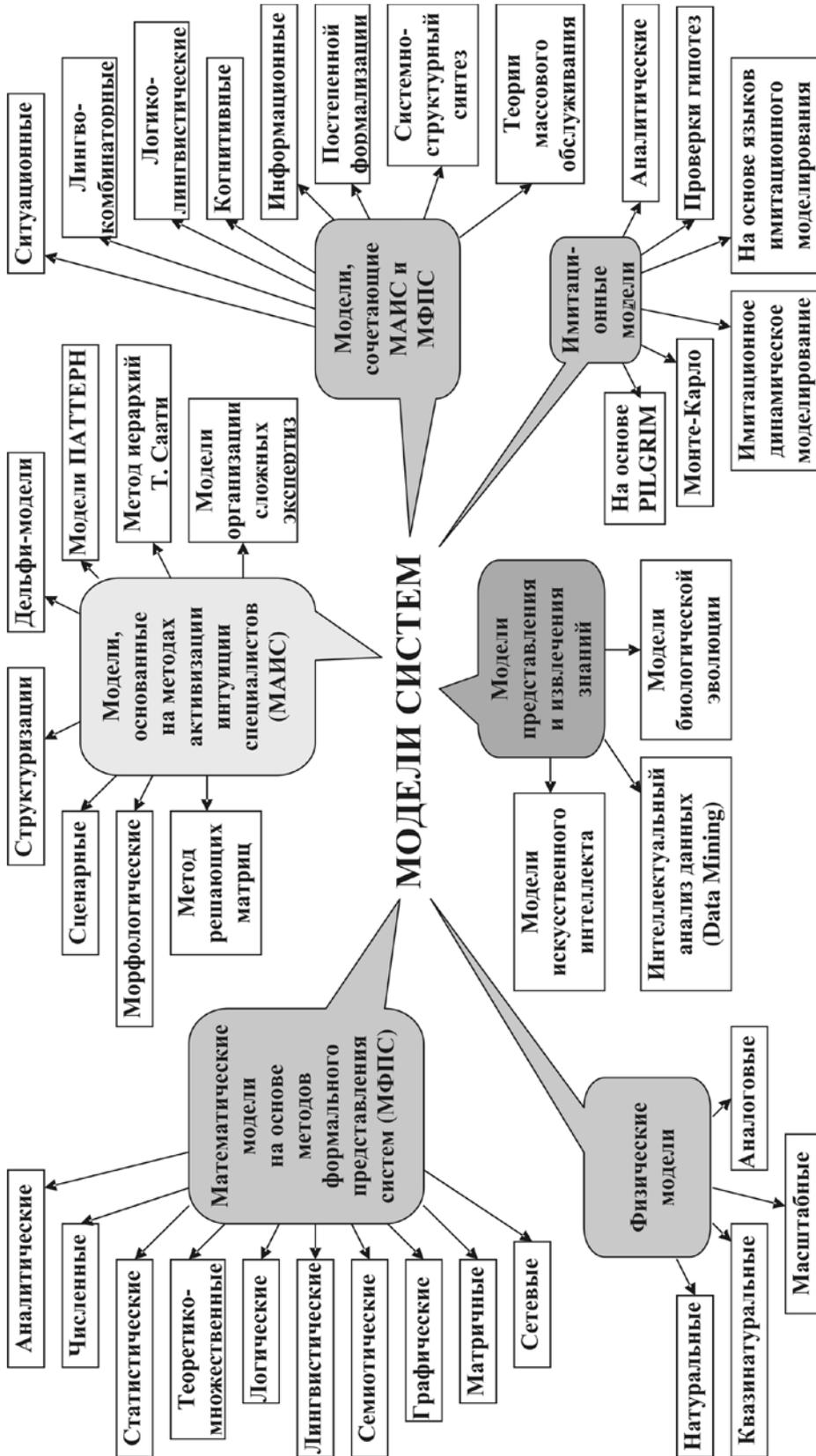
- *теоретико-множественные, логические, лингвистические, семиотические* представления (методы *дискретной математики*), составляющие теоретическую основу разработки языков моделирования, автоматизации проектирования, информационно-поисковых языков;

- *графические* (включающие теорию графов и разного рода графические представления информации типа диаграмм, гистограмм и других графиков).

В классификацию МАИС включены методы, которые активно исследовались в начальный период развития теории систем: «мозговая атака» (или коллективная генерация идей) и другие методы выработки коллективных решений; методы типа «сценариев», методы и методики структуризации, методы экспертных оценок, морфологического моделирования и т. д.

Разбиение методов на МАИС и МФПС соответствует основной идее системного анализа, состоящей в сочетании в моделях и методиках формальных и неформальных представлений, что помогает в разработке методик, выборе методов постепенной формализации отображения и анализа проблемной ситуации.

Предлагаемые названия групп методов более предпочтительны, чем используемые иногда термины – *качественные* и *количественные* методы, поскольку, например, методы, отнесенные к группе МАИС, могут использовать и формализованные представления. Так, при разработке *сценариев* могут применяться статистические данные, с которыми проводятся некоторые расчеты; с



Классификация моделей систем по признаку «методы моделирования»



формализацией связаны получение и обработка экспертных оценок, методы морфологического моделирования. Тем не менее согласно теореме К. Гёделя о неполноте [18], «в рамках любой формальной системы, сколь бы полной и непротиворечивой она не казалась, имеются положения (соотношения, высказывания), истинность или ложность которых нельзя доказать формальными средствами этой системы». Поэтому для преодоления неразрешимой проблемы нужно расширять формальную систему, опираясь на содержательный качественный анализ, например, используя показатель «системная информативность модели» (см. выше).

Результаты Гёделя были получены для арифметики, самого формального из направлений математики, и позволили предположить, что процесс логического, в т. ч. математического доказательства, не сводится к использованию только дедуктивного метода, что в нем всегда присутствуют неформальные элементы мышления. В дальнейшем исследования этой проблемы математиками и логиками показали, что «доказательства во все не обладают абсолютной, не зависящей от времени строгостью, и являются только культурно опосредованными средствами убеждения» [6].

Иными словами, *строгого разделения на формальные и неформальные методы не существует*. Можно говорить только о большей или меньшей степени формализованности или, напротив, большей или меньшей опоре на интуицию, здравый смысл.

Разумеется, на рисунке приведены лишь укрупненные группы-направления. Эти направления непрерывно развиваются, и в их рамках появляются методы с расширенными возможностями по сравнению с исходными.

Новые методы часто возникают на основе сочетания ранее существовавших. Так, методы, иногда называемые *комплексированными* (комбинаторика, топология), начинали развиваться параллельно в рамках линейной алгебры, теории множеств, теории графов, а затем оформились в самостоятельные направления. На пересечении аналитических и теоретико-множественных представлений возникла и развивается ал-

гебра групп; параллельно в рамках алгебры групп и теории множеств начала развиваться *комбинаторика*; теоретико-множественные и графические представления стали основой возникновения *топологии*; статистические и теоретико-множественные методы инициировали возникновение теории нечетких множеств Л. Заде, которая, в свою очередь, явилась началом развития нового направления — *нечетких формализаций*, и т. д.

Практически невозможно создать единую классификацию, включающую все разделы современной математики. В то же время приведенные направления помогают понять особенности конкретных методов, использующие средства того или иного направления или их сочетания, помогают выбирать методы для конкретных приложений.

В качестве самостоятельной на рисунке представлена группа *специальных методов* системного анализа, которые точнее было бы квалифицировать как подходы, базирующиеся на сочетании средств МАИС и МФПС. В то же время эти подходы в большинстве доведены до реализации в виде формальных алгоритмов, что позволяет их квалифицировать как методы.

Наибольшее распространение получили следующие специальные методы моделирования систем:

- *Имитационное динамическое моделирование (System Dynamics Simulation Modelling)*. Предложено Дж. Форрестером (США) в 1950-х гг. [19].

- *Ситуационное моделирование*. Идея предложена Д.А. Поспеловым [15] и реализована Ю.И. Клыковым и Л.С. Загадской (Болотовой) (см. ссылки в [16]).

- *Лингво-комбинаторное моделирование*. Предложено для моделирования плохо формализованных систем М.Б. Игнатьевым [7].

- *Логико-лингвистическое моделирование*. Является развитием *структурно-лингвистического* моделирования, широко распространенного в 1970-е гг. в инженерной практике и основанного на использовании для реализации идей комбинаторики структурных представлений разного рода, с одной стороны, и средств математической

лингвистики, с другой. В 2000-е гг. логико-лингвистическое моделирование применительно к анализу и развитию адаптивного управления социально-экономическими системами развивает Б.Л. Кукор [11], используя в качестве языковых (лингвистических) средств и другие методы дискретной математики (теоретико-множественные представления, средства математической логики и семиотики).

- *Теория информационного поля и информационных цепей (информационный подход к моделированию и анализу систем)*. Концепция информационного поля предложена А.А. Денисовым [5] и основана на использовании для активизации интуиции ЛПР законов диалектики, а в качестве средства формализованного отображения – применение аппарата математической теории поля и теории цепей.

- *Подход, базирующийся на идее постепенной формализации задач (проблемных ситуаций) с неопределенностью путем очередного использования средств МАИС и МФПС* [2].

- *Системно-структурный синтез*. Системно-структурные методы моделирования разрабатывались с самого начала развития теории систем на основе иерархических и сетевых структур как средства исследования объектов и процессов с неопределенностью, когда не могут быть сразу получены математические модели. Теория системно-структурного синтеза, основанная на многоуровневой модели, постепенно сужающей область допустимых решений, предложена Ю.И. Лыпарем (см. [16]).

- *Когнитивный подход* (от лат. *cognitio* – знание, познание). Базируется на идеях когнитивной психологии. Истоки когнитивного подхода прослеживаются, начиная с работ древнегреческих мыслителей («учение об универсалиях» Платона). Оформление когнитивного подхода как особой дисциплины связывают с именем У. Найссера [14]. В нашей стране это направление развивается в ИПУ РАН (В.И. Максимов [10], В.В. Кульба [12], Н.А. Абрамова [20] и др.) и в Южном федеральном университете школой Г.В. Гореловой [4] применительно к

системам управления муниципальными образованиями и социально-экономическими объектами.

Из математических моделей в ходе развития выделился самостоятельный класс имитационных моделей.

*Имитационная модель* – описание системы и внешних воздействий, алгоритмов функционирования системы или правил изменения состояния системы под влиянием внешних и внутренних возмущений в ситуациях, когда математические методы аналитического или численного решения неприменимы, но алгоритмы и правила позволяют имитировать процесс функционирования системы и вычислять ее характеристики.

В дальнейшем имитационные модели стали создавать для более широкого класса объектов и процессов, чем аналитические и численные. На базе статистических представлений разработаны:

- направление имитационного моделирования с целью определения функции распределения случайной величины;

- моделирование, основанное на методе Монте-Карло;

- имитационное моделирование в теории массового обслуживания.

Поскольку для реализации имитационных моделей применяются вычислительные системы, в качестве средств формализованного описания имитационной модели используют универсальные и специальные языки – языки имитационного моделирования (ЯИМ) и общего назначения (ЯОН).

Для исследования, разработки прогнозов и для решения других задач принятия решений разработаны специальные методы имитационного моделирования: имитационное динамическое моделирование Дж. Форрестера [19], имитационное моделирование с использованием автоматизированной системы Pilgrim, развиваемое А.А. Емельяновым [10].

В компьютерном моделировании выделяется самостоятельный класс *моделей представления и извлечения знаний*, в составе которых выделяют модели, развиваемые на базе теории искусственного интеллекта, мо-



дели, построенные на принципах, заимствованных у природы, модели интеллектуального анализа данных (ИАД) — *Data Mining*.

Классификация, подобная рассмотренной, помогает осознанно выбирать методы моделирования и должна входить в состав методического обеспечения работ по проектированию сложных технических комплексов, по управлению предприятиями и организациями. Она может развиваться, дополняться новыми методами, т. е. акку-

мулировать опыт, накапливаемый в процессе проектирования и управления.

Любая классификация условна и может быть подвергнута критике. Она — лишь средство, помогающее ориентироваться в огромном числе разнообразных методов и моделей. Однако, несмотря на условность классификации, она необходима для учета конкретных условий, особенностей моделируемых систем (процессов принятия решений) и предпочтений ЛПР, которым предлагается выбирать классификацию.

### СПИСОК ЛИТЕРАТУРЫ

1. **Волкова, В.Н.** Методы формализованного представления систем: Учеб. пособие [Текст] / В.Н. Волкова, А.А. Денисов, Ф.Е. Темников. — СПб.: Изд-во СПбГТУ, 1993. — 107 с.
2. **Волкова, В.Н.** Постепенная формализация моделей принятия решений [Текст] / В.Н. Волкова. — СПб.: Изд-во СПбГПУ, 2006. — 120 с.
3. **Волкова, В.Н.** Моделирование систем: Учеб. пособие [Текст] / В.Н. Волкова [и др.]; под ред. В.Н. Волковой, В.Н. Козлова. — СПб.: Изд-во Политехн. ун-та, 2012. — 440 с.
4. **Горелова, Г.В.** Исследование слабоструктурированных проблем социально-экономических систем: когнитивный подход [Текст] / Г.В. Горелова, Е.Н. Захарова, С.А. Радченко. — Ростов-на-Дону: Изд-во РГУ, 2006. — 330 с.
5. **Денисов, А.А.** Современные проблемы системного анализа: Учебник [Текст] / А.А. Денисов. — 3-е изд. — СПб.: Изд-во Политехн. ун-та, 2008. — 304 с.
6. **Ивин, А.А.** Строгий мир логики [Текст] / А.А. Ивин. — М.: Педагогика, 1988. — С. 125.
7. **Игнатьев, М.Б.** Кибернетическая картина мира. Теория сложных систем [Текст] / М.Б. Игнатьев. — СПб., 2011, 468 с.; Ignatyev M.B. Linguistic-Combinatorial Simulation of Complex Systems // J. of Mathematics and System Science. — USA, Jan. 2012. — Vol. 2. — № 1. — P. 58–66.
8. Когнитивный анализ и управление развитием ситуаций [Текст] / Тр. I Междунар. конф. CASC2001. — М.: ИПУ РАН, 2001. — № 1. — С. 4–18.
9. **Козлов, В.Н.** Вычислительная математика и теория управления [Текст] / В.Н. Козлов, В.Е. Куприянов, В.Н. Шашихин. — СПб.: Изд-во СПбГТУ, 1996. — 198 с.
10. Компьютерная имитация экономических процессов [Текст] / Под ред. А.А. Емельянова. — М.: Маркет ДС, 2010. — 464 с.
11. **Кукор, Б.Л.** Семиотика системного анализа и семантическая система логико-лингвистической модели предметной области [Текст] / Б.Л. Кукор // Сб. науч. тр. XIII Междунар. науч.-практ. конф. Системный анализ в проектировании и управлении: — СПб.: Изд-во Политехн. ун-та, 2009. — Ч. 1. — С. 164–169.
12. **Кульба, В.В.** Сценарный анализ динамики поведения социально-экономических систем / В.В. Кульба [и др.]. — М.: ИПУ РАН, 2002. — 122 с.
13. **Микони, С.В.** Основы системного анализа: Учеб. пособие / С.В. Микони, В.А. Ходаковский. — СПб.: СГУПС, 2011. — С. 47–48.
14. **Найссер, У.** Познание и реальность [Текст] / У. Найссер. — М., 1981.
15. **Поспелов, Д.А.** Ситуационное управление: Теория и практика [Текст] / Д.А. Поспелов. — М.: Наука, 1986. — 284 с.
16. Системный анализ и принятие решений: Словарь-справочник [Текст] / Под ред. В.Н. Волковой, В.Н. Козлова. — М.: Высш. школа, 2004. — 616 с.
17. **Тарасенко, Ф.П.** Прикладной системный анализ: Наука и искусство решения проблем: Учебник [Текст] / Ф.П. Тарасенко. — Томск.: Изд-во Томского ун-та. 2004. — 186 с.
18. **Успенский, В.А.** Теорема Гёделя о неполноте [Текст] / В.А. Успенский. — М.: Наука, 1982. — 112 с.
19. **Форрестер, Дж.** Мировая динамика [Текст] / Дж. Форрестер. — М.: Наука, 1978. — 167 с.
20. Человеческий фактор в управлении [Текст] / Под ред. Н.А. Абрамовой [и др.]. — М.: КомКнига, 2006. — 496 с.

## REFERENCES

1. **Volkova V.N., Denisov A.A., Temnikov F.E.** Metody formalizovannogo predstavleniia sistem: Ucheb. posobie. – St. Petersburg: Izd-vo SPbGTU, 1993. – 107 s. (rus)
2. **Volkova V.N.** Postepennaia formalizatsiia modelei priiniatiia reshenii. – St. Petersburg: Izd-vo SPbGPU, 2006. – 120 s. (rus)
3. **Volkova V.N. i dr.** Modelirovanie sistem: Ucheb. posobie; Pod red. V.N. Volkovoi, V.N. Kozlova. – St. Petersburg: Izd-vo Politekhn. un-ta, 2012. – 440 s. (rus)
4. **Gorelova G.V., Zakharova E.N., Radchenko S.A.** Issledovanie slabostrukturirovannykh problem sotsial'no-ekonomicheskikh sistem: kognitivnyi podkhod. – Rostov-na-Donu: Izd-vo RGU, 2006. – 330 s. (rus)
5. **Denisov A.A.** Sovremennye problemy sistemnogo analiza: Ucheb. posobie; 3-e izd. – St. Petersburg: Izd-vo Politekhn. un-ta, 2008. – 304 s. (rus)
6. **Ivin A.A.** Strogii mir logiki. – Moscow: Pedagogika, 1988. – 125 s. (rus)
7. **Ignat'ev M.B.** Kiberneticheskaia kartina mira. Teoriia slozhnykh sistem. – St. Petersburg, 2011. – 468 s. (rus); Ignatyev M.B. Lingual-Combinatorial Simulation of Complex Systems / J. of Mathematics and System Science. – USA, Jan. 2012. – Vol. 2. – № 1. – P. 58–66.
8. Kognitivnyi analiz i upravlenie razvitiem situatsii (CASC2001): tr. I Mezhdunar. konf. – Moscow: IPU RAN, 2001. – № 1. – С. 4–18. (rus)
9. **Kozlov V.N., Kupriianov V.E., Shashikhin V.N.** Vychislitel'naia matematika i teoriia upravleniia. – St. Petersburg: Izd-vo SPbGTU, 1996. – 198 s. (rus)
10. Komp'yuternaia imitatsiia ekonomicheskikh protsessov; Pod red. A.A. Emel'ianova. – Moscow: Market DS, 2010. – 464 s. (rus)
11. **Kukor B.L.** Semiotika sistemnogo analiza i semanticheskaia sistema logiko-lingvisticheskoi modeli predmetnoi oblasti / Sb. nauch. trudov XIII Mezhdunar. nauch.-prakt. konf. Sistemnyi analiz v proektirovanii i upravlenii. – St. Petersburg: Izd-vo Politekhn. un-ta, 2009. – Ch. 1. – S. 164–169. (rus)
12. **Kul'ba V.V. i dr.** Stsenarnyi analiz dinamiki povedeniia sotsial'no-ekonomicheskikh sistem. – Moscow: IPU RAN, 2002. – 122 s. (rus)
13. **Mikoni S.V., Khodakovskii V.A.** Osnovy sistemnogo analiza: ucheb. posobie. – St. Petersburg: SGUPS, 2011. – S. 47–48. (rus)
14. **Naisser U.** Poznanie i real'nost'. – Moscow: 1981. (rus); Neisser U. Cognitive psychology. – N.Y., 1967.
15. **Pospelov D.A.** Situatsionnoe upravlenie: teoriia i praktika. – Moscow: Nauka, 1986. – 284 s. (rus)
16. Sistemnyi analiz i priiniatie reshenii: Slovar'-spravochnik; Pod red. V.N. Volkovoi, V.N. Kozlova. – Moscow: Vyssh. shkola, 2004. – 616 s. (rus)
17. **Tarassenko F.P.** Prikladnoi sistemnyi analiz: Nauka i iskusstvo resheniia problem: Ucheb. posobie. – Tomsk: Izd-vo Tomskogo un-ta, 2004. – 186 s. (rus)
18. **Uspenskii V.A.** Teorema Gedelia o nepolnote. – Moscow: Nauka, 1982. – 112 s. (rus)
19. **Forrester Dzh.** Mirovaia dinamika. – Moscow: Nauka, 1978. – 167 s. (rus)
20. Chelovecheskii faktor v upravlenii; Pod red. N.A. Abramovoi i dr. – Moscow: Komkniga, 2006. – 496 s. (rus)

---

**ВОЛКОВА Виолетта Николаевна** – профессор кафедры системного анализа и управления Санкт-Петербургского государственного политехнического университета, доктор экономических наук, заслуженный работник высшей школы РФ.

195251, Россия, Санкт-Петербург, ул. Политехническая, д. 29.  
Тел. 8 (812) 294-42-14; e-mail: violetta\_volkova@list.ru

**VOLKOVA, Violetta N.** St. Petersburg State Polytechnical University.  
195251, Politekhnikeskaya Str. 29, St. Petersburg, Russia.  
E-mail: violetta\_volkova@list.ru

**ЧЕРНЕНЬКАЯ Людмила Васильевна** – директор департамента менеджмента качества Санкт-Петербургского государственного политехнического университета, профессор кафедры системного анализа и управления, доктор технических наук.

195251, Россия, Санкт-Петербург, ул. Политехническая, д. 29.  
Тел. 8 (812) 552-76-40; e-mail: Ludmila@qmd.spbstu.ru



**CHERNENKAYA, Liudmila V.** *St. Petersburg State Polytechnical University.*  
195251, Politekhnikeskaya Str. 29, St. Petersburg, Russia.  
E-mail: Ludmila@qmd.spbstu.ru

**МАГЕР Владимир Евстафьевич** – *начальник отдела департамента менеджмента качества Санкт-Петербургского государственного политехнического университета, доцент кафедры системного анализа и управления, кандидат технических наук.*

195251, Россия, Санкт-Петербург, ул. Политехническая, д. 29.  
Тел. 8 (812) 552-76-40; e-mail: mv@qmd.spbstu.ru

**MAGER, Vladimir E.** *St. Petersburg State Polytechnical University.*  
195251, Politekhnikeskaya Str. 29, St. Petersburg, Russia.  
E-mail: : mv@qmd.spbstu.ru

УДК 65.012.122

*Е.А. Ланцев, М.Г. Доррер*

## **ПОЛУЧЕНИЕ АГЕНТНОЙ ИМИТАЦИОННОЙ МОДЕЛИ ИЗ ДИСКРЕТНО-СОБЫТИЙНОГО ОПИСАНИЯ БИЗНЕС-ПРОЦЕССА**

*Y.A. Lantsev, M.G. Dorrer*

### **CREATING AGENT-BASED MODEL FROM THE BUSINESS PROCESS DISCRETE-EVENT MODEL**

Предложена методика трансляции модели из нотации ARIS eEPC в агентную имитационную модель AnyLogic. Поставлен эксперимент по трансляции модели описанной нотацией eEPC в агентную имитационную модель AnyLogic. Результаты работы могут использоваться аналитиками для сокращения времени проектирования в процессе реализации имитационных моделей, а также упрощения процесса последующей эксплуатации имитационных моделей.

**АГЕНТНОЕ ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ. eEPC. ARIS. AnyLogic. БИЗНЕС-ПРОЦЕСС.**

In this paper the technique for translating model in eEPC notation into AnyLogic agent-based simulation model are proposed and developed. Performed an experiment of translating model in eEPC notation into Anylogic agent-based simulation model based on the developed algorithm. The results can be used by analysts to reduce simulation models designing and implementing time, and simplifying the exploitation process.

**AGENT-BASED SIMULATION. EEPC. ARIS. ANYLOGIC. BUSINESS PROCESS.**

В современной практике организационного управления широкое распространение получили графические модели бизнес-процессов. Этот класс методов организационной науки изначально был предназначен для решения проблем избыточной сложности описания и исследования систем. Однако исследование графической модели, даже выполненной в соответствии с правилами структурного подхода, представляет значительную сложность. Вместе с тем существует давно известный и обладающий мощными аналитическими возможностями при исследовании дискретных систем аппарат имитационного моделирования бизнес-систем. Но само по себе имитационное моделирование не решает оптимизационных задач, а скорее представляет собой технику оценки значений функциональных характеристик моделируемой системы, позволяя выявлять проблемные места в системе [1].

В настоящее время для моделирования и анализа бизнес-процессов используются различные подходы и средства. В пер-

вую очередь следует отметить методологии описания бизнес-процессов IDEF, ARIS, BPMN. При этом в программных продуктах, предназначенных для описания бизнес-процессов, нередко присутствуют модули, обеспечивающие проведение имитационного эксперимента на построенных событийных моделях (здесь следует упомянуть об интеграции BPWin и Arena, ARIS Simulation).

Существующие разработки в области трансляции между формальным описанием бизнес-процессов и его имитационной моделью в основном нацелены на получение в конечном итоге дискретно-событийной имитационной модели. Агентные имитационные модели в этой области менее распространены, поскольку агентное моделирование как парадигма появилось относительно недавно. Но следует отметить растущий в последнее время интерес к агентному моделированию.

Обычно при трансляции в качестве исходной нотации формального описания

бизнес-процессов используется уже существующая нотация (IDEF0, IDEF3, UML, BPMN, EPC). В качестве целевой имитационной модели может использоваться как модель в формате существующей имитационной системы (Avena, GPSS, CPN Tools), так и формат для системы имитационного моделирования собственной разработки.

В основном (за счет исторически раннего своего появления и вследствие большей распространенности) в качестве исходной формальной модели бизнес-процессов используется методология IDEF0 и IDEF3 [2–4], менее распространены подходы, основанные на нотациях UML [5] и BPMN [6]. В качестве имитационной модели получили широкое распространение сети Петри [7–9] и GPSS. Все эти подходы в большинстве своем ориентированы на получение в конечном результате дискретно-событийной имитационной модели. Агентные имитационные модели в этой области используются мало, в качестве имитационной системы применяются собственные разработки.

Агент – это некоторая сущность, которая обладает активностью, автономным поведением, может принимать решения в соответствии с некоторым набором правил, может взаимодействовать с окружением и другими агентами, а также может изменяться (эволюционировать) [10].

Агентное имитационное моделирование – направление в имитационном моделировании, используемое для исследования децентрализованных систем, динамика функционирования которых определяется не глобальными правилами и законами (как в других парадигмах моделирования), а наоборот, эти глобальные законы и правила являются результатом индивидуальной активности членов группы. Цель агентных моделей – получить представление об этих глобальных правилах, общем поведении системы, исходя из предположений об индивидуальном, частном поведении ее отдельных активных объектов и взаимодействии этих объектов в системе [10].

Цель данной статьи – предложить методику получения агентной имитационной

модели AnyLogic, моделирующей микроэкономическую систему на основе применяемых для бизнес-моделирования графических методологий, на примере нотации ARIS eEPC.

### Методы и материалы исследования

Нотация ARIS eEPC (extended Event Driven Process chain) предназначена для детального описания бизнес-процесса и отражает логику его выполнения [11].

Модель в нотации eEPC определим как [12]  $G = \{X, V\}$ , где  $X$  – непустое множество объектов модели (вершин графа);  $V$  – непустое множество дуг  $V_n$  и ребер  $V_r$ , таких, что  $V = V_n \cup V_r$ .

Объекты модели (вершины графа):  $X = \{S, F, D, I, P, R\}$ , где  $S$  – множество объектов типа «событие»;  $F$  – множество объектов типа «функция»;  $D$  – множество объектов типа «документ» и «информационный носитель»;  $I$  – множество объектов типа «должность», «организационная единица»;  $P$  – множество объектов типа «продукт/услуга»;  $R$  – множество объектов типа «правило» исключающего ИЛИ (XOR), ИЛИ (OR), И (AND), задающих логику ветвления/слияния бизнес-процесса.

Множество правил ветвления  $R$  состоит из элементов множества  $R_{xor} \cup R_{or} \cup R_{and}$ , где  $R_{xor}$  – множество правил ветвления типа XOR;  $R_{or}$  – множество правил ветвления типа OR;  $R_{and}$  – множество правил ветвления типа AND.

Множество объектов типа  $I$  состоит из элементов  $I_{pos} \cup I_{org}$ , где  $I_{pos}$  – множество объектов типа «должность»;  $I_{org}$  – множество объектов типа «организационная единица».

Множество объектов типа  $D$  состоит из элементов  $D_{doc} \cup D_{inf}$ , где  $D_{doc}$  – множество объектов типа «документ»;  $D_{inf}$  – множество объектов типа «информационный объект».

Агентная модель в AnyLogic реализуется с помощью базового объекта – активного объекта. Активный объект имеет параметры, переменные, которые можно считать памятью агента, стейтчарты или диаграммы состояний выражают поведение: состояния

объекта и изменение состояний под воздействием событий и условий.

Определим агентную модель AnyLogic как  $ABM = \{E, A\}$ , где  $E$  – множество объектов, составляющих окружение агента;  $A$  – множество агентов.

Множество объектов  $E$ , составляющих окружение агента, определим как  $E = \{Tm, Var, Ds, Msg\}$ , где  $Tm$  – множество таймеров;  $Var$  – множество переменных;  $Ds$  – множество таблиц для сохранения статистических данных о работе модели и синхронизации;  $Msg$  – множество сообщений для взаимодействия агентов со средой и друг с другом.

Множество агентов  $A$  определим как  $A = \{Sch, Par, Var, Fn\}$ , где  $Sch$  – непустое множество стейтчартов (диаграмм состояний, соответствует стандарту UML);  $Par$  – множество параметров агента;  $Fn$  – множество функций агента (под функцией понимается специальный элемент AnyLogic для записи функций на языке Java).

Множество стейтчартов  $Sch$  определим как  $Sch = \{Bc, St, Trn\}$ , где  $Bc$  – непустое множество указателей начального состояния;  $St$  – непустое множество состояний;  $Trn$  – множество переходов.

Перевод графических элементов модели в нотации eEPC в агентную имитационную модель AnyLogic производится согласно табл. 1.

Рассмотрим подробнее методику преобразования модели в нотации eEPC в АИМ AnyLogic. В процессе изложения преобразований будем использовать соответствия для элементов моделей, приведенные в таблице.

В общем виде АИМ AnyLogic, основанная на событийных моделях eEPC, состоит из активных объектов (агентов), имеющих каждый свои правила поведения (задаются диаграммой состояний на языке UML) и взаимодействующих с внешней средой и друг с другом посредством пересылки сообщений (сообщение – специально реализованный программный класс на языке Java).

Итак, имеется модель в нотации eEPC. Необходимо на ее основе создать агентную

имитационную модель в AnyLogic. Вначале создаем структуру имитационной модели. Для этого необходимо следующее.

1. Создать классы агентов имитационной модели AnyLogic на основе организационных единиц модели eEPC. Поместить созданные классы на структурную диаграмму корневого объекта «Main», моделирующего внешнюю среду (объект «Main» создается по умолчанию вместе с созданием имитационной модели, в целом имя класса непринципиально).

2. Внутри класса каждого из агентов создать диаграмму состояний по следующим принципам:

- создать гиперсостояние. В гиперсостоянии создать только одно простое состояние – состояние ожидания, в котором агент находится в самом начале моделирования, а также переходит после каждого простого состояния;

- создать простые состояния – по одному на каждую функцию eEPC, которую выполняет организационная единица;

- создать переходы для каждого из простых состояний в гиперсостояние: от простого состояния к гиперсостоянию и от состояния ожидания, вложенного в гиперсостояние, к простому состоянию. Настроить, чтобы переход от простого состояния к гиперсостоянию срабатывал по таймауту. Настроить, чтобы переход от состояния ожидания к простому состоянию срабатывал по сигналу, и для каждого такого перехода указать уникальное (в пределах стейтчарта) название сигнала.

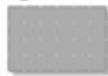
3. Создать Java-класс «сообщение» для описания взаимодействия агентов. В классе должно быть создано поле «fireEvent» строкового типа для указания сигнала перехода в диаграмме состояний.

4. На структурной диаграмме объекта «Main» создать статический таймер по количеству начальных событий в модели eEPC.

5. На структурной диаграмме объекта «Main» создать для каждого из классов агентов векторные переменные для хранения очередей сообщений (дисциплина очереди – FIFO).

Таблица 1

Описание соответствия объектов модели нотации eEPC и элементов агентной имитационной модели на языке AnyLogic

Объект модели eEPC	Графическое обозначение	Соответствующий элемент AnyLogic	Описание элемента AnyLogic
Функция $F \in X \in G$		Состояние $St \in Sch \in A \in ABM$ 	Простое состояние стейтчарта
Начальное событие $S_s \in S \in X \in G$		Таймер $Tm \in E \in ABM$ 	Наступление начальных событий в имитационной модели определяется таймером по заданному аналитиком расписанию
Событие $S_m, S_f \in S \in X \in G$		Переход $Trn \in Sch \in A \in ABM$ 	Переход из гиперсостояния стейтчарта в простое состояние
Материальные и информационные ресурсы			
Документ, информационный носитель $D \in X \in G$		Переменная $Var \in E \in ABM$ 	Переменные используются для моделирования изменяющихся характеристик, хранения результатов моделирования. Изменение количественных ресурсов происходит в состояниях стейтчарта, программируется на языке Java
Продукт/ Услуга $P \in X \in G$			
Исполнители – ресурсы, необходимые для выполнения бизнес-функции, но не изменяемые ею			
Должность $I_{pos} \in I \in X \in G$		Класс активного объекта $A \in ABM$ 	Поведение активного объекта описывается с помощью стейтчарта. В имитационной модели создаются классы активных объектов и далее в модели используется любое число экземпляров класса
Организационная единица $I_{org} \in I \in X \in G$			
Правила ветвления			
Правило ИЛИ $R_{or} \in R \in X \in G$		Переход $Trn \in Sch \in A \in ABM$ 	При переходе из простого состояния стейтчарта в гиперсостояние в методе перехода «Action» на языке Java программируется логика принятия агентом решения
Правило исключающего ИЛИ $R_{xor} \in R \in X \in G$			
Правило И $R_{and} \in R \in X \in G$			

6. На структурной диаграмме объекта «Main» создать переменные для хранения количества созданных документов — по одной на каждый объект документа в модели eEPC.

После описания структуры АИМ задать логику работы модели.

7. Срабатывание начальных событий в eEPC моделируется статическими таймерами, где прописывается расписание работы — циклическое срабатывание через определенные промежутки времени. При срабатывания таймера происходит создание и отсылка сообщений агентам, которые имеют состояния стейтчарта, соответствующие функциям модели eEPC, следующим за начальным событием. Сигнал для перехода в состояние стейтчарта записывается в поле «fireEvent» сообщения.

8. При получении агентом сообщения происходит переход в простое состояние согласно сигналу в сообщении, если имеется свободный экземпляр агента, иначе сообщение помещается в общую для всех агентов этого класса очередь. Анализ и обработка общей очереди производится каждым агентом после перехода в состояние ожидания (после выхода из простого состояния).

9. Вероятностная логика правил ветвления модели eEPC «ИЛИ» и «исключающего ИЛИ» (OR и XOR) реализуется в методе (под методом здесь подразумевается соответствующий термин объектно-ориентированного программирования) перехода «Action» на языке Java при смене состояний (простого состояния на гиперсостояние). Таким образом, моделируется принятие агентом решения — в какое следующее состояние он должен перейти далее или же он должен перейти в режим ожидания и отправить сообщение (с указанием сигнала для стейтчарта в поле «fireEvent») другому агенту для обработки.

10. Параллельное выполнение модели функций eEPC в имитационной модели также реализуется в методе перехода «Action» на языке Java, при смене состояний стейтчарта, рассылкой сообщений необходимым агентам. Синхронизация правила ветвления «И» (AND) происходит при помощи вну-

тренних таблиц (DataSet), в которые записывается факт произошедших событий.

Получившаяся в результате преобразований по предложенной методике АИМ является «заготовкой» для дальнейшего более детального моделирования: сосредоточение непосредственно на агентах, их поведении и коммуникациях.

В качестве примера применения методики рассмотрим создание агентной имитационной модели на основе модели бизнес-процесса «Формирование заказа на поставку товара», описанного в нотации eEPC (см. рис. 1). Это модель действующего бизнес-процесса на торговом предприятии.

Вначале строим модель в AnyLogic на основе предложенной методики. В представленной модели eEPC введено три организационные единицы: «Служба безопасности», «Отдел выписки» и «Диспетчер», соответственно они будут являться в имитационной модели агентами. На рис. 1 пунктирными стрелками показаны соответствия элементов модели eEPC и элементов агентной имитационной модели AnyLogic на примере агента «сотрудник службы безопасности» и агента «диспетчер».

Фрагменты созданной агентной имитационной модели AnyLogic на основе модели в нотации eEPC представлены на рис. 2.

На рис. 2 а показано дерево объектов модели, на рис. 2 б, в, г отображается состав объектов, вложенных в класс «Main». Среда (окружение) моделируется классом «Main» (содержит переменные, таймер, таблицы), в который вложены активные объекты (рис. 2 г) — агенты (sB\_depart, v\_depart, dispatcher), пример стейтчартов для агентов можно видеть на рис. 1. Поскольку активные объекты технически — это классы на языке Java, то не представляется возможным давать им имена на русском языке и для пояснений приходится ограничиваться текстовыми комментариями на самой модели. В данном примере, чтобы не перегружать модель комментариями, не сделаны комментарии к переменным (например, переменная «cancelOrders» хранит количество отмененных заказов, перемен-

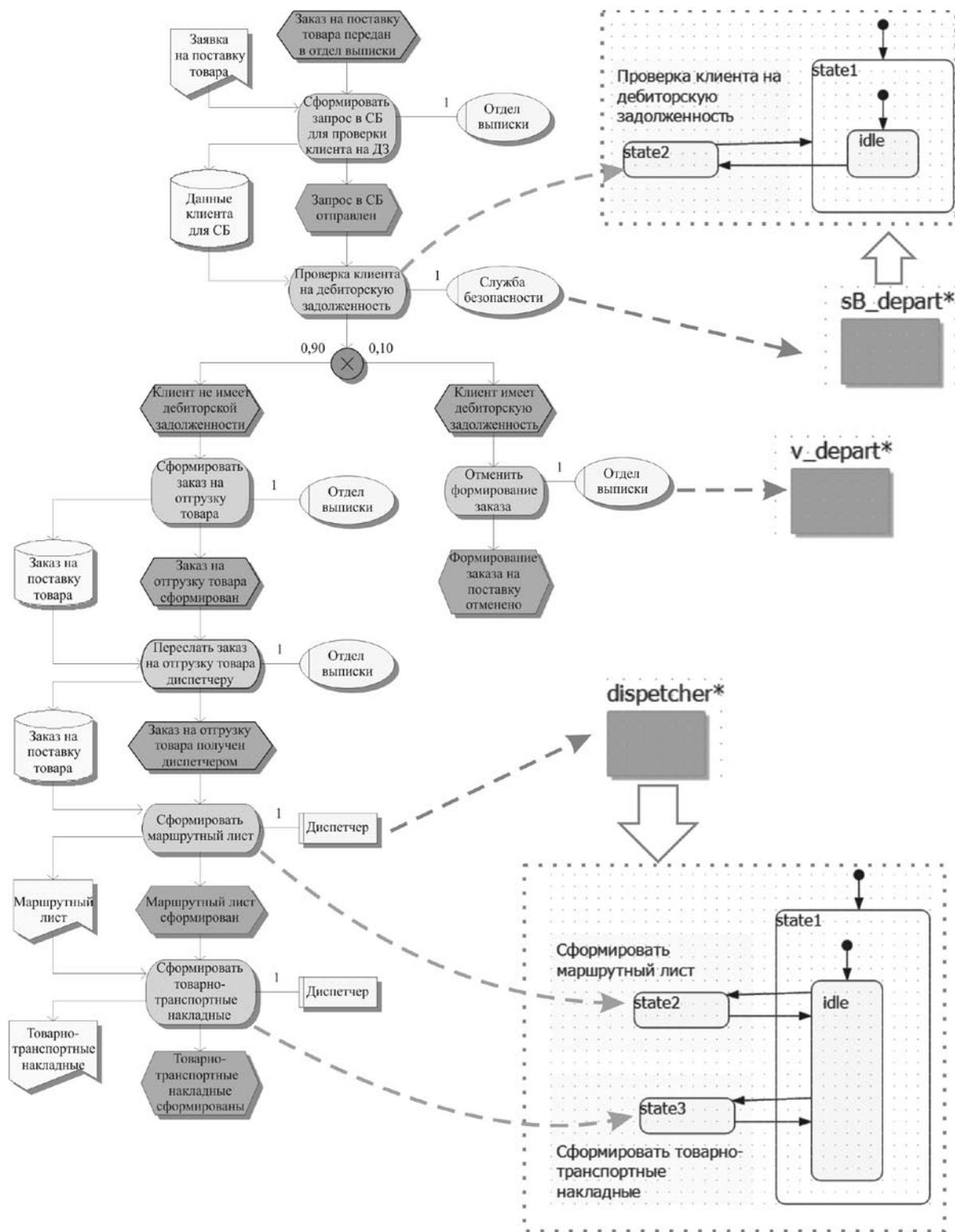


Рис. 1. Формирование заказа на поставку товара

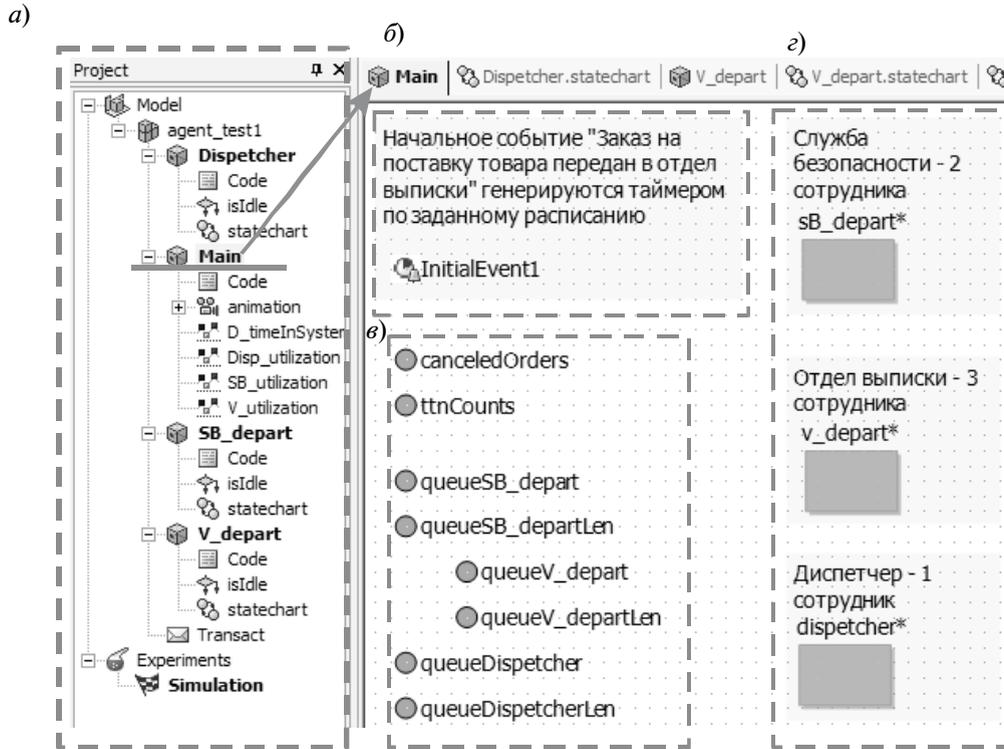


Рис. 2. Активные объекты и их окружение в агентной имитационной модели AnyLogic

ная «ttnCounts» хранит количество сообщений, сгенерированных таймером, и т. д.).

После получения структуры агентной имитационной модели определим численные характеристики объектов модели eEPC, необходимые для проведения имитационного моделирования. Определим время выполнения функций eEPC (зададим время константами на основе среднего значения), количество исполнителей, задействованных в выполнении функции (зададим константами, на рис. 1 в модели eEPC указано их количество на ребре, связывающем организационную единицу и функцию). Определим общее количество исполнителей (организационные единицы) и соответственно экземпляров агентов: «Отдел выписки» – три сотрудника, «Служба безопасности» – два сотрудника, «Диспетчер» – один сотрудник. Вероятности наступления событий в правилах ветвления «исключающего ИЛИ» (XOR) определим как вероятность события «Клиент не имеет дебиторской задолженности» – 0,95, вероятность события «Клиент

имеет дебиторскую задолженность» – 0,05. Наступление начального события «Заказ на поставку товара передан в отдел выписки» будет происходить один раз каждую единицу модельного времени.

Для верификации полученной агентной имитационной модели AnyLogic, сравним ее выходные данные с выходными данными имитационной модели встроенного в ARIS ToolSet имитатора ARIS Simulation.

Сравним усредненные выходные данные (табл. 2) по загруженности исполнителей после имитационных экспериментов в системах AnyLogic и ARIS Simulation за одинаковый модельный промежуток времени (восемь модельных часов) и с одинаковыми входными данными (описаны выше) после одинакового количества экспериментов – всего проводилось десять экспериментальных прогонов моделей в каждой из систем AnyLogic и ARIS Simulation.

Таким образом, сравнивая сводные данные по загруженности исполнителей

Таблица 2

Сводные усредненные данные по загруженности исполнителей, %

	ARIS Simulation	AnyLogic
Отдел выписки	97,8	98,6
Служба безопасности	88,5	83,7
Диспетчер	55,7	55,6

(см. табл. 2), полученные для одной и той же исходной модели eEPC, делаем вывод, что имитационная модель ARIS Simulation и модель AnyLogic, полученная посредством перевода по предложенной методике, показывают близкие результаты.

Следовательно, имитационная модель, разработанная в соответствии с предложен-

ной методикой, обладает прогностической способностью, не уступающей традиционным имитационным моделям.

Результаты работы могут использоваться аналитиками для сокращения времени реализации имитационных моделей, упрощения процесса эксплуатации имитационных моделей.

## СПИСОК ЛИТЕРАТУРЫ

1. **Таха, Хемди А.** Введение в исследование операций [Текст] / Хемди А. Таха. – М.: ИД «Вильямс», 2005. – 912 с.

2. **Замятина, О.М.** Моделирование и комплексный анализ бизнес-процессов предприятия [Текст] / О.М. Замятина. – Томск, 2004. – 174 с.

3. **Жевнерчук, Д.В.** Методика построения имитационных моделей с применением больших массивов данных на основе интеграции IDEF3, OLAP, GPSS технологий [Текст] / Д.В. Жевнерчук. – Ижевск, 2006. – 146 с.

4. **Тихонов, С.В.** Моделирование бизнес-процессов с использованием аппарата теории систем массового обслуживания [Текст] / С.В. Тихонов // Сб. тр. II конф. II школа-семинар молодых ученых. Управление большими системами. – Воронеж: Научная книга, 2007. – Т. 2. – С. 53–59.

5. **Ханова, А.А.** Организация принятия решений в виде цикла управления эффективностью организации [Текст] / А.А. Ханова, А.С. Пономарёва // Вестник астраханского государственного технического университета. Управление, вычислительная техника и информатика. – 2011. – № 2. – С. 171–177.

6. **Валькман, Ю.Р.** Определение базиса для построения мультиагентной системы имитационного моделирования бизнес-процессов в нотации BPMN [Текст] / Ю.Р. Валькман, К.В.

Муцаковская // Институту проблем моделювання в енергетиці. Збірник наукових праць. – 2012. – № 62. – С. 101–108.

7. **Van der Aalst, W.M.P.** The Application of Petri Nets to Workflow Management [Text] / W.M.P. Van der Aalst // J. of Circuits, Systems and Computers, 1998. – № 8(1). – P. 21–66.

8. **Van der Aalst, W.M.P.** Business Process Management [Text] / W.M.P. van der Aalst, A.H.M. ter Hofstede, M. Weske // A Survey. Internat. Conf. on Business Process Management, LCNS. – Berlin: Springer-Verlag, 2003. – Vol. 2678. – P. 1–12.

9. **Jensen, K.** Coloured Petri nets: Basic concepts, analysis methods and practical use [Text] / K. Jensen // Basic concepts. – Berlin: Springer-Verlag, 1996. – Vol. 1.

10. **Карпов, Ю.Г.** Имитационное моделирование систем. Введение в моделирование с AnyLogic 5 [Текст] / Ю.Г. Карпов. – СПб.: БХВ-Петербург, 2006. – 400 с.

11. **Шеер, А.В.** Бизнес-процессы. Основные понятия. Теория. Методы [Текст] / А.В. Шеер. – М.: Весть-МетаТехнология, 1999. – 182 с.

12. **Дорпер, М.Г.** Событийный анализ бизнес-процессов в нотации eEPC [Текст] / М.Г. Дорпер, Е.А. Ланцев, А.И. Шаргаева // Тр. XV Междунар. ЭМ конф. по эвентологической математике и смежным вопросам. – Красноярск: СФУ, НИИППБ, КГТЭИ, 2011. – С. 89–96.

## REFERENCES

1. **Takha Khemdi A.** Vvedenie v issledovanie operatsii. – Moscow: Izdatel'skii dom «Vil'iams», 2005. – 912 s. (rus)

2. **Zamiatina O.M.** Modelirovanie i komplek-

snyi analiz biznes-protsessov predpriiatiia. – Tomsk, 2004. – 174 s. (rus)

3. **Zhevnerchuk D.V.** Metodika postroeniia imitatsionnykh modelei s primeneniem bol'shikh mas-

sivov dannykh na osnove integratsii IDEF3, OLAP, GPSS tekhnologii. — Izhevsk, 2006. — 146 s. (rus)

4. **Tikhonov S.V.** Modelirovanie biznes-protsessov s ispol'zovaniem apparata teorii sistem massovogo obsluzhivaniia / Sb. trudov II konf. II shkola-seminar molodykh uchenykh Upravlenie bol'shimi sistemami. — Voronezh: Nauchnaia kniga, 2007. — Т. 2. — С. 53–59. (rus)

5. **Khanova A.A., Ponomareva A.S.** Organizatsiia priiniatia reshenii v vide tsikla upravleniia effektivnost'iu organizatsii / Vestnik Astrakhanskogo gosudarstvennogo tekhnicheskogo universiteta. Upravlenie, vychislitel'naia tekhnika i informatika. — 2011. — № 2. — С. 171–177. (rus)

6. **Val'kman Yu.R., Mutsakovskaia K.V.** Opredelenie bazisa dlia postroeniia mul'tiagentnoi sistemy imitatsionnogo modelirovaniia biznes-protsessov v notatsii BPMN / Institutu problem modeliuvaniia v energetitsi. Zbirnik naukovikh prats'. — 2012. — № 62. — С. 101–108.

7. **van der Aalst W.M.P.** The Application of Petri Nets to Workflow Management / The J. of Cir-

cuits, Systems and Computers. — 1998. — № 8 (1). — P. 21–66.

8. **van der Aalst W.M.P., ter Hofstede A.H.M., Weske M.** Business Process Management: A Survey / International Conf. on Business Process Management (BPM 2003). — Berlin: Springer-Verlag, 2003. — LCNS Vol. 2678. — P. 1–12.

9. **Jensen K.** Coloured Petri nets: Basic concepts, analysis methods and practical use. — Berlin: Springer-Verlag, 1996. — Vol. 1. Basic concepts.

10. **Karpov Yu.G.** Imitatsionnoe modelirovanie sistem. Vvedenie v modelirovanie s AnyLogic 5. — St. Petersburg: BKhV-Petersburg, 2006. — 400 s. (rus)

11. **Sheer A.V.** Biznes-protsessy. Osnovnye poniatiia. Teoriia. Metody. — Moscow: Vest'-MetaTekhnologiiia, 1999. — 182 s. (rus)

12. **Dorrer M.G., Lantsev E.A., Shargaeva A.I.** Sobytiinyi analiz biznes protsessov v notatsii eEPC // Tr. XV Mezhdunar. EM konf. po eventologicheskoi matematike i smezhnym voprosam. — Krasnoiarsk: SFU, NIIPPB, KGTEI, 2011. — С. 89–96. (rus)

---

**ЛАНЦЕВ Евгений Алексеевич** — аспирант кафедры схемотехники Сибирского государственного технологического университета.

660049, Россия, г. Красноярск, пр. Мира, д. 82.

E-mail: red\_snow@list.ru

**LANTSEV, Yevgeniy A.** *Siberian State Technological University.*

660049, pr. Mira, 82, Krasnoiarsk, Russia.

E-mail: red\_snow@list.ru

**ДОРРЕР Михаил Георгиевич** — доцент кафедры схемотехники Сибирского государственного технологического университета, кандидат технических наук.

660049, Россия, г. Красноярск, пр. Мира, д. 82.

E-mail: mdorrer@mail.ru

**DORRER, Mikhail G.** *Siberian State Technological University.*

660049, pr. Mira, 82, Krasnoiarsk, Russia.

E-mail: mdorrer@mail.ru



# Математическое моделирование: методы, алгоритмы, технологии

УДК 004.67, 519.688

*М.Ю. Ястребов*

## ПОИСК ФИБОНАЧЧИ – ОБОСНОВАНИЕ, СИСТЕМА СЧИСЛЕНИЯ И ХЕШИРОВАНИЕ

*M.Yu. Yastrebov*

### THE FIBONACCI SEARCH — THE SUBSTANTIATION, THE NUMBER SYSTEM AND THE HASHING

Доказана полнота поиска Фибоначчи. Алгоритм поиска позволяет представлять искомые числа кортежами специального вида из трехсимвольного алфавита. Преобразование троичной записи числа к допустимому кортежу дает процедуру хеширования.

ПОИСК ФИБОНАЧЧИ. ПОЛНОТА ПОИСКА. ПРЕДСТАВЛЕНИЕ ЧИСЕЛ. ТРОИЧНАЯ СИСТЕМА. ХЕШИРОВАНИЕ.

The article is aimed to prove the completeness of the Fibonacci search. The algorithm of the search allows representing the searched-for numbers as finite sequences ( $n$ -tuples) of a special type combined of the alphabet consisting of three symbols. The conversion of the tripartite record of a number into the admissible tuple results in the hashing procedure.

FIBONACCI SEARCH. COMPLETENESS OF THE SEARCH. REPRESENTATION OF NUMBERS. TERNARY SYSTEM. HASHING.

Изложение алгоритма поиска в отсортированном по возрастанию массиве  $a_i$  с использованием последовательно убывающих чисел Фибоначчи в качестве шагов смещения ( $\Phi$ -поиска), приведенное в основополагающих руководствах по программированию (прежде всего, у Д. Кнута в [1]), нуждается, на наш взгляд, в теоретическом обосновании. Как справедливо отмечает сам Д. Кнут, «Технология поиска Фибоначчи, на первый взгляд, представляется весьма загадочной и, если просто взять программу и постараться понять, как она работает, вам покажется, что это полное шаманство» [1, с. 450].

К сожалению, дальнейшее изложение у Д. Кнута не «превращает шаманство в обычный танец с бубном». Остаются открытыми следующие вопросы. Почему  $\Phi$ -поиск, реализованный для индексов ( $a_i = i$ ), позволяет

получить все заявленные в качестве возможного ключа поиска значения  $i$  от 1 до  $F_{k+1} - 1$ ? Как связаны алгоритм  $\Phi$ -поиска и структура дерева Фибоначчи?

Поскольку для упорядоченного массива неравенство между элементами равносильно такому же неравенству между индексами  $(a_i <, > a_j) \Leftrightarrow (i <, > j)$ , можно для упрощения обозначений считать, что  $a_i = i$ , так что ключ поиска сравнивается непосредственно с индексами.

Напомним формулировки рекурсивного определения дерева Фибоначчи  $\Delta_k$  порядка  $k$  с корнем  $F_k$  и алгоритма  $\Phi$ -поиска.

Определение. Если  $k = 0$  или  $k = 1$ , то  $\Delta_0$  – лист со значением 0 или 1 соответственно; если  $k \geq 2$ , то корнем является  $F_k$ , левым поддеревом –  $\Delta_{k-1}$ , правым –  $\Delta_{k-2}$  с увеличенными на  $F_k$  значениями.

Алгоритм

Шаг 1. Начальные присвоения:  $i := F_k$ ; шаг смещения вправо  $p := F_{k-1}$ ; шаг смещения влево  $q := F_{k-2}$ .

Шаг 2. Развилка:  $q = 0 \Rightarrow$  поиск завершен неудачно;  $K = i \Rightarrow$  элемент найден;  $K < i \Rightarrow$  переход к шагу 3;  $K > i \Rightarrow$  переход к шагу 4.

Шаг 3.  $q = 0 \Rightarrow$  поиск завершен неудачно; сдвиг влево:  $i := i - q$ ; уменьшение смещений:  $(p, q) := (q, p - q)$ ; переход к шагу 2.

Шаг 4.  $p = 1 \Rightarrow$  поиск завершен неудачно; сдвиг вправо:  $i := i + q$ ; уменьшение смещений:  $p := p - q$ ;  $q := q - p$ ; переход к шагу 2.

### Обоснование Ф-поиска

Утверждение 1. Наибольший индекс, который можно получить с помощью алгоритма Ф-поиска [1, с. 451], отправляясь от первого проверяемого индекса  $F_k$  при ключе поиска  $K > F_k$ , есть  $F_{k+1} - 1$ .

Доказательство. Если  $k$  четное, то есть  $k = 2t$ , постоянные сдвиги вправо для увеличения очередного  $i$  в процессе поиска (шаг 4) означают прибавление очередного по убыванию слагаемого  $F_{k-2j}$ . В результате получается сумма чисел  $F_k + F_{k-2} + \dots + F_2$ , равная, как известно,  $F_{k+1} - 1$  [2]. Если же  $k$  нечетное, то есть  $k = 2t - 1$ , постоянные сдвиги вправо для увеличения очередного  $i$  в процессе поиска (шаг 4) означают прибавление очередного по убыванию слагаемого  $F_{k-2j-1}$  до слагаемого  $F_3$ . В результате получается сумма чисел  $F_{2t-1} + F_{2t-3} + \dots + F_3 = F_{2t} - F_1 = F_{2t} - 1$ .

Лемма 1. Пусть  $k \geq 5$  и  $\beta(k) = F_{k-4} + F_{k-5} + \dots + F_1$ . Тогда

$$F_k + F_{k-2} - \beta(k) = F_k + 1.$$

Доказательство. Индукция по  $k$ . При  $k = 5$  (база индукции) имеем:

$$\beta(5) = F_1 = 1;$$

$$F_5 + F_3 - \beta(5) = 5 + 2 - 1 = 6 = F_5 + 1.$$

Индукционный переход:

$$\begin{aligned} & F_{k+1} + F_{k-1} - \beta(k+1) = \\ & = F_{k+1} + (F_k - F_{k-2}) - F_{k-3} - \beta(k) = \end{aligned}$$

(применяем к  $\beta(k)$  индукционное предположение)

$$\begin{aligned} & = F_{k+1} + F_k - F_{k-2} - F_{k-3} - (F_{k-2} - 1) = \\ & = F_{k+1} + 1 + ((F_k - F_{k-2}) - F_{k-3}) - \\ & \quad - F_{k-2} = F_{k+1} + 1. \blacksquare \end{aligned}$$

Лемма 2. Наименьший индекс, который можно получить с помощью алгоритма Ф-поиска, отправляясь от первого проверяемого индекса  $F_k$  при ключе поиска  $K > K_j$ , есть  $F_k + 1$ .

Доказательство. Постоянные сдвиги влево (после перехода на первой итерации алгоритма Ф-поиска к  $F_k + F_{k-2}$ ) с целью уменьшения очередного  $i$  (шаг 3) означают вычитание очередного по убыванию слагаемого  $F_{k-2j}$ , как это имеет место и в формуле для  $\beta(k)$ . ■

Очередная цель — убедиться, что таким путем можно получить и все промежуточные индексы между  $F_k + 1$  и  $F_{k+1} - 1$  правого поддерева. Поскольку индексы этого поддерева получаются увеличением на  $F_k$  левого поддерева уровня  $k - 2$ , исследуем предварительно Ф-поиск в левом поддереве исходного уровня  $k$ .

Теорема 1. Начинаясь с проверки  $F_k$  Ф-поиск с ключом  $K \in \{1, 2, \dots, F_{k+1} - 1\}$  позволяет получить все индексы  $\{1, 2, \dots, F_k - 1\} \cup \{F_k\} \cup \{F_k + 1, \dots, F_{k+1} - 1\}$ , причем ровно по одному разу.

Доказательство. Индукция по  $k$ . Базу индукции для  $k = 3$  (а также для  $k = 4, 5$ ) проверяется непосредственно. Индукционный переход. Поиск, начинающийся с проверки элемента  $F_{k+1}$ , после перехода влево на первой итерации (шаг 3) приводит к проверке числа  $F_{k+1} - F_{k-1} = F_k$  с исходными параметрами сдвига  $p = F_{k-1}$ ,  $q = F_{k-2}$ , что соответствует Ф-поиску для предшествующего стартового значения  $F_k$ . Тогда дальнейший поиск, по индукционному предположению, позволяет получить все индексы  $\{1, \dots, F_{k+1} - 1\}$ . Поиск, начинающийся с перехода вправо, приводит к индексу  $F_{k+1} + F_{k-1}$ . При этом уменьшаются параметры сдвига:  $p = F_{k-2}$ ,  $q = F_{k-3}$ . Эти параметры являются исходными для Ф-поиска, начинающегося с проверки  $F_{k-1}$ , что приводит к

Допустимые кортежи при  $k = 6$

$i$	1	2	3	4	5	6	7	8	9	10	11	12
$c_4$	-1	-1	-1	-1	-1	-1	-1	0	1	1	1	1
$c_3$	-1	-1	-1	-1	0	1	1	0	0	0	0	0
$c_2$	-1	-1	0	1	0	0	0	0	-1	-1	0	1
$c_1$	-1	0	0	0	0	-1	0	0	-1	0	0	0

отрезку индексов  $\{1, \dots, F_k - 1\}$ . Но поскольку разность между стартовыми индексами  $(F_{k+1} + F_{k-1}) - F_{k-1}$  есть  $F_{k+1}$ ,  $\Phi$ -поиск приводит к смещению этого отрезка:  $\{1 + F_{k+1}, \dots, F_k - 1 + F_{k+1}\} = \{F_{k+1}, \dots, F_{k+2} - 1\}$ . ■

**$\Phi$ -поисковое представление**

$\Phi$ -поиск индекса  $i \in \{1, \dots, F_{k+1} - 1\}$  со стартовым индексом  $F_k$  равносильно представлению  $i = F_k + c_{k-2}F_{k-2} + c_{k-3}F_{k-3} + \dots + c_1F_1$ , и задается кортежем  $g_k(i) = (c_{k-1,i}^{(k)}, c_{k-2,i}^{(k)}, \dots, c_{2,i}^{(k)}, c_{1,i}^{(k)})$ , в котором компоненты  $c_{j,i}^{(k)}$  могут принимать значения из множества  $\{-1, 0, 1\}$ . В то время как общее число таких кортежей равно  $3^{k-2}$ ,  $\Phi$ -поиску соответствуют лишь  $F_{k+1} - 1$  из них.

Назовем кортеж *допустимым*, если он соответствует  $\Phi$ -поиску некоторого  $i \in \{1, \dots, F_{k+1} - 1\}$ . В таблице приведены допустимые кортежи для  $k = 6$ .

По-видимому, до сих пор не было отмечено, что при фиксированном  $k$  (и, значит, фиксированном  $F_k$ ) кортеж  $g_k(i)$  можно рассматривать как представление числа  $i$  в своеобразной системе счисления с основанием  $F_k$ , которую назовем  *$\Phi$ -поисковым представлением*. Оно отличается от *фибоначчиевой записи* в виде последовательности нулей и единиц [2]. Последняя, в отличие от  $g_k(i)$ , опирается на последовательное выделение из  $n$  максимально возможных слагаемых-чисел Фибоначчи.

Назовем при заданном  $k$  кортеж  $(c_{k-2}, c_{k-3}, \dots, c_1)$  со значениями компонент  $c_j \in \{-1, 0, 1\}$  *правильным*, если он удовлетворяет следующим условиям:

1) если при некотором  $j$  выполняется  $c_j \neq 0, c_{j-1} = 0, c_{j-2} = 0$ , то  $c_{j-3} = c_{j-4} = \dots$

$\dots = c_1 = 0$ , т. е. два или более нулей подряд могут только замыкать кортеж;

2) если  $c_j = 1$ , то  $c_{j-1} = 0$ ;

3)  $c_1 \neq 1$ ;

4) если  $c_{k-2} = 0$ , то  $c_{k-3} = c_{k-4} = \dots = c_1 = 0$ , т. е. с нуля может начинаться только нулевой кортеж (что имеет место, когда ключ поиска равен  $F_k$ );

5) если  $c_j = -1, c_{j-1} = 0$ , то  $c_{j-2} \neq 1$  (запрещена комбинация  $-1, 0, 1$ ).

Лемма 3. При заданном  $k$  число  $m(k)$  правильных кортежей равно  $F_{k+2} - 1$ .

Доказательство. Пусть  $G(k)$  – множество правильных кортежей,  $G_{-1}(k)$  – подмножество кортежей, начинающихся с  $-1$ ,  $G_{1,0}(k)$  – подмножество кортежей, начинающихся с  $1, 0$ . Тогда

$$G(k) = \{(0, 0, \dots, )\} \cup G_{-1}(k) \cup G_{1,0}(k). \quad (*)$$

База индукции при  $k = 4$  и  $k = 5$  проверяется непосредственным перебором возможных кортежей.

Индукционный переход. Если к любому кортежу из  $G(k - 1)$  приписать слева минус единицу, получится кортеж из  $G_{-1}(k)$ , так что  $|G_{-1}(k)| \leq |G(k - 1)|$ . Обратно, если из кортежа, принадлежащего  $G_{-1}(k)$ , убрать начальную минус единицу, получится кортеж из  $G(k - 1)$ , так что  $|G_{-1}(k)| \geq |G(k - 1)|$ . Следовательно,  $|G_{-1}(k)| = |G(k - 1)|$ . Аналогично устанавливается, что  $|G_{1,0}(k)| = |G(k - 2)|$ . Применяя индукционное предположение, получаем в силу (\*):  $(F_{k+2} - 1) + 1 + F_{k+1} - 1 = F_{k+3} - 1$ . ■

Кортежи, отвечающие значению  $k + 1$ , получаются из кортежей, отвечающих значению  $k$  следующим образом.

Теорема 2. Для того чтобы кортеж

$(c_{k-1}, c_{k-2}, \dots, c_1)$  был допустимым, необходимо и достаточно, чтобы он был правильным

**Доказательство.** Необходимость условия следует из данных таблицы. Убедемся в его достаточности. Допустимые кортежи представляют числа  $\{1, 2, \dots, F_{k+2} - 1\}$ , так что их уже  $F_{k+2} - 1$ . В силу леммы они покрывают все возможные варианты правильных кортежей. ■

### Построение дерева поиска

Алгоритм Ф-поиска позволяет задать дерево, узлами которого являются индексы  $i \in \{1, \dots, F_{k+1} - 1\}$ . Именно, при  $k \geq 3$  корнем дерева объявляется  $u_1 = F_k$ , его левым потомком —  $u_2 = F_k - F_{k-2} = F_{k-1}$ , его правым потомком —  $u_3 = F_k + F_{k-2}$ . Это соответствует тому, что при  $i = u_1$  в кортеже  $g_k(u_1)$  имеем  $c_{k-1,i}^{(k)} = c_{k-2,i}^{(k)} = \dots = c_{1,i}^{(k)} = 0$ ; при  $i = u_2$  в кортеже  $g_k(i)$  имеем  $c_{k-1,i}^{(k)} = -1$ ,  $c_{k-2,i}^{(k)} = \dots = c_{1,i}^{(k)} = 0$ ; при  $i = u_3$  в кортеже  $g_k(i)$  имеем  $c_{k-1,i}^{(k)} = 1$ ,  $c_{k-2,i}^{(k)} = \dots = c_{1,i}^{(k)} = 0$ . Далее, при  $j \geq 2$  задаем рекурсивно: левый потомок для  $u_j$  есть индекс, который получается при одношаговом сдвиге из  $u_{j-1}$  на  $-q$  (шаг 3); правый потомок для  $u_j$  есть индекс, который получается при одношаговом сдвиге из  $u_{j-1}$  на  $+q$  (шаг 4). В силу утверждения 4, каждый из индексов окажется включенным в дерево, причем ровно один раз.

**Утверждение 4.** Дерево, структура которого задается алгоритмом Ф-поиска, является деревом Фибоначчи.

**Доказательство.** Индукция по  $k$ . База индукции при  $k = 3 \Leftrightarrow F_k = 2$  проверяется непосредственно. Индукционный переход. Рассмотрим дерево, которое строится по алгоритму Ф-поиска для  $k + 1$ . Согласно алгоритму, левым потомком узла  $F_{k+1}$  (шаг 3 с  $q = F_{(k+1)-2} = F_{k-1}$ ) является  $F_{k+1} - F_{k-1} = F_k$ , потомки которого, в силу индукционного предположения, образуют дерево Фибоначчи порядка  $k$ . Вместе с  $F_{k+1}$  в качестве родителя они оказываются левым поддеревом дерева Фибоначчи порядка  $k + 1$  в соответствии с определением последнего. Правым потомком является  $F_{k+1} + F_{k-1}$ ; дальнейший поиск от него происходит с исходными параметрами  $p = F_{k-2}$ ,  $q = F_{k-3}$ .

Это соответствует, согласно индукционному предположению, дереву Фибоначчи с корнем  $F_{k-1}$ . Сдвиг значений узлов последнего на  $F_{k-1}$  совпадает с условием, входящим в определение дерева Фибоначчи, поскольку разница индексов  $(k + 1) - (k - 1) = 2$ . ■

### Процедура хеширования, основанная на Ф-поисковом представлении

Записанное в троичной позиционной системе счисления число  $n \in \{1, 2, \dots, 3^k - 1\} = A(k)$  задается кортежем  $g$  из  $k$  нулей, единиц и двоек. Если заменить все двойки на минус единицы, получим кортеж  $g'$ , который может оказаться Ф-поисковым представлением некоторого другого числа  $n' \in \{1, 2, \dots, F_{k+3} - 1\}$ :  $g' = g_{k+2}(n')$ . В этом случае хеширование разреженного массива  $a_n$ ,  $n \in A(k)$  означает соответствие  $a_n \rightarrow b_{n'}$ , и мы задаем хеш-функцию в виде  $h(n) = n' = F_{k+1} + \sum_{i=1}^{k-1} c_i F_i$ .

Наряду с преобразованием кортежа троичной записи в потенциально Ф-поисковое представление по схеме  $(0 \rightarrow 0, 1 \rightarrow 1, 2 \rightarrow -1)$ , возможны еще пять других вариантов преобразования (т. е. всего 3!).

Если же кортеж  $g'$  не является допустимым, то для хеширования необходим механизм его преобразования в допустимый. Любой предложенный механизм должен пройти экспериментальную проверку на предмет равномерности в среднем распределения числа коллизий на один допустимый кортеж.

Априори представляется разумным следующий механизм преобразования. Запрещенными в кортеже являются:

- 1) тройки  $(0, 0, -1)$ ,  $(0, 0, 1)$ ,  $(0, 0, 0)$  — последняя, если не является концевой;
- 2) пары  $(1, -1)$ ,  $(1, 1)$ ;
- 3) конечная пара  $(\forall, 1)$ , где  $\forall = 0, 1, -1$ ;
- 4) начальная пара  $(0, 1)$ ,  $(0, -1)$ ;
- 5) тройка  $(-1, 0, 1)$ .

Поэтому сначала в конечной паре  $(\forall, 1)$ , если она присутствует,  $\forall$  заменяется на 0 при четном количестве минус-единиц в кортеже и на  $-1$  при нечетном. Затем начальная пара  $(0, 1)$ , если она присутству-

ет, заменяется на  $(-1, 1)$ , а начальная пара  $(0, -1)$  — на  $(-1, -1)$ . Далее случаи 3 и 4 уже встретиться не могут. Затем при движении слева направо с последовательным сдвигом на одну позицию тройка  $(0, 0, 1)$  заменяется на  $(0, 1, 0)$ , тройка  $(0, 0, -1)$  — на  $(0, -1, 0)$ ; неконцевая тройка  $(0, 0, 0)$  заменяется на  $(0, 1, 0)$  при четном количестве минус единиц в исходном кортеже и на  $(0, -1, 0)$  при нечетном. После прохода по всему кортежу случай 1 больше встретиться не может. Для устранения случая 2 при движении слева направо пары  $(1, -1)$ ,  $(1, 1)$  заменяются на пару  $(-1, -1)$ . Новых нулей при этом не появляется. На последнем этапе тройки  $(-1, 0, 1)$  заменяются на  $(-1, 0, -1)$ .

Заметим, принудительное введение в кортеж минус единиц в большем числе случаев, чем введение единиц, хорошо согласуется с тем, что правое поддерево имеет порядок на два меньший, чем левое

(в правом поддереве первая компонента кортежа является единицей, а в левом — минус единицей).

Поскольку  $F_i = \langle \alpha^i / \sqrt{5} \rangle$ , где  $\langle \dots \rangle$  обозначает ближайшее целое, а  $\alpha = (1 + \sqrt{5}) / 2$ , то коэффициент экономии памяти можно оценить как  $(\sqrt{5} / \alpha^2) \beta^k$ , где  $\beta = 3 / \alpha$ .

Резюмируя, следует сказать, что процедура поиска Фибоначчи может использоваться для формирования как новой трехсимвольной системы счисления, так и новой процедуры хеширования, не предполагающей выполнение «медленных» операций умножения и деления. Рассмотренный подход дает методологическую основу для использования в указанных целях и других поисковых процедур. Наконец, дополнительную вариативность  $\Phi$ -поисковому подходу придает возможность изменения корневого узла  $F_k$ .

#### СПИСОК ЛИТЕРАТУРЫ

1. **Кнут, В.** Искусство программирования [Текст] / В. Кнут // Сортировка и поиск. — 2-е изд. — М.: ИД «Вильямс», 2003. — Т. 3. — 820 с.
2. **Воробьев, Н.Н.** Числа Фибоначчи [Текст] / Н.Н. Воробьев. — М.: Наука, 1978. — 144 с.

#### REFERENCES

1. **Knut V.** Iskusstvo programmirovaniia; T. 3. Sortirovka i poisk; 2-e izd. — Moscow: Izdatel'skii dom «Vil'iams», 2003. — 820 s. (rus)
2. **Vorob'ev N.N.** Chisla Fibonachchi. — Moscow: Nauka, 1978. — 144 s. (rus)

**ЯСТРЕБОВ Михаил Юрьевич** — заведующий кафедрой математики Государственной морской академии имени адмирала С.О. Макарова, профессор.

198035, Россия, Санкт-Петербург, ул. Двинская, д. 5/7.

E-mail: mikyast@gmail.com

**YASTREBOV, Mikhail Yu.** Admiral Makarov State University of Maritime and Inland Shipping.

198035, Dvinskaya str., 5/7, St. Petersburg, Russia.

E-mail: mikyast@gmail.com

УДК 004.932

*В.Г. Шубников, С.Ю. Беляев*

## **ПОДАВЛЕНИЕ ШУМА И ОЦЕНКА РАЗЛИЧИЙ В ИЗОБРАЖЕНИЯХ**

*V.G. Shubnikov, S.Yu. Belyaev*

### **IMAGE DIFFERENCE ESTIMATION AND NOISE REMOVING**

Для оценки различий между двумерными изображениями обычно используется метрика PSNR (Отношение сигнала к шуму). Недавно была предложена более приближенная к человеческому восприятию мера для оценки различий – SSIM (Оценка структурного подобия). Данная статья посвящена улучшенной оценке разницы между изображениями, вычисляемой как сумма структурных параметров, взвешенная критерием наличия ребер в рассматриваемой точке изображения. Предлагаемая метрика значительно ближе к человеческому восприятию различий в изображениях по сравнению с общеизвестными оценками (PSNR, SSIM). Рассмотрена пара типовых случаев изменения изображения, когда PSNR и SSIM не могут зафиксировать различия между изображениями, а предлагаемая метрика хорошо справляется с задачей. Предложен новый алгоритм удаления шума из двумерных изображений, основанный на детектировании ребер и широко известном методе двусторонней (bilateral) фильтрации.

**СХОДСТВО ИЗОБРАЖЕНИЙ. ОБРАБОТКА ИЗОБРАЖЕНИЙ. ФИЛЬТРАЦИЯ. ДЕТЕКТИРОВАНИЕ ГРАНЕЙ. УДАЛЕНИЕ ШУМА.**

PSNR (Peak Signal to Noise Ratio) metric is typically used to estimate difference between two given images. Recent time more reliable metric (SSIM – Structural Similarity) was introduced to detect image changes/noise/etc. This article introduces more advanced approach for the image distance measurement, based on weighted sum of the image structure estimation and edges detection. Proposed metric (ESSIM – Edgeted Structural Similarity) is more close to the human image difference perception, and provide strong emphasize to pixels near edges. Paper illustrates two image modification cases, where PSNR and SSIM cannot find difference between images, but proposed metric can do it. Also this paper introduces a novel image noise removal approach, based on the weighted mixture of bilateral method and advanced edge detection approach.

**IMAGE SIMILARITY. IMAGE PROCESSING. FILTRATION. EDGE DETECTION. NOISE REMOVING.**

Алгоритмы удаления шума в изображениях широко используются в различных задачах: профессиональная и любительская фотография, аэрофотосъемка, обработка цифровых изображений в современной медицинской технике, классификация изображений, распознавание образов и т. д.

Обычно в литературе используют следующую последовательность действий при описании алгоритма подавления шумов: на первом этапе из какого-либо источника копируется исходное изображение. Как правило, берется изображение высокого

качества (оценивая его визуально), часто используются эталонные изображения из коллекций фотографий различных компаний и исследовательских центров.

На втором этапе в изображение внедряется шумовая помеха каким-либо искусственным методом. При этом обычно берется т. н. «белый шум», удовлетворяющий гауссовому распределению. Можно показать, что произвольный шум не всегда напоминает шум, возникающий при получении цифровых изображений в реальных условиях. Однако рассмотрение этой про-

блемы выходит за рамки данной статьи.

На третьем этапе искусственно зашумленное изображение подвергается очистке (фильтрации) предлагаемым алгоритмом, получая, таким образом, некое очищенное изображение. Если бы алгоритм удаления шума был бы идеален, то на выходе третьего этапа мы бы получили в точности исходное изображение. Так как идеального шумоподавления в изображениях еще не изобретено, то любой алгоритм такого рода восстанавливает исходное изображение с некоторой погрешностью.

На четвертом этапе полученное изображение сравнивается с исходным на основе какой-либо метрики, и алгоритм шумоподавления оценивается численным результатом такого сравнения. Крайне важно пользоваться «хорошей» метрикой, сравнивая два изображения. В работе [1] сделан исчерпывающий обзор популярных методик сравнения изображений. Также предложены интересные подходы к разработке других методик, однако есть только краткие словесные их описания, но нет их реализации. Одна из идей, основанная на измерениях, пропорциональных резкости граней в изображении, использована в настоящей статье. Статья [2] содержит отличное доказательство того факта, что самая популярная метрика PSNR не может использоваться для адекватного (человеческому) машинного сравнения изображений в различных типичных случаях изменения изображений. Также в [2] показано, что метрика SSIM намного ближе к человеческому восприятию по сравнению с PSNR. В нашей работе мы сделаем еще один шаг вперед от машинного (формального) сравнения изображений к человеческому.

На сегодняшний день известно множество алгоритмов удаления шума в изображениях. Казалось бы, задача уже давно решена. Однако и по сей день предлагаются новые алгоритмы удаления помех (шумов) в изображениях, которые дают еще более качественную очистку и работают еще более быстро. Наиболее простые алгоритмы со статической матрицей свертки позволяют осуществить очень быструю фильтрацию зашумленного изображения, но, к сожалению,

такой простой подход приводит к появлению размазанности изображения и потере четко видимых ребер. Удалить помехи и, одновременно с этим, сохранить четкость ребер, — главные цели любого алгоритма удаления шума. Метод двусторонней фильтрации [3] более-менее хорошо справляется с поставленной таким образом задачей. Мы предлагаем внести некоторые изменения в этот метод с той целью, чтобы результаты вычисления различий между изображениями еще лучше соответствовали восприятию различий между изображениями человеком.

### 1. Идея алгоритма вычисления различий между изображениями

В качестве основы вычисления разницы между изображениями можно взять метод вычисления SSIM (описанный подробно в [2]). Мы сделали небольшие, но существенные изменения, добавили еще один мультипликативный коэффициент, сильно влияющий на конечный результат вычислений. Согласно [1, 2] и многим другим источникам, для двух заданных изображений  $P_x$  и  $P_y$  величина SSIM вычисляется следующим образом:

$$\mu_x = \frac{1}{W \cdot H} \sum_{j=0}^{H-1} \sum_{i=0}^{W-1} P_x(i, j),$$

$$\mu_y = \frac{1}{W \cdot H} \sum_{j=0}^{H-1} \sum_{i=0}^{W-1} P_y(i, j),$$

где  $W, H$  — размер изображений (естественно, должны быть одинаковы для обоих сравниваемых изображений). Зная осредненную интенсивность, можно вычислить показатель контраста для каждого изображения:

$$C_x = \frac{1}{W \cdot H - 1} \sum_{j=0}^{H-1} \sum_{i=0}^{W-1} [P_x(i, j) - \mu_x]^2,$$

$$C_y = \frac{1}{W \cdot H - 1} \sum_{j=0}^{H-1} \sum_{i=0}^{W-1} [P_y(i, j) - \mu_y]^2.$$

После этого вычислим показатель корреляции между изображениями:

$$R = \frac{1}{W \cdot H - 1} \sum_{j=0}^{H-1} \sum_{i=0}^{W-1} |P_x(i, j) - \mu_x| \times |P_y(i, j) - \mu_y|.$$

Таким образом, для каждого изображения у нас есть три величины, описывающие качественные характеристики, на основе которых можно вычислить коэффициенты, отражающие яркость, контрастность и структурные различия между изображениями:

$$K_l = \frac{(2\mu_x\mu_y + C_1)}{(\mu_x\mu_x + \mu_y\mu_y + C_1)},$$

$$K_c = \frac{(2C_xC_y + C_2)}{(C_xC_x + C_yC_y + C_2)}, \quad K_s = \frac{(R + C_3)}{(C_xC_y + C_3)}.$$

В приведенных выше формулах  $C_1$ ,  $C_2$ ,  $C_3$  – константы, необходимые для того, чтобы избежать деление на ноль.

Итоговая величина SSIM вычисляется простым умножением коэффициентов  $K_l$ ,  $K_c$  и  $K_s$ . Позже, в разделе 3, мы проиллюстрируем недостатки SSIM. Попробуем представить, что для любого заданного изображения мы можем построить специальное изображение, интенсивность пикселей которого будет пропорциональна резкости ребер. Под словом «резкость» ребер имеется в виду степень размытости границ объектов/деталей в изображении. Значения, близкие к нулю, описывают смазанные участки изображения. Значения, близкие к единице, описывают фрагменты изображения, в которых имеется резкий переход интенсивности между группами пикселей. Под словом «ребра» мы понимаем значительную разницу интенсивности одной группы пикселей от другой, расположенной по соседству. Существует масса способов, при помощи которых можно постро-

ить «карту ребер» для заданного на входе изображения. Добавив весовые коэффициенты в расчет контрастности и корреляции, мы можем подсчитать реберно-зависимые контрастность и корреляцию:

$$C_x = \frac{\sum_{j=0}^{H-1} \sum_{i=0}^{W-1} W_x(i, j) [P_x(i, j) - \mu_x]^2}{\sum_{j=0}^{H-1} \sum_{i=0}^{W-1} W_x(i, j)},$$

$$C_y = \frac{\sum_{j=0}^{H-1} \sum_{i=0}^{W-1} W_x(i, j) [P_y(i, j) - \mu_y]^2}{\sum_{j=0}^{H-1} \sum_{i=0}^{W-1} W_x(i, j)},$$

где  $W_x$  – матрица весовых коэффициентов, значения которых лежат в интервале  $[0..1]$  и описывают наличие «ребра» в точке  $P_x(i, j)$ . Мы будем использовать только одну весовую матрицу, созданную из первого изображения, в то время как сравниваются два изображения. Такое допущение можно сделать, если считать первое изображение «эталонным», а второе – измененным, и наша цель – оценить количество этих изменений. Реберно-зависимая корреляция между изображениями может быть вычислена:

$$R_w = \frac{\sum_{j=0}^{H-1} \sum_{i=0}^{W-1} W_x(i, j) |P_x(i, j) - \mu_x| \cdot |P_y(i, j) - \mu_y|}{\sum_{j=0}^{H-1} \sum_{i=0}^{W-1} W_x(i, j)}.$$

Также мы предлагаем вычислить корреляцию контраста ( $D$ ) и взвешенную корреляцию контраста ( $D_w$ ):

$$D = \frac{\sum_{j=0}^{H-1} \sum_{i=0}^{W-1} [(P_x(i, j) - \mu_x) - (P_y(i, j) - \mu_y)]^2}{W \cdot H - 1};$$

$$D_w = \frac{\sum_{j=0}^{H-1} \sum_{i=0}^{W-1} W_x(i, j) [(P_x(i, j) - \mu_x) - (P_y(i, j) - \mu_y)]^2}{\sum_{j=0}^{H-1} \sum_{i=0}^{W-1} W_x(i, j)}.$$

Для этих новых (относительно алгоритма вычисления SSIM) параметров можно рассчитать специальный коэффициент, показывающий отношение между величинами  $D$  и  $D_w$ :

$$K_w = \frac{(D + C_4)}{(D + D_w + C_4)}.$$

Для тех пар изображений, в которых различия в «реберных» зонах существенны, величина  $K_w$  будет стремиться к нулю. Пред-

лагаемая новая характеристика  $K_w$  соответствует принципам человеческого восприятия разницы в изображениях – чем больше разницы в «реберных» зонах, тем сильнее различаются изображения в целом. В итоге можно рассчитать улучшенную меру различия между изображениями:

$$ESSIM = K_l \cdot K_c \cdot K_s \cdot K_w.$$

ESSIM (Edged Structure Similarity Image Metric – Реберно-структурная метрика похожести изображений) несет ту же смысловую нагрузку, что и SSIM: величина, равная нулю, означает совершенно не похожие изображения, величина, равная единице, описывает одинаковые изображения. Чем больше величина ESSIM ближе к единице, тем большим сходством обладают сравниваемые изображения.

## 2. Вычисления карты ребер

В предыдущем разделе мы ссылались на весовую матрицу  $W_x$ , которая характеризует наличие ребер для каждого пикселя изображения. Для простоты вычислений  $W_x$  мы будем использовать алгоритм, описанный в [4]: на первом этапе производится сглаживание изображения с целью удаления выбросов, связанных с шумом. Далее вычислим градиентные поля  $G_x$  и  $G_y$ :

$$G_x(x, y) = \sum_{j=-1}^{+1} \sum_{i=-1}^{+1} P_x(x+i, y+j) \cdot KG_x(1+i, 1+j);$$

$$G_y(x, y) = \sum_{j=-1}^{+1} \sum_{i=-1}^{+1} P_y(x+i, y+j) \cdot KG_y(1+i, 1+j).$$

Здесь  $KG_x$ ,  $KG_y$  – простые матрицы, участвующие в вычислении свертки матрицы:

$$KG_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \quad KG_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}.$$

Величины  $G_x$  и  $G_y$  описывают градиенты интенсивности изображения в обоих направлениях: горизонтальном и вертикальном.  $KG_x$ ,  $KG_y$  также хорошо известны как операторы Собеля. Окончательный расчет реберной характеристики

для точки  $(x, y)$  выполняется по формуле  $W_x(x, y) = \sqrt{G_x(x, y)^2 + G_y(x, y)^2}$ .

## 3. Синтетические тесты для ESSIM

Чувствительность предложенной оценки ESSIM может быть проиллюстрирована на специальных синтетических изображениях, полученных искусственным путем. Например, создадим специальное тестовое изображение, состоящее из простых градиентов интенсивности цвета и вертикальных периодических полос. Мы можем внедрить белый шум с распределением Гаусса в это тестовое изображение двумя различными способами: в первом случае мы изменим пиксели изображения в областях, у которых нет явных ребер. Во втором случае изменим изображение вокруг резких границ интенсивности (рис. 1).

В обоих случаях площадь поврежденных пикселей одна и та же, разброс псевдослучайного шума один и тот же, и в обоих изменениях шум имеет одну и ту же интенсивность. Если внимательно приглядеться, то можно заметить, что изображение справа выглядит более «испорченным» по сравнению с левым. Такое впечатление получается, если внимательно присмотреться к ребрам: на изображении слева ребра не тронуты, поэтому человеческий глаз «полагает», что слева изображение более качественно, чем справа. Этот эффект близок к хорошо известной иллюзии полос Маха, показывающей, что человеческий глаз видит один и тот же цвет по-разному в зависимости от наличия резких реберных границ. (исчерпывающий обзор оптических иллюзий можно прочитать в [5, 6]). Парное сравнение между изображениями на рис. 1 и исходным изображением (без внедренного шума) дает несущественное различие между оценками PSNR и SSIM, однако для предлагаемой величины ESSIM различия будут куда более значимыми (табл. 1).

Оценка PSNR наводит нас на неправильный вывод о том, что изображение справа на рисунке ближе к исходному (без внедренного шума), чем изображение слева. Оценка SSIM уверяет, что изменения чуть ли не одинаковые. И только величина

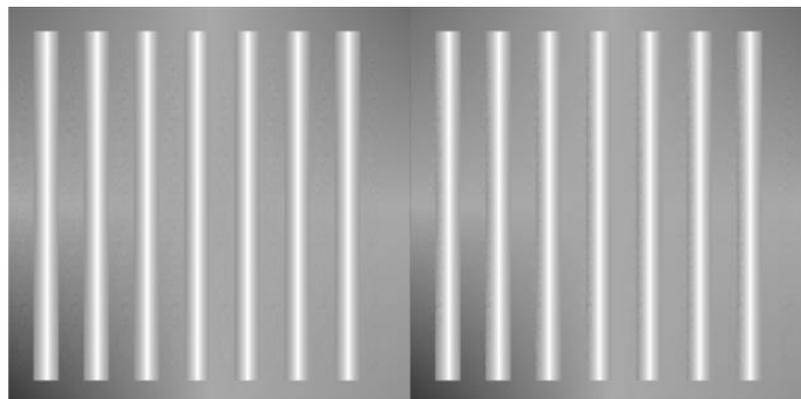


Рис. 1. Близкие изменения, внедренные в разные зоны исходного изображения

ESSIM показывает значительную разницу в изменениях, причем эта разница соответствует человеческому восприятию: изображение справа на рисунке повреждено более существенно. Ко всему прочему, мы протестировали, каким образом простые изменения изображения (которые легко выполняются в графических редакторах) отражаются на этих трех параметрах. Существенное изменение изображения – создание негатива. В табл. 2 приведены результаты измерений трех метрик в зависимости от простых изменений изображения. Метрика ESSIM наиболее чувствительна к изменениям.

Естественно, что размазывание изображения, изменение контраста и сжатие с потерями – наиболее серьезные модификации, которые затрагивают области ребер. Таблица иллюстрирует факт того, что именно эти изменения сильнее всего сказываются на ESSIM. Последнее изменение наиболее интересно: в каждом пикселе красная компонента цвета увеличена на 10 (диапазон изменения цвета считаем в  $[0..255]$ ), а зеленая и синяя компоненты уменьшены на 5. Обе широко используемые метрики (PSNR, SSIM) в этом случае не фиксируют изменений в изображении

(но визуально изменение очевидно), но в то же время предлагаемая метрика ESSIM показывает существенные различия.

#### 4. Метод удаления шума в изображении

Метод двусторонней фильтрации, предложенный в [3], базируется на более сложной идее по сравнению с простой взвешенной по Гауссу нелинейной фильтрацией. Большинство новых алгоритмов в этом классе основаны на сложном вычислении весов. В работе [7] приведен алгоритм, основанный на комбинации двух двусторонних фильтров. Существенный недостаток предложенного метода скрыт в самом вычислении весов, основанном на различии медиан групп пикселей, собранных в окрестностях двух сравниваемых пикселей. Разница медиан в некотором смысле повторяет разницу между интенсивностями самих сравниваемых пикселей, только в более «интегрированном» виде, поэтому эта характеристика не меняет акцент весов. В данной статье мы предлагаем новый способ вычисления весовых коэффициентов. Пусть  $W$  и  $H$  – размеры изображения;  $N$  – радиус квадратной окрестности вокруг каждого пикселя изображения  $P(x, y)$ . Окрестность

Таблица 1

Сравнение PSNR, SSIM, ESSIM

Изменение изображения	PSNR	SSIM	ESSIM
Шум в гладких зонах	29,53	0,9992	0,9682
Шум в реберных зонах	29,98	0,9990	0,5088

Таблица 2

Влияние простых изменений изображения на PSNR, SSIM, ESSIM

Способ изменения	PSNR	SSIM	ESSIM
Контраст	27,44	0,9767	0,3947
Негатив	10,87	0,9454	0,6778
Добавление гауссова шума	29,98	0,9892	0,5066
Сжатие с потерями	28,66	0,9886	0,3530
Добавление размазывания	26,43	0,9870	0,2489
Изменение каналов RGB	100,0	0,9997	0,6703

требуется для учета не только самого пикселя  $(x, y)$  но также и его соседей. Площадь этой квадратной окрестности равна  $S = (2N+1)(2N+1)$ .

Взяв за основу нелинейную фильтрацию, значение интенсивности пикселя в точке  $(x, y)$  вычисляется так:

$$P'(x, y) = \frac{\sum_{j=-N}^{N+1} \sum_{i=-N}^{N+1} w(i, j) P(x+i, y+j)}{\sum_{j=-N}^{N+1} \sum_{i=-N}^{N+1} w(i, j)}$$

Увеличивая значение  $N$ , мы делаем процесс вычисления более «охватывающим». Отрицательный эффект в увеличении  $N$  состоит в увеличении времени расчета. Классический двусторонний фильтр использует следующую формулу для расчета весов в точке  $(i, j)$ , находящейся в окрестности точки  $(x, y)$ :

$$w(i, j) = e^{-\frac{(i^2+j^2)}{2\sigma_s^2}} e^{-\frac{|P(x+i,y+j)-P(x,y)|^2}{2\sigma_r^2}},$$

где  $\sigma_s$  и  $\sigma_r$  – пространственные и радиометрические константы, оператор  $|P(x+i, y+j) - P(x, y)|$  – простой квадрат разности интенсивностей в пикселях  $(x, y)$  и текущем пикселе в его окрестности. Формула вычисления  $w(i, j)$  иллюстрирует простой принцип расчета веса: результат зависит от умножения двух компонент. Пиксель  $(i, j)$  в окрестности  $(x, y)$  будет обладать меньшим весом по мере увеличения расстояния между  $(i, j)$  и  $(x, y)$ . Меньший вес будет назначен также в случае, если различие между интенсивностями в рассматриваемом пикселе  $(i, j)$  и центральном пикселе окрестности  $(x, y)$  будет увеличиваться. Говоря более простым языком, если в окрестности  $(x, y)$  интенсивности мало различаются, то

результатом фильтрации будет осредненное значение интенсивностей. Если же у центрального пикселя окрестности интенсивность существенно отличается от пикселей по соседству, веса соседних пикселей будут стремиться к нулю, и значения их интенсивностей не будут учтены в результирующем пикселе.

В нашем подходе мы пойдем на один шаг дальше: будем сравнивать «реберность» центрального пикселя  $(x, y)$  и пикселей в его окрестности  $(i, j)$ . Если текущий рассматриваемый пиксель окрестности лежит на том же ребре, что и центральный пиксель (или наоборот, оба не лежат на ребре), мы будем учитывать его интенсивность и назначим больший вес для пикселя  $(i, j)$ . Если «реберность» текущего пикселя сильно отличается от такого же показателя центрального пикселя, то мы будем назначать меньший вес. В конечном итоге для каждого пикселя  $(i, j)$  в окрестности  $(x, y)$  мы будем рассчитывать пространственный, радиометрический и реберный коэффициенты:

$$C_s = i^2 + j^2;$$

$$C_r = |P(x, y) - P(x+i, y+j)|^2;$$

$$C_e = |W(x, y) - W(x+i, y+j)|,$$

где  $W(x, y)$  – реберная матрица, созданная методом Канни (описанным выше). В результате вес может быть вычислен как

$$w(i, j) = e^{-\frac{(C_s * C_r * C_e)}{\sigma}}$$

Важное различие между предлагаемым методом и классической двусторонней фильтрацией состоит в умножении компонент (а не сложении) – аргументов экспоненты. Такой прием позволит сделать ре-



Рис. 2. Результаты удаления шума из изображения Caracas<sup>1</sup>.  
Верхний ряд слева направо: исходное изображение, внедренный шум. Нижний ряд: изображения, восстановленные классическим двусторонним и предлагаемым фильтром

<sup>1</sup> Caracas: Коллекция изображений Flickr

зультирующий вес более чувствительным к изменениям каждой компоненты. Для практической реализации важно предоставить возможность изменять как размер квадратной окрестности, так и значение  $\sigma$ , влияющее на гладкость (большее значение приводит к более размытому результату).

### 5. Удаление шума в реальных изображениях

Для искусственно внедренных шумов в изображения предлагаемый метод не дает очень впечатляющих результатов, если сравнивать результат очистки. Однако по-

казатель ESSIM показывает лучшую очистку именно для двусторонне-реберного метода. На рис. 2 и 3 приведены несколько тестовых изображений, в которые внедрен шум и удален классическим и предлагаемым двусторонним методом. На этих иллюстрациях хорошо заметно, что контрастные зоны (около ребер) практически одинаковы, а зоны в смазанных частях зашумленного изображения очищены лучше предлагаемым фильтром.

Помимо очистки искусственно внедренных шумов, мы применили предлагаемый

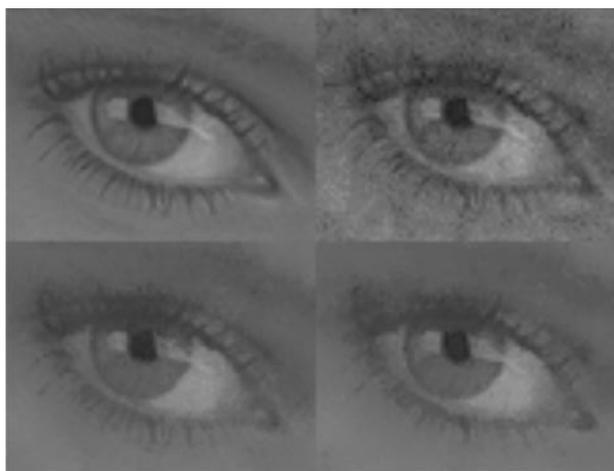


Рис. 3. Результаты удаления шума из изображения Girl<sup>2</sup> классическим двусторонним и предлагаемым фильтром (нижний ряд).

В верхнем ряду исходное изображение (слева) и внедренный искусственный шум (справа)

<sup>2</sup> Girl, Light House, City, Masaw: Коллекция изображений компании Kodak (<http://r0k.us/graphics/kodak/>)



Рис. 4. Изображение Face<sup>3</sup>, полученное с высоким ISO до и после реберной двусторонней фильтрации,  $N = 6$  (радиус окрестности),  $\sigma = 900$  (параметр вычисления веса)

<sup>3</sup> Любительское фото, цифровая камера Canon, предоставлено Иваном Крыловым

фильтр к реальным зашумленным изображениям, полученным в цифровых камерах, одно из которых показано на рис. 4.

В данной статье предложен новый метод оценки сходства между изображениями, основанный на принципах, схожих с человеческим восприятием. Проиллюстрировано несколько случаев, в которых общепринятые оценки (PSNR, SSIM) не отражают визуально заметных изменений, а предлагаемая оценка ESSIM справляется с поставленной задачей. Предлагаемая оценка может использоваться в профессиональных программах/плагинах и вряд ли может применяться в приложениях, работающих в масштабе реального времени. Также пред-

ложено улучшение известного метода двусторонней фильтрации для удаления шумов в изображении.

В табл. 3 приведены оценки качества очистки тестовых изображений предлагаемым методом.

Легко заметить, что метрика ESSIM всегда соответствует лучшему результату очистки. Небольшие изменения во втором значащем разряде не должны наводить на мысль о том, что улучшение незначительное. Во-первых, даже предложенная метрика ESSIM не может считаться идеально адекватной человеческому восприятию (она лучшая из трех, но все же не идеальная), а, во-вторых, даже небольшое улучшение в алгоритме удаления шумов является очень важным.

Таблица 3

**Результаты очистки искусственно внедренных шумов**

Тестовое изображение	Метод очистки шума	PSNR	SSIM	ESSIM
Caracas	Двусторонний	26,44	0,9792	0,3643
	Реберный двусторонний	25,73	0,9735	0,3896
City	Двусторонний	26,21	0,9900	0,4167
	Реберный двусторонний	25,22	0,9874	0,4508
Macaw	Двусторонний	27,54	0,9932	0,3344
	Реберный двусторонний	27,73	0,9931	0,3569
Girl	Двусторонний	27,18	0,9746	0,3625
	Реберный двусторонний	28,46	0,9754	0,3788
Light House	Двусторонний	27,04	0,9859	0,4078
	Реберный двусторонний	26,58	0,9837	0,4281

### СПИСОК ЛИТЕРАТУРЫ

1. **Al-Najar, Y.A.Y.** Comparison of Image Quality Assessment: PSNR, HVS, SSIM, UIQI [Text] / Y.A.Y. Al-Najar, D.C.Soong // International J. of Scientific & Engineering Research. – 2008. – Vol. 3. – Iss. 8.
2. **Ndajah, Peter.** Watanabe, Shogo Muramatsu, SSIM Image Quality Metrics for Denoised Image [Text] / Peter Ndajah, Hisakazu Kikuchi, Masahiro Yukawa [et al.] // 2010.
3. **Tomasi, C.** Bilateral filtering for gray and color images [Электронный ресурс] / C. Tomasi, R. Manduchi // In Proc. Int. Conf. Computer Vision. –1998. –P. 839–846.
4. **Zhou, P.** An Improved Canny Algorithm for Edge Detection [Text] / W.Ye, Y.Xia, Q.Wang // J. of Computational Systems 7:5. –2011.
5. **Adelson, E.H.** Lightness Perception and Lightness Illusions [Text] / E.H. Adelson // In the New Cognitive Neurosciences; 2<sup>nd</sup> ed. – MIT press. –2000. –P. 339–351.
6. **Andraos, J.** Named Optical Illusions; Department of Chemistry [Text] / J. Andraos. –New York University, 2003–2011.
7. **Vijaya, G.** Noise Reduction Method using Double Bilateral Filtering [Text] / G.Vijaya, V.Vasudevan, A Novel // European J. of Scientific Research. – 2010. –Vol. 46. –№ 3. –P.331–338.

### REFERENCES

1. **Al-Najar Y.A.Y., Soong D.C.** Comparison of Image Quality Assessment: PSNR, HVS, SSIM, UIQI / International Journal of Scientific & Engineering Research. –2008. –Vol. 3. –Iss. 8.
2. **Peter Ndajah, Hisakazu Kikuchi, Masahiro Yukawa et al.** SSIM Image Quality Metrics for Denoised Image, 2010.
3. **Tomasi C., Manduchi R.** Bilateral filtering for gray and color images; in Proc. Internat. Conf. Computer Vision, 1998. –P. 839–846.
4. **Zhou P., Ye W., Xia Y., Wang Q.** An Improved Canny Algorithm for Edge Detection / J. of Computational Systems 7:5. –2011.
5. **Adelson E.H.** Lightness Perception and Lightness Illusions / In the New Cognitive Neurosciences; 2<sup>nd</sup> ed. –MIT press, 2000. –P. 339–351.
6. **Andraos J.** Named Optical Illusions, Department of Chemistry; New York University, 2003–2011.
7. **Vijaya G., Vasudevan V.** A Novel Noise Reduction Method using Double Bilateral Filtering / European J. of Scientific Research. –2010. –Vol. 46. –№ 3. –P. 331–338.

---

**ШУБНИКОВ Владислав Германович** – доцент кафедры прикладной математики Института прикладной математики и механики Санкт-Петербургского Политехнического Университета.

195251, Россия, Санкт-Петербург, ул. Политехническая, д. 29.  
E-mail: vlad.shubnikov@gmail.com

**SHUBNIKOV, Vladislav G.** St. Petersburg State Polytechnical University.

195251, Politekhnikeskaya Str. 29, St. Petersburg, Russia.  
E-mail: vlad.shubnikov@gmail.com

**БЕЛЯЕВ Сергей Юрьевич** – профессор кафедры прикладной математики Института прикладной математики и механики Санкт-Петербургского Политехнического Университета.

195251, Россия, Санкт-Петербург, ул. Политехническая, д. 29.  
E-mail: sergey.belyaev@d-inter.ru

**BELYAEV, Sergey Yu.** St. Petersburg State Polytechnical University.

195251, Politekhnikeskaya Str. 29, St. Petersburg, Russia.  
E-mail: sergey.belyaev@d-inter.ru

УДК 004.825

*Е.И. Антонова, И.А. Васильев*

**ПРОБЛЕМА ОРГАНИЗАЦИИ ГРУЗОВЫХ РАБОТ  
НА ЖЕЛЕЗНОЙ ДОРОГЕ КОНТЕЙНЕРНОГО ТЕРМИНАЛА**

*E.I. Antonova, I.A. Vasilev*

**THE PROBLEM OF CARGO OPERATION MANAGEMENT  
ON CONTAINER TERMINAL RAILWAY**

Исследованы существующие проблемы в работе железной дороги контейнерного терминала на примере Владивостокского контейнерного терминала (ВКТ). Проанализированы технологические процессы, протекающие на нем, связанные с погрузкой и выгрузкой контейнеров на железной дороге. Изучена возможность автоматизации части технологических процессов с целью оптимизации управления работой контейнерного терминала на железной дороге. Цель исследования – разработка методов автоматизации процесса управления грузовыми работами на железнодорожном транспорте контейнерного терминала. Приведена постановка задач, решение которых необходимо найти для реализации цели по разработке методов автоматизации грузовых процессов на железной дороге контейнерного терминала.

**КОНТЕЙНЕРНЫЕ ПЕРЕВОЗКИ. РАБОТА ЖЕЛЕЗНОЙ ДОРОГИ. СИСТЕМЫ УПРАВЛЕНИЯ КОНТЕЙНЕРНЫМИ ТЕРМИНАЛАМИ. АВТОМАТИЗАЦИЯ РАБОТЫ ЖЕЛЕЗНОЙ ДОРОГИ. ПОСТАНОВКА ЗАДАЧИ ПЛАНИРОВАНИЯ ТЕХНОЛОГИЧЕСКОГО ПРОЦЕССА.**

The article covers the research of existing problems with container terminal railway functioning. Technological processes are analyzed that are in progress when discharging and loading to railway transport using Vladivostok container terminal as an example. Author researches ability to automate technological process to get managing container terminal railway more optimized. The goal of research is development of automating methods for cargo processes management at container terminal. The article contains formal description of the tasks that should be solved to achieve the goal of developing the methods for automating cargo loading processes at container terminal railway.

**CONTAINER TRANSPORTATION. RAILWAY PROCESSES. CONTAINER TERMINAL MANAGEMENT SYSTEMS. RAILWAY PROCESSES AUTOMATING. FORMAL PLANNING OF TECHNOLOGICAL PROCESS.**

**Характеристика контейнерных  
железнодорожных перевозок**

Железнодорожный транспорт в РФ имеет исключительно важное значение в жизнеобеспечении многоотраслевой экономи-

ки и реализации социально значимых услуг по перевозке пассажиров. На его долю приходится более 75 % грузооборота и 40 % пассажирооборота, выполняемого транспортом общего пользования [1].

Успешность развития транспортной отрасли определяется эффективностью управления и грамотной организацией деятельности, а также оперативным контролем и качественным анализом ее осуществления. Ежесуточное оперативное планирование местной работы осуществляется в целях стабильного выполнения заданий по погрузке и выгрузке, ускорения доставки вагонов с местным грузом на станции назначения, равномерного распределения грузовой работы по периодам суток, избегания нарушений сроков доставки грузов. Организация информационного взаимодействия между отделением железной дороги и предприятием-грузоотправителем (грузополучателем) является ключевым фактором, определяющим интенсивность развития отрасли и активности бизнеса в ней.

Одно из важнейших направлений транспортировки грузов по железной дороге – контейнерные грузоперевозки. По информации журнала «Эксперт» объем железнодорожных контейнерных перевозок по России в 2011 г. составил 2,6 млн TEU (что на 12 % превосходит объем за 2010 г.) [2]. По мнению ведущих экспертов железнодорожной отрасли, в течение следующих 10 лет объемы контейнерных перевозок могут увеличиться в 2–2,5 раза. Основные грузопотоки контейнеров в России проходят через порт Санкт-Петербурга (35 %), порты Восточный и Владивосток Дальневосточного бассейна (23 %).

Грамотное планирование работы контейнерного терминала непременно стабилизирует работу таможенных органов. Контейнерные терминалы не только обрабатывают экспортные и импортные контейнерные грузы, но и хранят их под таможенным контролем на складах временного хранения и на таможенных лицензионных складах. Формирование, сортировка, перепакровка, изменение вида транспорта для дальнейшей перевозки товаров в разных таможенных режимах, таможенное оформление, посреднические услуги, транспортно-экспедиционное обслуживание – все это

нередко происходит на площадях контейнерных терминалов, а значит, оперативная выгрузка и погрузка грузов – необходимое условие для своевременной таможенной очистки груза.

### **Проблемы деятельности железнодорожного направления контейнерного терминала**

Особенности грузового процесса, связанного с выгрузкой и погрузкой контейнеров на железнодорожный транспорт, можно отследить на примере работы контейнерного терминала Владивостокского морского торгового порта (ВМТП). Владивостокский морской торговый порт (входит в Транспортную группу FESCO) оказывает услуги по обработке генеральных, навалочных, контейнерных грузов. ВМТП является лидером по перевалке контейнеров, автомобилей, спецтехники, тяжеловесных грузов среди портов российского Дальнего Востока. ВМТП обладает уникальным, среди портов региона, опытом переработки зерновых грузов. Порт имеет соответствующую мировым стандартам сервисную структуру, включающую в себя стивидорные, агентские, экспедиторские, буксирные, сюрвейерские, тальманские и другие компании [3]. Владивостокский контейнерный терминал (ВКТ) расположен на территории ВМТП. Это крупнейший терминал на Дальнем Востоке, имеющий многолетний опыт работы в области переработки контейнеров и использующий на данный момент электронную систему для обработки контейнеров (ИС ВКТ).

ВКТ включает все три пути обработки контейнеров: морской, автомобильный и железнодорожный. Грузооборот ВКТ за октябрь 2012 г. составил 33 055 TEU, из них 6 916 TEU составило перемещение контейнеров с железнодорожного (ЖД) транспорта и 7 783 TEU пришлось на долю отгрузки контейнеров на ЖД транспорт. В сумме это составляет 44,5 % от общего объема грузовых перевозок. По состоянию на октябрь 2012 г. ВКТ имеет 20 железнодорожных подъездных путей, из них 12 пригодны для

обработки, т. е. вагоны, находящиеся на этих подъездных путях, могут обрабатываться погрузочной техникой.

Основной объем контейнеропотока, проходящего через железнодорожный транспорт, отправляется на экспорт морским путем (45 503 TEU за 11 месяцев 2012 г. или 75,8 % от общего числа контейнеров, прибывших по железнодорожным путям). Контейнеры, приходящие с ЖД вагонов, формируются в партии на площадках контейнерного терминала в ожидании дальнейшей отправки на море. В связи с различными факторами, оказывающими влияние на процесс перевозки грузов по железным дорогам, процесс комплектования партий контейнеров может занимать длительное время, вплоть до нескольких месяцев. Это обуславливает необходимость планировать размещение контейнеров при выгрузке их с железнодорожного транспорта в складские секции хранения. Необходимо хранить контейнеры, предназначенные для отправки одному грузополучателю или направляющиеся в один порт, в одной секции (либо в соседних секциях).

Грузы, отправляющиеся в контейнерах по железной дороге, также накапливаются на складских площадях контейнерного терминала. В процессе погрузки нередко случается, что контейнеры, которые должны быть погружены на одну платформу, находятся в разных частях терминала. Помимо этого, если нужный контейнер находится в нижних ярусах штабеля, то для его погрузки приходится перемещать все контейнеры, находящиеся сверху. Все это приводит к затратным и излишним операциям перестановки контейнеров, что увеличивает как себестоимость погрузки контейнеров, так и время выполнения этих действий.

Несмотря на необходимость развития железнодорожных контейнерных перевозок, качество оказания транспортных услуг на железной дороге оставляет желать лучшего. Невыполнение сроков подачи подвижного состава на станции, простой вагонов на железнодорожных тупиках, задержка

сроков доставки грузов грузополучателям, сложности с оформлением грузовых документов — все это не способствует успешному развитию железнодорожной отрасли. Невыполнение сроков доставки грузов по железной дороге оказывает влияние и на работу контейнерных терминалов, получающих в свой адрес контейнеры по железной дороге. Не последнее место среди причин такого положения дел занимает человеческий фактор. Исполнители на местах не справляются с объемом работ, которые необходимо выполнять для своевременного и бесперебойного функционирования системы. Планирование местной работы на сортировочной станции контейнерного терминала выполняется периодически, по мере поступления и обработки информации, достаточной для принятия решения ответственным человеком. Иногда эти интервалы времени могут достигать нескольких часов, что недопустимо в виду высокой оперативности работ по всем транспортным направлениям на контейнерном терминале.

Сотрудникам контейнерного терминала приходится сталкиваться с подобными проблемами ежедневно в процессе своей профессиональной деятельности. Ежедневный процесс погрузки и выгрузки контейнеров сопровождается постоянным контролем мест размещения контейнеров, их наполненности и специализации по направлениям грузов. Помимо этого сотрудник терминала должен учитывать наличие погрузочной техники и возможность обработать тот или иной подъездной путь, а также пригодность подходящей секции хранения для обработки. Для обработки такого объема информации сотрудник контейнерного терминала должен обладать большим опытом работы, а также иметь достаточную степень квалификации. Безусловно, при интенсивной нагрузке неизбежны ошибки или накладки в составлении графиков обработки железнодорожных вагонов. Оптимизация производственного процесса в такой ситуации остается на втором месте. Все это говорит

о крайне высокой степени актуальности перечисленных проблем, решение которых существенно могло бы повысить качество услуг, предоставляемых контейнерным терминалом.

Перечисленные проблемы могут быть решены путем внедрения автоматизированных систем учета. Автоматизация деятельности людей, занятых в процессах планирования работы контейнерного терминала, позволит систематизировать производственные процессы, обеспечить полный учет информации о транспортно-экспедиторской деятельности, сократить время, необходимое для принятия решений, и повысить эффективность управления. Однако для успешной автоматизации подобной профессиональной деятельности, характеризующейся высокими темпами изменения оперативной обстановки, большим количеством обрабатываемой информации и сложными взаимосвязями внутри управляемых объектов, необходимы мощные теоретические интеллектуальные методы и практически реализованные инструменты, позволяющие решать подобные задачи в режиме реального времени. Проблема отсутствия достаточных знаний о методах решения подобных задач выходит на первый план в свете данных рассуждений.

#### **Современные подходы к решению проблем функционирования железной дороги**

В настоящее время большое количество исследований уделяется проблемам оптимизации технологических процессов на транспорте. Большой вклад в исследование проблем оптимизации взаимодействия различных видов транспорта при интермодальных перевозках внесли труды В.Н. Зубкова, П.А. Козлова, А.Ф. Котляренко, Е.М. Кожанова, А.П. Кузнецова, А.В. Харитоновна, А.Г. Черняева, А.В. Рыженкова и др. [4, 5].

В работе [6] исследован вопрос оптимизации подвода грузов к припортовой станции. Отдельное внимание уделено разработке имитационной модели функ-

ционирования транспортного узла, позволяющей выполнить оценку статистических характеристик и законов распределения времени выполнения технологических операций при интермодальных перевозках, а также прогнозировать статистические характеристики показателей качества транспортных процессов (например, простой вагона на припортовой станции). В качестве теоретического аппарата построена модель оптимального подвода грузов к припортовой станции, выполненная в терминах квадратической задачи о назначениях, позволяющая автоматизировать процесс распределение нитей графика движения поездов между составами, находящимися на подходе к транспортному узлу.

В [7] рассмотрен вопрос определения оптимального числа контейнерных терминалов на полигоне одной железной дороги, а также решена проблема оптимального расположения базового контейнерного терминала на полигоне дороги методом «подвижной точки». В работе использованы методы статистической обработки данных и законы распределения. К теоретическим методам, использованным в [7], относится «метод определения центра тяжести» системы распределения, который может применяться в очень ограниченном числе случаев. Распределительная система должна иметь только один распределительный центр, все его клиенты должны быть равнодоступны, что предполагает наличие в обслуживаемом районе развитой автодорожной сети. Ограниченное применение на практике имеет и другой метод, называемый «методом пробной точки». Имеются два ограничения: наличие развитой транспортной сети и прямоугольная форма обслуживания области. В [7] предложена новая методика определения места расположения базового контейнерного терминала «методом подвижной точки», основанная на методе половинного деления (дихотомии).

Зарубежная система управления контейнерным терминалом компании Port IT & Security Solutions [8] отображает гра-

фическую информацию о загруженности склада, предоставляя возможность вручную сгруппировать партии контейнеров для планирования отгрузки или выгрузки. По окончании составления плана система предоставляет возможность сохранить и распечатать списки контейнеров, запланированные на выгрузку или погрузку, для передачи их сотрудникам, ответственным за проведение грузовых работ.

#### **Нерешенные проблемы автоматизации железнодорожной деятельности контейнерного терминала**

Проведенное исследование современных разработок в области автоматизации железнодорожной работы контейнерного терминала позволяет говорить о динамичном развитии данного направления. Существует ряд информационных систем [9, 10], успешно применяемых на предприятиях, которые позволяют автоматизировать процесс сбора и учета информации о перевозимых контейнерах, грузах, находящихся в контейнерах, собственниках подвижного состава и т. д. Современные информационные системы позволяют учитывать и отслеживать время выполнения основных технологических процессов: начало/окончание погрузки и выгрузки контейнеров, время подачи и уборки железнодорожных составов на подъездные пути.

Однако современные системы автоматизации железнодорожной деятельности не учитывают оперативную обстановку на терминале и погрузочных площадях при организации управления подачей и уборкой железнодорожных платформ, крайне редко принимается во внимание влияние множества внешних факторов на формирование последовательности выполняемых работ на терминале и сортировочной станции. Например, при подаче вагонов на подъездные пути не учитываются последующие затраты на выгрузку контейнеров и их перемещение по местам хранения. Контейнеры необходимо размещать в складских секциях терминала, исходя

из их направления движения, и таким образом, чтобы сократить расстояние при перемещении контейнера с подъездного пути до секции хранения. Учет оперативной обстановки на терминале, например график работы техники, приписанной к обработке определенных подъездных путей и складских секций, мог бы повысить эффективность технологических операций в процессе выгрузки и погрузки контейнеров. Еще одна нерешенная проблема при организации погрузочных работ – излишнее количество перемещений контейнеров внутри одной или нескольких складских секций. Нередко для погрузки одного контейнера на железнодорожную платформу необходимо переместить несколько других контейнеров, поскольку последние находятся над контейнером, который необходимо погрузить. Вполне возможно, что перемещаемые контейнеры впоследствии также будут погружены на железнодорожный транспорт в рамках одного перегрузочного процесса. Однако отсутствие в системах автоматизации планирования, учитывающего большой перечень факторов, не позволяет полноценно оптимизировать процесс погрузки контейнеров на железнодорожный транспорт.

#### **Перспективные пути решения задач управления железнодорожной деятельностью контейнерного терминала**

Для исследования возможных путей автоматизации железнодорожной составляющей деятельности контейнерного терминала необходимо рассмотреть грузовые процессы, протекающие на железнодорожном транспорте контейнерного терминала в условиях организации управленческих процессов, влияющих на ход выполнения погрузочно-разгрузочных работ. Цель проводимого исследования – разработка методов автоматизации процесса управления грузовыми работами на железнодорожном транспорте контейнерного терминала. В результате будет получена адекватная модель предметной области, максимально

приближенная к реальным представлениям экспертов; сформулирована задача планирования погрузочно-разгрузочных работ в терминах разработанной модели; предложены методы решения задач автоматизации процесса управления отгрузкой контейнеров на железнодорожный транспорт; спроектировано и реализовано программное средство, помогающее специалистам грамотно планировать работу железнодорожного отдела контейнерного терминала.

Исследование проблемы автоматизации железнодорожной деятельности контейнерного терминала сопряжено с решением следующих задач:

- анализ технологических процессов, происходящих на контейнерном терминале в процессе грузовых работ на железнодорожном транспорте контейнерного терминала;
- формальное определение специфических состояний и процессов, происходящих на контейнерном терминале в рамках деятельности по выгрузке и погрузке контейнеров на железнодорожный транспорт;
- постановка задачи расстановки подвижного состава по подъездным путям с учетом оптимизации процесса выгрузки контейнеров;
- формализация задачи комплектования контейнеров по железнодорожным вагонам с учетом критериев совместной отправки контейнеров на одной платформе;
- формализация задачи планирования процесса выгрузки и погрузки контейнеров на железнодорожный транспорт с оптимизацией использования погрузочной техники и минимизации количества перестановок контейнеров.

Применение результатов исследования будет способствовать сокращению времени обработки железнодорожных составов

в процессе выгрузки и погрузки контейнеров, позволит более эффективно использовать погрузочную технику, уменьшив пробег техники по контейнерному терминалу. Реализация метода решения задачи планирования процесса погрузки контейнеров даст возможность уменьшить число излишних перестановок контейнеров в складских секциях терминала, что позволит сократить время погрузки контейнера на железнодорожный вагон, уменьшить себестоимость процедуры погрузки контейнера, уменьшить износ техники, связанный с лишними операциями перемещения контейнеров внутри секций.

Метод размещения подвижного состава по подъездным путям контейнерного терминала позволит минимизировать временные и технические затраты на выгрузку контейнеров.

Метод формирования комплекта контейнеров для совместной отгрузки на железнодорожный вагон даст возможность снизить влияние человеческого фактора в процессе комплектации перечня контейнеров, предназначенных для отгрузки.

Метод планирования процесса отгрузки контейнеров позволит минимизировать количество лишних перемещений контейнеров по складским секциям.

Программный комплекс, предназначенный для поддержки принятия решений персоналом контейнерного терминала, позволит эффективно организовать процесс погрузки и выгрузки контейнеров с железнодорожного транспорта.

Разработанные методы и программные решения могут быть интегрированы в существующую систему управления контейнерным терминалом ИС ВКТ, используемую на контейнерном терминале ВКТ во Владивостокском морском торговом порту.

#### СПИСОК ЛИТЕРАТУРЫ

1. Организация контейнерных перевозок грузов. Транспортно-грузовые системы [Элек-

тронный ресурс] / Режим доступа: <http://www.tgs-jd.ru/8-transportno-gruzovye-kompleksy-dlya->

kontejnerov/8-2-organizaciya-kontejneryx-perevo-zok-gruzov/

2. Объемы железнодорожных контейнерных перевозок по России [Электронный ресурс] / Электронный журнал «Эксперт». – Режим доступа: <http://expert.ru/expert/2012/23/soobrazili-na-dvoih/media/145174/>

3. Официальный сайт ОАО «ВМТП» [Электронный ресурс] / Режим доступа: <http://www.vmtp.ru>

4. **Кожанов, Е.М.** Методика расчета текущего плана пропуска грузовых поездов по однопутному участку на основе математического моделирования [Текст] / Е.М. Кожанов // Транспорт. Наука, техника, управление: науч. инф. сб. – М.: ВИНТИ, 2008. – № 12. – С. 5–10.

5. **Кузнецов, А.П.** Грузопотоки на транспортных сетях (анализ, прогнозирование, управление) [Текст] / А.П. Кузнецов. – СПб.: ИПТ РАН, 2000. – 126 с.

6. **Ломаш, Д.А.** Мультиагентный метод

оптимизации подвода составов к припортовой-станции при конфликтующих целях [Текст] / Д.А. Ломаш // Тр. Всерос. науч.-практ. конф. Транспорт–2004. – Ростов н/Д: РГУПС, 2004.

7. **Поспелов, А.М.** К вопросу об управлении запасами парка контейнеров на контейнерных терминалах ОАО «РЖД» [Текст] / А.М. Поспелов // Тр. науч.-практ. конф. Неделя Науки–2007. – МГУПС, 2007. – 4.2: Наука МИИТа транспорту. – С. VI-18–VI-19.

8. Port IT & Security Solutions. Container Terminal Management System. [Электронный ресурс] / Режим доступа: <http://www.portek.com/ctms.html>

9. Railman – железнодорожный метроном [Текст] / РЖД-Партнер. – 2008. – № 3 (127).

10. CargoPrime: Контейнерный терминал. Управление обработкой железнодорожного транспорта. [Электронный ресурс] / Режим доступа: <http://www.cargoprime.ru/cargoprime/cpkt/functions/03.php>

## REFERENCES

1. Organizatsiia konteynerykh perevozkov gruzov. Transportno-gruzovye sistemy. Available <http://www.tgs-jd.ru/8-transportno-gruzovye-kompleksy-dlya-kontejnerov/8-2-organizaciya-kontejneryx-perevozkov-gruzov/>

2. Ob"emy zheleznodorozhnykh konteynerykh perevozkov po Rossii; elektronnyi zhurnal «Ekspert». Available <http://expert.ru/expert/2012/23/soobrazili-na-dvoih/media/145174/>

3. Ofitsial'nyi sait ОАО ВМТП. Available <http://www.vmtp.ru>

4. **Kozhanov E.M.** Metodika rascheta tekushchego plana propuska gruzovykh poezdov po odnopushtnomu uchastku na osnove matematicheskogo modelirovaniia / Transport. Nauka, tekhnika, upravlenie: nauch. inf. sb. – Moscow: VINITI, 2008. – №12. – S. 5–10. (rus)

5. **Kuznetsov A.P.** Gruzopotoki na transportnykh setiakh (analiz, prognozirovanie, upravlenie). – St. Petersburg: IPT RAN, 2000. – 126 s. (rus)

6. **Lomash D.A.** Mul'tiagentnyi metod optimizatsii podvoda sostavov k priportovoistantsii pri konfliktuiushchikh tseliakh / Tr. Vseros. nauch.-prakt. konf. Transport–2004. –Rostov n/D: RGUPS, 2004. (rus)

7. **Pospelov A.M.** K voprosu ob upravlenii zapasami parka konteynerov na konteynerykh terminalakh ОАО «RZhD» / Tr. nauch.-prakt. konf. Nedelia Nauki-2007. –Moscow: MGUPS, 2007; 4.2: Nauka MIITa transportu. –С. VI-18–VI-19. (rus)

8. Port IT & Security Solutions. Container Terminal Management System. Available <http://www.portek.com/ctms.html>

9. Railman – zheleznodorozhnyi metronome / RZhD-Partner. –2008. –№ 3 (127). (rus)

10. SargoRrime: Konteyneryni terminal. Upravlenie obrabotkoi zheleznodorozhnogo transporta. Available <http://www.cargoprime.ru/cargoprime/cpkt/functions/03.php>

**АНТОНОВА Елена Ивановна** – доцент кафедры ПММУиПО Дальневосточного федерального университета, кандидат технических наук.

690950, Россия, г. Владивосток, ул. Суханова, д. 8.

E-mail: [ant\\_vladivostok@mail.ru](mailto:ant_vladivostok@mail.ru)

**ANTONOVA, Elena I.** – Far Eastern Federal University.

690950, Sukhanova Str. 8, Vladivostok, Russia.

E-mail: [ant\\_vladivostok@mail.ru](mailto:ant_vladivostok@mail.ru)

**ВАСИЛЬЕВ Илья Александрович** — *соискатель ученой степени кандидата наук Дальневосточного федерального университета.*

690065, Россия, г. Владивосток, ул. Леонова, д. 66.

E-mail: vasyhin@list.ru

**VASILEV, Ilya A.** — *Far Eastern Federal University.*

690065, Leonova Str. 66, Vladivostok, Russia.

E-mail: vasyhin@list.ru



# Информационные и телекоммуникационные технологии в образовании

УДК 007:681.512.2

*И.Ю. Каширин, Р.Е. Медведев*

## **ОНТОЛОГИЧЕСКОЕ НАКОПЛЕНИЕ УЧЕБНЫХ ЗНАНИЙ НА ОСНОВЕ СЕРВИСА ИНФОРМАЦИОННЫХ СЕТЕЙ**

*I.Yu. Kashirin, R.E. Medvedev*

### **ONTOLOGICAL ACCUMULATION OF EDUCATIONAL KNOWLEDGE BASED ON THE SERVICE INFORMATION NETWORKS**

---

Предложена онтологическая модель представления знаний для их накопления в системах дистанционного обучения. При сборе информации использованы WSDL и OWL-S технологии описания сервисов в глобальных информационных сетях. Полученная технология стала эффективной при практическом использовании в качестве подсистемы Internet-портала общего образования.

**ОНТОЛОГИЧЕСКАЯ МОДЕЛЬ. INTERNET-СЕРВИС. НАКОПЛЕНИЕ ЗНАНИЙ. ДИСТАНЦИОННОЕ ОБУЧЕНИЕ. ДЕСКРИПТИВНАЯ ЛОГИКА.**

This article discusses an ontological model of knowledge representation for their accumulation in distance learning systems. When collecting information used WSDL and OWL-S technology service descriptions in global information networks. The resulting technology has been effective in practical use as a subsystem of Internet-portal of general education.

**ONTOLOGICAL MODEL. INTERNET-SERVICE. ACCUMULATION OF KNOWLEDGES. DISTANCE LEARNING. DESCRIPTION LOGICS.**

---

Технологии, использующие в качестве одного из инструментальных средств Web-сервисы, завоевывают свою популярность за счет возможности объединения знаний специалистов одних и тех же или родственных областей знаний. Такие сервисы могут использоваться и в системах дистанционного обучения.

Возможность автоматизированного или даже автоматического накопления знаний в заданной тематической области с целью структуризации, интеграции знаний и их повторного использования в последующем обучении становится очередной задачей, требующей применения новых подходов, в частности, концепции Semantic Web [1, 5, 7].

Информационные ресурсы, содержащие необходимые знания, могут представлять собой не только интернет-сайты, но и порталы, использующие инструментарий данных CMS (Content Management System) [3], а также хранилища информации, оформленные в качестве удаленных информационных банков, доступных посредством специальных интернет-сервисов.

Информация на таких ресурсах оформлена в виде готовых единиц учебного материала (курсов, вопросов, заданий, медиа-файлов и т. д.) представленных в цифровом формате. Каждый информационный ресурс располагает своей моделью представления данных, тем самым усложняя процесс ин-

теграции накопленных знаний в информационную базу существующей системы дистанционного обучения. Решение задачи интеграции знаний целесообразно возложить на интернет-сервисы, являющиеся программными средствами, дающими возможность использования удаленных ресурсов в том виде, в котором они необходимы пользователю. Сервисы, как правило, содержат мета-описание соответствующей области знаний, которое может использоваться для автоматизированного поиска необходимой информации, а также ее накопления с целью повторного использования.

В этой связи возникает актуальная задача предварительного описания предметных областей обучения и соответствующих динамических интернет-сервисов средствами онтологических моделей для представления знаний.

Для проектирования таких моделей в настоящее время применяются языковые

средства XML, RDF, WSDL, OWL-S. Разработка новых технологий, использующих перечисленные языки, даст возможность существенно повысить надежность и эффективность механизмов накопления знаний для их применения в дистанционном обучении.

Цель работы – получение оригинальной технологии накопления знаний в системах дистанционного обучения на основе онтологической модели представления знаний, с применением инструментария OWL-S и WSDL для повышения эффективности использования Internet-ресурсов.

**Теоретические исследования.** Онтологическая модель знаний для систем дистанционного обучения предполагает проектирование такой прикладной онтологии, которая была бы пригодной для адекватного описания предметных областей какой-либо отрасли знаний, например, программирования. Требованием адекватности является полное соответствие модели знаний основ-

Таблица 1

Семантика дескриптивной логики SHOIN(D)

ППФ	Пример	Семантика	Наименование
C	Problem	$C(x)$	Именованный концепт
$E_1 \cap E_2$	System $\cap$ Instrument	$E_1(x) \& E_2(x)$	Конъюнкция
$E_1 \cup E_2$	Problem $\cap$ Subsection	$E_1(x) \vee E_2(x)$	Дизъюнкция
$\neg C$	$\neg$ SolvedProblem	$\neg C(x)$	Отрицание
$\exists R.F$	$\exists$ isSolved.Problem	$\exists y (R(x,y) \& F(y))$	Квантификация существования
$\forall R.F$	$\forall$ hasElement. Subsection	$\forall y (R(x,y) \rightarrow F(y))$	Квантификация всеобщности
$\forall R.F$	$\forall$ hasElement $\neg$ . Problem	$\forall y (R(y,x) \rightarrow F(y))$	Обратная квантификация всеобщности
$\exists R.F$	$\forall$ isApplied $\neg$ .SolvedProblem	$\exists y (R(y,x) \& F(y))$	Обратная квантификация существования
$\geq n.R$	$\geq 2$ .hasSubsection	$\exists \geq 2y R(x,y)$	Нижнее ограничение
$\leq n.R$	$\leq 12$ .hasSubproblem	$\exists \leq 12y R(x,y)$	Нижнее ограничение
{o}	{ класс «стек» }	$x = o$	Индивид
{v}	{ 30 }	$x = v$	Значение
$\top$	$\exists$ isSolved	Все индивиды	Истина
$\perp$	$\forall$ hasLearned. $\perp$	Пусто	Ложь
a:F	Array_B: Array	Индивид	Представитель

ным технологическим принципам технологии. В нашем случае выбирается задачный подход, в котором структура подразделов изучаемой предметной области является изоморфной структуре тестовых примеров, задаваемых для контроля усвоения знаний.

Исходя из сказанного, спроектируем прикладную онтологию, в рамках которой можно было бы сформулировать ключевую фразу задачного подхода: «для решения задачи студент должен усвоить все понятия, которые необходимы при ее решении».

Для описания онтологии предметной области обучения студентов на множестве тестовых задач [2] воспользуемся концепцией дескриптивной логики SHOIN(D), семантика которой может быть задана табл. 1 [3], где обозначение ППФ читается как «правильно построенные формулы».

В качестве общей онтологии как системы предельно абстрактных понятий примем общую ICF-онтологию, описанную в [4]. Эта онтология графически представлена на рис. 1. В прикладной онтологической системе будем использовать следующие концепты: «Система», «Элемент», «Причина», «Следствие», «Инструмент», «Актор (Действующее лицо)», «Динамика», «Процесс».

Для конструктивной работы с формулами дескриптивной логики необходимо определить два раздела выражений: TBox и ABox [5], описывающие абстрактную и конкретную составляющие прикладной онтологии.

Разработаем основные формулы концептуального раздела дескриптивной логики TBox, адекватные цели получения абстрактной части базы знаний «Обучение». Приведем далее их описания в русской транскрипции.

Система =  $\forall$ содержатьЭлемент.Элемент,  
что соответствует концептуальному описанию «понятие Система в качестве всех своих составляющих содержит Элементы».

Причина =  $\exists$ иметьСледствием.Следствие,  
что аналогично предыдущему «Причина имеет хоть какое-нибудь следствие».

Инструмент =  $\exists$ ИметьАктора.Человек  $\cap$

$\exists$ Применим  $\neg$ .Задача

соответствует высказыванию «Инструмент в качестве использующего его имеет хоть какого-то Человека, этот Инструмент применим для решения хоть какой-то задачи».

Решивший =  $\exists$ ужеРешивший.Задача  $\cap$   
Студент

соответствует тому, что «Решившим является Студент, у которого существует в активе уже решенная задача».

РешеннаяЗадача =  $\exists$ ужеРешивший  $\neg$ .  
Задача  $\cup$   $\exists$ ужеРешивший.Решивший,

НеРешеннаяЗадача =  $\exists$ ужеРешивший  $\neg$ . $\perp$

представляют собой соответственно «Решенная задача имеет хотя бы одного решившего ее» и «Нерешенная задача в качестве решивших ее студентов имеет пустое множество».

Задача = РешеннаяЗадача  $\cup$  НеРешенная-  
Задача

трактуются как «Задача является представителем объединения множеств решенных и нерешенных задач».

Учебная Дисциплина  $\subseteq$  Система  $\cap$  Ин-  
струмент,

что по аналогии с предыдущим выражением означает «Учебная дисциплина является подмножеством Инструментов, рассматриваемых одновременно как сложные Системы».

Учебная Дисциплина =  $\forall$ содержатьЭле-  
мент.Подраздел

Подраздел  $\subseteq$  Учебная Дисциплина

Задача  $\subseteq$  Учебная Дисциплина

Эти выражения соответствуют толкованиям: «Учебная дисциплина содержит подразделы», «Подраздел входит в учебную дисциплину», «Задача входит в учебную дисциплину».

Подраздел =  $\forall$ содержатьЭлемент.Задача  $\cap$   
 $\forall$ содержатьЭлемент.Понятие

соответствует упрощенному пониманию того, что «Подраздел в качестве элементов

Таблица 2

Нумерация формул прикладной онтологии обучения

Понятие / Концепт	Формула
System	$= \forall \text{hasElement. Element}$
Causa	$= \exists \text{hasConsequence. Consequence}$
Instrument	$= \exists \text{hasActor. Person} \cap \exists \text{isApplied} \neg . \text{Problem}$
Solver	$= \exists \text{isSolved. Problem} \cap \text{Student}$
SolvedProblem	$= \exists \text{isSolved} \neg . \text{Problem} \cup \exists \text{isSolved. Solver}$
NotSolvedProblem	$= \exists \text{hasSolver.} \perp$
Problem	$= \exists \text{isSolved} \neg . \text{Problem} \cup \exists \text{isSolved. Solver}$
SubjectMatter	$\subseteq \text{System} \cap \text{Instrument}$
SubjectMatter	$= \forall \text{hasElement. Subsection}$
Subsection	$\subseteq \text{SubjectMatter}$
Problem	$\subseteq \text{SubjectMatter}$
Subsection	$= \forall \text{hasElement. Problem} \cap \forall \text{hasElement. Concept}$
Learned Subsection	$= \text{Subsection} \cap (\exists \text{hasLearned} \neg . \text{Subsection} \cup \exists \text{hasLearned. Student})$
Un Learned Subsection	$= \text{Subsection} \cap \exists \text{hasLearned.} \perp$
Concept	$\subseteq \text{Instrument}$
Student	$\subseteq \text{Person}$
Process	$= \text{Dynamic}$
Learning	$\subseteq \text{Process} \cap \exists \text{hasConsequence. SolvedProblem}$
Learn_the_Concept	$= \exists \text{hasLearn. Concept}$
ProblemConcepts	$= \text{Concept} \cap \forall \text{isApplied} \neg . \text{SolvedProblem}$
LearnProblemConcepts	$= \text{Learning} \cap \forall \text{hasLearn} \neg . \text{ProblemConcepts}$

содержит только Понятия и Задачи».

УсвоенныйПодраздел = Подраздел  $\cap$   
(  $\exists$ ужеУсвоивший $\neg$ .Подраздел  $\cup$   $\exists$ ужеУсвоивший.Студент )

НеУсвоенныйПодраздел = Подраздел  $\cap$   
ужеУсвоивший.  $\perp$

соответственно «Усвоенный подраздел относится к подразделам, для которых есть студенты, которые усвоили этот подраздел» и «Неусвоенный подраздел не имеет усвоивших его».

Студент  $\subseteq$  Человек

Понятие  $\subseteq$  Инструмент

Процесс = Динамика

— выражения, соответствующие базовым элементам родовидовой таксономии [5], сопрягаемой с общей онтологией.

Обучение  $\subseteq$  Процесс  $\cap$   $\exists$ иметьСледствием. РешеннаяЗадача

соответствует формуле «Обучение входит в Процесс, который имеет следствием решение хоть одной задачи».

Усвоение\_понятия =  $\exists$ Усвоить.Понятие

Понятия Задачи = Понятие  $\cap$   $\forall$ Применим $\neg$ . РешеннаяЗадача

УсвоениеПонятийЗадачи = Обучение  $\cap$   
 $\forall$ Усвоить Понятие.Понятия Задачи

соответственно: «Усвоение понятия значит, что хотя бы одно понятие усвоено», «Понятия одной задачи — это Понятия, применимые в решении этой задачи», «Усвоение понятия из какой-то задачи — это процесс Обучения, в результате которого усваиваются все понятия этой задачи».

Далее для нумерации формул и графи-

Таблица 3

Раздел ABox прикладной онтологии обучения

Индивидуальная константа	Концепт (формула) онтологии
Интерфейс с TBox	
Student_122	$\subseteq$ Student
Subsection_74	$\subseteq$ Subsection
Problem_74	$\subseteq$ Problem
Факты ABox	
«Васильев П.В.»	: Student_122
«Производные типы данных. Массивы»	: Subsection_74
«Цикл с параметром»	: Concept_74
«Оператор присваивания»	: Concept_74
«Арифметические выражения»	: Concept_74
«Структура программы»	: Concept_74
«Операции ввода и вывода»	: Concept_74
«Массив Одномерный»	: Concept_74
«Массив Двумерный»	: Concept_74
«Массив Многомерный»	: Concept_74
«Квадратная матрица»	: Concept_74
«Ввод Вывод Массива»	: Problem_74
«Транспонирование элементов»	: Problem_74
«Вычисление суммы главной диагонали»	: Problem_74

ческого представления онтологии введем ее англоязычный эквивалент (табл. 2).

Разработаем вторую часть прикладной онтологии ABox [5, 6], предназначенную для описания конкретных знаний. В случае с предметной областью «Обучение» можно выбрать учебный материал и задачи дисциплины «Алгоритмизация и языки программирования» из раздела «Производные типы данных. Массивы». Приведем далее лишь один из кратких фрагментов ABox, в то же время достаточный для демонстрации основных возможностей описания разделов дисциплин, а также оперирования составляющих таких разделов (табл. 3).

**Экспериментальные исследования.** Алгоритм унификации OWL-выражений для случая полиморфических термов был получен в 2008 г. и описан в [4]. Задача использования этого алгоритма в качестве одного из сервисов накопления знаний в качестве учебных материалов может быть решена с помощью WSDL-описания или OWL-S-описания сервиса для предметной области

«Обучение».

Далее приведем WSDL-описание, для обработки которого используется впоследствии инструментарий SOAP-протокола.

Пространство имен WSDL использует раздел <wSDL:definitions>, содержащий следующие производные подразделы для описания сервиса накопления знаний:

<wSDL:import> – ссылка на документ WSDL с описаниями, подлежащими включению в этот документ;

<wSDL:types> – определяет типы XML или элементы, используемые для обмена сообщениями;

<wSDL:message> – определяет фактическое сообщение;

<wSDL:portType> – определяет абстрактный набор операций, осуществленных сервисом;

<wSDL:binding> – определяет реализацию <wSDL:portType> с помощью конкретных протоколов и форматов;

<wSDL:service> – определяет сервис в целом, как правило, включая один или не-

сколько элементов `<wsdl:port>` с информацией доступа для элементов `<wsdl:binding>`.

В описании задействован еще элемент `<wsdl:documentation>`, который используется в целях документирования сервиса.

```
<wsdl:definitions ... xmlns:tns = «http://EduAccumulation.ru/SubSections /SearchServer»
```

```
targetNamespace= " http://EduAccumulation.ru/SubSections /SearchServer "
```

```
<wsdl:document> Описание сервиса выбора учебного материала.
```

```
</wsdl:document> <wsdl:types>
```

```
<xs:schema ...
```

```
targetNamespace= « http://EduAccumulation.ru/SubSections /SearchServer "
```

```
<xs:import namespace= " http://EduAccumulation.ru/SubSections /SearchServer /Types"
```

```
schemaLocation="PapersTypes.xsd"/>
```

```
</xs:schema>
```

```
</wsdl:types>
```

```
<wsdl:message name="getSubsectionMessage">
```

```
<wsdl:part name="part" element="tns:getSubsection"/>
```

```
</wsdl:message>
```

```
<wsdl:message name="getSubsectionResponseMessage">
```

```
<wsdl:part name="part" element="tns:getSubsectionResponse"/>
```

```
</wsdl:message>
```

```
<wsdl:message name="addSubsectionMessage">
```

```
<wsdl:part name="part" element="tns:addSubsection"/>
```

```
</wsdl:message>
```

```
<wsdl:message name="addSubsectionResponseMessage">
```

```
<wsdl:part name="part" element="tns:addSubsectionResponse"/>
```

```
</wsdl:message>
```

```
<wsdl:message name="addDuplicateFault">
```

```
<wsdl:part name="fault" element="tns:addDuplicate"/>
```

```
</wsdl:message>
```

```
<wsdl:portType name="SubsectionServerPortType">
```

```
<wsdl:documentation>
```

Далее описан сервис, реализующий накопление материалов задач на специализи-

рованном сервере.

```
</wsdl:documentation>
```

```
<wsdl:operation name="getSubsectionProblem">
```

```
<wsdl:documentation>
```

Поиск материала задач с необходимым описанием.

```
</wsdl:documentation>
```

```
<wsdl:input message="tns: SubsectionProblem Message"/>
```

```
<wsdl:output message="tns:getSubsectionProblemeMessage"/>
```

```
</wsdl:operation>
```

```
<wsdl:operation name="addSubsectionProblem">
```

```
<wsdl:documentation>Добавление задачи.</wsdl:documentation>
```

```
<wsdl:input message="tns:addSubsectionProblemMessage"/>
```

```
<wsdl:output message="tns:addSubsectionProblemResponseMessage"/>
```

```
<wsdl:fault
```

```
message="tns:addDuplicateFault" name="addDuplicateFault"/>
```

```
</wsdl:operation>
```

```
</wsdl:portType>
```

```
</wsdl:definitions>
```

Здесь в разделе `<wsdl:definitions>` определяется пространство имен описанной ранее онтологии. Аналогичным образом может быть описан сервис накопления учебного материала подразделов дисциплин, соответствующих найденным задачам. Кроме того, предполагается, что реализация сервиса включает поисковый алгоритм, рассмотренный в [4].

Приведем более эффективное сокращенное описание сервиса накопления учебных материалов с помощью языка OWL-S:

```
<service:Service rdf:ID="Intell Subsection FinderService">
```

```
<service:presents rdf:resource="#Intell Subsection FinderProfile"/>
```

```
<service:describedBy rdf:resource="#Intell Subsection FinderProcess"/>
```

```
<service:supports rdf:resource="#Intell Subsection FinderGrounding"/>
```

```
</service:Service>
```

Профиль сервиса (входные и выходные данные) описываются, например, так:

```
<profile:Profile rdf:ID="Intell Subsection
```

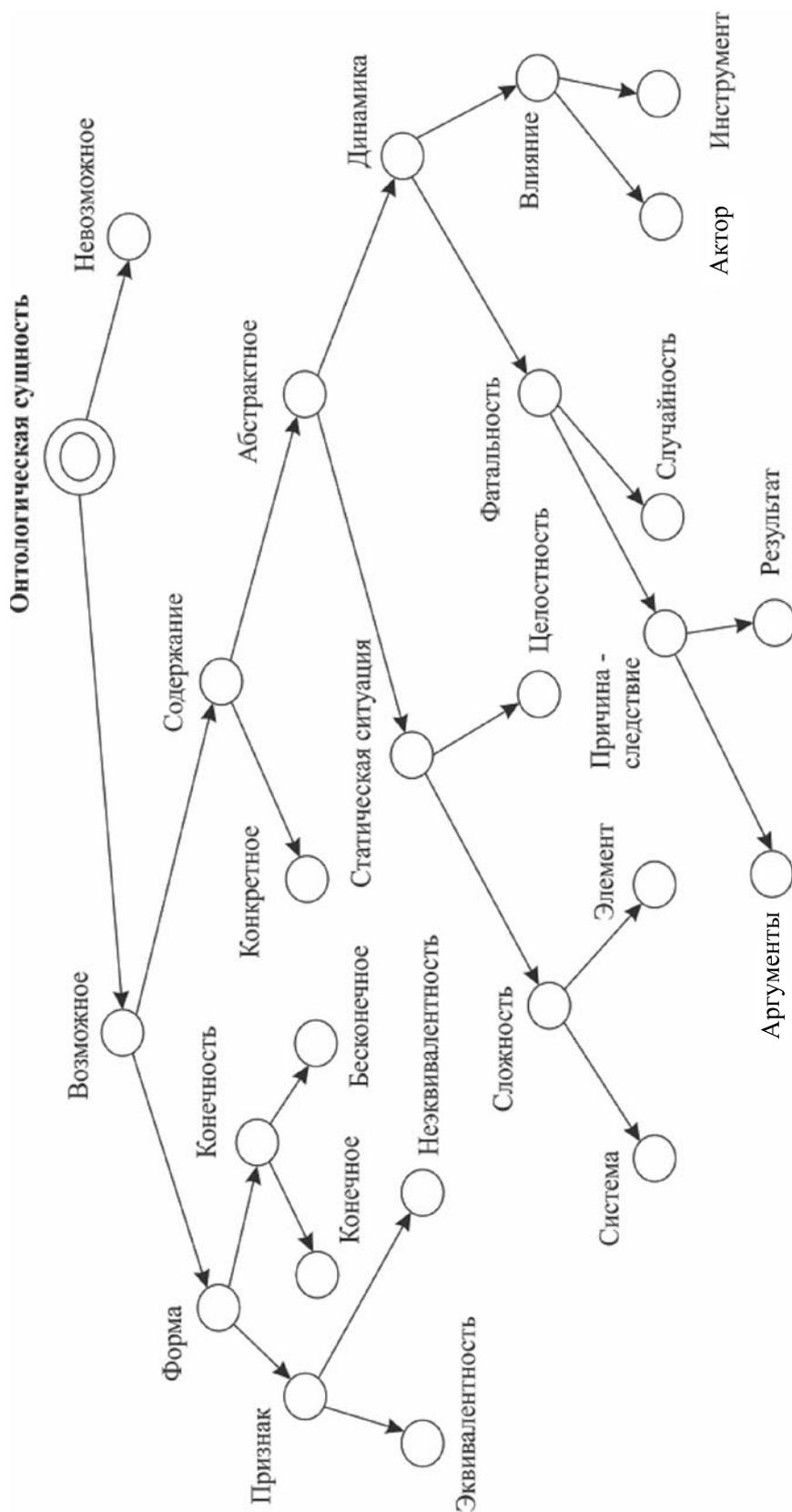


Рис. 1. Общая онтология системы ICF

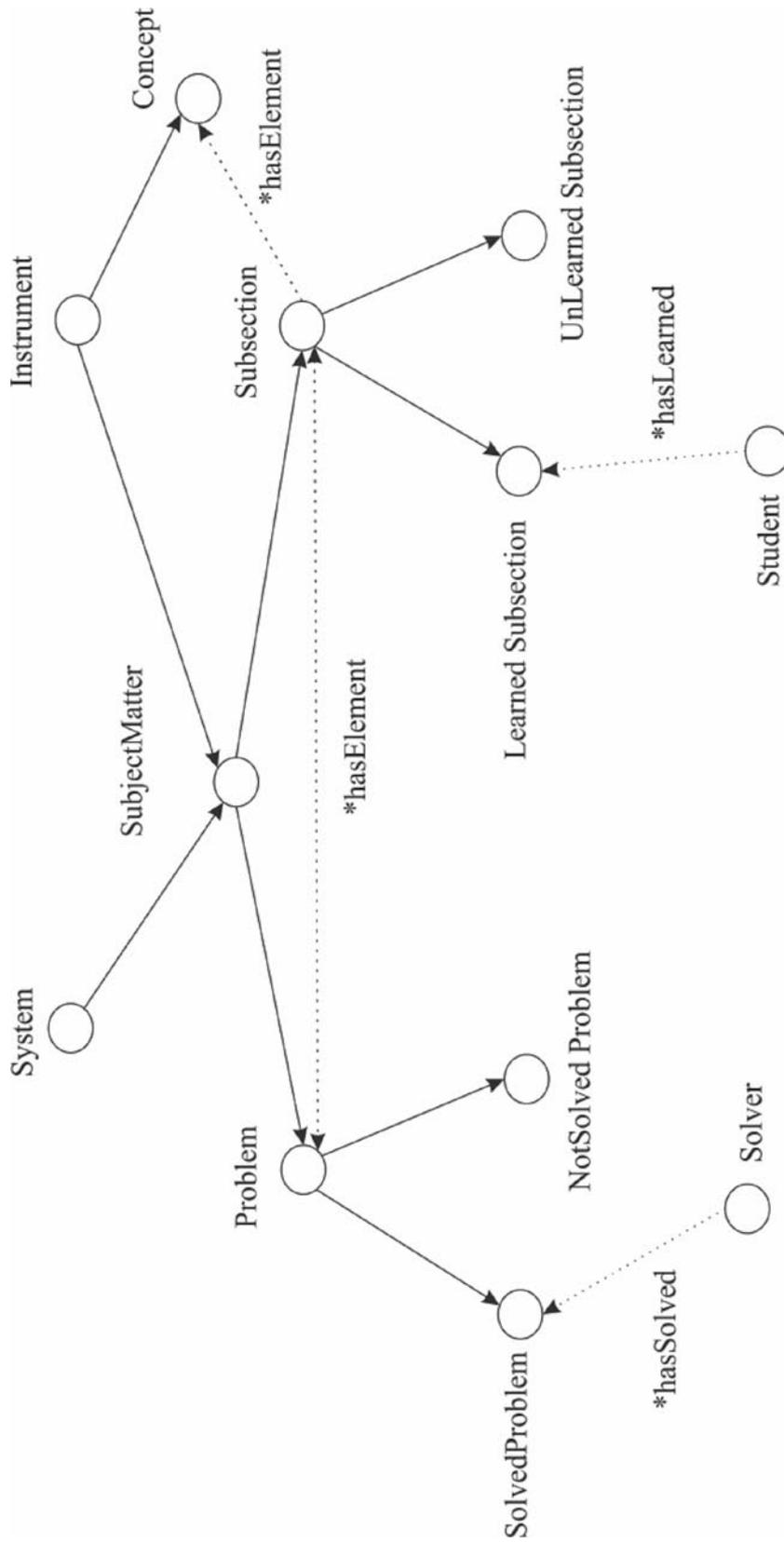


Рис. 2. Общая семантическая сеть фрагмента онтологии «Problem, Subsection»

```

FinderProfile»»
  <service:presentedBy rdf:resource=#Intell
Subsection FinderService»/>
  <profile:serviceName xml:lang=#en»>Intell
Subsection
                                Finder</profile:
serviceName>
  <profile:textDescription xml:lang=#en»>
Сервис будет возвращать информацию
о материале, заголовок которого соответ-
ствует строке, описывающей задачу, и за-
данной через Intell-сервис.
  </profile:textDescription>
  <profile:hasInput rdf:resource=# Subsec-
tion Name»/>
  <profile:hasOutput  rdf:resource=#Item-
Info»/>
  </profile:Profile>
Процесс передачи информации будет
выглядеть следующим образом:
  <process:AtomicProcess  rdf:ID=#Intell
Subsection FinderProcess»>
  <service:describes rdf:resource=#Intell
Subsection FinderService»/>
  <process:hasInput  rdf:resource=##
Subsection Name»/>
                                <process:hasOutput
rdf:resource=##ItemInfo»/>
  </process:AtomicProcess>
Теперь описание входных/выходных па-
раметров процесса накопления будет опи-
сано следующим образом:
  <process:Input  rdf:ID=# Subsection
Name»>
  <process:parameterType rdf:datatype=#&
xsd:#anyURI»>&xsd:#string>
  </process:parameterType>

```

```

<rdfs:label> Subsection Name</rdfs:label>
</process:Input>
<process:Output rdf:ID=#ItemInfo»>
  <process:parameterType rdf:datatype=#&
store;#StockItem>
  </process:parameterType>
  <rdfs:label>Item Info</rdfs:label>
</process:Output>

```

Разработанная прикладная онтология и полученные описания Web-сервисов в нотации языковых средств WSDL и OWL-S описания дают возможность использовать новую технологию для автоматического сбора и накопления знаний для современных дистанционных систем обучения.

Структура предложенной онтологии позволяет повторно использовать существующие единицы учебного материала. Более детальное описание разработанной прикладной онтологии, выходящее за рамки данной статьи, позволяет говорить о возможности выделения информационных единиц, необходимых для построения двух основных моделей, по которым строится индивидуальная траектория обучения: модели предметной области и модели обучаемого.

Практика применения предложенной технологии с соответствующим описанием сервиса показала свою эффективность при проектировании сервисов портала системы общего образования. Достигнуто повышение релевантности искомым обучающих материалов примерно на 3,7 %.

Интернет-ресурс внедрен на сервере управления образования городской администрации Рязани.

#### СПИСОК ЛИТЕРАТУРЫ

1. **Bruijn, J.** OWL DL vs. OWL Flight: Conceptual modelling and reasoning on the Semantic Web [Электронный ресурс] / J. Bruijn, A. Polleres, R. Lara [et al.] / In Proc. of the 14th International World Wide Web Conf. – WWW2005. – Chiba, Japan, 2005.
2. **Морозевич, А.И.** Стратегия автоматизации управления познавательной деятельностью на основе информационной модели образовательного процесса [Текст] / А.И. Морозевич, В.Н. Комличенко, В.В. Гедранович // Информационные технологии. – 2000. – № 5. – С. 47–52.
3. **Малых, А.А.** Объектно-ориентированная

- дескриптивная логика [Текст] / А.А. Малых, А.В. Манцивода // Изв. Иркутского гос. ун-та. – Иркутск, 2011. – С. 57–72.
4. **Каширин, Д.И.** Полиморфическое представление знаний в Semantic Web: Монография [Текст] / Д.И. Каширин, И.Ю. Каширин, А.Н. Пылькин. – М.: Горячая линия-Телеком, 2009. – 136 с.
5. **Рассел, С.** Искусственный интеллект: современный подход [Текст] / С. Рассел, П. Норvig. – М.: ИД «Вильямс», 2006. – 1408 с.
6. **Soergel, D.** Reengineering Thesauri for New Applications: the AGROVOC Example [Электрон-

ный ресурс] / D. Soergel, B. Lauser, A. Liang [et al.] // Article № 257. – 2004-03-17.

7. **Abdul-Ghafour, S.** A Common Design-Features Ontology for Product Data Semantics In-

teroperability [Электронный ресурс]/ S. Abdul-Ghafour, P. Ghodous, B. Shariat [et al.] // IEEE/WIC/ACM International Conf. on Web Intelligence – WI'07. – 2007. – P. 443–446.

#### REFERENCES

1. **Bruijn J., Polleres A., Lara R., Fensel D.** OWL DL vs. OWL Flight: Conceptual modelling and reasoning on the SemanticWeb / In Proc. of the 14th International World Wide Web Conf. (WWW2005). – Chiba, Japan, 2005.

2. **Morozevich A.I., Komlichenko V.N., Gedranovich V.V.** Strategy automation cognitive activity based information model of the educational process / Information technologies. – 2000. – № 5. – S. 47–52. (rus)

3. **Malih A.A., Mantsivoda A.V.** Object-oriented descriptive logic / News of Irkutsk State University. – Irkutsk, 2011. – S. 57–72. (rus)

4. **Kashirin D.I., Kashirin I.Y., Pylkin A.N.**

Polymorphic representation of knowledge in Semantic Web: Monograph. – Moscow: HotLine-Telecom, 2009. – 136 s. (rus)

5. **Rassel, S., Norvig, P.** Iskusstvennyy intellect: sovremennyy podhod. – Moscow: ID «Vil'yams», 2006. – 1408 s. (rus)

6. **Soergel D., Lauser B., Liang A. et al.** Reengineering Thesauri for New Applications: the AGRO-VOC Example; Article № 257 2004-03-17.

7. **Abdul-Ghafour S., Ghodous P., Shariat B. et al.** A Common Design-Features Ontology for Product Data Semantics Interoperability; IEEE/WIC/ACM International Conf. on Web Intelligence (WI'07), 2007. – P. 443–446.

---

**КАШИРИН Игорь Юрьевич** – профессор кафедры вычислительной и прикладной математики Рязанского государственного радиотехнического университета, доктор технических наук.

390005, Россия, г. Рязань, ул. Гагарина, д. 59/1.

E-mail: igor\_kashirin@mail.ru

**KASHIRIN, Igor Yu.** – *Ryazan State Radioengineering University.*

390005, Gagarina Str. 59/1, Ryazan, Russia.

E-mail: igor\_kashirin@mail.ru

**МЕДВЕДЕВ Роман Евгеньевич** – старший преподаватель кафедры вычислительной и прикладной математики Рязанского государственного радиотехнического университета.

390005, Россия, г. Рязань, ул. Гагарина, д. 59/1.

E-mail: roman\_medvedev@inbox.ru

**MEDVEDEV, Roman E.** – *Ryazan State Radioengineering University*

390005, Gagarina Str. 59/1, Ryazan, Russia.

E-mail: roman\_medvedev@inbox.ru

УДК 004.7

*Д.А. Иванченко, И.А. Хмельков, Д.Ю. Райчук,  
А.М. Митрофанов, А.В. Самочадин, П.А. Rogov*

## **ПРИМЕНЕНИЕ ПОДХОДОВ BYOD ДЛЯ ПОСТРОЕНИЯ СТРАТЕГИИ ИНФОРМАТИЗАЦИИ ВЫСШЕГО УЧЕБНОГО ЗАВЕДЕНИЯ**

*D.A. Ivanchenko, I.A. Khmelkov, D.Yu. Raychuk,  
A.M. Mitrofanov, A.V. Samochadin, P.A Rogov*

### **THE USE OF BYOD APPROACHES FOR IMPLEMENTATION OF UNIVERSITY IT-STRATEGY**

Представлены современные подходы к применению мобильных технологий и сервисов в системе высшего профессионального образования, раскрыто содержание BYOD в качестве компонента IT-стратегии высшего учебного заведения, выявлены технологические и организационные проблемы внедрения подходов BYOD в образовательный процесс.

ИНФОРМАТИЗАЦИЯ ОБРАЗОВАНИЯ. МОБИЛЬНЫЕ ТЕХНОЛОГИИ. BYOD. MOBILE DEVICE MANAGER.

The article presents the current approaches to the use of mobile technologies and services in the higher education system, disclosed the contents of BYOD as a component of IT-strategy for the higher education institution, identified technological and organizational problems of implementing BYOD approaches to the educational process.

INFORMATIZATION OF EDUCATION. MOBILE TECHNOLOGIES. BYOD. MOBILE DEVICE MANAGER. MDM.

Рынок мобильных решений – одно из самых перспективных и динамично развивающихся направлений современного бизнеса, что обуславливает большие возможности по развитию и применению новых технологий и сервисов для различных сфер деятельности.

Широкое применение средств мобильной связи и интернет-коммуникаций оказывает существенное влияние на различные сферы жизни общества и меняет формы общения, трудовой, образовательной и творческой деятельности, способы торговли, методы управления и др. Использование мобильных приложений и интерфейсов, считывание касаний и жестов, улучшенные поисковые технологии, управление голосом, богатые интерактивные и мультимедийные возможности, позволяют на качественно новом уровне удовлетворять информационные и коммуникационные потребности индивидов, определяют новые направления формирования инфор-

мационного пространства.

**Постановка задачи.** В статье предполагается изучить современное состояние рынка BYOD-решений и MDM-систем с целью определения перспективности данного направления автоматизации бизнес-процессов в контексте деятельности высшего образовательного учреждения.

#### **Обзор рыночной ситуации**

Сегодня можно утверждать, что в полной мере сбылся прогноз Г. Рейнгольда, который утверждал что «через десять лет основные места средоточия населения Земли будут наводнены микросхемами, способными общаться друг с другом <...> люди, оснащенные такими устройствами, составят «умные толпы», и их общение обретет невиданные прежде формы и возможности» [1].

Беспроводные и мобильные технологии все чаще используются сотрудниками предприятий и организаций в своей трудовой деятельности: многие работодатели предостав-

ляют персоналу специальные устройства или позволяют использовать собственные для удаленного доступа к корпоративным информационным ресурсам и сервисам. Такой подход получил название BYOD (Bring Your Own Device – «Принеси свое собственное устройство») и, по мнению ряда экспертов, своим появлением обязан системе высшего профессионального образования. Вчерашние университетские выпускники, привыкшие использовать собственные смартфоны, планшеты, ноутбуки и нетбуки во время учебы, столкнулись с необходимостью отказаться от привычного и комфортного образа деятельности из-за жестких корпоративных правил и ограничений систем управления, почти победивших «лоскутное одеяло автоматизации» и совершенно неготовых к еще более пестрому набору персональных устройств и мобильных сервисов.

В качестве концепции и корпоративной IT-политики BYOD впервые сформировалась в 2009 г. в корпорации Intel под влиянием нарастающего тренда на использование сотрудниками собственных мобильных устройств в рабочем процессе [2]. Окончательно термин BYOD укрепился уже в 2011 г. в результате совместной работы в данном направлении крупного североамериканского поставщика IT-решений – компании Unisys и таких известных производителей программного обеспечения, как VMware и Citrix.

Согласно опросу, проведенному аналитической компанией IDC (<http://www.idc.com>), в настоящий момент около 34 % компаний считает своей первоочередной технологической задачей реализацию доступа сотрудников к корпоративным приложениям с помощью мобильных устройств, которые позволяют пользователям осуществлять свою работу независимо от местонахождения [3].

По данным исследования Gartner ([www.gartner.com](http://www.gartner.com)), проведенного в октябре 2012 г., уже в 2017 г. более 65 % предприятий во всем мире перейдут на использование мобильных устройств в работе с корпоративными информационными системами. При этом 90 % предприятий будут поддерживать две и более мобильных операционные системы для работы своих сотрудников [4].

К сожалению, в настоящий момент на-

блюдается достаточно невысокий уровень проникновения мобильных приложений в корпоративную среду отечественных компаний среднего и крупного размера. Российским пользователям более понятна модель, при которой работодатель выдает им устройство, полностью контролируемое администраторами, поэтому в отечественных компаниях стремятся реализовать поддержку бизнес-модели GYOD (Give me your own device – «Дай мне свое устройство») [5]. Недостатком данного подхода нередко является снижение эффективности коммуникаций.

На наш взгляд, в России сегодня наиболее перспективными являются вопросы применения подходов BYOD при определении стратегии информатизации в высших учебных заведениях с целью использования персональных мобильных устройств (смартфонов, планшетов, ноутбуков, нетбуков, ультрабуков, электронных книг) учащихся, преподавателей, ученых и административного персонала для доступа к общесетевым и специализированным ресурсам и сервисам учебного заведения и Интернету.

Такой подход подразумевает внедрение в образовательное пространство вуза технологий, сервисов и политик безопасности, ориентированных на предоставление персонализированного доступа к информационным и справочным ресурсам посредством мобильных устройств (учебные планы и программы, отчеты об успеваемости и посещаемости, результаты сессии, учебная нагрузка, расписание учащегося и преподавателя и пр.); организацию распределенного онлайн доступа к контенту (подкаст-вещание; вебинары; электронные журналы; персональная библиотека образовательных и научных ресурсов; социальные медиа и пр.); обеспечение академической мобильности учащихся, преподавателей и ученых; использование мобильных устройств как средства идентификации и платежного инструмента; мобильное геопозиционирование и пр. [6]

Мировой опыт показывает, что мобильные сервисы уже несколько лет рассматриваются и используются в качестве одного из перспективных направлений развития информационных технологий в образовании, но BYOD как IT-политика существует

лишь у некоторых западных вузов. В качестве примера приведем сводные результаты проведенного нами анализа пяти ведущих университетов США (см. таблицу).

Успешность использования подходов BYOD в качестве компонента IT-стратегии высшего учебного заведения обусловлена, на наш взгляд, следующими предпосылками:

во-первых, высокий уровень и динамика распространения мобильных устройств в студенческой и преподавательской среде (не редкость, когда один пользователь является владельцем двух и более устройств) и устойчивый интерес к их применению, уже сформированный внешними социально-психологическими факторами;

во-вторых, существенный когнитивный потенциал аудитории высшего учебного заведения, гибко и адекватно реагирующей на изменения в устоявшейся практике организации образовательного процесса и легко адаптирующейся к использованию новых подходов и технологий;

в-третьих, учебные материалы сравнительно легко превратить в медиа-контент и содержание для интерактивных мобильных сервисов;

в-четвертых, мобильные сервисы и контент как технологически, так и методологически достаточно просто интегрируются в инфраструктуру образовательного и научно-исследовательского пространства.

### Сервисная модель

При построении IT-стратегии вуза наиболее целесообразным представляется применение сервисной модели – построения структурированного каталога сервисов, предоставляемых различным категориям пользователей (учащимся, профессорско-преподавательскому составу, административно-хозяйственному персоналу). Проиллюстрируем данный тезис примерным перечнем сервисов, ориентированных на студентов.

#### Общие сервисы:

мобильная версия портала учебного заведения;

сервис централизованного управления мобильными устройствами.

#### Образовательные сервисы:

приемная комиссия;

каталоги учебных планов и учебных курсов;

регистрация на курсы (семинары, от-

Информационные сервисы крупных вузов мира

	Мобильный сайт	Клиент под iPhone	Клиент под Android	Библиотечный каталог	Заказ книг онлайн	Каталог курсов	Регистрация на курсы	Платежная система	Адресная книга	Интерактивная карта	Общественное питание	Новости спорта	Экстренная информация	Расписание автобусов	Трекинг автобусов	Эксплуатационные службы	Медиа-архив	iTunes U	Онлайн-радио
Massachusetts Institute of Technology	x		x	x	x	x	x	x	x	x	x		x	x	x	x			
Princeton University		x	x	x		x			x	x	x	x		x					
Stanford University	x	x	x	x		x			x	x	x	x	x				x	x	x
Duke University	x	x	x	x	x	x		x	x	x	x	x					x	x	
Harvard University	x	x	x	x	x	x			x	x	x	x			x		x		

крытые лекции, факультативы и др.);  
онлайн расписание;  
персональные достижения, портфолио;  
проектная деятельность;  
резервирование помещений и оборудования для проектной работы.

*Коммуникационные сервисы:*

электронная почта;  
социальные медиа;  
аудио- и видеоконференцсвязь.

*Библиотечные сервисы:*

поиск по каталогу;  
онлайн заказ книг;  
доступ в медиатеку (библиотеку электронных изданий).

*Справочные сервисы:*

календарь событий;  
телефонный справочник;  
интерактивная карта, путеводитель по кампусу;

места общественного питания;  
места парковки, загрузка парковки;  
экстренная информация.

*Мобильный и интернет-банкинг:*

баланс персонального счета;  
платежи за услуги;  
отчеты о проделанных операциях.

Доступ пользователей к информационным сервисам может осуществляться посредством широкого класса устройств, включая мобильные платформы, такие, как iOS, Android, Windows Phone, Blackberry, Windows Mobile и др.

Таким образом, использование подходов BYOD в сочетании с принципами e-learning дает возможность организовать для студента своего рода мультиплатформенный мобильный образовательный офис, что позволит реализовывать более свободные формы учебной работы, новые варианты использования образовательного пространства вуза, решать современные творческие, педагогические и методические задачи:

- контролируемый доступ к информационным и образовательным ресурсам, подкаст-вещанию, вебинарам, социальным сетям в любом месте и в любое время;

- использование привычных устройств и интерфейсов – нет необходимости запоминать персональные настройки и данные (учетные записи, логины, пароли);

- применение мобильного устройства

в качестве персональной библиотеки учебных, методических и справочных материалов; плеера для записи и прослушивания аудиолекций; мультимедийного гида в музеях и галереях; фотоаппарата и видеокамеры для фиксирования визуальной информации в цифровом виде и пр.;

- беспроводное подключение к различным измерительным приборам/сенсорам и использование встроенных датчиков, например гироскопа (позволяет использовать как угломер), вибрации, освещенности, влажности, температуры и др. в образовательных и исследовательских целях;

- включение в инфраструктуру образовательного и исследовательского процессов магазинов приложений и учебного контента (Apple AppStore, Google Play и т. п.), обеспечивающих возможность удаленного доступа к электронным образовательным и научным ресурсам, их загрузки, воспроизведения, рейтингования, редактирования, обмена опытом по их использованию;

- применение мобильного устройства как средства идентификации и платежного инструмента: регистрация учащихся, ППС и административного персонала на занятиях и мероприятиях; оплата дополнительных услуг, общественного питания, учет пользования инфраструктурой и т. д.;

- мобильная навигация и геопозиционирование (навигация по кампусу; определение местоположения; поиск географических объектов; получение справочной картографической информации; построение треков передвижения и пр.).

Менеджмент учебного заведения, в свою очередь, сможет не только привлечь активную, мотивированную, подготовленную молодежь, но и решить ряд организационных и технологических вопросов, включая:

- повышение мобильности учащихся и сотрудников;

- повышение мотивации студентов и преподавателей;

- увеличение производительности за счет уменьшения временных затрат;

- сокращение расходов на приобретение компьютерной техники, IT-инфраструктуру, приобретение и разработку специализированного программного обеспечения и пр.;

- сокращение затрат на содержание службы поддержки (Help Desk), содержа-

ние в штате высококвалифицированных специалистов и др.

Рост рынка мобильных устройств и потребность в специальных системах для контроля за ними вызывает увеличение спроса на решения по управлению мобильными устройствами (MDM-решения, Mobile Device Manager), ориентированные на выполнение следующих технических и организационных задач:

- определение типа и технических возможностей подключенных устройств;
- управление возможностями устройств, обеспечение доступа к корпоративным и интернет-ресурсам, включение/отключение технических компонентов (видеокамера, сенсоры);
- вывод информации с мобильных устройств на сетевое оборудование и устройства отображения информации;
- обмен данными с системами управления пользователями;
- автоматизированная загрузка на устройство информационных материалов;
- автоматизированное удаление корпоративной информации по команде владельца или администратора корпоративной системы (например, в случае утери или кражи мобильного устройства);
- массовое оповещение владельцев устройств (в случае экстренных ситуаций).

По оценке аналитиков компании Gartner, мировой рынок решений класса MDM насчитывает порядка 60 компаний, из них не более 20 разработчиков, решения которых наиболее полно реализуют функциональность управления жизненным циклом мобильного устройства [7].

Все представленные сегодня на рынке MDM-решения можно разделить на два типа. В первом, «легком», варианте для подключения нового мобильного устройства необходимо установить на него программу-клиент, поддерживающую связь с корпоративным сервером управления и контролирующую все действия пользователей. При этом в клиенте хранится описание корпоративной политики безопасности и средства мониторинга по ее соблюдению, а пользователь работает с привычными ему приложениями.

Второй, «тяжелый», вариант предполагает установку на устройство специального

набора программного обеспечения, с помощью которого осуществляется возможность подключения к корпоративным сервисам. В этом случае на мобильном устройстве создаются две рабочие среды: персональная, доступ к которой имеет пользователь, и корпоративная, контролируемая ИТ-администраторами компании [4].

Среди отечественных решений класса MDM необходимо упомянуть систему защиты корпоративной сотовой связи с централизованным управлением SafePhone (<http://www.safe-phone.ru>), разработанную в НИИ СОКБ (Научно-испытательный институт систем обеспечения комплексной безопасности), и продукт MobileIron от компании Крок (<http://www.croc.ru/>), предназначенный для обеспечения безопасного доступа к корпоративным ресурсам с различных мобильных платформ. В начале 2013 г. о выпуске MDM-решения, ориентированного на безопасность и защиту от вредоносного ПО, заявила лаборатория Касперского (<http://www.kaspersky.com>).

В настоящее время, в рамках реализации Постановления Правительства РФ от 09.04.2010 № 218 по развитию кооперации российских высших учебных заведений и производственных предприятий [8], компания IBS совместно с Санкт-Петербургским государственным политехническим университетом реализуют комплексный проект по разработке программного обеспечения для систем централизованного управления мобильными устройствами (менеджер мобильных устройств) по функциональным характеристикам превосходящего зарубежные аналоги и ориентированного для использования в высших учебных заведениях.

Помимо решения традиционных для MDM задач, рассмотренных выше, разрабатываемый менеджер мобильных устройств будет обеспечивать расширенный набор возможностей, позволяющих реализовать дополнительную функциональность по управлению мобильными устройствами, в том числе:

- массовое оповещение/информирование пользователей;
- обеспечение доступа к корпоративным информационным ресурсам и сервисам;
- ограничение доступа к корпоративным ресурсам и сервисам на уровне групп пользователей, в т. ч. по заданному расписанию;

взаимодействие с облачными ресурсами (облачное хранилище, вычислительная платформа, портал корпоративных приложений, удаленные виртуальные рабочие места).

### **Особенности реализации концепции BYOD в вузах**

Мы считаем, что использование подходов BYOD в высших учебных заведениях имеет свою специфику, в частности, необходимо регулировать и контролировать доступ к ресурсам корпоративной сети в зависимости от физической локации пользователя (например, мобильное устройство должно автоматически отключаться от Интернета и подключаться к хранилищу учебных материалов, если студент находится на лекции); осуществлять централизованную загрузку образовательного контента на мобильные устройства в привязке к расписанию занятий; при необходимости блокировать входящие и исходящие звонки, SMS и MMS-сообщения и т. д. Аналогов подобной функциональности среди существующих MDM-решений не существует.

Вместе с этим необходимо отметить трудности применения подходов BYOD, которые особенно актуальны для вузов с учетом современного состояния российской высшей школы:

необходимость инвестиций в первоначальную модернизацию инфраструктуры;

необходимость в высоком быстродействии и пропускной способности средств коммуникации и маршрутизации;

недостаточно разработанная нормативно-правовая база, регулирующая особенности применения информационных технологий в образовании;

необходимость разработки соответствующего образовательного контента;

увеличение интернет-трафика и др.

Помимо этого вузовская IT-инфраструктура должна обеспечивать нормальную работу широкого спектра устройств на различных платформах (iOS, Windows, Android, Symbian и др.); информационную безопасность и защиту информации; доступ к внутренним и к внешним ресурсам; централизованную техническую поддержку пользователей и др.

На первом этапе внедрения подходов BYOD быстрый эффект при сравнитель-

но небольших затратах могут дать проекты по разработке и запуску версии портала вуза для мобильных устройств; внедрение системы управления мобильными устройствами; разработка мобильных политик; создание собственного приложения для наиболее популярной среди учащихся и сотрудников мобильной платформы. Более значительные затраты и отложенный эффект будут у таких мероприятий, как развитие существующей IT-инфраструктуры и проектирование новых сервисов, сопровождающихся внедрением или интеграцией с системами управления образовательной и административно-хозяйственной деятельностью (ERP, SIS, LMS, CMS и т. п.<sup>1</sup>).

С организационной точки зрения, реализация подходов BYOD начинается с того момента, когда лица, принимающие решения в сфере развития информационных технологий в вузе, начинают рассматривать данную модель как комплексный сервис. Сложность практической реализации заключается не в проблемах управления, определения конфигурации и доставки контента на конкретное устройство, а в необходимости обеспечения гибкой модели безопасности для данных и сервисов. Вместе с этим запуск сервиса в вузе невозможен без учета таких вопросов, как управление эксплуатацией, расширение возможностей IT-инфраструктуры (чтобы «принять» новых пользователей), подготовки и привлечения специалистов для обслуживания и технической поддержки и др.

Для эффективного решения указанных задач вузу необходима стратегия информатизации, призванная определить роль и место информационных технологий в обеспечении деятельности и решении задач развития учебного заведения на среднесрочную (3–5 лет) и долгосрочную (5–10 лет) перспективы.

В минимальном варианте в качестве базового документа можно использовать каталог информационных сервисов, определяющий согласованную совокупность решений с определенными приоритетами и перспективой реализации от 3–5 лет. Как

<sup>1</sup> Enterprise Resource Planning, Student Management System, Learning Management System, Content Management System.

показывает практика, на первых позициях в таком документе должны стоять сервисы, срок реализации которых составляет от двух до шести месяцев.

В заключение отметим, что несмотря на указанные проблемы и сложности по построению стратегии информатизации высшего учебного заведения с использованием подходов BYOD, успешный опыт реализа-

ции IT-проектов в системе высшего профессионального образования последних лет [9–11] позволяет рассчитывать, что разрабатываемое решение будет востребовано в образовательных учреждениях, а российские университеты займут достойное место в ряду вузов, использующих мобильные сервисы и технологии в своей деятельности.

Работа выполняется при финансовой поддержке Министерства образования и науки РФ.

#### СПИСОК ЛИТЕРАТУРЫ

1. **Rheingold, H.** Smart Mobs. The Next Social Revolution [Text] / H. Rheingold. – Cambridge, MA: Basic Books, 2002.
2. **Harkins, M.** Mobile: Learn from Intel's CISO on Securing Employee-Owned Devices [Электронный ресурс] / M. Harkins // Government info security. Режим доступа <http://www.govinfosecurity.com/webinars/mobile-learn-from-intels-ciso-on-securing-employee-owned-devices-w-264> (Дата обращения 26.02.2013)
3. **Кукушкин, А.** Мобильная безопасность [Текст] / А. Кукушкин // Директор по безопасности. – 2012. – № 12. – С. 80–82.
4. По материалам Hype Cycle for Education 2012 [Электронный ресурс] / Gartner Inc., Oct. 2012.
5. **Коржов, В.** Платформы поддержки мобильности [Текст] / В. Коржов // Открытые системы. СУБД. – 2012. – № 6. – С. 17–22.
6. **Иванченко, Д.А.** Формирование образовательной среды вуза в условиях информатизации: проблемы и перспективы [Текст] / Д.А. Иванченко, Л.А. Марцинович // Инновации на основе информационных и коммуникационных технологий: Матер. Междунар. науч.-практ. конф.; Под ред. С.У. Увайсова. – М.: МИЭМ НИУ ВШЭ, 2012. – С. 280–284.

7. По материалам Hype Cycle for Education, 2012 [Электронный ресурс] / Gartner Inc., May 2012.
8. О мерах государственной поддержки развития кооперации российских высших учебных заведений и организаций, реализующих комплексные проекты по созданию высокотехнологичного производства [Текст] / Постановление Правительства России от 9 апреля 2010 г. № 218 // Рос. газета. – 16.04.2010. – № 5160.
9. **Хмельков, И.А.** Реализация модели и технологии распространения образовательного мультимедиа контента в российской системе образования [Текст] / И.А. Хмельков, С.О. Попов, О.А. Изосимов [и др.] // Открытое и дистанционное образование. – 2012. – Т. 3. – № 47. – С. 18–23.
10. **Рудской, А.И.** Информатизация вуза – ключевое направление инновационной образовательной программы [Текст] / А.И. Рудской // Высшее образование в России. – 2007. – № 12. – С. 8–12.
11. **Ильин, И.В.** Использование облачных технологий при построении информационных систем кластера [Текст] / И.В. Ильин, А.Б. Анисифоров // Экономика и управление. – 2012. – № 7. – С. 22–27.

#### REFERENCES

1. **Rheingold N.** Smart Mobs. The Next Social Revolution. – Cambridge, MA: Basic Books, 2002.
2. **Harkins M.** Mobile: Learn from Intel's CISO on Securing Employee-Owned Devices / Government info security. Available <http://www.govinfosecurity.com/webinars/mobile-learn-from-intels-ciso-on-securing-employee-owned-devices-w-264> (Accessed 26.02.2013)
3. **Kukushkin A.** Mobil'naia bezopasnost' /Direktor po bezopasnosti. –2012. –№ 12. –S. 80–82. (rus)
4. По материалам Hype Cycle for Education, 2012; Gartner Inc., Oct. 2012.
5. **Korzov V.** Platformy podderzhki mobil'nosti // Otkrytye sistemy. SUBD. – 2012. – № 6. – S. 17–22. (rus)
6. **Ivanchenko D.A., Martsinovich L.A.** Formiro-

- vanie obrazovatel'noi sredy vuza v usloviakh informatizatsii: problemy i perspektivy / Innovatsii na osnove informatsionnykh i kommunikatsionnykh tekhnologii: Mater. mezhdunar. nauch.-prakt. konf.; Pod red. S.U. Uvaisova. – Moscow: MIEM NIU VShE, 2012. – S. 280–284. (rus)
7. По материалам Hype Cycle for Education, 2012; Gartner Inc., May 2012.
8. О мерах государственной поддержки развития кооперации российских высших учебных заведений и организаций, реализующих комплексные проекты по созданию высокотехнологичного производства; Postanovlenie Pravitel'stva Rossii ot 9 apreliia 2010 g. № 218 // Ros. gazeta. –16.04.2010. –№ 5160. (rus)
9. **Khmel'kov I.A., Popov S.O., Izosimov O.A. i dr.** Realizatsiia modeli i tekhnologii raspros-

traneniia obrazovatel'nogo mul'timedia kontenta v rossiiskoi sisteme obrazovaniia / Otkrytoe i dis-tantsionnoe obrazovanie. – 2012. – Т. 3. – № 47. – S. 18–23. (rus)

10. **Rudskoi A.I.** Informatizatsiia vuza – kliu-chevoe napravlenie innovatsionnoi obrazovatel'noi

programmy // Vysshee obrazovanie v Rossii. – 2007. – № 12. – S. 8–12. (rus)

11. **Il'in I.V., Anisiforov A.B.** Ispol'zovanie oblachnykh tekhnologii pri postroenii informat-sionnykh sistem klastera / Ekonomika i upravlenie. – 2012. – № 7. – S. 22–27. (rus)

---

**ИВАНЧЕНКО Дмитрий Алексеевич** – *ведущий консультант IBS.*

127434, Россия, Москва, Дмитровское ш., д. 9Б.

E-mail: DIvanchenko@ibs.ru

**IVANCHENKO Dmitry A.** – *IBS.*

9B Dmitrovskoe Sh., 127434 Moscow, Russia.

E-mail: DIvanchenko@ibs.ru

**ХМЕЛЬКОВ Игорь Александрович** – *директор по консалтингу IBS.*

127434, Россия, Москва, Дмитровское ш., д. 9Б.

E-mail: IKhmelkov@ibs.ru

**KHMELKOV Igor A.** – *IBS.*

9B Dmitrovskoe Sh., 127434 Moscow, Russia.

E-mail: IKhmelkov@ibs.ru

**РАЙЧУК Дмитрий Юрьевич** – *проректор по научной работе Санкт-Петербургского государственного политехнического университета, кандидат технических наук.*

195251, Россия, Санкт-Петербург, ул. Политехническая, д. 29.

E-mail: vicerector.sc@spbstu.ru

**RAYCHUK Dmitry Yu.** – *St. Petersburg State Polytechnical University.*

195251, Politekhnikeskaya Str. 29, St. Petersburg, Russia.

E-mail: vicerector.sc@spbstu.ru

**МИТРОФАНОВ Александр Михайлович** – *директор департамента научно-организационной деятельности Санкт-Петербургского государственного политехнического университета, кандидат технических наук.*

195251, Россия, Санкт-Петербург, ул. Политехническая, д. 29.

E-mail: mitrofanov@spbstu.ru

**MITROFANOV Aleksandr M.** – *St. Petersburg State Polytechnical University.*

195251, Politekhnikeskaya Str. 29, St. Petersburg, Russia.

E-mail: mitrofanov@spbstu.ru

**САМОЧАДИН Александр Викторович** – *профессор кафедры распределительных вычислений и компьютерных сетей Санкт-Петербургского государственного политехнического университета, кандидат технических наук.*

195251, Россия, Санкт-Петербург, ул. Политехническая, д. 29.

E-mail: Samochadin@soft-consult.ru

**SAMOCHADIN Aleksandr V.** – *St. Petersburg State Polytechnical University.*

195251, Politekhnikeskaya Str. 29, St. Petersburg, Russia.

E-mail: Samochadin@soft-consult.ru

**РОГОВ Петр Александрович** – *заместитель начальника организационного отдела департамента научно-организационной деятельности Санкт-Петербургского государственного политехнического университета.*

195251, Россия, Санкт-Петербург, ул. Политехническая, д. 29.

E-mail: petr.rogov@spbstu.ru

**ROGOV Petr A.** – *St. Petersburg State Polytechnical University.*

195251, Politekhnikeskaya Str. 29, St. Petersburg, Russia.

E-mail: petr.rogov@spbstu.ru



## Конференция «Технологии Microsoft в теории и практике программирования»

Начиная с 2004 года Институт информационных технологий и управления (ранее — факультет технической кибернетики) Санкт-Петербургского государственного политехнического университета проводит конференцию студентов, аспирантов и молодых ученых Северо-Запада «Технологии Microsoft в теории и практике программирования». Цели конференции — подготовка участников к будущей работе в профессиональных программистских коллективах, создающих программный продукт высокого качества, поддержка изучения современных информационных технологий и инструментальных средств в соответствии с мировыми стандартами и действующими международными сертификационными требованиями, а также выявление талантливых молодых специалистов в области разработки и использования программных систем.

Особенность конференции в том, что участник должен проявить свои знания и умения не столько в области программирования различных математических головоломок, сколько в области разработки и использования программных продуктов и систем в условиях, максимально приближенных к реальным процессам проектирования и разработки современных систем различной степени сложности.

Конференция проводится в два этапа: на первом отбираются и ревьюируются работы участников, присылающих короткие статьи — заявки на участие. На их основе отбираются приглашенные докладчики на второй этап. Присланные статьи публикуются в сборнике материалов конференции. Второй этап проходит в виде докладов на секциях. Кроме того, предусмотрены пленарные доклады и мастер-классы по новациям в мире инструментов и технологий программирования.

Все участники секционных презентаций награждаются дипломами первой, второй и третьей степени. Решение о премировании участников конкурса принимает конкурсная комиссия, в которую входят руководители секций и представители организаций спонсоров.

В 2013 году спонсорами конференции являются всемирно известные компании Microsoft и EMC.

В данном журнале публикуется одиннадцать статей победителей — участников конференции 2013 года. Тематика этих статей соответствует тематике трех секций: 1) программные продукты и системы (статьи 3, 5, 6, 7); 2) применение перспективных методов и технологий разработки программного обеспечения (статьи 1, 2, 4, 8); 3) перспективные методы и алгоритмы теории программирования (статьи 9, 10, 11). Публикуемые работы отличаются инновационным характером и ориентацией на реализацию полученных результатов, широкое применение технологий MS при создании программного обеспечения и документации.

Директор ИИТУ, профессор,  
доктор технических наук  
И.Г. Черноруцкий

УДК 519.685.3

*Я.А. Кириленко, С.В. Григорьев, Д.А. Авдюхин*

## **РАЗРАБОТКА СИНТАКСИЧЕСКИХ АНАЛИЗАТОРОВ В ПРОЕКТАХ ПО АВТОМАТИЗИРОВАННОМУ РЕИНЖИНИРИНГУ ИНФОРМАЦИОННЫХ СИСТЕМ**

*I.A. Kirilenko, S.V. Grigoriev, D.A. Avdiukhin*

### **SYNTAX ANALYZERS DEVELOPMENT IN AUTOMATED REENGINEERING OF INFORMATIONAL SYSTEM**

Рассмотрены особенности разработки синтаксических анализаторов при автоматизации реинжиниринга. Сформулированы требования, предъявляемые к генераторам парсеров. Описан инструмент, удовлетворяющий этим требованиям.

РЕИНЖИНИРИНГ. СИНТАКСИЧЕСКИЙ АНАЛИЗ. ТРАНСЛЯЦИЯ. ГЕНЕРАТОРЫ ПАРСЕРОВ.

In this article features of syntax analyzer development in automatic reengineering problems are considered. Requirements, laid on parser generators, are stated. Tool, satisfying these requirements, is described.

REENGINEERING. SYNTAX ANALYSIS. TRANSLATION. PARSER GENERATORS.

Проекты по автоматизации реинжиниринга программ [1] выдвигают особые требования к генераторам синтаксических анализаторов по сравнению с требованиями при разработке оптимизирующих компиляторов. Причины принципиального отличия кроются в процессе разработки средства автоматизации реинжиниринга по сравнению с процессом разработки транслятора, но немаловажную роль играет история развития мигрируемых языков: большинство из них уже существовали, когда теория синтаксически управляемой трансляции только развивалась. Такие языки — сейчас их относят к устаревшим (legacy) языкам — проектировались, когда еще не были получены основные теоретические результаты, положенные в основу наиболее распространенных современных генераторов. Поэтому устаревшие языки не учитывают эти результаты и трудны для синтаксического анализа с использованием современных инструментов, которые в свою очередь значительно упрощают создание анализаторов, но лишь для относительно современных или заново проектируемых языков.

В отличие от проектов по разработке оптимизирующих компиляторов, при реинжиниринге систем актуальна проблема отсутствия подходящей спецификации языка. Нередко документация вовсе неполна или содержит ошибки — работающие программы даже синтаксически не соответствуют документации. Кроме того, для описания грамматик устаревших языков использовались синтаксические диаграммы, обладающие значительно большими выразительными возможностями, чем популярная ныне и хорошо формализованная форма Бэкуса—Наура (БНФ) [2].

Отдельного внимания заслуживает проблема множества схожих вариантов синтаксиса — диалектов (ярким примером является COBOL<sup>1</sup>).

Если посмотреть на жизненный цикл синтаксического анализатора в проекте по реинжинирингу информационной системы,

<sup>1</sup> Common Business Oriented Language — один из старейших языков программирования (первая версия увидела свет в 1959 г.), широко используемый в разработке бизнес-приложений.



то разница видна уже на этапе предпродажной подготовки: необходимо быстро реализовать прототип инструмента, позволяющий продемонстрировать заказчику возможность создания конечного инструмента. В связи с этим важным требованием является возможность быстрого создания простой версии парсера, обрабатывающего основные конструкции языка. Однако при продолжении работ иногда выясняется, что были выбраны неподходящие инструменты, а их смена приводит к необходимости переписывания грамматики заново, в формате, принимаемом новыми инструментами. Чтобы этого избежать, необходима возможность автоматического преобразования грамматики в новый формат.

В реинжиниринге нет необходимости полностью описывать входной язык, поскольку многие конструкции не встретятся в исходном коде. Кроме того, обрабатываемый код часто меняется в процессе работы над ним, и в нём появляются новые синтаксические конструкции. Поэтому разработка грамматики происходит итеративно и представляет собой не разовое действие, а процесс, идущий на протяжении всего проекта по реинжинирингу системы. Из-за этого парсер часто не способен сразу обработать весь исходный код, однако нужно попытаться продолжить разбор и получить информацию о всех возможных ошибках, а в случае небольшого числа ошибок необходимо продолжить работу над разобранным кодом, при этом покрыв большую часть функциональности системы.

#### **Требования к генератору синтаксических анализаторов**

Важными характеристиками генераторов синтаксических анализаторов, влияющими на простоту разработки и качество создаваемых анализаторов при разработке средства автоматизации реинжиниринга ИС, являются следующие:

- класс принимаемых языков;
- разрешение неоднозначностей в грамматике;
- скорость работы порождаемого парсера/транслятора;
- возможности языка описания грам-

матики (как предметно-ориентированного языка программирования);

- возможность быстрого изменения грамматики;
- простота отладки сгенерированного парсера/транслятора;
- наличие восстановления после ошибок;
- сопровождаемость, качество документации;
- возможность модификации инструмента под свои нужды;
- простота интеграции и использования.

Многие из этих свойств связаны с алгоритмом разбора. При использовании инструментов, порождающих LL [3] и LALR [4] парсеры, необходимо избавляться от конфликтов в грамматике. Однако в реинжиниринге грамматика постоянно меняется, и добавление одного правила может породить десятки конфликтов [5]. Некоторые типы неоднозначностей можно устранить путем преобразований грамматики, но при этом она значительно усложняется, что отрицательно сказывается на ее сопровождаемости. Поэтому входной язык генератора и внутренний алгоритм должны предоставлять возможности по эффективному разрешению конфликтов: использование предикатов, управляющих процессом разбора, либо использование GLR-алгоритма [6] анализа, предназначенного для работы с неоднозначными грамматиками.

При поддержке нескольких диалектов языка нужно в соответствии с принципами качественного кода [7] переиспользовать общие части их грамматик. Помимо этого, многие языки имеют синтаксически схожие конструкции, например, арифметические выражения, поэтому можно повторно использовать некоторые части грамматики одного языка при описании другого. То есть инструмент должен обладать средствами повторного использования элементов грамматик. Одним из них является модульность. Разделение грамматики на модули необходимо не только для повторного использования каких-то ее частей, но и для повышения управляемости и снижения сложности процесса разработки и сопровождения.

Другим способом переиспользования грамматики является параметризация одних правил другими для введения общих концепций. Проводя параллель с языками программирования, правила, параметризованные другими правилами, соответствуют шаблонам (templates) в C++ и generics типам в C# и Java. Одним из примеров является правило для разбора списка элементов с разделителем между ними:

```
list<item delimiter>: item (delimiter
item) *
func: typeName funcName list<param
','> funcBody
funcBody: '{' list<stmt ';'> '}'
```

В синтаксических диаграммах, которые используются для описания устаревших языков, часто встречаются перестановки, с помощью которых определяются списки опций или атрибутов, где каждый элемент встречается только один раз. Обозначим перестановку как  $[| a, b, c, \dots |]$ , тогда упрощенное описание данных в языке COBOL можно задать так:

```
dataEntry: levelNumber dataName
[|pictureClause, occursClause,
usageClause|]
```

Однако обычно такой список записывают менее строгим образом, что приводит к большему числу допущений и конфликтов:

```
dataEntry: levelNumber dataName
(pictureClause | occursClause |
usageClause)+
```

В настоящее время ни один из популярных инструментов не отвечает всем перечисленным выше требованиям. Тем не менее современные инструменты обладают такими удобными средствами для задания грамматик, как

- конструкции EBNF [2] (обычно их обозначают через «\*», «+» и «?»);
- правила с параметрами, которые значительно упрощают трансляцию с учетом контекста (в терминах атрибутивной грамматики такие параметры называются *наследуемыми атрибутами*);
- специальные конструкции для разрешения неоднозначностей в грамматике — предикаты;

- именование семантических значений (вместо использования \$1, \$2, \$3, ...), что уменьшает число возможных ошибок при обращении к ним.

Таким образом, генератор синтаксических анализаторов должен обладать как хорошо зарекомендовавшими себя возможностями, так и предоставлять описанные выше средства для задания грамматики и разрешения неоднозначностей в ней.

### Инструмент для создания синтаксических анализаторов YaccConstructor

На кафедре системного программирования математико-механического факультета Санкт-Петербургского государственного университета (СПбГУ) разрабатывается инструмент для создания генераторов синтаксических анализаторов YaccConstructor [8], который имеет модульную структуру, позволяющую создавать анализаторы с различными языками описания грамматик и алгоритмами разбора. Такая структура во многом решает проблему изменения алгоритма разбора: если пользователь понимает, что изначально выбран неподходящий алгоритм, ему достаточно взять другой генератор.

Для создания генератора анализаторов необходимы следующие компоненты:

- парсер входной грамматики (фронтенд), преобразующий грамматику языка, записанную в некотором формате, во внутреннее представление;
- генератор (бэкенд), при помощи которого на основе внутреннего представления создается синтаксический анализатор или генерируется исходная грамматика на определенном языке, что позволяет породить эквивалентное описание языка, принимаемое другим инструментом;
- преобразования грамматики, которые приводят грамматику к виду, удовлетворяющему ограничениям, накладываемым генератором. Например устранение левой рекурсии или раскрытие конструкций EBNF.

В рамках проекта разрабатывается язык описания трансляций Yard, удовлетворяющий описанным ранее требованиям, благодаря мощному синтаксису (EBNF, перестановки, литералы), реализации механизмов переиспользования грамматики (модуль-



ность, параметризованные правила) и поддержке средств спецификации трансляции (наследуемые атрибуты, предикаты, именование семантических значений).

Основываясь на описанных преимуществах GLR алгоритма разбора перед LL и LR, в рамках проекта также реализован GLR-генератор. В результате анализа существующих алгоритмов был выбран алгоритм RNLGR [9], работающий со всеми контекстно-свободными грамматиками. В нем реализована встроенная поддержка наследуемых атрибутов, предикатов и литералов. Также для него реализован алгоритм восстановления после ошибок. На основе его практического использования реализованы различные средства отладки, среди которых вывод информации о неоднозначностях и визуальное представление результатов разбора.

Несмотря на то, что построение синтаксически управляемых трансляций яв-

ляется хорошо изученной областью, особенности проектов по реинжинирингу выдвигают новые требования и формулируют новые задачи в сфере автоматизации построения синтаксических анализаторов. Современные инструменты, известные как «компиляторы компиляторов», предоставляют широкие возможности для автоматического создания синтаксических анализаторов (парсеров), но недостаточно отвечают специальным требованиям проектов по реинжинирингу. Это послужило мотивацией для разработки инструмента, предназначенного для создания генераторов синтаксических анализаторов, в рамках которого предъявленные требования будут выполнены.

Дальнейшая исследовательская работа над инструментом ведется в направлении поддержки различных диалектов языков, а также его совершенствования по результатам практического использования.

#### СПИСОК ЛИТЕРАТУРЫ

1. **Терехов, А.Н.** Автоматизированный реинжиниринг программ [Текст] / А.Н. Терехов. — Изд-во СПбГУ, 2000. — 332 с.
2. **Ахо, А.** Компиляторы: принципы, технологии, инструменты [Текст] / А. Ахо, Р. Сети, Дж. Ульман. — М.: ИД «Вильямс», 2003. — 768 с.
3. **Lewis, P.M. II, Stearns R.E.** Syntax-Directed transduction [Text] / P.M. Lewis II, R.E. Stearns —ACM 15. —July 1968. —Vol. 3. —P. 465–488.
4. **Deremer, F.L.** Practical translators for LR(k) languages [Text] / F.L. Deremer // Ph.D. thesis. —Massachusetts Institute of Technology, 1969.
5. **Van den Brand, M.G.J.** Current Parsing Techniques in Software Renovation Considered Harmful [Text] /M.G.J. Van den Brand, M.P.A. Sellink, C. Verhoef // IEEE Proc. of the 6th International Workshop on Program Comprehension. —1998. —P. 108–117.
6. **Tomita, M.** Efficient Parsing for Natural Language [Text] / M. Tomita. —1986.
7. **Макконнелл, С.** Совершенный код [Текст] / С. Макконнелл. —СПб.: Питер, 2005. —896 с.
8. Домашняя страница YaccConstructor [Электронный ресурс] / Режим доступа: <https://code.google.com/p/recursive-ascent/> (Дата обращения 2008)
9. **Scott, Elizabeth.** Right nulled GLR parsers [Text] / Elizabeth Scott, Adrian Johnstone // ACM Transactions on Programming Languages and Systems. —July 2006. —№ 28 (4). —P. 577–618.

#### REFERENCES

1. **Terekhov A.N.** Avtomatizirovannyi reinzhiniring program. — St Petersburg: Izdatel'stvo Sankt-Peterburgskogo gosudarstvennogo un-ta, 2000. — 332 s. (rus)
2. **Akho A., Seti R., Ul'man Dzh.** Kompilyatory: printsipy, tekhnologii, instrumenty. — Moscow: Izdatel'skii dom «Vil'iams», 2003. — 768 s. (rus)
3. **Lewis P.M. II, Stearns R.E.** Syntax-Directed transduction/ ACM. — July 1968. — Vol. 15. — № 3. — P. 465–488.
4. **Deremer F.L.** Practical translators for LR(k) languages / Ph.D. thesis. — Massachusetts Institute of Technology, 1969.
5. **van den Brand M.G.J, Sellink M.P.A., Verhoef C.** Current Parsing Techniques in Software Renovation Considered Harmful / Proc. of the 6th Internat. Workshop on Program Comprehension. — 1998. — P. 108–117.
6. **Tomita M.** Efficient Parsing for Natural Language, 1986.
7. **Makkonnell S.** Sovershennyi kod. — St. Petersburg: Piter, 2005. — 896 s. (rus)

8. Home page YaccConstructor. Available <https://code.google.com/p/recursive-ascent/> (Accessed 2008)

parsers / ACM Transactions on Programming Languages and Systems. – July 2006. – № 28 (4). – P. 577–618.

9. **Scott E., Johnstone A.** Right nulled GLR

---

**КИРИЛЕНКО Яков Александрович** – старший преподаватель Санкт-Петербургского государственного университета.

198504, Россия, Санкт-Петербург, Старый Петергоф, Университетский пр., д. 28.

E-mail: [jake@math.spbu.ru](mailto:jake@math.spbu.ru)

**KIRILENKO, Iakov A.** – St. Petersburg State University.

198504, Universitetsky prospekt 28, Peterhof, St. Petersburg, Russia.

E-mail: [jake@math.spbu.ru](mailto:jake@math.spbu.ru)

**ГРИГОРЬЕВ Семен Вячеславович** – аспирант Санкт-Петербургского государственного университета.

198504, Россия, Санкт-Петербург, Старый Петергоф, Университетский пр., д. 28.

E-mail: [rsdpisuy@gmail.com](mailto:rsdpisuy@gmail.com)

**GRIGORIEV Semen V.** – St. Petersburg State University.

198504, Universitetsky prospekt 28, Peterhof, St. Petersburg, Russia.

E-mail: [rsdpisuy@gmail.com](mailto:rsdpisuy@gmail.com)

**АВДЮХИН Дмитрий Алексеевич** – студент Санкт-Петербургского государственного университета.

198504, Россия, Санкт-Петербург, Старый Петергоф, Университетский пр., д. 28.

E-mail: [dimonbv@gmail.com](mailto:dimonbv@gmail.com)

**AVDIUKHIN, Dmitry A.** – St. Petersburg State University.

198504, Universitetsky prospekt 28, Peterhof, St. Petersburg, Russia.

E-mail: [dimonbv@gmail.com](mailto:dimonbv@gmail.com)



УДК 004.415

*П.Д. Дробинцев, И.В. Никифоров, В.П. Котляров*

## **МЕТОДИКА ПРОЕКТИРОВАНИЯ ТЕСТОВ СЛОЖНЫХ ПРОГРАММНЫХ КОМПЛЕКСОВ НА ОСНОВЕ СТРУКТУРИРОВАННЫХ UCM МОДЕЛЕЙ**

*P.D. Drobintsev, I.V. Nikiforov, V.P. Kotliarov*

### **FORMAL MODELS STRUCTURIZATION BASED TECHNIQUE OF COMPLEX SOFTWARE PROJECTS TESTING**

Описана методика тестирования, основанная на применении структуризации к формальной модели разрабатываемого программного комплекса. В рамках методики описаны подходы и критерии структуризации, а также приведен пример автоматической генерации тестов на основе структурированной модели.

Предлагаемая методика позволяет существенно упростить процесс анализа формальной модели и обеспечивает возможность автоматической генерации тестов под контролем пользователя.

UCM. СТРУКТУРИЗАЦИЯ. ПОКРЫТИЕ. ТЕСТОВЫЙ НАБОР.

The paper describes a technique of software testing based on structurization of formal model. In the scope of the technique approaches and criterions of structurization are described. An example of automated tests generation is shown.

Suggested technique allows to resolve problems of formal model understanding and gives an opportunity of automated tests generation with usage of branch coverage criterion.

UCM. STRUCTURIZATION. COVERAGE. TEST SUI.

Растущая сложность разрабатываемых программных комплексов существенно осложняет обеспечение высокого уровня их качества. При этом коллективный подход к разработке программного продукта (ПП) как правило ограничивается компетентностью разработчиков и тестировщиков проектной команды, знанием исключительно запланированных ими конкретных модулей и интерфейсов внутримодульного взаимодействия и взаимодействия с окружением. В связи с этим при проектировании сложных систем развивается т. н. *инкрементальный подход*, позволяющий интегрировать проверку корректности программного обеспечения (ПО) по частям (по компонентам) с проверкой корректности системы в целом, что существенно снижает суммарную трудоемкость разработки.

Данная статья посвящена описанию применения инкрементального подхода к тестированию и верификации сложного программного продукта. В основе подхода

лежит идея автоматизированного получения структурированной формальной модели разрабатываемой системы. Структурированная формальная модель наряду с упрощением понимания функциональности и поведенческих свойств системы, обеспечивает упрощение анализа, верификации и служит основой для автоматической генерации кодов и тестов разрабатываемого ПП.

#### **Описание UCM нотации**

Для описания сложных программных систем все чаще используются формальные языки проектирования, в частности, в данной работе использован язык UCM (Use Case Maps) [1].

Модель UCM проекта (рис. 1) представляет собой набор связанных и структурированных диаграмм, каждая из которых состоит из последовательности элементов нотации UCM. В совокупности набор диаграмм задает возможные поведения систе-

мы, описанные в требованиях. К основным элементам UCM нотации относятся: компоненты Team, в границах которых задается поведение объекта; элементы Responsibility (X), отражающие точки выполнения каких-либо действий; StartPoint (●) и EndPoint (⊥), задающие начальную и конечную точки поведенческого сценария; элементы OrFork (λ) и OrJoin (∧), описывающие альтернативы и недетерминизм; элементы AndFork (⊕) и AndJoin (⊖), описывающие параллельные сценарии; элемент FailurePoint (≡), описывающий механизм генерации и обработки исключений; элемент Timer (⊙), задающий временную задержку, в т. ч. и со сложным логическим поведением, структурный элемент Stub (◇), описывающий иерархию в поведении системы, что позволяет вести разработку системы покомпонентно от верхнего уровня абстракции до детального описания низкоуровневых диаграмм.

Следует отметить, что использование формальной модели открывает возможность различного рода преобразований, так, например, во входной язык верификатора [2] или во входной язык системы автоматической генерации тестов [6].

### Структуризация формальной модели системы

Структуризация – это этап системного анализа, который состоит в том, что вся совокупность объектов и процессов, имеющих отношение к поставленной цели, сначала

разделяется на собственно изучаемую систему и внешнюю среду, затем выделяются отдельные составные части – подсистемы и элементы изучаемой системы. С точки зрения UCM единицей структуризации может быть любой элемент или выделенная группа элементов поведения, лишь бы это позволяло упростить процесс анализа исходной модели. В рамках настоящей работы в роли единиц структуризации используется элемент Stab или явно выделенная совокупность элементов UCM диаграммы.

Определим UCM диаграмму как ориентированный граф с вершинами из множества элементов диаграммы и дугами, определяющими достижимость одного элемента диаграммы из других. То есть UCM диаграмма – это четверка  $UCM = (U, S, E, R)$ , где  $U$  – конечное множество элементов;  $S \subseteq U$  – множество стартовых точек;  $E \subseteq U$  – множество конечных точек;  $R \subseteq U \times U$  – отношение переходов, определяющее достижимость одного элемента диаграммы из других.

С точки зрения анализа поведения системы для процессов верификации и тестирования важным является понятие пути ( $\pi$ ), которое определяется как  $\pi = u_1 \rightarrow u_2 \rightarrow \dots \rightarrow u_n$ , где  $u_i \in U$ . При проведении тестирования также важно условие, что в тестовых сценариях каждый путь начинается в стартовой точке и заканчивается в конечной точке  $u_1 \in S \wedge u_n \in E$ .

Структурированная UCM диаграмма отличается от исходной только тем, что ис-

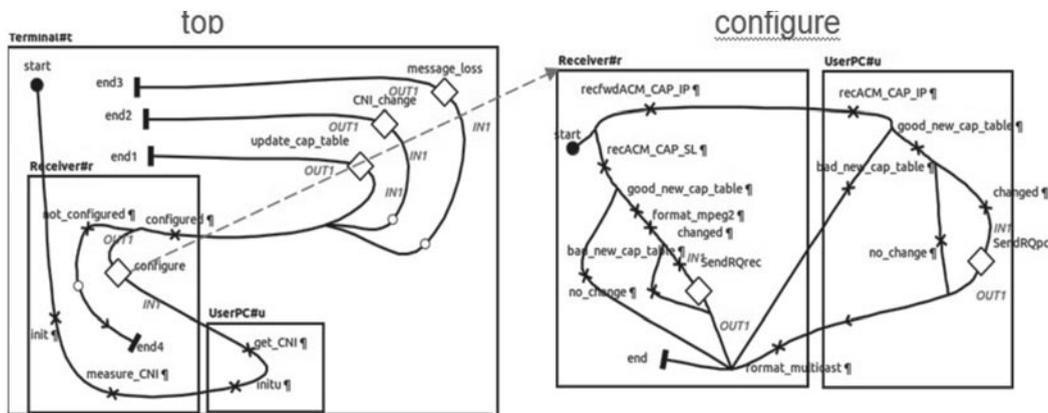


Рис. 1. UCM диаграмма для телекоммуникационного проекта Satellite

пользует элементы Stub для декомпозиции поведения на основании функциональных или структурных критериев. При этом элемент Stub является вложенной UCM диаграммой и определяется как шестерка  $ST = (U_{st}, S_{st}, E_{st}, R_{st}, Sr, Er)$ , в которой определение  $U_{st}, S_{st}, E_{st}, R_{st}$  аналогично определению UCM диаграммы;  $Sr \subseteq S_{st} \times U$  – отношение переходов, определяющее связь стартовых точек элемента Stub с элементами диаграммы верхнего уровня, и  $Er \subseteq E_{st} \times U$  – отношение переходов, определяющее связь конечных точек элемента Stub с элементами диаграммы верхнего уровня.

Таким образом, процесс структуризации можно определить как преобразование:  $UCM \rightarrow UCM'$ , в котором  $U' = U \cup ST_1 \cup ST_2 \cup \dots \cup ST_k$ . Очевидно, что существует множество способов реализации подобного преобразования, однако с точки зрения анализа сценариев поведения системы оно может проводиться на основании двух дополняющих критериев – функционального и структурного.

Функциональный критерий определяется создателем формальной спецификации и подразумевает выделение в элемент Stub тех элементов исходной UCM диаграммы, которые служат выполнению определенной функциональности, заданной в требованиях на ПО. Использование этого критерия требует понимания как исходной спецификации требований системы, так и того, какими поведенческими сценариями каждое из требований может быть проверено при тестировании системы. Ограничением подобного подхода является обязательное участие специалиста предметной области (представителя заказчика), участвующего в формализации и утверждающего предлагаемое преобразование  $UCM \rightarrow UCM'$ . Заметим, что декомпозиция формализованной спецификации практически не поддается автоматизации.

На рис. 1 представлен пример двух уровней структурированной UCM диаграммы, описывающей протокол взаимодействия между спутником и оператором, предоставляющим IP услуги на основе стандарта ETSI [3]. В данном примере в соответствии

с требованиями поведение системы разбито на компоненты top и configure.

Структурный критерий в отличие от функционального использует структуру самой UCM диаграммы для построения элементов Stub и проведения декомпозиции. Характерным примером использования данного критерия является объединение в Stub элементов диаграммы, принадлежащих к различным путям, приводящим систему в одно и то же состояние. Следует отметить, что композиция поведенческих сценариев вложенных диаграмм разных уровней на основе структурного критерия может быть автоматизирована.

Вне зависимости от применяемого критерия структурирования любая UCM диаграмма может быть подвержена декомпозиции с целью упрощения дальнейшего анализа.

#### Автоматическая генерация тестового набора по структурной модели

Для автоматической генерации тестового набора по структурированной модели необходимо определить возможные критерии покрытия, применение которых будет гарантировать качество тестирования данной модели. В силу того факта, что UCM диаграмма содержит информацию о структуре разрабатываемого ПО, для обеспечения ее покрытия удобно использовать структурные критерии. Наиболее известными из них являются критерий покрытия по ветвям, определяющий необходимость покрытия каждой ветви поведения системы хотя бы одним из тестов, и критерий покрытия по путям, определяющий необходимость покрытия каждого пути поведения системы хотя бы одним тестом [4].

Рассмотрим критерий покрытия по ветвям. С точки зрения UCM диаграммы ветвь определяется как отрезок пути, удовлетворяющий следующим требованиям:

- каждая ветвь начинается в стартовой точке диаграммы или в точке альтернативного выбора OrFork, или в точке начала параллельного поведения AndFork;
- каждая ветвь заканчивается или в конечной точке диаграммы, или в точке окончания альтернативного поведения OrJoin,

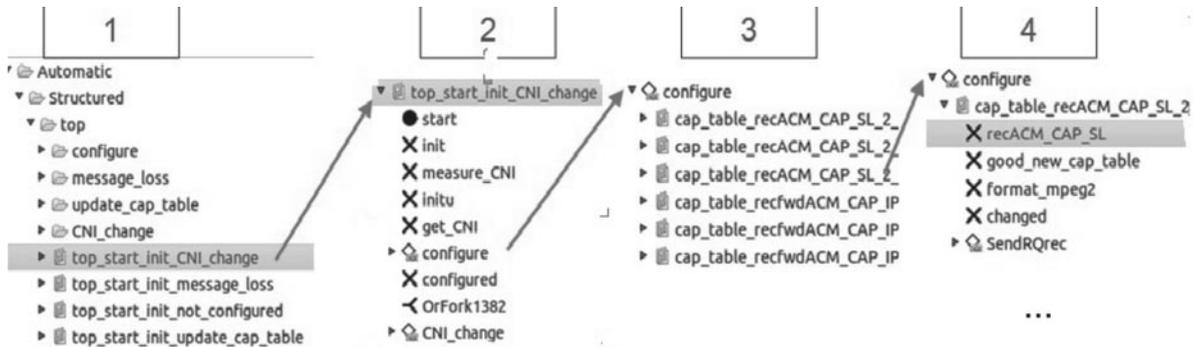


Рис. 2. Структурные гиды UCM проекта

или в точке синхронизации параллельных сценариев AndJoin.

С точки зрения структурированного представления UCM диаграммы для выполнения критерия покрытия по ветвям необходимо покрытие всех ветвей основной диаграммы (диаграммы верхнего уровня), а также всех ветвей диаграмм, находящихся в элементах Stub. При этом тесты для подобного покрытия могут быть построены независимо для каждого элемента Stub.

В рассматриваемом подходе генерация тестов основана на использовании гидов [5], которые могут быть получены путем обхода структурированной модели.

Технология генерации тестовых сценариев для промышленных проектов [7] предусматривает три этапа: 1) генерация т. н. гидов [8, 9], кортежей событий однозначно направляющих генерацию верифицированных тестовых сценариев; 2) генерация множества детальных символьных сценариев [8], обеспечивающих покрытие функциональности системы по заданному критерию; 3) генерация множества конкретных тестовых сценариев или трасс [6, 9], по которым генерируются тестовые наборы для автоматического тестирования.

В настоящей статье описывается первый этап. Например, по UCM проекту Satellite генерируются структурные гиды (рис. 2). Для диаграммы top генерируется соответствующая директория top, которая содержит в себе папки с именами элементов Stub и гиды, покрывающие возможные поведения системы на верхнем уровне аб-

стракции (1). Каждый из гидов раскрывается как последовательность UCM элементов (2) и содержит в себе элементы Reference (🔗), которые ссылаются на гиды детального описания соответствующего уровня (3). Элемент Reference может содержать множество гидов детального уровня. Каждый из гидов детального уровня также раскрывается как последовательность UCM элементов со ссылками на нижние уровни (4). Такая детализация может быть осуществлена вплоть до самого низкого уровня абстракции системы. В результате структурного подхода образуется дерево гидов проектируемой системы.

Полученное множество гидов может использоваться как для покрытия ветвей отдельного элемента Stub, так и для покрытия системы в целом, в этом случае гиды «склеиваются» таким образом, чтобы каждый полученный результирующий гид начинался в стартовой точке диаграммы и заканчивался после покрытия определенной ветви системы. Например, гид top\_start\_init\_CNI\_change может быть определен как гид, проходящий через ветку корректной конфигурации системы good\_new\_cap\_table с последующим прохождением элемента Stub CNI\_change.

Если «склейку» выполнить для всех комбинаций гидов из каждого Stub, то получится нежелательный «взрыв» тестовых сценариев. Поэтому для тестирования необходим избирательный подход к «склейке» гидов отдельных Stub. Для этого предусмотрен механизм селекции актуальных гидов в множестве гидов каждого Stub (рис. 3),

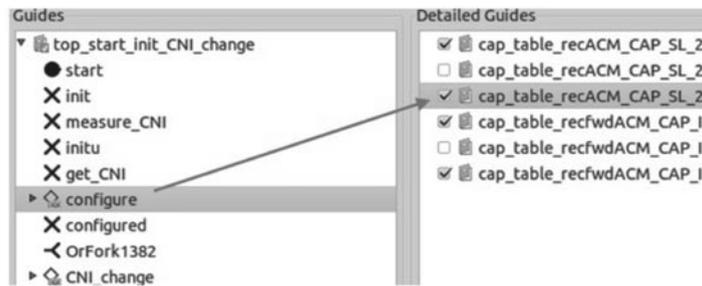


Рис. 3. Выбор гидов из множества гидов, сгенерированных для configure

реализуемый с помощью пометки check-box в выбранном гиде.

Механизм селекции позволяет из всего множества поведений системы выбрать только интересующие: режимы функционирования и соответствующие им поведенческие сценарии, обеспечивающие полное покрытие исходных требований.

Как в случае покрытия ветвей, так и в случае покрытия селектированных пользователем режимов поведения, использование структурированного представления облегчает анализ модели и дает пользователю

возможность контролировать процесс генерации гидов, используемых для получения тестового набора.

Использование структурированного представления UCM модели в процессе тестирования сложных программных комплексов позволяет существенно сократить время на анализ сгенерированного тестового набора, а также предоставляет пользователю возможности по управлению процессом генерации тестов с использованием гидов.

### СПИСОК ЛИТЕРАТУРЫ

1. Z.151 : User requirements notation (URN) – Language definition [Электронный ресурс] / Режим доступа: <http://www.itu.int/rec/T-REC-Z.151-200811-I/en>
2. Никифоров, И.В. Генерация формальной модели системы по требованиям, заданным в нотации USE CASE MAPS [Текст] / И.В. Никифоров, А.В. Петров, Ю.В. Юсупов // Научно-технические ведомости СПбГПУ. Информатика. Телекоммуникации. Управление. –СПб.: Изд-во Политехнического ун-та, 2010. –№ 4 (103). –С. 191–195.
3. ETSI TS 102 441 V 1.1.1 [Электронный ресурс] / Режим доступа: [http://www.etsi.org/deliver/etsi\\_ts/102400\\_102499/102441/01.01.01\\_60/ts\\_102441v010101p.pdf](http://www.etsi.org/deliver/etsi_ts/102400_102499/102441/01.01.01_60/ts_102441v010101p.pdf)
4. Котляров, В.П. Основы тестирования программного обеспечения: Учеб. пособие [Текст] / В.П. Котляров, Т.В. Коликова. –М.: Интернет-Университет Информационных Технологий, 2006. – 256 с.
5. Дробинцев, П.Д. Автоматизация тестирования на основе покрытия пользовательских сценариев [Текст] / П.Д. Дробинцев, В.П. Котляров, И.Г. Черноруцкий // Научно-технические ведомости СПбГПУ. Информатика. Телекоммуникации. Управление. –СПб.: Изд-

- во Политехнического ун-та, 2012. –№ 4 (152). –С. 123–126
6. Recommendation ITU-T Z.120. Message Sequence Chart (MSC ), 11/2000 [Электронный ресурс].
7. Baranov, S. The technology of Automation Verification and Testing in Industrial Projects [Электронный ресурс] / S. Baranov, V. Kotlyarov, A. Letichevsky, P. Drobintsev // Proc. of St. Petersburg IEEE Chapter, International Conf. –May 18-21, 2005. – P. 81–86.
8. Ануреев, И.С. Средства поддержки интегрированной технологии для анализа и верификации спецификаций телекоммуникационных приложений [Текст] / И.С. Ануреев, С.Н. Баранов, Д.М. Белоглазов, Е.В. Бодин, П.Д. Дробинцев, А.В. Колчин, В.П. Котляров, А.А. Летичевский, А.А. Летичевский мл., В.А. Непомнящий, И.В. Никифоров, С.В. Потиеенко, Л.В. Прийма, Б.В. Тютин // Труды СПИИРАН. –2013. –№ 1. –28 с.
9. Колчин, А. Метод генерации тестовых сценариев в среде инсерционного моделирования [Текст] / А. Колчин, В. Котляров, П. Дробинцев // Управляющие системы и машины. –Киев: Академперіодика, 2012. –Т. 6. –С. 43–48.

## REFERENCES

1. Z.151: User requirements notation (URN) – Language definition. Available <http://www.itu.int/rec/T-REC-Z.151-200811-I/en>
2. **Nikiforov I.V., Petrov A.V., Yusupov Yu.V.** Generatsiia formal'noi modeli sistemy po trebovaniiam, zadannym v notatsii USE CASE MAPS [Generation of formal model of a system from requirements specified in use case map] / Nauchno-tehnicheskie vedomosti SPbGPU. Informatika. Telekommunikatsii. Upravlenie [St. Petersburg State Polytechnical University Journal. Computer Science. Telecommunications and Control Systems]. – St. Petersburg: Izd-vo Politehnicheskogo un-ta, 2010. – № 4 (103). – S. 191–195. (rus)
3. ETSI TS 102 441 V 1.1.1. Available [http://www.etsi.org/deliver/etsi\\_ts/102400\\_102499/102441/01.01.01\\_60/ts\\_102441v010101p.pdf](http://www.etsi.org/deliver/etsi_ts/102400_102499/102441/01.01.01_60/ts_102441v010101p.pdf)
4. **Kotliarov V.P., Kolikova T.V.** Osnovy testirovaniia programmogo obespecheniia: Ucheb. posobie. – Moscow: Internet-Universitet Informatsionnykh Tekhnologii, 2006. – S. 256. (rus)
5. **Drobintsev P.D., Kotliarov V.P., Chernorutskii I.G.** Avtomatizatsiia testirovaniia na osnove pokrytiia pol'zovatel'skikh stsenariiev [Approach for testing automation based on user scenarious] / Nauchno-tehnicheskie vedomosti SPbGPU. Informatika. Telekommunikatsii. Upravlenie [St. Petersburg State Polytechnical University Journal. Computer Science. Telecommunications and Control Systems]. – St. Petersburg: Izd-vo Politehnicheskogo un-ta, 2012. – № 4 (152). – S. 123–126. (rus)
6. Recommendation ITU-T Z.120. Message Sequence Chart (MSC), 11/2000.
7. **Baranov S., Kotlyarov V., Letichevsky A., Drobintsev P.** The technology of Automation Verification and Testing in Industrial Projects / Proc. of St. Petersburg IEEE Chapter, International Conf.; May 18–21, 2005. – P. 81–86. (rus)
8. **Anureev I.S., Baranov S.N., Beloglazov D.M., Bodin E.V., Drobintsev P.D., Kolchin A.V., Kotliarov V.P., Letichevskii A.A., Letichevskii A.A. ml., Nepomniashchii V.A., Nikiforov I.V., Potienko S.V., Priima L.V., Tiutin B.V.** Sredstva podderzhki integrirovannoi tekhnologii dlia analiza i verifikatsii spetsifikatsii telekommunikatsionnykh prilozhenii / Trudy SPIIRAN. – 2013. – № 1. – 28 s. (rus)
9. **Kolchin A., Kotliarov V., Drobintsev P.** Metod generatsii testovykh stsenariiev v srede insertsiionnogo modelirovaniia / Upravliaiushchie sistemy i mashiny. – Kiev: Akadempriodika, 2012. – T. 6. – S. 43–48.

**ДРОБИНЦЕВ Павел Дмитриевич** – доцент кафедры информационных и управляющих систем Санкт-Петербургского государственного политехнического университета, кандидат технических наук.

195251, Россия, Санкт-Петербург, ул. Политехническая, д. 29.

**DROBINTSEV, Pavel D.** St. Petersburg State Polytechnical University.

195251, Politehnicheskaya Str. 29, St. Petersburg, Russia.

**НИКИФОРОВ Игорь Валерьевич** – аспирант кафедры информационных и управляющих систем Санкт-Петербургского государственного политехнического университета.

195251, Россия, Санкт-Петербург, ул. Политехническая, д. 29.

**NIKIFOROV, Igor V.** St. Petersburg State Polytechnical University.

195251, Politehnicheskaya Str. 29, St. Petersburg, Russia.

**КОТЛЯРОВ Всеволод Павлович** – профессор кафедры информационных и управляющих систем Санкт-Петербургского государственного политехнического университета, кандидат технических наук.

195251, Россия, Санкт-Петербург, ул. Политехническая, д. 29.

E-mail: vpk@spbstu.ru

**KOTLYAROV, Vsevolod P.** St. Petersburg State Polytechnical University.

195251, Politehnicheskaya Str. 29, St. Petersburg, Russia.

E-mail: vpk@spbstu.ru



УДК 004.722+004.415.23

*А.С. Колосов, Ю.А. Богоявленский*

**ПАРАЛЛЕЛЬНЫЙ АЛГОРИТМ  
ПОСТРОЕНИЯ ГРАФА ИКТ-ИНФРАСТРУКТУРЫ  
ИНТЕРНЕТ-ПРОВАЙДЕРА**

*A.S. Kolosov, Yu.A. Bogoyavlensky*

**A PARALLEL ALGORITHM FOR CONSTRUCTING A GRAPH OF A LOCAL  
INTERNET SERVICE PROVIDER'S ICT-INFRASTRUCTURE**

Граф ИКТ-инфраструктуры (Сеть) необходим для решения большинства задач управления Сетью. При наличии в Сети большого числа современных мобильных устройств необходимо постоянное обновление графа Сети для поддержания его в актуальном состоянии, что требует повышения быстродействия существующих подходов построения таких графов.

Представлен параллельный алгоритм построения графа Сети на основе данных из MIB маршрутизаторов, получаемых по протоколу SNMP, и его реализация. Описанные эксперименты подтверждают трехкратное ускорение построения графа Сети ПетрГУ при использовании параллельного алгоритма.

**АНАЛИЗ СЕТЕЙ. ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОЕКТИРОВАНИЕ. ГРАФ СЕТИ. РАСПАРАЛЛЕЛИВАНИЕ. СЕТЕВОЕ УПРАВЛЕНИЕ.**

An ICT-infrastructure (Network) graph is a necessary tool for solving of most network management problems. While the number of modern mobile devices in a Network grow up, permanent update of the Network graph is required to keep it actual and consistent. Therefore, it's required to develop more efficient and fast methods of building such graphs.

In this paper we present parallel algorithm of Network graph discovery using routers' MIB data obtained with SNMP and its implementation. Described experiments confirm three-fold speed-up of the PetrSU Network graph building when using parallel algorithm.

**NETWORK ANALYSIS. OBJECT-ORIENTED MODELLING. NETWORK TOPOLOGY. CONCURRENCY. NETWORK MANAGEMENT.**

Данные об аппаратных элементах ИКТ-инфраструктуры и их взаимосвязях (далее – граф Сети) являются базовым инструментом систем сетевого управления локальных поставщиков сетевых услуг (лПСУ).

Разрабатываемая в Петрозаводском государственном университете (ПетрГУ) экспериментальная платформа Nest [1, 2] предоставляет исследователю модели и методов сетевого управления средства автоматизированного построения графа Сети, его визуализации, выполнения запросов. Граф Сети определяется как подграф объектного графа модели архитектуры лПСУ SON [2]. Подсистема NesToro [1] обеспечивает автоматизированное построение и хранение

графа Сети (сетевой уровень) в БД с помощью последовательного алгоритма.

Экспериментальная эксплуатация подсистемы NesToro показала ее приемлемую производительность, однако, принимая во внимание рост количества мобильных устройств, способных быстро менять свое положение в графе, необходимость оперативного отслеживания подключений и потенциальный рост количества элементов и связей в графах Сетей, следует признать актуальной задачу фиксации быстрых изменений графа, для решения которой необходимо увеличить быстродействие алгоритма [1].

В данной статье предлагается парал-

лельный алгоритм построения и хранения в объектной БД графа Сети. Описанные в статье результаты экспериментов свидетельствуют о примерно трехкратном увеличении быстродействия процедуры построения графа Сети ПетрГУ по сравнению с традиционным подходом.

### Параллельный алгоритм построения графа Сети

Распространенным способом определения возможности распараллеливания алгоритма [3] является выявление в его ярусно-параллельной форме (ЯПФ) независимых вершин одного яруса, каждая из которых может выполняться параллельно.

На рис. 1 представлена ЯПФ графа алгоритма [1] для Сети, содержащей один маршрутизатор с двумя канальными интерфейсами, один из которых подключен к двум IP-подсетям. Каждая вершина графа алгоритма представляет набор операций построения вершины графа Сети, выполняющих получение данных и запись в БД для маршрутизатора (R), канального интерфейса (LI), сетевого интерфейса (NI), IP-подсети (N) и устройства (D). Здесь вершины одного яруса являются независимыми, т. е. могут выполняться параллельно.

Ширина данной ЯПФ равна максимальному количеству подключенных сете-

вых устройств среди всех маршрутизаторов Сети, которое в современных ЛПСУ может достигать нескольких тысяч. Таким образом, непосредственная реализация параллелизма, представленного в ЯПФ (рис. 1), потребует реализации такого же количества параллельных ветвей. В то же время важным требованием к параллельному алгоритму является возможность его эффективной работы на современных серверных ЭВМ среднего класса (десятки ядер), что обеспечит возможность его широкого применения. Другое важное требование к алгоритму – масштабируемость, т. е. независимость его реализации от количества ядер.

Для выполнения двух этих требований был предложен алгоритм 1 (рис. 2), описывающий процедуру построения графа Сети как взаимодействие процесса формирования вершин графа при его обходе с параллельно выполняющимися задачами получения данных от сетевых устройств. Число этих задач ограничивается размером пула рабочих потоков.

Для текущей вершины создается задача формирования данных вершины (функция graphTraverse), выполняющая их обновление по протоколу SNMP (функция update) согласно [1]. Для обнаруженных при обновлении вершин, связанных с текущей, создаются аналогичные задачи, ко-

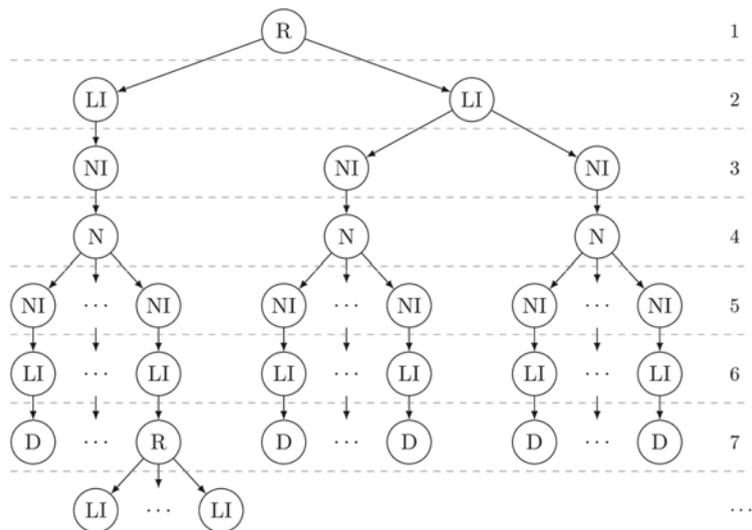


Рис. 1. Ярусно-параллельная форма алгоритма построения графа Сети

**Вход:** *threadPool, startNode, seenSet*

```

1: function GRAPHTRAVERSE(node)
2: for all e in CHILDREN(node) \ seenSet do
3:   seenSet ← seenSet + e
4:   UPDATE(e)
5:   task ← {GRAPHTRAVERSE(e)}
6:   ADDTASK(threadPool, task)
7: end for
8: end function
9: task ← {GRAPHTRAVERSE(startNode)}
10: INVOKE(threadPool, task)

```

Рис. 2. Алгоритм 1: параллельный алгоритм построения графа Сети

торые добавляются в очередь пула потоков (вызов функции `addTask`) и выполняются по мере его освобождения. Запуск процедуры построения графа Сети начинается с инициализации и запуска в контексте пула потоков (вызов функции `invoke`) задачи `graphTraverse` для корневой вершины `startNode`. Предотвращение заикливания обеспечивается за счет сопровождения множества просмотренных вершин `seenSet`. Отметим также, что построение всех связей вершины внутри одного потока позволяет не синхронизировать доступ к объекту данной вершины.

### Реализация подсистемы NesToro

Реализация параллельного алгоритма построения графа Сети, а также ряд архитектурных проблем последовательной версии, не позволявших расширять процедуры сбора данных о Сети независимо друг от друга, потребовали разделения подсистемы NesToro [1] на три независимых модуля (рис. 3): управление процедурой построения графа Сети, получение данных об элементах Сети и связях между ними, построение отдельных элементов графа в соответствии с моделью SON.

С целью отделения процедур построения элементов графа в БД от процедур получения данных применена идея индуктивных графов [4], в соответствии с которой при продвижении по графу Сети для каждого устройства и связанных с ним вершин графа Сети известен контекст, предоставляющий следующую информацию: описание устройства; тип устройства (маршрутизатор

или конечный узел); таблица интерфейсов и их сетевых адресов; таблица маршрутизации; список устройств, разделяющих широкоэвещательный домен с данным устройством.

Рассматриваемый подход реализован в модуле получения данных в базовом классе `ProviderContext` и его наследниках `SnmpContext` (получение данных по SNMP) и `IsolatedContext` (получение данных об оконечных устройствах).

Процедуры создания и связывания вершин графа Сети на основе данных, получаемых от конкретного контекста, реализованы в т. н. классах-поставщиках (базовый класс — `NElementProvider`) модуля построения элементов графа. Экземпляры классов-поставщиков реагируют на события обращений к свойствам обслуживаемых вершин графа, реализуя паттерн «Издатель-Подписчик» [5].

Интерфейс для настройки и запуска процедуры построения графа Сети предоставляется классом `Collector`. Последовательный алгоритм [1] построения реализован в классе `BasicCollector` и используется для тестирования. Параллельный алгоритм реализован в классе `ConcurrentCollector` с помощью программного каркаса `Fork/Join` из состава пакета `java.util.concurrent JDK7`.

### Тестирование и эксперименты

Процесс тестирования был разделен на три части: 1) тестирование процедур построения узлов графа Сети (классы-поставщики) с помощью специальной тестовой реализации контекста, которая предсказуемо отве-

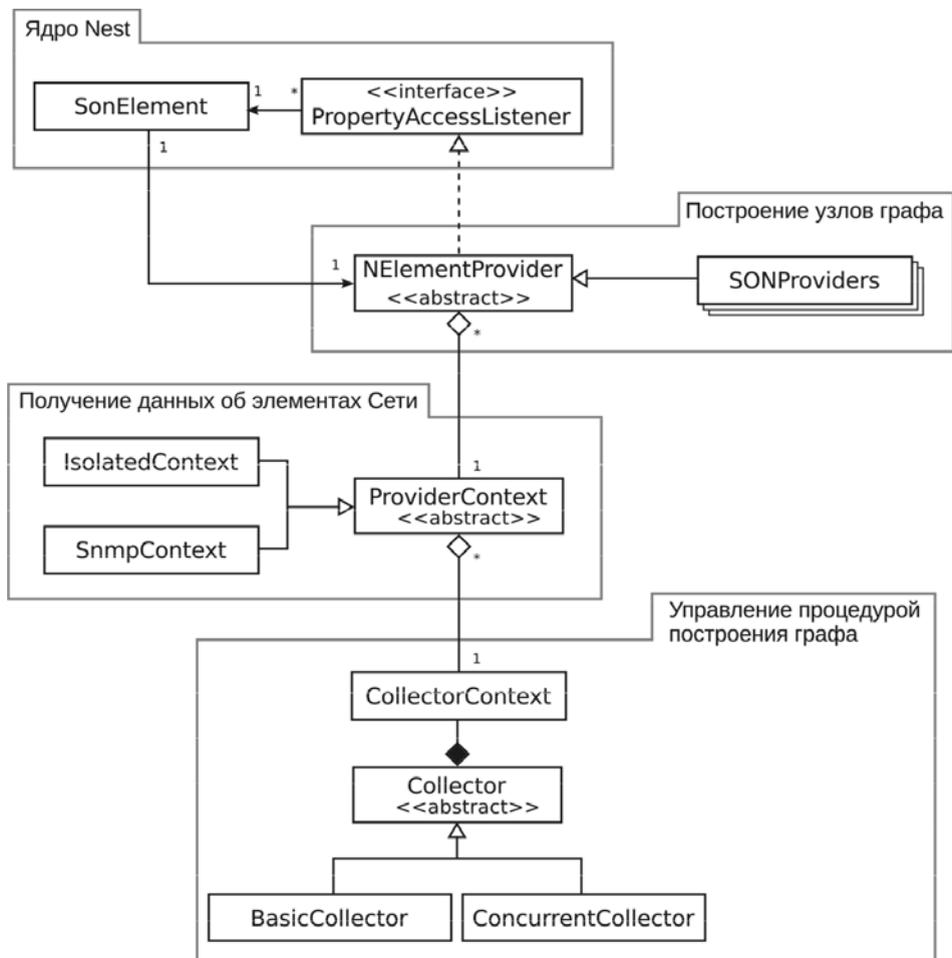


Рис. 3. UML-диаграмма классов подсистемы NesToro

**Результаты экспериментов**

Участок сети	Количество ядер	Размер пула	Время построения (параллельный алгоритм), с	Время построения (последовательный алгоритм), с
1	1	4	4,70	7,12
2	1	2	15,49	22,30
3	1	9	21,53	41,17
1	2	9	4,62	7,12
2	2	16	18,10	22,30
3	2	15	21,54	41,17
1	4	14	4,59	7,12
2	4	15	14,62	22,30
3	4	15	14,42	41,17
1	8	11	4,56	7,12
2	8	16	8,90	22,30
3	8	16	12,14	41,17

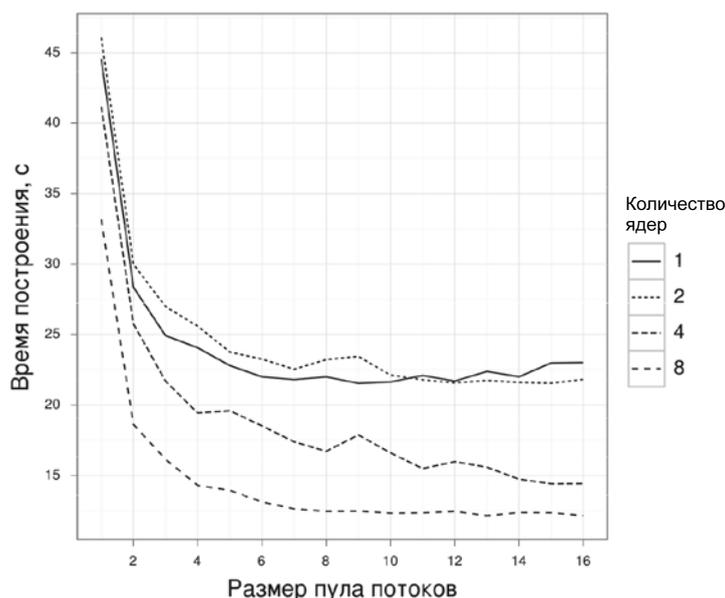


Рис. 4. Зависимость времени построения графа Сети ПетрГУ от размера пула потоков при работе на разном количестве ядер

чает на запросы классов-поставщиков, позволяя оценить корректность построения узлов графа путем сравнения ожидаемых результатов с реальными; 2) тестирование процедур получения данных из MIB сетевых устройств по протоколу SNMP с использованием «поддельного» (*mock*) объекта [6], имитирующего поведение SNMP-сессии; 3) сверка двух графов Сети ПетрГУ, построенных с помощью последовательного и параллельного алгоритмов, в ходе которой различий обнаружено не было.

Всего было разработано 55 тестов, из них 21 тест для классов-поставщиков и 13 тестов для SNMP-контекста. Обнаружено и исправлено 16 ошибок. Разработка и выполнение тестов производились в среде JUnit.

В ходе экспериментов выполнялось построение графа для различных участков Сети ПетрГУ: 1) ЛВС кафедры информатики и математического обеспечения (один опрашиваемый маршрутизатор, 50 устройств); 2) ЛВС учебных корпусов ПетрГУ (два маршрутизатора, 1 500 устройств); 3) вся вычислительная сеть ПетрГУ (четыре маршрутизатора, 1 700 устройств). Эксперименты проводились на ЭВМ с процессором Intel Xeon 2,5 ГГц (8 ядер), ОЗУ 1 ГБ,

ОС Linux 3.1.0 в среде выполнения Java JRE 1.7.0\_09.

Оценки времени производились для каждой комбинации параметров: номер участка сети, количество выделенных для задачи ядер процессора, размер пула потоков. Каждый эксперимент выполнялся три раза, в качестве окончательных оценок взяты средние значения полученных характеристик. В таблице приведены результаты для комбинаций с теми размерами пула, при которых получено минимальное время построения графа.

По сравнению с результатами эксперимента, проводимого на идентичной аппаратной конфигурации с использованием последовательного алгоритма [1], время построения графа Сети ПетрГУ без учета сетевого взаимодействия и работы с БД сократилось с 41 до 12 с.

Из таблицы и графика (рис. 4) видно, что увеличение размера пула потоков позволяет получить выигрыш в производительности при работе даже на одном ядре процессора.

Представленный параллельный алгоритм автоматизированного построения графа Сети имеет существенно более высокую производительность, чем его последо-

вательная версия, что подтверждается экспериментами. Так, время построения графа Сети ПетрГУ при использовании пула из

16 потоков примерно в три раза меньше, чем время построения того же графа с помощью последовательного алгоритма.

#### СПИСОК ЛИТЕРАТУРЫ

1. **Богоявленский, Ю.А.** Организация и автоматизированная поддержка объектной базы данных графа ИКТ-инфраструктуры поставщика услуг Интернета [Текст] / Ю.А. Богоявленский, А.С. Колосов // Научно-технические ведомости СПбГПУ. Информатика. Телекоммуникации. Управление. —СПб.: Изд-во Политехнического университета, 2011. —№ 3 (126). —С. 27–36.
2. **Богоявленский, Ю.А.** Прототип экспериментальной платформы Nest для исследования моделей и методов управления ИКТ-инфраструктурами локальных поставщиков услуг Интернет [Текст] / Ю.А. Богоявленский // Программная инженерия. —2013. —№ 2.

—С. 11–20.

3. **Воеводин, В.В.** Параллельные вычисления [Текст] / В.В. Воеводин, Вл.В. Воеводин. —СПб.: ВВХ-Петербург, 2002. —608 с.

4. **Erwig, M.** Inductive graphs and functional graph algorithms [Text] / M. Erwig // J. of Functional Programming. —2001.

5. **Гамма, Э.** Приемы объектно-ориентированного проектирования. Паттерны проектирования [Текст] / Э. Гамма, Р. Хелм, Р. Джонсон [и др.]. —СПб.: Питер, 2001. —368 с.

6. **Бек, К.** Экстремальное программирование: разработка через тестирование [Текст] / К. Бек. —СПб.: Питер, 2003.

#### REFERENCES

1. **Bogoiavlenskii Iu.A., Kolosov A.S.** Organizatsiia i avtomatizirovannaia podderzhka ob"ektnoi bazy dannykh grafa IKT-infrastruktury postavshchika uslug Interneta [*Organization and automated maintenance of an ict-infrastructure graph object database of an Internet service provider*] / Nauchno-tekhicheskie vedomosti SPbGPU. Informatika. Telekommunikatsii. Upravlenie [*St. Petersburg State Polytechnical University Journal. Computer Science. Telecommunications and Control Systems*]. St. Petersburg: Izd-vo Politekhnikheskogo un-ta, 2011. — № 3 (126). — S. 27–36. (rus)

2. **Bogoiavlenskii Iu. A.** Prototip eksperimental'noi platformy Nest dlia issledovaniia modelei i metodov upravleniia IKT-infrastrukturami lokal'nykh

postavshchikov uslug Internet / Programmnaia inzheneriia. —2013. — № 2. — S. 11–20. (rus)

3. **Voevodin V.V., Voevodin V.V.** Parallel'nye vychisleniia. — St. Petersburg: BVKh-Petersburg, 2002. — 608 s. (rus)

4. **Erwig M.** Inductive graphs and functional graph algorithms / J. of Functional Programming. — 2001.

5. **Gamma E., Khelm R., Dzhonson R. i dr.** Priemy ob"ektno-orientirovannogo proektirovaniia. Patterny proektirovaniia. —St. Petersburg: Piter, 2001. — 368 s. (rus)

6. **Bek K.** Ekstremal'noe programmirovanie: razrabotka cherez testirovanie. — St. Petersburg: Piter, 2003. (rus)

**КОЛОСОВ Александр Сергеевич** — старший преподаватель кафедры информатики и математического обеспечения Петрозаводского государственного университета.

185910, Россия, Республика Карелия, г. Петрозаводск, пр. Ленина, д. 33.

E-mail: akolosov@cs.karelia.ru

**KOLOSOV, Aleksandr S.** *Petrozavodsk State University.*

185910, Lenin Str. 33, Petrozavodsk, Republic of Karelia, Russia.

E-mail: akolosov@cs.karelia.ru

**БОГОЯВЛЕНСКИЙ Юрий Анатольевич** — заведующий кафедрой информатики и математического обеспечения Петрозаводского государственного университета, кандидат технических наук.

185910, Россия, Республика Карелия, г. Петрозаводск, пр. Ленина, д. 33.

**BOGOYAVLENSKIY, Yuriy A.** *Petrozavodsk State University.*

185910, Lenin Str. 33, Petrozavodsk, Republic of Karelia, Russia.



УДК 004.415

*И.В. Никифоров, П.Д. Дробинцев, В.П. Котляров*

## **ИНТЕГРАЛЬНЫЕ КРИТЕРИИ ПРОВЕРКИ ТРЕБОВАНИЙ К ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ**

*I.V. Nikiforov, P.D. Drobintsev, V.P. Kotliarov*

### **INTEGRATED CRITERIA OF SOFTWARE REQUIREMENT CHECKING**

Описаны методы разработки критериев проверки требований, которые позволяют обеспечить стопроцентное покрытие тестовыми сценариями исходных спецификаций на систему. Предложено использовать подход критериальных цепочек, используемых в матрице отслеживания покрытия. По критериальным цепочкам в автоматическом режиме создаются тестовые сценарии.

Изучены методы генерации сценариев и отбор сценариев, удовлетворяющих выбранному интегральному критерию покрытия.

Использование предложенных интегральных критериев проверки требований на этапе разработки формальной спецификации системы позволяет покрыть тестовыми сценариями все исходные требования системы.

**ДИЗАЙН СПЕЦИФИКАЦИЙ. ТРЕБОВАНИЯ. МАТРИЦА ОТСЛЕЖИВАНИЯ. ГИДЫ.**

The paper describes developing methods of tests for the requirements checking that provide 100 % coverage of the initial specifications for the system by test cases. Proposed approach of criterion chains are used in the traceability matrix. The criteria chains are used to automatic generation of test scenarios. Also the article describes methods for generating scenarios and selection of scenarios that satisfy the selected integral coverage criteria.

The use of suggested integrated tests criteria for the requirements checking at the stage of a formal specification of the system, allows to cover by test scenarios all the initial requirements of the system.

**DESIGN SPECIFICATION. REQUIREMENTS. TRACEABILITY MATRIX. GUIDES.**

Среди основных проблем автоматизации разработки и тестирования программного обеспечения промышленных приложений в виде сосредоточенных модулей или распределенных сетей непременно отмечается проблема обработки сложных и больших по объему спецификаций требований. Документы, фиксирующие спецификации требований, пишутся, как правило, на естественном языке и могут содержать сотни и тысячи пунктов требований. В силу этого задача формализации требований для описания поведенческих сценариев, используемых для разработки автоматических тестов или ручных тестовых процедур, характеризуется как задача огромной сложности и трудоемкости.

Применимость формальных методов в промышленности в огромной степени определяется тем, насколько адекватен язык формализации принятой инженерной

практике, в которую вовлечены не только собственно разработчики кода и тестирующие, но и заказчики, руководители проектов разных уровней, маркетологи и другие специалисты. Ясно, что никакой чисто логический язык не подходит для адекватной формализации требований, которая бы одновременно сохранила семантику разрабатываемого приложения и удовлетворяла всех «причастных лиц» [1].

В современной проектной документации формулировка исходных требований задается либо конструктивно, когда из текста требования на естественном языке удается реконструировать процедуру контроля или сценарий проверки выполнения данного требования, либо неконструктивно, когда заданное в требовании свойство не содержит пояснения способа его проверки.

**Проверка выполнения требований.** Процедура проверки требования – это точная

последовательность причин и следствий некоторых активностей (кодируемых действиями, сигналами, состояниями), в результате анализа которой можно утверждать, что данное требование выполнено или нет. Подобная процедура проверки может использоваться в качестве критерия выполнения конкретного требования, т. е. стать критериальной процедурой. В дальнейшем изложении для критериальной процедуры будем использовать термин *последовательность* или «цепочка» событий.

Отслеживая в поведенческом сценарии системы (гипотетическом, реализованном в модели или в реальной системе) факты выполнения критериальной процедуры, можно утверждать, что соответствующее требование в анализируемой системе удовлетворено.

Процедура проверки требования (цепочка) формулируется путем задания для всех элементов цепочки следующей информации:

- условий (причин), требующихся для активизации некоторой активности;
- самой активности, подлежащей исполнению при данных условиях;
- следствий – наблюдаемых (измеряемых) результатов исполнения указанной активности.

Для описания причин и следствий используются сигналы, сообщения или транзакции, обычные в коммуникациях инстанций реактивной системы [2], а также состояния переменных в виде значений или ограничений на области допустимых значений. Отслеживая изменения в состояниях, производимых активностями цепочек, можно наблюдать за покрытием соответствующих цепочек. При анализе допустимо рассматривать прямой переход из состояния в состояние с пустой активностью, а в случае недетерминизма – альтернативные варианты изменения состояний.

Проблемы неконструктивного задания требований преодолеваются в процессе разработки процедур проверки выполнения требований на пользовательских или межкомпонентных интерфейсах. То есть сформулированное требование о наличии некоторой функциональности или свойства надо

проинтерпретировать на функциях пользовательского интерфейса или API (Application Program Interface) компонент, описывающих или именующих закодированную функциональность так, чтобы на основе этой интерпретации можно было задать процедуру проверки наличия соответствующей функциональности в приложении.

Таким образом, критерием выполнения требований могут служить цепочки, содержащие последовательности активностей и состояний; помимо этого возможны случаи, в которых критерий выполнения некоторого требования задается не одной, а несколькими цепочками.

**Исходные документы, фиксирующие требования к приложению.** При формулировке технических требований в промышленных проектах обычно используются следующие виды документов:

- Marketing Requirement Specification (MRS) – перечисление новой функциональности с точки зрения ее потребления пользователем, часто описываемой в виде пользовательского сценария;
- Technical Requirement Specification (TRS) – перечисление всей функциональности, подлежащей разработке, с кратким уточнением каждой функции;
- Functional Requirement Specification (FRS) – детальное описание всей функциональности, подлежащей разработке, достаточное для разработки тестовых наборов для проверки программного продукта на всех этапах жизненного цикла.

Формулировка каждого требования в этих документах чаще всего осуществляется на естественном языке и выражается одним из двух способов:

• в виде поведенческого требования, когда из текста требования на естественном языке удается реконструировать сценарий (процедуру) выполнения данного требования;

• в виде неповеденческого требования, задающего состав, структуру или желаемое наличие некоторого свойства без пояснения способа его потребления (способа проверки).

Поведенческие требования в силу конструктивности своего задания допускают

для проверки своей реализованности статические и/или динамические методы верификации и тестирования.

Неповеденческие требования по определению заданы неконструктивно и требуют дополнительной информации для воссоздания конструктивного сценария их проверки, чтобы верификация и тестирование стали применимы.

Формализация любых конструктивно заданных требований возможна, как возможен и эффективный автоматизированный анализ требований к программному изделию, и реализована в технологии VRS/TAT [3].

В технологии VRS/TAT для высокоуровневого описания модели, в рассматриваемой технологической цепочке используется нотация Use Case Maps (UCM) [4], а инструменты автоматизации проверки и генерации работают с моделью на языке базовых протоколов [5].

**Матрица отслеживания.** Формализация требований верификационного проекта начинается с разработки матрицы отслеживания TRM – Traceability matrix (на рис. 1 TRM для конкретного проекта представлена в виде таблицы). Столбцы 4 и 6 содержат идентификатор требования, употребляемый в TRS или FRS, а также полный текст тре-

бования, подлежащий формализации.

Следующий важный этап создания TRM – разработка критериальных цепочек для требований. Примеры цепочек представлены в колонке 8 «Scenario of requirement verifying». Например, в строке 291 цепочка для покрытия требования TRS\_18081-2209 обозначена в колонке 10 «Scenario ID» как scen#045 и состоит из сценария scen#064, за которым исполняются еще две указанные активности. Для покрытия требования TRS\_18081-2212 (строка 293) в столбце 8 указаны две цепочки: первая для автоматического, вторая для ручного тестирования. В требовании TRS\_18081-2209 использованы сообщения «Do Not Disturb» и «My Availability changed to Do Not Disturb», появляющиеся на экране мобильного телефона. Информация о том, в каком месте экрана они появляются и с какими активностями связаны, извлечена из дополнительной технической документации, описывающей пользовательский интерфейс телефона. Следует заметить, что при создании критериальных цепочек строится модель верифицируемой функциональности, в процессе чего вводится много переменных состояния, типов, агентов, инстанций и т. п.

**Разработка интегральных критериев покрытия требований.** Выше отмечалась ха-

1	4	6	8	10	12	28
Req Id	Source Text of the Title, Requirement, or Comment	Scenario of requirement verifying	Scenario ID	Traceability	Traces	
290	TRS_18081-2210[7] - See Section "Self Presence Reporting" and Section "Silent Mode Settings".				N/A	
291	TRS_18081-2209[6] - When My Availability is set to Do Not Disturb, the user's presence status is reported to the server as Do Not Disturb.	scen#64 + 1. Select the option 'Do Not Disturb' in the KPIT_MyStatus menu and press the LSK 'Select' or the HK 'Center Select'. 2. Check that the KPIT_Confirmation_Notice menu (body: My Availability changed to Do Not Disturb) appeared	scen#045	:TRS_18081_2209 KPIT_MyStatus06 KPIT_Confirmation_N		2209_v1_ verdict0001.mpr
292	TRS_18081-2211[7] - See Section "Self Presence Reporting".				N/A	
293	TRS_18081-2212[7] - When My Availability is set to Do Not Disturb, the user's ability to originate PTT calls shall not be affected.	1. Check that state of the handset is DND 2. Go to the KPIT_Contacts menu, select a contact (it's own state shall be not DND - for more information see 8.2.2.1.1 Individual Contact Presence) 3. Press the PTT button ----- manual test: Check that a PTT session starts		:TRS_18081_2212 KPIT_Contacts39		2212_v3_ verdict0002.mpr

Рис. 1. TRM-матрица отслеживания

рактерная черта технологии VRS/TAT – наличие специальных критериев проверки покрытия для каждого требования. Перечислим критерии, относящиеся к требованиям, по степени роста их мощности (задача каждого критерия – стопроцентное покрытие требований):

- степень покрытия сгенерированными сценариями подмножества событий, использованных в критериальных цепочках;
- степень покрытия сгенерированными сценариями подмножества цепочек (состоящих их событий и состояний переменных), не менее чем по одной для каждого требования;
- степень покрытия сгенерированными сценариями всего множества цепочек, задающего интегральный критерий покрытия требований.

Критерии должны применяться гибко и могут изменяться в зависимости от условий генерации сценариев.

Для высокоуровневого описания модели, в рассматриваемой технологической цепочке используется нотация UCM (рис. 2), а инструменты автоматизации проверки и генерации работают с моделью на языке базовых протоколов.

**Генерация сценариев и отбор сценариев, удовлетворяющих выбранному интегральному критерию покрытия.** Генерация трасс осуществляется символьным и конкретным трассовыми генераторами STG (Symbolic Trace Generator) и CTG (Concrete Trace Generator), реализующими эффектив-

ные алгоритмы проверки моделей (Model Checking). Основной проблемой трассовой генерации является «взрыв» вариантов перебора при генерации сценариев (трасс) из ВР, которые формализуют сценарные события, условия их реализации и соответствующее изменение состояния модели после реализации. Инструментом решения является фильтрация вариантов генерации на основе многочисленных ограничений, специально выбираемых перед циклом генерации трасс [6]: ограничения на число конечных (Goal) и промежуточных (Visited) состояний; ограничения на максимальное количество протоколов, используемых в трассе; максимальное количество трасс, генерируемых в одном цикле генерации.

Специальные ограничения задаются последовательностями событий UCM модели, направляющими процесс генерации в интересующем пользователя поведении модели (т. н. гиды – Guides). Используется два этапа для генерации тестовых сценариев по гидам. На первом этапе на основе UCM модели создаются гиды, обеспечивающие заданные критерии покрытия поведения системы. На втором этапе гиды в нотации UCM (рис. 3 а) преобразуются в гиды на языке базовых протоколов (рис. 3 б), и под их управлением производится генерация трасс (рис. 3). Важно отметить, что в гидах фиксируются лишь главные контрольные точки поведения, в то время как трасса, полученная по гиду, содержит детальную последовательность элементов поведения.

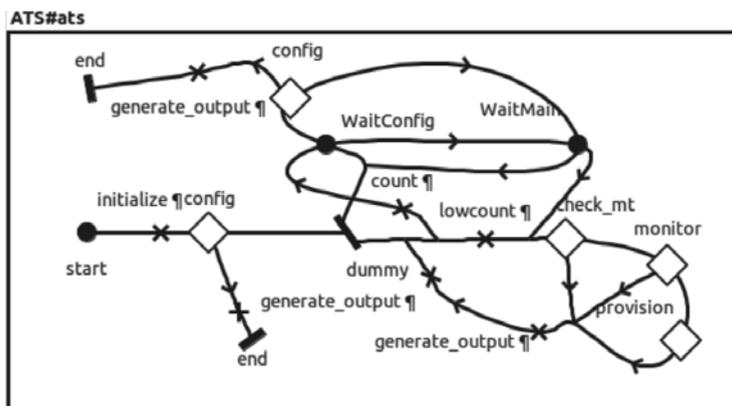


Рис. 2. Пример описания поведенческой модели на языке UCM

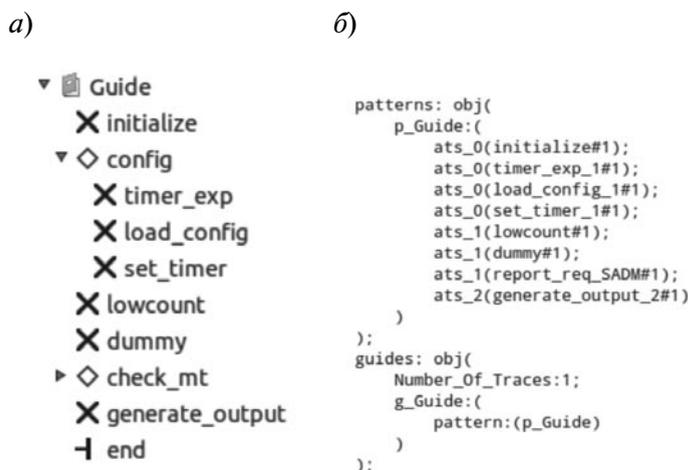


Рис. 3. Гиды: *a* – в нотации UCM; *б* – на языке Guide Language VRS

Такой подход к генерации существенно уменьшает влияние комбинаторного взрыва на время генерации трасс при обходе дерева поведения проектируемой системы.

**Результаты применения в пилотных проектах.** В таблице приведены результаты применения интегрированной технологии проектирования и тестирования в области разработки беспроводных телекоммуникационных приложений. В результате получено существенное снижение трудозатрат и увеличение качества программного продукта.

Результатом работы является усовершенствованная технология, интегрирующая верификацию и тестирование программных проектов, которая обеспечивает:

полную автоматизацию процесса разработки промышленного программного продукта с контролем реализации семантики требований;

генерацию модели приложения и символьных поведенческих сценариев, стопроцентно покрывающих поведенческие свойства приложения;

**Результаты применения технологии в пилотных проектах**

Проект	Количество требований	Количество базовых протоколов	Степень покрытия требований, %	Количество найденных и исправленных ошибок (всего/существенных )	Трудоемкость (человеко-неделя)
Модуль 1 беспроводной сети (БС)	400	127	75	142/11	5,6
Модуль 2 БС	730	192	80	106/18	12
Модуль верхнего уровня БС	148	205	100	68/23	11
Модуль связи клиентов и администратора БС	106	163	100	42/8	6
Модуль ПО мобильного телефона	200	170	100	96/10	7
Модуль интеграции в системе управления автомобилем	2220	1533	65	573/57	17
Радиоприемник автомобиля	26	26	100	14/4	3

автоматическую конкретизацию символьных трасс в соответствии с планом тестирования;

высокую автоматизацию процесса разработки и управления качеством программного продукта.

#### СПИСОК ЛИТЕРАТУРЫ

1. **Баранов, С.** Индустриальная технология автоматизации тестирования мобильных устройств на основе верифицированных поведенческих моделей проектных спецификаций требований [Текст] / С. Баранов, В. Котляров, А. Летичевский // Тр. Междунар. науч. конф. Космос, астрономия и программирование. – СПб.: Изд-во СПбГУ, 2008. – С. 134–145.
2. **Manna, Z.** The Temporal Logic of Reactive and Concurrent Systems [Text] / Z. Manna, A. Pnueli. – Springer-Verlag, 1992.
3. **Baranov, S.** The technology of Automation Verification and Testing in Industrial Projects [Электронный ресурс] / S. Baranov, V. Kotlyarov, A. Letichevsky, P. Drobintsev // Proc. of St. Petersburg IEEE Chapter, International Conf. May 18-21, 2005. – P. 81–86.
4. Recommendation ITU-T Z.151 [Электрон-

ный ресурс] / User requirements notation (URN), 11/2008.

5. **Letichevsky, A.** Basic Protocols, Message Sequence Charts, and the Verification of Requirements Specifications [Text] / A. Letichevsky, J. Kapitonova, A. Letichevsky Jr., V. Volkov, S. Baranov, V. Kotlyarov, T. Weigert // Proc of ISSRE04 Workshop on Integrated-reliability with Telecommunications and UML Languages (ISSRE04:WITUL). – IRISA Rennes France, 02.11.2004.

6. **Колчин, А.** Направленный поиск в верификации формальных моделей [Текст] / А. Колчин // Тези доп. міжнар. конф. Теоретичні та прикладні аспекти побудови програмних систем ТАAPSD'2007. – Бердянск: НАУКМА, Національний ун-т ім. Т.Г. Шевченка, Ін-т програмних систем НАН України, 2007. – С. 256–258.

#### REFERENCES

1. **Baranov S., Kotliarov V., Letichevskii A.** Industrial'naia tekhnologiiia avtomatizatsii testirovaniia mobil'nykh ustroystv na osnove verifitsirovaniykh povedencheskikh modelei proektnykh spetsifikatsii trebovaniy / Trudy mezhdunar. nauch. konf. Kosmos, astronomiia i programmirovaniye. – St Petersburg: Izd-vo SPbGU, 2008. – S. 134–145. (rus)
2. **Manna Z., Pnueli A.** The Temporal Logic of Reactive and Concurrent Systems. – Springer-Verlag, 1992.
3. **Baranov S., Kotlyarov V., Letichevsky A., Drobintsev P.** The technology of Automation Verification and Testing in Industrial Projects / Proc. of St. Petersburg IEEE Chapter, International Conf., May 18-21, St. Petersburg, 2005. – P. 81–86. (rus)
4. Recommendation ITU-T Z.151. User re-

quirements notation (URN), 11.2008.

5. **Letichevsky A., Kapitonova J., Letichevsky A. Jr., Volkov V., Baranov S., Kotlyarov V., Weigert T.** Basic Protocols, Message Sequence Charts, and the Verification of Requirements Specifications. Proc of ISSRE04 Workshop on Integrated-reliability with Telecommunications and UML Languages (ISSRE04: WITUL). – IRISA Rennes France, 02.11.2004.

6. **Kolchin A.** Napravlennyi poisk v verifikatsii formal'nykh modelei // Tezi dop. mizhnar. konf. «Teoretichni ta prikladni aspekti pobudovi programnikh sistem TAAPSD'2007». – Berdiansk: NaUKMA, Natsional'nii un-t im. T.G. Shevchenka, In-t programnikh sistem NAN Ukraïni, 2007. – S. 256–258.

---

**НИКИФОРОВ Игорь Валерьевич** – аспирант кафедры информационных и управляющих систем Санкт-Петербургского государственного политехнического университета.

195251, Россия, Санкт-Петербург, ул. Политехническая, д. 29.

**NIKIFOROV, Igor V.** St. Petersburg State Polytechnical University.

195251, Politekhnikeskaya Str. 29, St. Petersburg, Russia.

**ДРОБИНЦЕВ Павел Дмитриевич** – доцент кафедры информационных и управляющих систем Санкт-Петербургского государственного политехнического университета, кандидат технических наук.

195251, Россия, Санкт-Петербург, ул. Политехническая, д. 29.

**DROBINTSEV, Pavel D.** St. Petersburg State Polytechnical University.

195251, Politekhnikeskaya Str. 29, St. Petersburg, Russia.

**КОТЛЯРОВ Всеволод Павлович** – профессор кафедры информационных и управляющих систем Санкт-Петербургского государственного политехнического университета, кандидат технических наук.

195251, Россия, Санкт-Петербург, ул. Политехническая, д. 29.

E-mail: vpk@spbstu.ru

**KOTLYAROV, Vsevolod P.** *St. Petersburg State Polytechnical University.*

195251, Politekhnikeskaya Str. 29, St. Petersburg, Russia.

E-mail: vpk@spbstu.ru

УДК 004.4'22

*Б.В. Тютин, А.О. Веселов, В.П. Котляров*

## **МАСШТАБИРОВАНИЕ ВЫПОЛНЕНИЯ ТЕСТОВОГО НАБОРА ПРИ АВТОМАТИЗИРОВАННОМ ТЕСТИРОВАНИИ**

*B.V. Tyutin, A.O. Veselov, V.P. Kotlyarov*

### **TEST SUITE EXECUTION SCALING FOR AUTOMATED TESTING**

Рассмотрено понятие масштабирования выполнения тестового набора и подход к решению этой задачи, реализованный в системе тестирования TestCommander. Проведена оценка результатов использования разработанного подхода.

**АВТОМАТИЗАЦИЯ ТЕСТИРОВАНИЯ. ЮНИТ-ТЕСТИРОВАНИЕ. ВЕРИФИКАЦИЯ. УПРАВЛЕНИЕ ТРЕБОВАНИЯМИ. МАСШТАБИРОВАНИЕ.**

This article reviews the test execution scaling problem and the approach for its solution implemented in TestCommander, an automated testing system. Results of the system piloting are analyzed and discussed.

**TESTING AUTOMATION. UNIT-TESTING. VERIFICATION. REQUIREMENT MANAGEMENT. SCALING.**

В общем виде масштабирование означает способность системы увеличивать свою производительность при добавлении ресурсов пропорционально их объему. Количественно это свойство можно оценить через отношение прироста производительности системы к приросту используемых ресурсов. Чем ближе это отношение к единице, тем лучше.

Применительно к тестированию понятие масштабирования можно определить как способность тестирующей системы ускорять выполнение тестового набора за счет параллельного выполнения его частей. Цель в снижении временных затрат на выполнение тестов, что приводит к возможности более полного покрытия кода тестами, повышению качества тестирования и более эффективному использованию имеющихся аппаратных средств. В существующих системах тестирования и библиотеках для их построения данная задача решена в лучшем случае частично за счет интеграции со сторонними инструментами. Это повышает сложность организации тестирования и, следовательно, связанные с ним затраты.

Основная идея работы состоит в создании метода масштабирования выпол-

нения тестового набора и его реализации в системе автоматического тестирования TestCommander [1], ориентированной на функциональное и модульное тестирование. Для параллельного выполнения тестов осуществлена модификация ядра системы с использованием MPI, что позволило адаптировать TestCommander к работе на многопоточной и кластерной инфраструктуре. Для масштабирования тестового набора применен комбинированный подход, основанный на кластеризации набора тестов и изоляции среды их исполнения.

**Возможные подходы к масштабированию.** Выделяют два типа масштабирования: вертикальное и горизонтальное [2]. Вертикальное масштабирование – увеличение производительности каждого компонента системы с целью повышения общей производительности. Горизонтальное масштабирование – разбиение системы на более мелкие структурные компоненты и разнесение их по отдельным физическим машинам (или их группам) или увеличение количества серверов, параллельно выполняющих одну и ту же функцию.

В зависимости от типа тестирования и логики тестов можно выделить два подхода



к решению данной задачи. Первый – параллельное выполнение тестов. Он подразумевает разбиение исходного тестового набора на части и их выполнение в различных процессах, расположенных как на одной, так и на разных рабочих станциях. Данный подход может успешно применяться в области модульного тестирования, а также тестирования серверных приложений. Для того чтобы успешно использовать параллельное выполнение тестов необходимо, чтобы выполнение одних тестов не влияло на выполнение других. Второй способ масштабирования тестового набора – дублирование тестового стенда. Это горизонтальное масштабирование процесса тестирования, когда взаимодействующие тестирующая и тестируемая системы развертываются на нескольких платформах. Данный подход требует больших аппаратных ресурсов, однако позволяет выполнять любые виды тестирования.

При проектировании расширения системы автоматизации тестирования TestCommander, поддерживающей масштабирование выполнения тестового набора, были учтены следующие аспекты.

1. Балансировка нагрузки и критерии разделения тестового набора на части. Балансировка может производиться на доступных рабочих узлах в зависимости от нагрузки и их количества, сложности тестов или прогнозируемого времени выполнения.

2. Изоляция выполнения. При выполнении масштабирования требуется обеспечить отсутствие сторонних эффектов при выполнении отдельных тестов, влияющих на весь процесс тестирования.

3. Сбор и обработка результатов. Необходимо обеспечить целостность журналов событий, а также любой другой информации, регистрируемой в ходе тестирования.

4. Прозрачность для пользователя. Процесс масштабирования должен контролироваться системой тестирования, а не тестирующим, с тем, чтобы выполнение тестового набора без масштабирования было идентично параллельному выполнению с точки зрения пользователя.

**Система тестирования.** TestCommander представляет собой генератор кода тестово-

го окружения на основе тестовых сценариев, конфигурационного XML файла и целевого кода, отвечающего за сериализацию/десериализацию сообщений, отправляемых/получаемых через внешние интерфейсы окружения. В TestCommander входит набор инструментов, используемых на разных этапах тестирования: макроподстановщик, анализатор трасс, кодогенерирующий шаблон, генератор обертки [1].

В качестве языка тестовых сценариев используется Message Sequence Chart (MSC) [3]. Данный язык позволяет определить взаимодействие различных сущностей, в т. ч. и параллельное, в терминах отправки и приема сигналов. На основании набора тестов в формате MSC и конфигурационного файла автоматически создается код тестов в виде машины состояний, а также прослойка, обрабатывающая сигналы и реализующая взаимодействие с тестируемой системой [4]. Далее полученный код собирается в готовый тестовый набор. Запуск тестов и управление тестированием осуществляется автоматически средствами TestCommander. Результаты тестирования (журнал ошибок, отчеты по каждому пройденному тесту, сводные таблицы) представляются в формате HTML.

Для создания тестов, адекватных к исходным требованиям к продукту, а также автоматизированного создания MSC с контролируемым уровнем покрытия тестами TestCommander интегрирован со средствами разработки Use-case maps (UCM) [1] и верификатором VRS [1].

TestCommander был успешно применен в ряде проектов по автоматизации тестирования телекоммуникационных систем. По результатам его использования можно сказать об удобстве данного инструмента при решении задач функционального и регрессионного тестирования [4].

**Кластеризация тестового набора.** В контексте задач тестирования под кластеризацией подразумевается разбиение набора тестовых сценариев на отдельные группы тестов с целью их независимого выполнения. Это позволяет эффективно использовать имеющиеся аппаратные ресурсы, распределяя нагрузку более равномерно.

При разбиении тестов на группы необходимо обеспечить выполнение связанных между собой тестов в рамках одного кластера в необходимом порядке. Данное требование напрямую следует из требования воспроизводимости и однозначности процесса тестирования [5].

В системе TestCommander разбиение тестового сценария осуществляется на основании указываемого пользователем числа групп. При этом разделение происходит на основании одного из двух следующих показателей:

- Количество тестов в группе. При использовании данного показателя тестовый набор делится на группы, содержащие указанное пользователем число тестов.

- Сложность тестов. При использовании данного показателя суммарная сложность тестов в различных группах примерно одинакова. Данную величину можно выразить формулой  $C = \sum_{i=1}^N n_i c_i$ , где  $C$  – суммарная сложность тестов;  $n_i$  – множитель для конкретного события или управляющей конструкции на диаграмме;  $c_i$  – весовой коэффициент, соответствующий выражению. Весовые коэффициенты и множители для событий выбираются таким образом, чтобы учитывать требования к времени прохождения теста и интенсивности обмена информацией. При кластеризации тестов TestCommander также учитывает наличие взаимосвязей между тестами.

**Изолирование среды исполнения тестов.** Изолирование среды исполнения необходимо при параллельном запуске тестов в том случае, если возможно взаимное влияние таких тестов друг на друга, например, при обращении к разделяемому ресурсу или одновременном взаимодействии с одним и тем же экземпляром тестируемой системы [6]. Такое влияние может оказать воздействие на вердикт прохождения теста и привести к нестабильным результатам прогона тестового набора.

Для предотвращения нежелательного взаимодействия параллельно исполняемых тестов система тестирования автоматически создает для каждого теста полностью изолированное тестовое окружение и от-

дельный экземпляр тестируемой системы (или ее компонент). Изолирование происходит прозрачно для пользователя без необходимости дублирования информации или ручных манипуляций.

Тестирование происходит в рамках каждой из изолированных систем путем взаимодействия компонент тестового окружения с компонентами тестируемой системы, что подразумевает необходимость указания связей между интерфейсами компонент, что является одним из этапов конфигурирования тестового окружения.

В простейшем случае, при отсутствии какой-либо информации о связях, считается, что тестовое окружение состоит всего из одной компоненты, которая напрямую подключается ко всем интерфейсам всех компонент тестируемой системы. В более сложных случаях пользователь имеет возможность при помощи конфигурационного файла описать связи между компонентами. Следует отметить, что такое конфигурирование не влияет на свойство изолированности, т. к. единожды описанное в конфигурационном файле, оно применяется многократно для каждого из тестов в отдельности.

**Параллельное выполнение тестов.** После успешного завершения этапа кластеризации и конфигурирования возможна генерация кода тестового набора, его сборка и запуск. Данные этапы автоматизированы в системе TestCommander. Генерация и сборка кода осуществляется на основном узле, после чего тестовые наборы распределяются по имеющимся рабочим станциям, либо запускаются на текущей. Запуск тестов координируется системой тестирования. Результаты выполнения тестов собираются вместе, при этом сохраняется структура журналов, что делает данную операцию прозрачной для пользователя.

Для возможности параллельного выполнения тестового набора на кластерных вычислительных системах в TestCommander реализован режим кластеризации тестов на этапе выполнения с помощью вызовов MPI. Выполнение отдельных тестов происходит на различных процессорах системы, при этом сохраняются свойства изолиро-

ванности тестов и воспроизводимости тестового набора.

**Ограничения используемого подхода.** Разработанная в ходе выполнения данной работы методика была испытана в различных проектах тестирования. По результатам ее использования определен ряд ограничений и особенностей реализации. Наиболее важными являются следующие из них:

- сложности в настройке конфигурации, в которой тестирующая система выступает в роли мастера;
- необходимость создания обертки для систем, закрытых от взаимодействия;
- наличие только одного критерия разбиения при выполнении тестов в кластерных системах.

Данные ограничения определяют границы эффективного применения системы тестирования и дальнейшие направления ее развития.

Традиционно тестированию отводится второстепенная роль в цикле создания программного продукта. Некоторые современные подходы к разработке, такие, как *test-driven development*, привлекают внимание разработчиков к проблеме обеспечения качества программного продукта на всех этапах его создания, что приводит к раз-

витию инструментов тестирования. Новые возможности в данной области позволяют сделать процесс тестирования более удобным, дешевым и полезным [7].

Уменьшение времени выполнения тестового набора и автоматизация запуска тестов ведет к снижению издержек на организацию тестирования и его проведение. Масштабирование выполнения тестов — один из наиболее эффективных способов для достижения этого результата. Реализованный в системе *TestCommander* подход к решению данной задачи, включающий в себя кластеризацию тестов и их параллельное выполнение на рабочих станциях или узлах кластера, опробован в ряде проектов тестирования компонент телекоммуникационных систем. По результатам данных проектов можно сделать вывод о том, что наиболее эффективно задача масштабирования решалась при организации модульного тестирования за счет возможности параллельного запуска тестов, и при интеграционном тестировании, где главную роль сыграла возможность запуска изолированных многокомпонентных тестов. Прозрачность процесса масштабирования для пользователя системы является фактором, снижающим сложность работы с ней.

#### СПИСОК ЛИТЕРАТУРЫ

1. Тютин, Б.В. Построение системы автоматизации статической и динамической проверки требований к программному продукту [Текст] / Б.В. Тютин, И.В. Никифоров, В.П. Котляров // Научно-технические ведомости СПбГПУ. Информатика. Телекоммуникации. Управление. — СПб.: Изд-во Политехнического ун-та, 2012. — № 4 (152). — С 119–123.
2. Таненбаум, Эндрю. Распределенные системы. Принципы и парадигмы [Текст] / Эндрю Таненбаум, Мартин ван Стеен. — СПб.: Питер, 2003.
3. ITU-T Recommendation Z.120: Message sequence chart (MSC). Geneva, Switzerland, October 1996 [Электронный ресурс] /Режим доступа <http://eu.sabotage.org/www/ITU/Z/Z0120e.pdf>
4. Веселов, А.О. Автоматическая настройка тестового окружения телекоммуникационных проектов [Текст] / А.О. Веселов, А.С. Иванов, Б.В. Тютин, В.П. Котляров // Научно-технические ведомости СПбГПУ. Информатика. Телекоммуникации. Управление. — СПб.: Изд-во Политехнического ун-та, 2011. — № 4 (128). — С. 149–152.
5. Oshero, R. The Art of Unit Testing: With Examples in .Net. [Text] / R. Oshero. — Greenwich: Manning Publications Co, 2010.
6. Калбертсон, Р. Быстрое тестирование [Текст] / Р. Калбертсон, К. Браун, Г. Кобб. — М.: ИД «Вильямс», 2002.
7. Beck, K. Test-Driven Development: By Example [Text] / K. Beck. — Addison-Wesley, 2002.

#### REFERENCES

1. Tyutin B.V., Nikiforov I.V., Kotlyarov V.P. Implementation of static and dynamic requirement analysis system [*Elaboration of the toolsuite for automation of the static and dynamic software requirement checking*] / Nauchno-tekhicheskie vedomosti SPbGPU. Informatika. Telekommunikatsii.

Upravlenie [*St. Petersburg State Polytechnical University Journal. Computer Science. Telecommunications and Control Systems*]. St. Petersburg: Izd-vo Politehnicheskogo un-ta, 2012. – № 4 (152). – S. 119–123. (rus)

2. **Tanenbaum Andrew S., van Steen Maarten.** Distributed systems. Principles and paradigms. – St. Petersburg: Piter, 2003. (rus)

3. ITU-T Recommendation Z.120: Message sequence chart (MSC). Geneva, Switzerland, October 1996; Available <http://eu.sabotage.org/www/ITU/Z/Z0120e.pdf>

4. **Veselov A.O., Ivanov A.S., Tyutin B.V., Kotlyarov V.P.** Avtomaticheskaya nastroyka testovogo okruzeniya telekommunikatsionnih proektov [*Automa-*

*tation of test environment configuring for telecom projects*] / Nauchno-tehnicheskie vedomosti SPb-GPU. Informatika. Telekommunikatsii. Upravlenie [*St. Petersburg State Polytechnical University Journal. Computer Science. Telecommunications and Control Systems*]. St. Petersburg: Izd-vo Politehnicheskogo un-ta, 2011. – № 4 (128). – S. 149–152. (rus)

5. **Osherove R.** The Art of Unit Testing: With Examples in .Net. – Greenwich: Manning Publications Co, 2010.

6. **Culbertson Robert, Brown Chris, Cobb Gary.** Byistroe testirovanie [*Rapid Testing*]. – Moscow: ID «Williams», 2002 (rus)

7. **Beck K.** Test-Driven Development: By Example. – Addison-Wesley, 2002.

---

**ТЮТИН Борис Викторович** – аспирант кафедры информационных и управляющих систем Санкт-Петербургского государственного политехнического университета.

195251, Россия, Санкт-Петербург, ул. Политехническая, д. 29.

E-mail: b.tyutin@gmail.com

**TYUTIN, Boris V.** – *St. Petersburg State Polytechnical University.*

195251, Politehnicheskaya Str. 29, St. Petersburg, Russia.

E-mail: b.tyutin@gmail.com

**ВЕСЕЛОВ Алексей Олегович** – аспирант кафедры информационных и управляющих систем Санкт-Петербургского государственного политехнического университета.

195251, Россия, Санкт-Петербург, ул. Политехническая, д. 29.

E-mail: veselov.alexey@gmail.com

**VESELOV, Alexey O.** – *St. Petersburg State Polytechnical University.*

195251, Politehnicheskaya Str. 29, St. Petersburg, Russia.

E-mail: veselov.alexey@gmail.com

**КОТЛЯРОВ Всеволод Павлович** – профессор кафедры информационных и управляющих систем Санкт-Петербургского государственного политехнического университета, кандидат технических наук.

E-mail: vpk@ics2.ecd.spbstu.ru

**KOTLYAROV, Vsevolod P.** – *St. Petersburg State Polytechnical University.*

195251, Politehnicheskaya Str. 29, St. Petersburg, Russia.

E-mail: vpk@ics2.ecd.spbstu.ru



УДК 621.391:004.421

*В.С. Тутыгин, И.А. Пелконен, К.С. Кириленко*

## СПОСОБ И ПРОГРАММНАЯ РЕАЛИЗАЦИЯ ПРЕЦИЗИОННОГО ИЗМЕРЕНИЯ ВРЕМЕНИ ЗАДЕРЖКИ РАДИОИМПУЛЬСА

*V.S. Tutygin, I.A. Pelkonen, K.S. Kirilenko*

### METHOD AND SOFTWARE FOR PRECISION MEASUREMENT OF TIME DELAY ECHO-SIGNALS OF RADARS

Рассмотрен новый способ и его программная реализация для прецизионного измерения времени задержки радиолокационного отраженного сигнала относительно зондирующего радиоимпульса, основанный на приблизительной оценке временного сдвига отраженного сигнала и итерационного алгоритма, включающего вычисление коэффициентов корреляции отраженного сигнала с несколькими эталонными сигналами, сплайн-интерполяцию, передискретизацию, обеспечивающего уточнение значения времени задержки.

РАДИОЛОКАЦИОННЫЙ ЭХО-СИГНАЛ. ЭТАЛОННЫЕ СИГНАЛЫ. КОРРЕЛЯЦИЯ. СПЛАЙН-ИНТЕРПОЛЯЦИЯ. ПЕРЕДИСКРЕТИЗАЦИЯ. ИТЕРАЦИОННЫЙ АЛГОРИТМ.

A new approach to the decision of a problem of delay definition noised radar echo-signals from an exciting radio impulse based on initial approach of delay received by means of autocorrelation procedure, and additional digital iterative approach of exact delay using nonconventional combination of normalized cross-correlation echo-signal with reference signals, spline-interpolation, oversampling are considered.

RADAR ECHO-SYGNAL. REFERENCE SIGNALS. CROSS-CORRELATION. SPLINE-INTERPOLATION. OVERSAMPLING. ITERATIVE ALGORITHM.

Известный способ измерения расстояния до объекта основан на измерении времени задержки отраженного радиолокационного сигнала относительно зондирующего. Такой способ применяется, например, в электромагнитно-акустических (ЭМА) толщиномерах [1], которые измеряют время прохождения ультразвукового импульса, порождаемого радиоимпульсом, через металлическое изделие.

Источник зондирующего радиоимпульса помещается у одной поверхности металлического изделия, сигнал, отраженный от другой поверхности, регистрируется. По времени задержки отраженного сигнала относительно зондирующего определяется толщина металла. При коррозии металлических изделий отраженный сигнал оказывается значительно зашумлен (см. рис. 1), что приводит к уменьшению точности определения временной задержки и, следовательно, к уменьшению точности определения толщины металла.

Для улучшения отношения сигнал/шум в отраженном сигнале могут применяться различные методы, такие, как оптимальная фильтрация (например, оптимальный

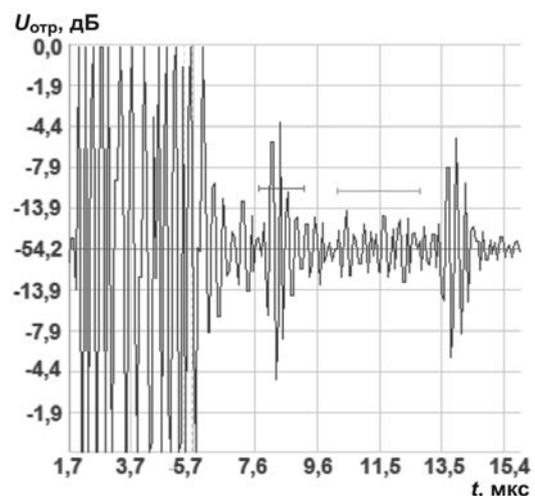


Рис. 1. Типичный отраженный сигнал ЭМА толщиномера

фильтр Колмогорова–Винера), прямое и обратное преобразование Фурье, автокорреляция.

Влияние шума в регистрируемом отраженном сигнале можно уменьшать и за счет многократного повторения эксперимента и синхронного накопления отраженных сигналов.

Однако при ограничении времени проведения измерения количество возможных накоплений сигнала должно быть ограничено или вообще должно отсутствовать. Поэтому задача повышения точности измерения временного сдвига зашумленного отраженного сигнала актуальна.

Использование традиционного подхода определения временного сдвига по максимуму отраженного сигнала или автокорреляционной функции для зашумленного сигнала не позволяет получить точное значение временного сдвига. Но, поскольку форма и начальная фаза отраженных сигналов известны, это дает возможность создать эталонные сигналы, соответствующие ожидаемому отраженному сигналу по форме и начальной фазе, и производить корреляционное сравнение отраженного сигнала с эталонными. Коэффициент корреляции отраженного сигнала с эталонным сигналом будет равен единице, если временной сдвиг отраженного и эталонного сигнала равен нулю и отраженный сигнал не зашумлен. Поэтому при отсутствии шумов найти временной сдвиг отраженного сигнала можно, производя корреляционное сравнение с эталонными сигналами, временной сдвиг эталонных сигналов подбирать до выполнения условия, когда коэффициент корреляции будет равен единице. Однако коэффициент корреляции уменьшается как при разнице времени сдвига отраженного и эталонного сигналов, так и при совпадении времени сдвига, но из-за наличия шума. Поэтому таким способом определить временную задержку зашумленного отраженного сигнала невозможно.

Предложенный нами в [2] способ определения времени сдвига отраженного сигнала основан на сочетании положительных качеств известного подхода и дополнительной цифровой обработки. Идея предлагае-

мого способа цифровой обработки заключается в том, что в небольшой окрестности от предполагаемого временного сдвига сигнала (приближенное значение временного сдвига сигнала может быть найдено с помощью автокорреляции или по максимуму амплитуды) вычисляются коэффициенты корреляции отраженного сигнала с несколькими эталонными сигналами в некоторой окрестности от приближенного значения временного сдвига, затем с помощью сплайн-интерполяции и передискретизации строится функция, выражающая зависимость коэффициента корреляции от временного сдвига эталонов, и находится максимум этой функции, по положению максимума определяется уточненное значение временного сдвига эталонного сигнала.

Теоретический анализ этого способа определения времени задержки отраженного радиоимпульса приведен в [3].

Коэффициент ковариации двух гармонических сигналов, имеющих одинаковую частоту и длительность, но отличающихся сдвигом во времени, определяется как

$$k_{\text{cov}} = \frac{1}{T} \int_0^T x(t)y(t - \Delta t)dt = \\ = \frac{1}{T} \int_0^T \sin\left(\frac{2\pi nt}{T}\right) \sin\left(\frac{2\pi n(t - \Delta t)}{T}\right) dt,$$

а коэффициент корреляции

$$k_{\text{cor}} = \frac{k_{\text{cov}}}{\sqrt{\sigma_x^2 \sigma_y^2}},$$

где

$$\sigma_x^2 = \frac{1}{T} \int_0^T \sin^2\left(\frac{2\pi nt}{T}\right) dt; \\ \sigma_y^2 = \frac{1}{T} \int_0^T \sin^2\left(\frac{2\pi n(t - \Delta t)}{T}\right) dt.$$

Функция  $k_{\text{cor}} = k_{\text{cor}}\left(\frac{\Delta t}{T}\right)$  может быть адекватно аппроксимирована параболой шестого порядка, обращенной вершиной вверх, и ее максимальное значение равно единице (рис. 2 а).

Если же второй сигнал зашумлен, вид функции коэффициентов взаимной корреляции сохранится (рис. 2 б), а коэффици-

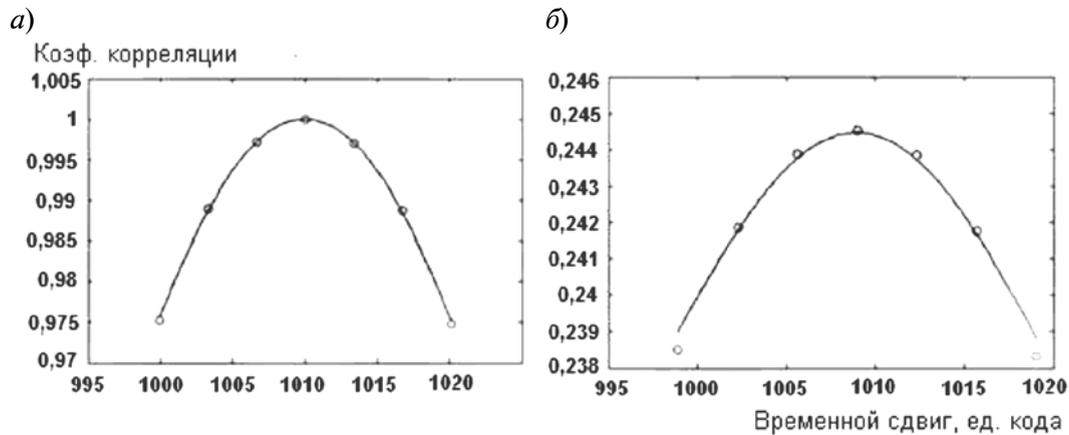


Рис. 2. Зависимость коэффициента корреляции от времени задержки при отсутствии шума (а) и при отношении сигнал/шум 1/3 (б). Точное значение времени задержки равно 1010. Аппроксимирующие кривые построены с помощью функции сплайн-аппроксимации `splaps` в MATLAB

ент взаимной ковариации может быть вычислен по формуле:

$$\begin{aligned}
 k_{cov2} &= \frac{1}{T} \int_0^T x(t)(y(t - \Delta t) + \xi(t)) dt = \\
 &= \frac{1}{T} \int_0^T \sin\left(\frac{2\pi nt}{T}\right) \left(\sin\left(\frac{2\pi n(t - \Delta t)}{T}\right) + \xi(t)\right) dt = \\
 &= k_{cov} + \frac{1}{T} \int_0^T \left(\sin\left(\frac{2\pi nt}{T}\right) \xi(t)\right) dt = \\
 &= k_{cov} + \xi(\theta) \frac{1}{T} \int_0^T \left(\sin\left(\frac{2\pi nt}{T}\right) dt\right),
 \end{aligned}$$

где  $0 < \theta < T$ , т. к. по теореме о среднем  $\int_a^b f(x)g(x)dx = f(\theta) \int_a^b g(x)dx, a < \theta < b$ .

Если  $n$  – целое число, то

$$\begin{aligned}
 \int_0^T \left(\sin\left(\frac{2\pi nt}{T}\right) dt\right) &= -\frac{T}{2\pi n} \cos\left(\frac{2\pi nt}{T}\right) \Big|_0^T = \\
 &= -\frac{T}{2\pi n} (1 - 1) = 0,
 \end{aligned}$$

отсюда  $k_{cov2} = k_{cov}$ .

Функция коэффициентов взаимной корреляции будет иметь вид:

$$k_{cor2} = \frac{k_{cov2}}{\sqrt{\sigma_{x2}^2 \sigma_{y2}^2}} = \frac{k_{cov}}{\sqrt{\sigma_x^2 \sigma_y^2}},$$

где

$$\sigma_{y2}^2 = \frac{1}{T} \int_0^T \left(\sin^2\left(\frac{2\pi n(t - \Delta t)}{T}\right) + \xi(t)\right) dt =$$

$$\begin{aligned}
 &= \frac{1}{T} \left\{ \int_0^T \left[\sin^2\left(\frac{2\pi n(t - \Delta t)}{T}\right) + \right. \right. \\
 &\quad \left. \left. + 2 \sin\left(\frac{2\pi n(t - \Delta t)}{T}\right) \xi(t) + \xi^2(t)\right] dt \right\} = \\
 &= \sigma_y^2 + \frac{2}{T} \int_0^T \sin\left(\frac{2\pi n(t - \Delta t)}{T}\right) \xi(t) dt + \frac{1}{T} \int_0^T \xi^2(t) dt.
 \end{aligned}$$

Применив теорему о среднем ко второму слагаемому, получим, что для случая, когда  $n$  – целое число, второе слагаемое равно нулю.

Определим  $\xi_1(t)$  как шум со средним квадратическим отклонением (СКО), равным единице, а  $\xi(t) = A\xi_1(t)$ , тогда получим  $\frac{1}{T} \int_0^T \xi^2(t) dt = A^2$ , где  $A$  – уровень шума, выраженный в долях СКО, т. е.  $A = \sigma_{noise}$ .

Более точно интеграл от квадрата шума или, другими словами, функция автоковариации шума, является случайной величиной и характеризуется математическим ожиданием, средним квадратическим отклонением и доверительным интервалом при заданном уровне доверительной вероятности.

Приведенные выше аналитические выражения позволяют количественно оценивать ожидаемую погрешность определения времени задержки отраженного радиолокационного сигнала предложенным способом.

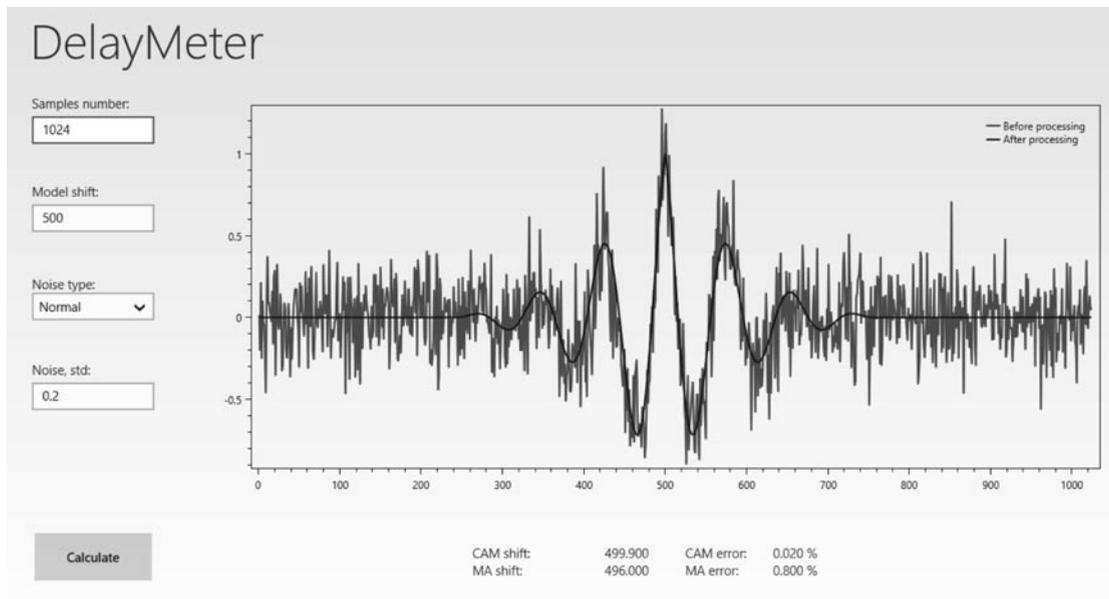


Рис. 3. Модельный сигнал (1024 отсчета, точное значение сдвига 500, нормальный шум с СКО 0,2)  
 Результат измерения задержки по максимуму амплитуды (МА): 496,0 (погрешность 0,8 %);  
 результат предложенного способа (САМ): 499,9 (погрешность 0,02 %)

В [4] приведена программа, реализующая предложенный способ в среде MATLAB. Для широкого практического применения нового способа нами разработана программа на языке C++, описанная в [5]. Реализация на языке C++ дает возможность переноса программы на микроконтроллер или цифровой сигнальный процессор, что позволяет создавать портативные цифровые ЭМА толщинометры высокой точности.

Функции коэффициентов корреляции и сплайн-интерполяции, необходимые при программной реализации способа измере-

ния времени задержки радиопульса, в стандартных библиотеках C++ отсутствуют, поэтому коэффициенты корреляции вычисляются в программе по приведенным выше формулам, а сплайн-интерполяция реализована согласно [6].

Работа программы проверена с использованием нормированного по максимуму амплитуды модельного сигнала, близкого к реальному сигналу ЭМА толщинометра, приведенному на рис. 1. Для моделирования нормально распределенного шума использован Mersenne Twister [7] – генератор

**Оценки погрешностей определения времени задержки отраженного радиолокационного сигнала традиционным и предложенным способами**

СКО шума	Выборочное среднее относительной погрешности, %		Выборочная дисперсия относительной погрешности, %	
	Традиционный способ	Предложенный способ	Традиционный способ	Предложенный способ
0,00	0,000	0,000	0,000	0,000
0,04	0,094	0,019	0,062	0,004
0,08	0,186	0,024	0,175	0,005
0,12	0,276	0,041	0,274	0,013
0,16	0,312	0,054	0,389	0,016
0,20	0,326	0,060	0,500	0,019

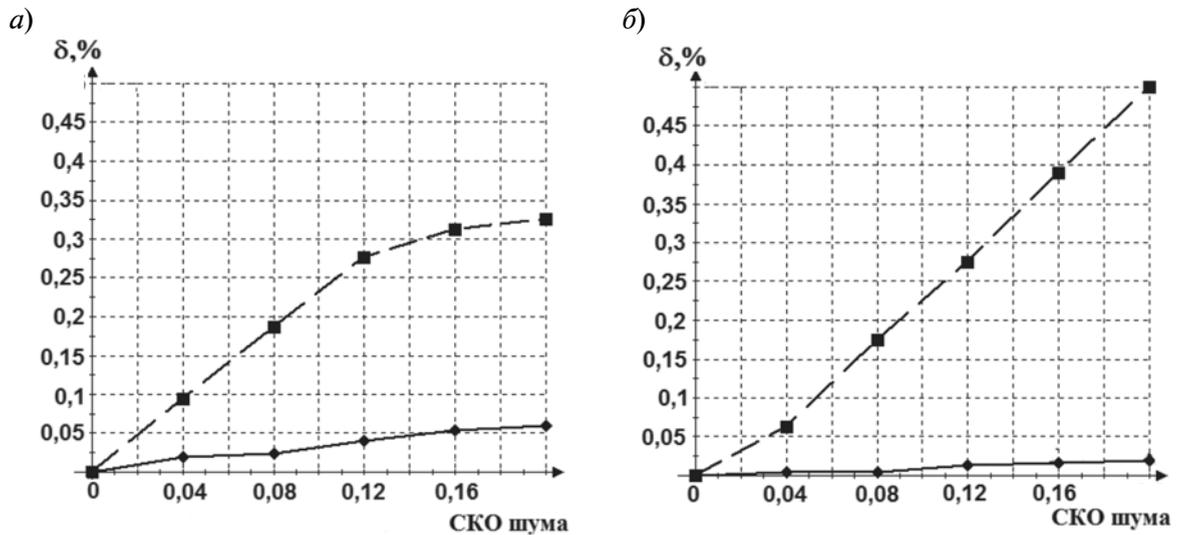


Рис. 4. Выборочное среднее относительной погрешности (а) и выборочная дисперсия относительной погрешности (б) измерения временной задержки радиоимпульса (◆◆◆◆) предложенный способ; (■■ ■■) традиционный способ

псевдослучайных чисел, из библиотеки random стандарта C++11, поддерживаемого компилятором Microsoft Visual C++ 11.0.

Предложенный способ реализован нами в среде Visual Studio 2012 в виде динамической библиотеки на языке C++. Для реализации графического интерфейса выбран тип приложения Windows Store Application, доступный в операционной системе Windows 8, и язык C#. Это позволяет использовать разработанное приложение для разных типов устройств (настольный компьютер, ноутбук, планшет и т. д.) и для разных архитектур процессоров (x86, x64, ARM) без изменения исходного кода. Графический интерфейс приложения приведен на рис. 3.

Для оценки погрешности измерения за-

держки предложенным и традиционным способами проведено сто опытов для каждого СКО шума от 0,0 до 0,2 с шагом 0,04. Результаты оценки относительной погрешности приведены в таблице и на рис. 4.

Использование данной программы в ЭМА толщиномерах позволяет значительно повысить точность измерения толщины металла, а использование операционной системы Windows 8 при создании графического интерфейса позволяет создать компактные недорогие системы для измерения толщины металла на базе планшетных компьютеров и USB-устройств сбора данных, пригодных для работы в полевых условиях.

#### СПИСОК ЛИТЕРАТУРЫ

1. Себко, В.П. Оптимизация параметров ЭМА толщиномеров для контроля тонкостенных изделий [Текст] / В.П. Себко, Г.М. Сучков, Е.А. Алексеев // Дефектоскопия. –2002. –№ 12. –С. 21–28.
2. Тутыгин, В.С. Способ измерения временных интервалов между радиоимпульсами [Текст] / В.С. Тутыгин // Патент РФ на изобретение № 2456632. –Приоритет от 21.03.2011. –Зарег. 20.07.2012.
3. Тутыгин, В.С. Цифровая обработка коротких сигналов [Текст] / В.С. Тутыгин. –СПб.: Изд-во Политехн. ун-та, 2012. –164 с.
4. Тутыгин, В.С. Программа точного определения времени задержки радиолокационного отраженного сигнала [Текст] / В.С. Тутыгин // Свид. № 2011612678 о гос. рег. от 01.04.2011.
5. Пелконен, И.А. Программа определения времени задержки радиолокационного сигнала [Текст] / И.А. Пелконен, К.С. Кириленко, В.С. Тутыгин // В сб. Технологии Microsoft в теории и практике программирования. Матер. межвуз. конкурса-конф. студентов, аспирантов и молодых ученых Северо-Запада. –СПб.: Изд-во Политехн. ун-та, 2013. –С. 43–45.
6. Форсайт, Дж. Машинные методы матема-

тических вычислений [Текст] / Дж. Форсайт, М. Малькольм, К. Мулер. — М.: Мир, 1980. — 276 с.  
7. Mersenne Twister Engine — C++ Reference

[Электронный ресурс] / Режим доступа [http://www.cplusplus.com/reference/random/mersenne\\_twister\\_engine/](http://www.cplusplus.com/reference/random/mersenne_twister_engine/)

#### REFERENCES

1. **Sebko V.P., Suchkov G.M., Alekseev E.A.** Optimizatsiia parametrov EMA-tolshchinomerov dlia kontroliia tonkostennykh izdelii / Defektoskopiia. — 2002. — № 12. — S. 21–28. (rus)

2. **Tutygin V.S.** Sposob izmereniia vremennykh intervalov mezhdu radioimpul'sami; Patent RF № 2456632; Prioritet 21.03.2011; Zareg. 20.07.2012. (rus)

3. **Tutygin V.S.** Tsifrovaia obrabotka korotkikh signalov. — St. Petersburg: Izd-vo Politekhn. un-ta, 2012. — 164 s. (rus)

4. **Tutygin V.S.** Programma tochnogo opredeleniia vremeni zaderzhki radiolokatsionnogo otrazhennogo signala; Svid. № 2011612678 o gos.

reg. ot 01.04.2011. (rus)

5. **Pelkonen I.A.** Programma opredeleniia vremeni zaderzhki radiolokatsionnogo signala /V sb. Tekhnologii Microsoft v teorii i praktike programmirovaniia. Mater. mezhvuz. konkursa-konf. studentov, aspirantov i molodykh uchenykh Severo-Zapada. — St. Petersburg: Izd-vo Politekhn. un-ta, 2013. — S. 43–45. (rus)

6. **Forsait Dzh., Mal'kol'm M., Mouler K.** Mashinnye metody matematicheskikh vychislenii. — Moscow: Mir, 1980. — 276 s. (rus)

7. Mersenne Twister Engine — C++ Reference. Available [http://www.cplusplus.com/reference/random/mersenne\\_twister\\_engine/](http://www.cplusplus.com/reference/random/mersenne_twister_engine/)

---

**ТУТЫГИН Владимир Семенович** — доцент кафедры информационных и управляющих систем Санкт-Петербургского государственного политехнического университета, кандидат технических наук.

195251, Россия, Санкт-Петербург, ул. Политехническая, д. 29.

E-mail: vs4141@mail.ru

**TUTYGIN, Vladimir S.** *St. Petersburg State Polytechnical University.*

195251, Politekhnikeskaya Str. 29, St. Petersburg, Russia.

E-mail: vs4141@mail.ru

**ПЕЛКОНЕН Игнатий Андреевич** — студент кафедры информационных и управляющих систем Санкт-Петербургского государственного политехнического университета.

195251, Россия, Санкт-Петербург, ул. Политехническая, д. 29.

**PELKONEN, Ignatij A.** *St. Petersburg State Polytechnical University.*

195251, Politekhnikeskaya Str. 29, St. Petersburg, Russia.

**КИРИЛЕНКО Кирилл Сергеевич** — студент кафедры информационных и управляющих систем Санкт-Петербургского государственного политехнического университета.

195251, Россия, Санкт-Петербург, ул. Политехническая, д. 29.

**KIRILENKO, Kirill S.** *St. Petersburg State Polytechnical University.*

195251, Politekhnikeskaya Str. 29, St. Petersburg, Russia.



УДК 681.3.068

*Н.В. Воинов, О.В. Александрова, Л.П. Котлярова***МОДЕРНИЗАЦИЯ МЕТОДИКИ РАЗРАБОТКИ МОБИЛЬНЫХ ПРИЛОЖЕНИЙ ДЛЯ ПЛАТФОРМЫ WINDOWS PHONE 8***N.V. Voinov, O.V. Aleksandrova, L.P. Kotlyarova***MODERNIZATION OF THE METHODIC OF WINDOWS PHONE 8 MOBILE APPLICATIONS DEVELOPMENT**

Описана модернизированная методика создания приложений для современных мобильных устройств, работающих на новой операционной системе компании Microsoft – Windows Phone 8. Данная методика обучает основным принципам разработки приложений и использованию наиболее важных возможностей платформы.

МЕТОДИКА. МОБИЛЬНАЯ ПЛАТФОРМА. МОБИЛЬНОЕ УСТРОЙСТВО. ПРИЛОЖЕНИЕ. РАЗРАБОТКА. MICROSOFT WINDOWS PHONE.

The paper describes the modernized methodic of an applications development for modern mobile devices working on new operating system of Microsoft Corporation – Windows Phone 8. The methodic teaches the main principles of applications development and usage of the most important platform's capabilities.

METHODIC. MOBILE PLATFORM. MOBILE DEVICE. APPLICATION. DEVELOPMENT. MICROSOFT WINDOWS PHONE.

Современный рынок мобильных устройств требует подготовки специалистов, способных быстро осваивать и создавать новые приложения под новые платформы. Для этих целей в Институте информационных технологий и управления СПбГПУ для специалистов по направлению «Программная инженерия» разработан новый курс [1], в основе которого методика создания приложений для мобильной платформы Windows Phone 7 [2], являющейся на тот момент наиболее перспективной. Однако в октябре 2012 г. вышло второе поколение операционной системы Windows Phone от Microsoft – Windows Phone 8 [3], что потребовало модернизации разработанной методики для отражения особенностей и новых функций мобильной платформы.

Модернизированная методика демонстрирует студентам основные принципы разработки приложений и учит использованию следующих важных возможностей платформы.

1. Варианты разметки StackPanel, Grid и Canvas. С помощью данных элементов

управления происходит группировка и позиционирование остальных элементов в xaml [4]. StackPanel выстраивает внутренние элементы последовательно по вертикали или горизонтали. Grid позволяет разделить рабочую область на строки и столбцы. Canvas предоставляет возможности абсолютного позиционирования внутренних элементов по координатам.

2. Основные элементы управления: кнопки, текстовые поля и блоки, поля, изображения, списки и т. п. Данные элементы входят в состав большинства приложений под Windows Phone 8. Они имеют различные атрибуты, такие, как цвет фона, толщина и цвет окантовки, размер, стиль и цвет шрифта, выравнивание и т. п. Каждый из элементов может иметь обработчики событий. Например, нажатие на кнопку, выделение нового элемента в списке, движение пальцем по Canvas.

3. Контекст ввода текстовых полей. При попадании фокуса на текстовое поле в Windows Phone 8 появляется клавиатура для ввода. С помощью контекста ввода можно

настраивать параметры кнопок на клавиатуре. Например, можно задать, чтобы на клавиатуре были только цифры или появились дополнительные кнопки «@» и «.com» и т. п. Также можно активировать и деактивировать подсказки слов.

4. Навигация между страницами. Так как большинство приложений под Windows Phone 8 являются многостраничными, имеются стандартные средства обеспечения навигации. Существует также возможность передачи параметров при переходе от одной странице к другой.

5. Элементы Pivot и Panorama. Данные элементы являются базовыми в Windows Phone 8. Они обеспечивают разделение информации по тематикам или представлениям. Pivot и Panorama имеют много общих черт: они оба используют несколько экранов для представления информации. Аналогом Pivot в приложениях Windows Forms являются вкладки. Panorama отличается тем, что задний фон и заголовок растягиваются на все используемые экраны. Также у элемента Panorama на каждой «вкладке» слегка видно соседнюю, что может побудить пользователя использовать прокрутку для просмотра всей информации.

6. Меню приложения. Меню (ApplicationBar) позволяет получить быстрый доступ к определенным функциям приложения. Оно может содержать до четырех кнопок на главной панели и не более пяти на всплывающей (возможно и больше, но тогда пользователю необходимо будет использовать прокрутку). Меню не является частью дерева визуальных элементов страницы, поэтому не перекрывается появляющейся клавиатурой.

7. Элемент управления WebBrowser. Данный элемент позволяет в заданной области отображать HTML-страницы, в т. ч. загруженные из сети Интернет. На базе данного элемента можно разрабатывать кросс-платформенные приложения, однако в рамках методики обучения будет рассмотрена основная его функция — отображение интернет-страниц.

8. Задачи запуска и выбора. С помощью задач запуска можно открывать стандартные приложения Windows Phone 8, такие,

как телефон, веб-обозреватель, проигрыватель медиафайлов и т. д. С помощью задач выбора можно получить доступ к различной информации на телефоне: контактам, изображениям, аудиофайлам и т. п.

9. Элемент управления Map и определение местоположения. С помощью данного элемента (для его использования следует подключить библиотеку Microsoft.Phone.Controls.Maps) можно загружать и отображать карты местности. Map поддерживает масштабирование и режимы воздушный и дорожный. Кроме того, Windows Phone 8 поддерживает определение текущего местоположения средствами GPS, сотовой сети и Wi-Fi.

10. Акселерометр. Позволяет определять изменение положения телефона относительно одной из осей. Для доступа к возможностям акселерометра необходимо подключить к проекту библиотеку Microsoft.Devices.Sensors.

11. Работа с HTTP посредством интерфейса WebClient. WebClient API позволяет разработчикам удобно работать с Get- и Post-запросами. Наилучшим образом подходит для загрузки документов из сети Интернет.

12. Хранение данных и привязка. Доступ к файловой системе из приложения ограничен изолированным хранилищем. Windows Phone 8 предоставляет различные средства для сохранения разного рода информации. Если требуется сохранить настройки приложения, можно воспользоваться классом IsolatedStorageSettings, который предоставляет доступ к словарю настроек. При этом данные хранятся в виде «ключ-значение». Для сохранения файлов следует воспользоваться классом IsolatedStorageFile. Для работы с большими объемами данных наилучшим образом подойдет локальная база данных. Общение приложения и базы осуществляется с помощью технологии Linq.

Важной особенностью Windows Phone 8 является привязка данных (binding). Она позволяет менять атрибуты одного элемента управления автоматически при изменении атрибутов другого. Так, например, можно выводить значение элемента Slider в текстовый блок.



13. Состояния приложения *Dormant* и *Tombstoned*. *Dormant* – спящее состояние, в которое переходит приложение при нажатии на кнопку «Пуск» во время работы. То есть приложение перестает быть активным, но продолжает находиться в памяти телефона. Вернуться к нему можно путем продолжительного нажатия на кнопку «Назад». *Tombstoned* – состояние захоронения, в которое приложение переходит из спящего состояния в случае нехватки памяти. При этом система может сохранить состояние приложения (одновременно сохраняются состояния только для пяти из них). Для пользователя вывод из *Tombstoned* аналогичен *Dormant*.

14. Фоновые службы. Позволяют приложению работать в фоне системы даже после закрытия. Службы могут выполнять различные функции, например, загружать в фоне файлы из сети Интернет (*BackgroundTransferRequest*) или проигрывать музыку (*AudioPlayerAgent*). Управление будильником и напоминаниями также выполняются средствами фоновых служб.

15. Живые тайлы (плитки). Тайлы – это ссылки на приложения, располагающиеся на главной странице Windows Phone 8. Они совмещают в себе функциональность обычных ярлыков и виджетов. Тайлы могут иметь несколько изображений, сменяющих друг друга с течением времени. Также они могут нести в себе некоторую полезную информацию, например, о количестве пропущенных вызовов.

16. *Push Notification*. Удобное средство, позволяющее облачным сервисам общаться с приложением и уведомлять пользователя о каком-либо событии.

17. Анимация. *Silverlight* обладает большим набором стандартных классов, позволяющих анимировать приложения как с помощью временной шкалы, так и по ключевым кадрам. Анимация может быть описана в файле разметки *xaml* или непосредственно в коде. Удобным инструментом, позволяющим быстро создавать различные анимационные эффекты, является *Microsoft Expression Blend*.

Предлагаемые изменения в методике позволят студентам без особых трудностей

научиться разрабатывать приложения для Windows Phone 8 разного уровня сложности.

Для иллюстрации применения модернизированной методики приведено описание лабораторной работы «Телефонная книга».

Задание: создать простую телефонную книжку для Windows Phone 8 с возможностью добавления новых контактов, просмотра, редактирования и удаления имеющихся.

Требования. В работе необходимо обеспечить: 1) навигацию между страницами; 2) основные элементы управления и разметки; 3) события; 4) контекст ввода; 5) меню приложения.

Поскольку лабораторные работы подразумевают создание приложений с большим числом строк кода, в данной статье далее приведен только порядок действий без детализации программной логики.

В самом начале кратко опишем структуру приложения: «Приложение, которое мы будем создавать, основано на четырех основных функциях управления данными – *CRUD* (*create, read, update, delete*). Оно будет состоять из двух окон: основного со списком контактов и окна редактирования/добавления контактов». Таким образом, обучающиеся получают представление о том, как будет выглядеть разрабатываемое приложение: две страницы, между которыми будут организованы переходы и передачи информации; меню с возможностью выбора действия; элемент управления списком, работа с которым будет вестись в коде.

Далее перейдем к разработке интерфейса пользователя: к изменению файла разметки. Начнем с изменения заголовков и плавно перейдем к размещению элементов управления: «Откроем файл разметки главной страницы *MainPage.xaml*. Изменим текст названия приложения *ApplicationTitle*, например, на «КОНТАКТЫ»...

Внутри объекта *Grid* с именем *ContentPanel* поместим элемент *ListBox*. Назовем его *listContacts* (атрибут *Name* необходим тем элементам, к которым будет обращение непосредственно из кода программы) и определим высоту списка (атрибут *Height*). Внутри *ListBox* поместим не-

сколько первых контактов, которые будут доступны сразу после запуска приложения. Каждый элемент списка необходимо заключить в тег `ListBoxItem`. Поскольку контакт состоит из имени и номера телефона, добавим внутрь объекта `ListBoxItem` два текстовых блока `TextBlock`. Определим для них атрибуты `FontSize` (размер шрифта) и `Margin` (отступы). Так как данные блоки идут последовательно друг за другом, их необходимо заключить в тег `StackPanel`».

Выполнив указанные действия, обучающиеся смогут размещать элементы управления в `xaml` файлах, задавать атрибуты, позиционировать и объединять их.

Далее опишем создание меню:

«Создадим меню на главной странице нашего приложения. Для этого в самом конце файла `MainPage.xaml` уберем комментарии, в которые заключен тег `phone:PhoneApplicationPage.ApplicationBar` и его содержимое. Добавим еще один элемент `shell:ApplicationBarIconButton` и изменим текст (атрибут `Text`) для всех пунктов меню... Добавим обработчики нажатия на кнопки меню. Для этого добавим в теги `shell:ApplicationBarIconButton` атрибут `Click`.

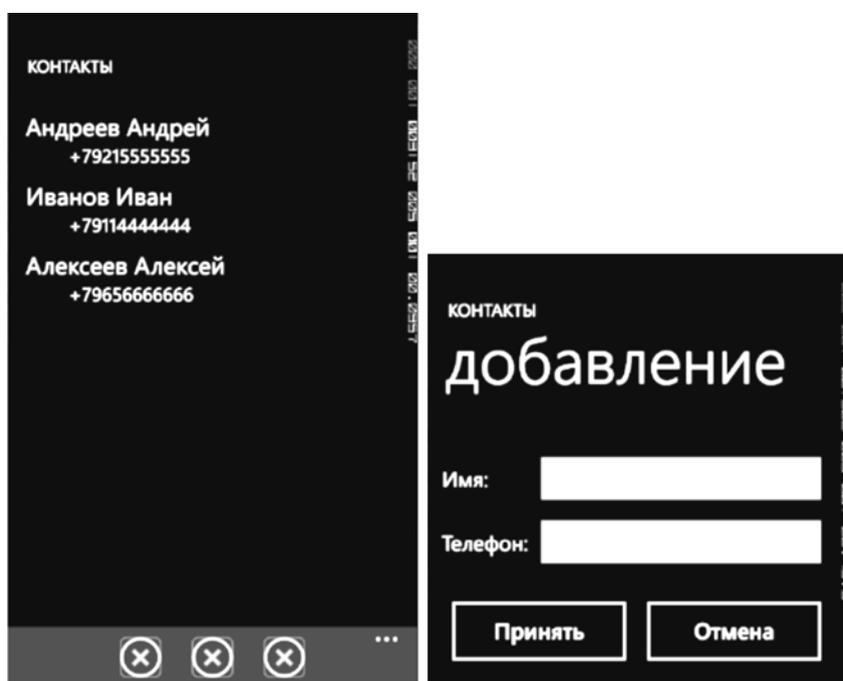
При этом в файле `MainPage.xaml.cs` автоматически будут добавлены соответствующие пустые обработчики».

Таким образом, обучающиеся смогут не только создавать меню, но также добавлять новые элементы, изменять текст, объявлять события, которые будут происходить при нажатии на кнопки меню.

Далее опишем добавление новой страницы и редактирование ее пользовательского интерфейса:

«Добавим в проект приложения новую страницу – `Windows Phone Portrait Page`. Назовем ее `PageEdit`. Откроем файл `PageEdit.xaml`...

Элемент `Grid` состоит из части объявления строк и столбцов и части содержимого. Элементы содержимого должны включать атрибуты `Grid.Column` и `Grid.Row`, которые определяют индексы столбца и строки таблицы, в которой они находятся. Атрибут `ShowGridLines` позволяет сделать видимой или невидимой границу таблицы. Определим также вертикальное выравнивание элементов (`VerticalAlignment`) и размер шрифта (`FontSize`). Для текстового поля `txtPhone` определим контекст ввода (`InputScope`).



Приложение «Телефонная книга»



Контекст ввода служит для того, чтобы пользователю было удобнее вводить те или иные данные в текстовое поле...

Для кнопок `btnApply` и `btnCancel` определим атрибут `Click` – «обработчик кнопок».

Данная часть работы поясняет особенности добавления в проект новых Silverlight-страниц, использование элемента управления `Grid`, применение контекста ввода для

текстовых полей и объявление событий для элементов управления.

В результате описанных действий будет разработано приложение (см. рисунок).

Внедрение модернизированной методики в учебный процесс запланировано на 2013–2014 учебный год и рассчитано на студентов 4-5 курсов.

#### СПИСОК ЛИТЕРАТУРЫ

1. **Воинов, Н.В.** Методика разработки мобильных приложений для платформы Windows Phone 7 [Текст] / Н.В. Воинов, Л.П. Котлярова, И.Е. Поддубный // Научно-технические ведомости СПбГПУ. Информатика. Телекоммуникации. Управление. –СПб.: Изд-во Политехнического ун-та, 2012. –№ 4 (152). –С.102–105.
2. Microsoft Windows Phone 7 [Электронный

- ресурс] / Режим доступа <http://www.microsoft.com/design/toolbox/tutorials/windows-phone-7/metro/>
3. Microsoft Windows Phone 8 [Электронный ресурс] / Режим доступа <http://www.microsoft.com/windowsphone>
4. XAML [Электронный ресурс] / Режим доступа <http://msdn.microsoft.com/en-us/library/ms752059.aspx>

#### REFERENCES

1. **Voinov N.V., Kotliarova L.P., Poddubnyi I.E.** Metodika razrabotki mobil'nykh prilozhenii dlia platformy Windows Phone 7 [*Methodic of WINDOWS PHONE 7 mobile applications development*] / Nauchno-tekhnicheskie ведомosti SPbGPU. Informatika. Telekommunikatsii. Upravlenie [*St. Petersburg State Polytechnical University Journal. Computer Science. Telecommunications and Control Systems*]. –St. Petersburg: Izd-vo

- Politekhnikeskogo un-ta, 2012. – № 4 (152). – S. 102–105. (rus)
2. Microsoft Windows Phone 7. Available <http://www.microsoft.com/design/toolbox/tutorials/windows-phone-7/metro/>
3. Microsoft Windows Phone 8. Available <http://www.microsoft.com/windowsphone>
4. XAML. Available <http://msdn.microsoft.com/en-us/library/ms752059.aspx>

---

**ВОИНОВ Никита Владимирович** – доцент кафедры информационных и управляющих систем Санкт-Петербургского государственного политехнического университета, кандидат технических наук.  
195251, Россия, Санкт-Петербург, ул. Политехническая, д. 29.

**VOINOV, Nikita V.** *St. Petersburg State Polytechnical University.*  
195251, Politekhnikeskaya Str. 29, St. Petersburg, Russia.

**АЛЕКСАНДРОВА Ольга Всеволодовна** – старший преподаватель кафедры информационных и управляющих систем Санкт-Петербургского государственного политехнического университета.  
195251, Россия, Санкт-Петербург, ул. Политехническая, д. 29.

**ALEKSANDROVA, Olga V.** *St. Petersburg State Polytechnical University.*  
195251, Politekhnikeskaya Str. 29, St. Petersburg, Russia.

**КОТЛЯРОВА Лина Павловна** – старший преподаватель кафедры информационных и управляющих систем Санкт-Петербургского государственного политехнического университета.  
195251, Россия, Санкт-Петербург, ул. Политехническая, д. 29.

**KOTLYAROVA, Lina P.** *St. Petersburg State Polytechnical University.*  
195251, Politekhnikeskaya Str. 29, St. Petersburg, Russia.

УДК 004.42

*М.М. Заславский, К.В. Кринкин, Э.М. Рябиков,  
В.В. Черкалова, А.Б. Цамутали*

## **ИНСТРУМЕНТАРИЙ НАДЕЖНОГО ХРАНЕНИЯ ДАННЫХ**

*M.M. Zaslavskiy, K.V. Krinkin, E.M. Ryabikov,  
V.V. Cherkalova, A.B. Tsamutali*

### **TOOLSET FOR SECURE DATA STORAGE**

Рассмотрены подходы к созданию программного обеспечения для функционирования и отладки систем надежного хранения данных. Изучены подходы к неинвазивному профилированию динамических библиотек в Linux. Представлен алгоритм обновления ресурсов в кластере, позволяющий обновлять ресурсы без нарушения целостности кластера (принудительной остановки узлов). Рассмотрена методика автоматизации просмотра файлов журнала, позволяющая оптимизировать просмотр и обработку данных тестирования.

**КЛАСТЕР. ОБНОВЛЕНИЕ РЕСУРСОВ. ПРОФИЛИРОВАНИЕ. РАЗДЕЛЯЕМЫЕ БИБЛИОТЕКИ. АНАЛИЗ ЖУРНАЛОВ ТЕСТИРОВАНИЯ.**

Different approaches to data storage systems software work and debugging tools creation are presented in this work. Approaches to non-invasive profiling of Linux dynamic libraries. Also, the algorithm of cluster resources update which allows to update the cluster resources without cluster consistency disruption (i.e. forced nodes stop) is considered. Moreover, the methods for log-file view automation, allowing the optimization of test results analysis.

**CLUSTER. RESOURCE UPDATE. PROFILING. SHARED LIBRARIES. TEST LOG ANALYSIS.**

#### **Разработка инструмента неинвазивного профилирования разделяемых библиотек в ОС Linux**

**Ограничения стандартных методов профилирования разделяемых библиотек в Linux.** Одним из специальных случаев профилирования является выявление временных и частотных характеристик вызовов подпрограмм из динамически компокуемых библиотек (shared libraries). Стандартные методы, основанные на инструментации исходного кода изучаемых программ, имеют широкий диапазон применения, однако не позволяют работать при отсутствии возможности компиляции библиотек, что особенно *актуально* при отладке встраиваемых приложений и систем.

Разработанный инструмент позволяет выполнять профилирование при отсутствии исходного кода и привилегий суперпользователя в операционной системе Linux [2]. В его основе лежит методика неинвазивного профилирования.

**Принципы неинвазивного профилирования разделяемых библиотек.** Методика неинвазивного (noninvasive) профилирования обеспечивает решение описанных выше проблем. Перечислим ее основные положения.

1. Инструмент профилирования не должен внедряться в программный код, а профилирование не должно зависеть от прав на использование системных средств.

2. Профилирование проводится над выбранным множеством функций в конкретные моменты их вызова.

3. Действия по профилированию не должны затрагивать алгоритм работы анализируемого приложения.

4. Профилировщик должен потреблять минимальное количество ресурсов процессора и выполнять измерение с максимальной точностью.

Реализация данных идей основана на внедрении в процесс релокации символов при работе с разделяемыми библиотеками

на стадии компоновки и загрузки. Модификация релокации состоит в реализации средств перехвата вызовов профилируемых функций.

**Реализация механизма перехвата вызовов.** Перехват вызовов профилируемых функций является основой реализации методики неинвазивного профилирования. В общем случае данный механизм состоит из двух частей: кода перенаправления вызова (редиректор, *redirector*) и функции-обертки (*wrapper*).

Редиректор представляет собой набор машинных команд для следующих ассемблерных инструкций (пример для платформы x86) [3]:

```
push $fcnPtr
jmp $wrapper_addr
```

Функции редиректора: сохранение адреса профилируемой функции в стеке программы и безусловный переход по адресу функции-обертки. Особенность кода перенаправления вызовов состоит в том, что он создается индивидуально для каждой профилируемой функции и размещается непосредственно в сегменте данных виртуальной памяти процесса. При этом средства ОС позволяют сделать данный набор инструкций исполняемым, что позволяет ему выполняться при вызове вместо интересующей функции (подробнее это описано в [4]).

Функция-обертка служит для вызова профилируемой функции и измерения времени ее работы при передаче управления от редиректора.

Алгоритм работы данной функции:

1. Получение управления от редиректора с сохранением состояния всех регистров в стеке программы и вызовом функции предварительного профилирования.

2. Создание функцией предварительного профилирования специальной структуры – контекста вызова, в которую сохраняется адрес профилируемой функции, адрес возврата и время начала работы.

3. Замена адреса возврата профилируемой функции, находящегося в стеке процесса, на адрес некоторой метки внутри функции-обертки и восстановление сохраненного состояния регистров.

4. Безусловный переход по адресу про-

филируемой функции.

5. Выполнение профилируемой функции и возврат в функцию-обертку по замененному адресу возврата.

6. Сохранение состояния регистров в стеке завершающего профилирования.

7. Восстановление функцией завершающего профилирования контекста вызова и измерение времени окончания работы, а также передача в функцию-обертку настоящего адреса возврата профилируемой функции.

8. Размещение настоящего адреса возврата в стеке и восстановление сохраненных регистров.

9. Возврат в вызывающую программу.

Главная особенность механизма «обертки» функций – то, что при профилировании не нарушается состояние регистров процессора и стека программы. Это позволяет работать с большим множеством функций, написанных на языках C/C++, независимо от архитектуры, соглашения о вызовах либо способа оптимизации программ.

**Ограничения разработанного инструмента при измерении времени.** Задача измерения времени работы функций ложится на функцию-обертку. При этом для измерения времени с максимальной разрешающей способностью применяется чтение состояния счетчика тактов процессора (Time Stamp Counter – TSC).

Однако использование такого метода измерений в многопроцессорной системе имеет побочный эффект, связанный с немонотонностью работы отдельно взятого процессорного таймера и асинхронностью совместной работы нескольких таймеров [5].

В связи с этим общие результаты, получаемые при профилировании с таким измерением времени, следует считать относительными. Иными словами, более точные результаты профилирования можно получить, измеряя время работы функций для двух состояний системы: нагруженного и ненагруженного.

В результате исследования и разработки инструмента неинвазивного профилирования разработан инструмент профилирования вызовов функций из разделяемых библиотек, основанный на использовании

динамического компоновщика и библиотеки динамической загрузки в ОС Linux; инструмент протестирован при профилировании системных утилит и сторонних приложений в ОС Debian GNU/Linux 6.0.

В дальнейшем планируется решить проблемы измерения времени в многопроцессорных системах и распространить решения на другие версии ОС Linux.

#### **Адаптивный алгоритм обновления ресурсов в отказоустойчивых кластерах**

Кластеры, состоящие из нескольких равноправных вычислительных систем (узлов), могут использоваться для оптимизации распределения вычислительной мощности, а также для повышения отказоустойчивости приложений (ресурсов). На каждом вычислительном узле кластерной системы запускается ряд служебных ресурсов, предназначенных для поддержания целостности служб кластера, слежения за состоянием аппаратного обеспечения и мониторинга работоспособности приложений, непосредственно выполняющих задачи по обработке информации.

Существующие системы управления кластерами требуют наличия идентичных версий ресурсов на всех узлах кластера [7], что не позволяет обновлять ресурсы не прерывая синхронизацию состояния узлов в кластере, что в свою очередь может привести к некорректному запуску ресурсов. Возможность автоматического обновления ресурсов с минимизацией времени простоя узла необходима для реализации отказоустойчивых систем, в которых длительное отключение одного узла ведет к возможной потере функциональности на других узлах кластера.

В данной статье представлен алгоритм, позволяющий обновлять ресурсы без нарушения целостности кластера (принудительной остановки узлов) на примере кластер-менеджера Pacemaker. Этот кластер-менеджер представляет собой одну из последних реализаций Linux High Availability (HA) кластеров, следующих спецификации Open Cluster Framework (OCF) [6].

В процессе разработки алгоритма обновления были выделены следующие этапы: 1) определение факторов, существенных в процессе обновления ресурсов кластера;

2) реализация алгоритма для кластера с двумя узлами; 3) расширение алгоритма для кластера с произвольным числом узлов. Кроме того, были определены способы задания версий кластерной конфигурации ресурса и его кода на каждом узле.

Процесс установки новой версии приложения на узел кластера характеризуется следующим: все ресурсы мигрируют на другие узлы; узлы поочередно отключаются от кластера; происходит обновление приложения; узел присоединяется к кластеру; алгоритм обновления записывает новые версии ресурсов в конфигурацию узла; выполняются необходимые действия по обновлению кластерной конфигурации ресурса; ресурсам позволяется стартовать на данном узле.

Для реализации предложенного процесса необходимы следующие функции кластер-менеджера: функция единовременной миграции ресурсов с конкретного узла кластера с сохранением зависимостей между ресурсами (в Pacemaker это свойство выполняется автоматически при отключении узла от кластера) и функция старта кластер-менеджера на узле без запуска ресурсов [9] (в Pacemaker эта функциональность не реализована [7]).

Предложенная функциональность автоматизирует следующие действия:

- при запуске процесса обновления на узле кластера происходит остановка кластер-менеджера, влекущая за собой миграцию ресурсов с узла;
- после установки новых версий приложений и конфигурационных файлов выполняется старт кластер-менеджера;
- первым стартует служебный ресурс, который сравнивает версии ресурсов на системе и в конфигурационных файлах;
- служебный ресурс запускает обновление кластерной конфигурации изменившихся ресурсов, обновляет версии ресурса на узле и зависимости, позволяющие блокировать одновременную работу несовместимых версий одного и того же ресурса;
- по завершении запуска служебного ресурса другие ресурсы могут стартовать на кластере в соответствии со вновь определенными зависимостями.

В итоге можно сформулировать список свойств, необходимых для автоматического обновления ресурсов кластера и отсутствующих как в спецификации OCF, так и в текущих реализациях Linux HA кластер-менеджерах:

- 1) возможность задания версий конфигурации и версий кода ресурсов;
- 2) возможность запускать кластер-менеджер на определенном узле без автоматического запуска ресурсов на этом узле;
- 3) возможность добавлять ресурсам зависимости от фантомных ресурсов.

При добавлении указанных свойств появляется возможность реализовать общий алгоритм обновления для произвольного количества узлов в кластере.

#### Анализ файлов журнала

Файл регистрации, протокол, журнал или log-файл – синонимы файла, хранящего записи о событиях в хронологическом порядке. Сведения хранятся с необходимой детальностью в определенном стандартном формате в текстовом виде.

Такие файлы (отчеты) содержат, в т. ч. и записи об ошибках, простоях системы и прочих проблемах. Часто анализа подобных отчетов достаточно для нахождения искомой неисправности или ошибки. Поэтому при разработке и модификации программного обеспечения анализ журнала тестирования является важной задачей. Как правило, ручная процедура анализа выполняется разработчиком или системным администратором и крайне трудозатратна. Есть ряд

моментов, которые делают «ручной» поиск необходимых строк в log-файле длительной процедурой.

Log-файлы, обычно, достаточно велики и содержат большое количество однообразных строк. Например, чтобы разобрать одну запись методом визуального просмотра нужно не менее 5 с, но на практике приходится прибегать к одновременному просмотру нескольких файлов, что существенно увеличивает время просмотра. Сложность анализа усугубляется тем, что трудно уловить причинно-следственные связи между событиями, приходится вручную производить поиск ошибок и отладочной информации, кроме того, в процессе просмотра невозможно оставлять пометки.

В рамках данной статьи сформулирован подход к решению описанных проблем, заключающийся в автоматизации, позволяющей:

- избежать проблемы обработки громоздких журналов;
- автоматически выделить важную информацию из строк журнала;
- создать сервисы для автоматизированного поиска и анализа журнала;
- создать удобное визуальное представление информации;
- создать возможность пользователю оставлять пометки и комментарии в тексте файлов, не нарушая целостности исходных файлов журнала.

В настоящее время подобный инструмент находится в разработке.

#### СПИСОК ЛИТЕРАТУРЫ

1. The ELF Object File Format by Dissection [Электронный ресурс] / Режим доступа <http://www.linuxjournal.com/node/1060/print>
2. **Michael Kerrisk**. The Linux Programming Interface [Text] / Kerrisk Michael. –San-Francisco: No Starch Press, 2010. –P. 833–859.
3. Intel x86 Function-call Conventions – Assembly View [Электронный ресурс] / Режим доступа <http://www.unixwiz.net/techtips/win32-call-conv-asm.html>
4. SSE (Streaming SIMD Extentions) [Электронный ресурс] / Режим доступа <http://www.songho.ca/misc/sse/sse.html>
5. **Matz Michael**. System V Application Binary Interface. AMD64 Architecture Processor Supple-

ment [Электронный ресурс] / Michael Matz, Jan Hubicka, Andreas Jaeger [et al.] // AMD64 ABI Draft 0.96 – June 14, 2005.

6. Corosync Configuration [Электронный ресурс] / Режим доступа <http://www.corosync.org/>

7. **Beekhof Andrew**. Pacemaker 1.1 Configuration Explained. An A-Z guide to Pacemaker's Configuration, Options Edition [Электронный ресурс] / Andrew Beekhof, 2012.

8. **Geist Al**. PVM: Parallel Virtual Machine. A Users' Guide and Tutorial for Networked Parallel Computing [Text] / Al Geist, Adam Beguelin, Jack Dongarra [et al.]. –Massachusetts Institute of Technology, 1994.

## REFERENCES

1. The ELF Object File Format by Dissection. Available <http://www.linuxjournal.com/node/1060/print>
2. **Kerrisk M.** The Linux Programming Interface. — San-Francisco: No Starch Press, 2010. — P. 833–859.
3. Intel x86 Function-call Conventions — Assembly View. Available <http://www.unixwiz.net/techtips/win32-callconv-asm.html>
4. SSE (Streaming SIMD Extentions). Available <http://www.songho.ca/misc/sse/sse.html>
5. **Matz Michael, Hubicka Jan, Jaeger Andreas, Mitchell Mark.** System V Application Binary Interface. AMD64 Architecture Processor Supplement; AMD64 ABI Draft 0.96; 14.06.2005.
6. Corosync Configuration. Available <http://www.corosync.org/>
7. **Andrew Beekhof.** Pacemaker 1.1 Configuration Explained. An A-Z guide to Pacemaker's Configuration, Options Edition, 2012.
8. **Geist Al, Beguelin Adam, Dongarra Jack, Jiang Weicheng, Manchek Robert, Sunderam Vaidy.** PVM: Parallel Virtual Machine. A Users' Guide and Tutorial for Networked Parallel Computing. — Massachusetts Institute of Technology, 1994.

---

**ЗАСЛАВСКИЙ Марк Маркович** — студент кафедры системного анализа и управления Санкт-Петербургского государственного политехнического университета.

195251, Россия, Санкт-Петербург, ул. Политехническая, д. 29.

E-mail: mark.zaslavskiy@gmail.com

**ZASLAVSKIY, Mark M.** St. Petersburg State Polytechnical University.

195251, Politekhnikeskaya Str. 29, St. Petersburg, Russia.

E-mail: mark.zaslavskiy@gmail.com

**КРИНКИН Кирилл Владимирович** — доцент кафедры математического обеспечения и применения ЭВМ Санкт-Петербургского государственного электротехнического университета «ЛЭТИ», кандидат технических наук.

197376, Россия, Санкт-Петербург, ул. Профессора Попова, д. 5.

**KRINKIN, Kirill V.** Saint Petersburg Electrotechnical University «LETI».

197376, Professora Popova Str. 5, St. Petersburg, Russia.

**РЯБИКОВ Эдуард Михайлович** — студент кафедры математического обеспечения и применения ЭВМ Санкт-Петербургского государственного электротехнического университета «ЛЭТИ».

197376, Россия, Санкт-Петербург, ул. Профессора Попова, д. 5.

E-mail: Edward.ryabikov@gmail.com

**RYABIKOV, Eduard M.** Saint Petersburg Electrotechnical University «LETI».

197376, Professora Popova Str. 5, St. Petersburg, Russia.

E-mail: Edward.ryabikov@gmail.com

**ЧЕРКАЛОВА Виктория Владимировна** — магистр кафедры информационных и управляющих систем Санкт-Петербургского государственного политехнического университета.

195251, Россия, Санкт-Петербург, ул. Политехническая, д. 29.

E-mail: Vitoria@bk.ru

**SHERKALOVA, Viktoria V.** St. Petersburg State Polytechnical University.

195251, Politekhnikeskaya Str. 29, St. Petersburg, Russia.

E-mail: Vitoria@bk.ru

**ЦАМУТАЛИ Алексей Борисович** — магистр кафедры информационных и управляющих систем Санкт-Петербургского государственного политехнического университета.

195251, Россия, Санкт-Петербург, ул. Политехническая, д. 29.

E-mail: indrik\_rush@mail.ru

**TSAMUTALI, Aleksey B.** St. Petersburg State Polytechnical University.

195251, Politekhnikeskaya Str. 29, St. Petersburg, Russia.

E-mail: indrik\_rush@mail.ru



УДК 621.319

*А.Ю. Жигулин, В.О. Сафонов*

## **ПРОЕКТИРОВАНИЕ ВИДЕОКОНТЕНТА НА СЛОЖНЫЕ ПОВЕРХНОСТИ С ПОМОЩЬЮ НЕСКОЛЬКИХ УСТРОЙСТВ ВЫВОДА**

*A.Yu. Zhigulin, V.O. Safonov*

### **VIDEO CONTENT PROJECTING ON COMPLEX SURFACES USING MULTIPLE OUTPUT DEVICES**

Рассмотрено приложение, проигрывающее контент на сложных поверхностях, не искажая его, а также сохраняя целостность контента. Представлено описание действия данного приложения, предложены перспективные пути развития данной области.

ПОВЕРХНОСТИ БЕЗЬЕ. ЭТАЛОННОЕ АЛЬФА-ЗНАЧЕНИЕ. ЦЕЛОСТНОСТЬ КОНТЕНТА. DIRECTX. DIRECTSHOW.

The article discusses the application, which plays content on complex surfaces without distorting it, as well as maintains the integrity of the content. The description of the application steps is presented, the development perspectives of this area are proposed.

BEZIER SURFACES. THE REFERENCE ALPHA VALUE. INTEGRITY OF THE CONTENT. DIRECTX. DIRECTSHOW.

В области медиа одной из самых сложных проблем является представление контента на любой поверхности без искажения изображения. В зависимости от предполагаемого устройства вывода контента на целевую поверхность, расстояние между устройством и поверхностью, а также площадь отображения контента данным устройством может варьироваться. За этой проблемой следует и другая: при наличии нескольких устройств вывода необходимо не только настроить каждое устройство по отдельности, но и правильно состыковать выводимые этими устройствами изображения.

При данной стыковке изображение искажается, а также может иметь неоднородный альфа-канал, что разрушает целостность контента. На данный момент существующие приложения-проигрыватели либо полностью игнорируют проблемы целостности изображения, либо не поддерживают стыковку в целом.

#### **Описание реализации**

В данной статье рассматривается пакет приложений, предназначенный для следующих задач:

стыковка отдельных частей отображаемого контента, в результате которой получается единое изображение;

установка правильной яркости изображений — приведение их к общему значению альфа-канала;

проигрывание изображений в распараллеленном режиме, позволяющее сохранить целостность контента.

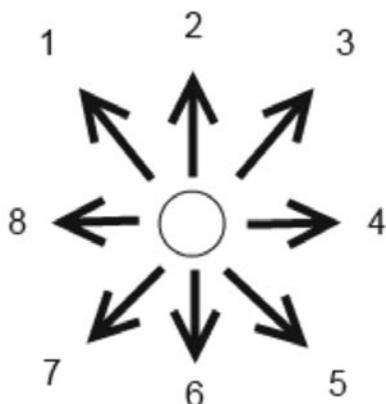
Суть работы системы описывается следующими шагами.

1. Запуск проигрывателя, на который подается статичная однотонная картинка.

2. Запуск настройщика поверхностей, который

- подключается к камере с высоким разрешением, анализирующей область отображения;

- перемещает точки поверхностей Безье, определяющие форму итогового изображения. В каждый момент времени настройщик хранит в себе координаты всех определяющих точек Безье, выбирает две соседние поверхности, не удовлетворяющие заданной точности, подсвечивает определяющие точки данных поверхностей и начинает



Направления смещения точки

смещать их в направлении стыковки. При соблюдении точности стыковки пары соответствующих друг другу точек, алгоритм переключается на следующую пару. После соединения всех пар точек данных поверхностей запускается вариативный алгоритм, смещающий точку, наиболее близкую к самой несостыкованной части изображения (определяется на основании данных камеры итоговая площадь несостыковки  $S$ , равная сумме площадей недоведенной до стыковки части и части, ушедшей в перекрытие). На рисунке показаны направления смещения точки на один из восьми близлежащих пикселей.

Каждое смещение продолжает цепочку вариативного смещения, запоминая промежуточный результат как наилучший в случае уменьшения общей площади несостыковки. Если после достаточно большого количества шагов все результаты ведут к ухудшению результата, данная ветка отбрасывается как бесперспективная. После успешной стыковки двух поверхностей алгоритм выбирает следующую пару поверхностей. Алгоритм также запоминает уже смещенные ранее точки и сопоставленные им точки соседних поверхностей как «пары соответствия», что позволяет учитывать сразу несколько точек в случае, когда в особо сложном месте необходимо стыковать сразу несколько поверхностей Безье;

- останавливается в случае, если соблюдена необходимая стыковка изображений (по заданной точности).

3. Запуск анализатора альфа-значений контента, который

- определяет минимальное альфа-значение на поверхности, достигнутое отображением контента на максимальной яркости;
- устанавливает данное альфа-значение как эталонное;
- подсвечивает другим цветом один из пикселей изображения;
- фиксирует камерой область, полностью охватывающую данный пиксель;
- варьирует альфа-значение пикселя до максимального соответствия среднего альфа-значения области с эталонным альфа-значением. Выбирается такое альфа-значение, которое наименее отклоняется от эталонного в большую либо меньшую сторону, что считается допустимой погрешностью, неразличимой для человеческого глаза;
- обрабатывает следующий пиксель.

Рассматриваемая система разделена на несколько частей:

1. Проигрыватель, реализованный на платформе .NET (язык C#), использующий следующие технологии Microsoft:

DirectShow [1] (отвечает за проигрывание полученного контента, а также за наложение фильтров — масок необходимого разрешения, у которых присутствует только альфа-канал, — уменьшающих альфа-канал у отдельных пикселей исходного контента);

DirectX [2] (отвечает за создание поверхностей вывода с полученными данными поверхностей Безье, на которые накладывается модифицированный контент).

2. Настройщик поверхностей, на которые отображается контент (реализован на платформе .NET (язык C#), с использованием технологии DirectX). Настройщик варьирует определяющие точки поверхностей Безье, опираясь на данные камеры.

3. Анализатор альфа-значений контента (реализован на платформе .NET (язык C#)), варьирующий альфа-значение отдельных пикселей изображения, опираясь на данные камеры.

В ходе проведенных исследований получены следующие результаты.



- Создан проигрыватель, позволяющий отображать контент на любом количестве устройств вывода (основные ограничения накладываются из-за мощности используемого оборудования: требуются видеокарты с большим количеством видеовыходов, а также мощный процессор, от которого зависит скорость определения итоговых координат точек поверхностей Безье и альфа-значений).

Также была рассмотрена дополнительная перспективная версия проигрывателя. Проигрыватель с клиент-серверной системой: главный компьютер-синхронизатор, на котором запущена серверная версия проигрывателя, отвечающая за синхронизацию изображений, а также несколько компьютеров с клиентской версией, получающих координаты для своей поверхности Безье и отвечающие за проигрывание своей части контента. Помимо соединения клиентских компьютеров с устройствами вывода, можно соединять с этими устройствами и серверный проигрыватель. В таком случае приложение в случае отказа одного из клиентов сможет перевести часть управления на себя, запустив резервную копию клиентского приложения и переключив канал входа на устройстве вывода с помощью протокола управления устройством (например, посредством передачи команд через RS-232).

Плюс данной системы заключается в

упрощении поиска подходящего оборудования для компьютера: вместо многоканальных видеокарт можно использовать одноканальные.

- Создан анализатор альфа-значений, позволяющий при наличии камеры с высоким разрешением стабилизировать альфа-канал на любой охватываемой поверхности.

- Создан прототип настройщика поверхностей, который оперирует определяющими точками поверхностей Безье, тем самым стыкуя различные участки изображения и получая единую картинку.

Являясь перспективным направлением, рассмотренная система позволяет уменьшить требования при проигрывании контента, т. к. она способна функционировать на любых поверхностях, опираясь лишь на алгоритм стыковки отдельных изображений.

Тем не менее данная система очень зависима от предоставленного оборудования: необходимо и мощное оборудование для функционирования алгоритмов системы, и дорогостоящие устройства вывода, камеры с высоким разрешением.

Также при сложных поверхностях необходимо продолжительное время в случае большой точности стыковки: вариативному алгоритму потребуется множество шагов, чтобы определить подходящее положение точек Безье, особенно в местах стыков нескольких поверхностей.

#### СПИСОК ЛИТЕРАТУРЫ

1. **Pesce, Mark D.** Programming Microsoft DirectShow for Digital Video and Television [Text] / Mark D. Pesce. — USA, 2003. — 448 p.

2. **Thorn, A.** DirectX 9 Graphics: The Definitive Guide to Direct 3D [Text] / A. Thorn. — USA, 2005. — 349 p.

#### REFERENCES

1. **Pesce Mark D.** Programming Microsoft DirectShow for Digital Video and Television. — USA, 2003. — 448 p.

2. **Thorn A.** DirectX 9 Graphics: The Definitive Guide to Direct 3D. — USA, 2005. — 349 p.

---

**ЖИГУЛИН Андрей Юрьевич** — аспирант Санкт-Петербургского государственного университета. 199034, Россия, Санкт-Петербург, Университетская наб., д. 7-9.  
E-mail: dron.subzero@gmail.com

**ZHIGULIN Andrei Yu.** — St. Petersburg State University. 199034, Universitetskaya nab. 8-9, St. Petersburg, Russia.  
E-mail: dron.subzero@gmail.com

**САФОНОВ Владимир Олегович** — доктор технических наук Санкт-Петербургского государственного университета.

199034, Россия, Санкт-Петербург, Университетская наб., д. 7-9.

E-mail: vosafonov@gmail.com

**SAFONOV, Vladimir O.** — *St. Petersburg State University.*

199034, Universitetskaya nab. 8-9, St. Petersburg, Russia.

E-mail: vosafonov@gmail.com



УДК 004.942

А.М. Батюков, Н.Б. Ампилова

## МОДИФИЦИРОВАННЫЙ АЛГОРИТМ МОДЕЛИРОВАНИЯ АГРЕГАЦИИ, ОГРАНИЧЕННОЙ ДИФФУЗИЕЙ

А.М. Batyukov, N.B. Ampilova

### A MODIFIED DIFFUSION-LIMITED AGGREGATION MODELLING ALGORITHM

Рассмотрен метод моделирования процессов агрегации, ограниченной диффузией (DLA), основанный на броуновском движении частиц по узлам решетки на плоскости. Реализован базовый метод, отслеживающий блуждание по решетке последовательно бросаемых частиц. Разработана и реализована оптимизация этого метода, основанная на априорной оценке коэффициентов выбора для брошенной на плоскость частицы. Приведены результаты численных экспериментов.

АГРЕГАЦИЯ, ОГРАНИЧЕННАЯ ДИФФУЗИЕЙ. ФРАКТАЛЬНЫЙ КЛАСТЕР. БРОУНОВСКОЕ ДВИЖЕНИЕ.

For processes of diffusion-limited aggregation (DLA) a simulation method based on Brownian motion of particles on a plane lattice has been considered. The basic method to trace the walking of sequentially dropped particles has been realized. The optimized method using a priori estimation of particle selection coefficients has been designed and implemented. The results of numerical experiments are given.

DIFFUSION-LIMITED AGGREGATION (DLA). FRACTAL CLUSTER. BROWNIAN MOTION.

Как показывают многочисленные исследования, многие процессы, происходящие в природе и обществе, обладают хаотической динамикой, которая характеризуется возникновением фрактальных структур. Эти структуры необычайно разнообразны: от строгого самоподобия до статистического (неполная повторяемость). Последние часто называют *хаотическими фракталами*.

Важный класс среди хаотических фракталов составляют т. н. *фрактальные кластеры* — класс физических объектов, плотность которых уменьшается по мере роста с увеличением размера кластера. Исследование фрактальных кластеров началось с появления теоретической модели агрегации, ограниченной диффузией (diffusion limited aggregation — DLA), которая описывает агрегацию частиц в условиях их случайного блуждания. Потом были предложены и другие модели — cluster-cluster aggregation (CCA) и ее модификации, основанные на различных вероятностях прилипания частиц к кластеру. Их подробный обзор можно найти в [4].

Модель DLA оказалась применимой для имитационного моделирования различных явлений: осаждение металла при электролизе, электрический разряд при пробое диэлектрика, диффузионные процессы в жидкости. В работе [5] эта модель использована для имитации роста минеральных дендритов, бактериальных колоний, а также образования кластера при высыхании на стекле коллоидного раствора.

**Описание базовой модели.** Мы рассматриваем модель DLA, предложенную Э. Пинским в [1]. Согласно этой модели блуждание частиц происходит по линиям дискретной прямоугольной сетки. Кроме того, предполагается, что частицы присоединяются к агрегату последовательно, одна за другой. Процесс построения агрегата начинается с единственной частицы, входящей в агрегат и располагающейся в начале координат прямоугольной сетки. Случайным образом вычисляется первоначальное положение другой частицы. Вторая частица начинает блуждание по линиям сетки, при этом на каждом шаге может переместиться

на единицу вправо, либо влево, вверх или вниз относительно текущего положения на координатной сетке. Блуждание продолжается до тех пор, пока частица не окажется соседней с какой-либо из частиц, входящих в агрегат. В соответствии с некоторым коэффициентом, называемым *коэффициентом прилипания*, определяется, остановится ли частица или продолжит свое блуждание по сетке. Остановившаяся частица соединяется отрезком с агрегатом и, начиная с этого момента, входит в него. После этого происходит бросание новой частицы. Описанный процесс повторяется многократно.

При практической реализации описанного алгоритма обычно вводят некоторые дополнительные ограничения. Так, точки бросают на плоскость случайным образом в некоторую ограниченную окрестность агрегата. Если точка при блуждании отходит слишком далеко от агрегата, ее отбрасывают. Кроме того, ограничивают число шагов при блуждании: точка не присоединяется к агрегату, если число шагов превышает некоторый заданный параметр.

Возможна реализация алгоритма с различными значениями коэффициента прилипания, для простоты мы рассматриваем случай, когда при приближении точки к агрегату она присоединяется к нему с вероятностью, равной единице.

**Реализация базовой модели.** При моделировании процесса агрегации, ограниченной диффузией, реализованы алгоритмы с

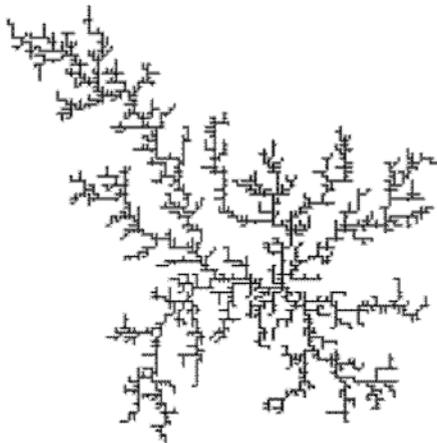


Рис. 1. Результат моделирования по базовому алгоритму, 10 000 бросаний

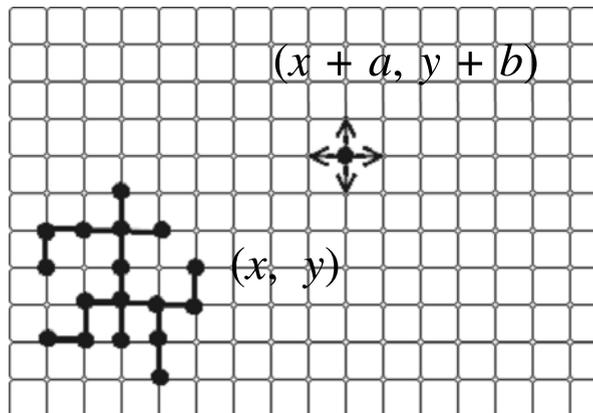


Рис. 2. Моделирование процесса DLA на прямоугольной решетке

описанными выше ограничениями:

- отбрасывание точек, удалившихся от агрегата в результате блуждания на расстояние, большее заданного параметра;

- отбрасывание точек, которые после 500 шагов не присоединились к агрегату.

Для каждого из алгоритмов проведена серия испытаний, в каждом из которых на плоскость бросали по 10 000 точек. Пример результата моделирования представлен на рис. 1. Среднее число точек в конечном агрегате для первого алгоритма равно 2034, время расчета составило около 40 ч. Среднее число точек в конечном агрегате для второго алгоритма равно 1357, время расчета составило около 20 ч.

**Оптимизация базовой модели.** Большое значение времени расчета объясняется большим числом обрабатываемых данных: число точек велико и, кроме того, траектория блуждания каждой точки может быть достаточно длинной. Заметим, что согласно работе [3] О.Н. Граничина, рассматриваемые нами алгоритмы представляют собой классический пример детерминированных алгоритмов, обрабатывающих большие объемы данных. Их можно ускорить за счет перехода к решению упрощенной задачи с частичной информацией.

Обратимся к рис. 2, на котором показан шаг процесса блужданий. Предположим, что при бросании частицы на координатную сетку ее координаты оказались равны  $(x + a, y + b)$ . Сделаем оценку того, что дан-

ная частица присоединится к точке агрегата с координатами  $(x, y)$ . Если при блуждании частицы по плоскости она сделала  $n$  шагов в сторону от точки  $(x, y)$  по оси абсцисс, то чтобы попасть в  $(x, y)$  частица должна сделать  $(n + a)$  шагов в сторону точки  $(x, y)$ . Соответственно, если при блуждании частица сделала  $m$  шагов в сторону от точки  $(x, y)$  по оси ординат, то она должна сделать  $(m + b)$  шагов в сторону точки  $(x, y)$ . На каждом шаге блуждания сторона следующего перехода определяется равновероятно между четырьмя вариантами. Это позволяет написать следующую оценку того, что через  $(2n + a + 2m + b)$  шагов координатами блуждающей точки окажутся  $(x, y)$ :

$$\frac{1}{4^n 4^{n+a} 4^m 4^{m+b}}.$$

Число шагов, через которые блуждающая точка попадет в  $(x, y)$ , определяется значениями  $n$  и  $m$ . Классы траекторий, определяемых конкретными значениями  $n$  и  $m$ , представляют собой разбиение множества всех траекторий на непересекающиеся классы. Поэтому, чтобы получить оценку для всего множества траекторий, подсчитаем сумму ряда:

$$\begin{aligned} & \sum_{m=0}^{\infty} \sum_{n=0}^{\infty} \frac{1}{4^n 4^{n+a} 4^m 4^{m+b}} = \\ & = \sum_{m=0}^{\infty} \sum_{n=0}^{\infty} \frac{1}{4^{2n+2m+a+b}} = \\ & = \frac{1}{4^{a+b}} \sum_{m=0}^{\infty} \sum_{n=0}^{\infty} \frac{1}{16^{n+m}} = \\ & = \frac{1}{4^{a+b}} \sum_{m=0}^{\infty} \frac{1}{16^m} \sum_{n=0}^{\infty} \frac{1}{16^n} = \left(\frac{16}{15}\right)^2 \frac{1}{4^{a+b}}. \end{aligned}$$

Полученное значение дает верхнюю оценку величины, определяющей среднюю скорость, с которой блуждающая точка с координатами  $(x + a, y + b)$  может попасть в точку  $(x, y)$ . Назовем эту величину коэффициентом выбора и положим  $P(a, b) \sim \frac{1}{4^{(a+b)}}$ .

Полученный коэффициент выбора зависит только от первоначального взаиморасположения брошенной точки и точки агрегата (параметры  $a$  и  $b$  — расстояние между точками по соответствующим осям координат). Это позволяет для брошенной

на плоскость частицы априори вычислить коэффициент выбора для каждой из граничных точек агрегата. Для определения, к какой точке агрегата присоединится блуждающая точка, поступаем следующим образом. Каждой граничной точке сопоставляем отрезок, длина которого — коэффициент выбора для этой точки. Из всех таких отрезков составляем один общий отрезок суммарной длины. Производим случайное бросание точки на этот общий отрезок. Тот из составляющих отрезков, в который попала точка, и определяет, к какой из граничных точек агрегата присоединится блуждающая точка.

Особо стоит отметить, что для каждого бросания число вычислений, необходимых для определения к какой именно точке агрегата присоединится новая точка, не зависит от каких-либо неизвестных на момент вычисления данных (таких, как число шагов в маршруте). Кроме того, ни одна из брошенных точек не отбрасывается.

При таком методе оценки практически не учитывается влияние на коэффициент выбора расположения других точек агрегата, которые могут оказывать влияние на траекторию. Кроме того, учтены также и те траектории, в которых точка при блуждании по линиям решетки попадает в точку агрегата неоднократно. Эти ограничения являются неизбежным следствием того

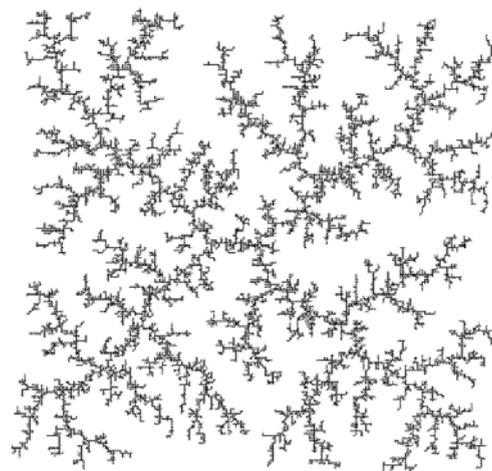


Рис. 3. Результат моделирования по оптимизированной модели, 10 000 бросаний

Результаты моделирования по описанным алгоритмам

Алгоритм	Число бросаний	Число точек в агрегате	Время моделирования
Базовый	10 000	2 034	40 ч
Базовый с ограничением числа шагов (500)	10 000	1 357	20 ч
Оптимизированный	2 000	2 001	4 мин
	10 000	10 001	26 мин
Оптимизированный с предподсчетом	10 000	10 001	9 мин (6 мин без учета времени предподсчета)

факта, что мы отказываемся от обработки части информации для ускорения вычислений.

**Результаты вычислений по оптимизированной модели.** По представленному выше алгоритму реализована программа построения агрегатов. Проведены две серии испытаний: в первой из них на плоскость кидали 2000 точек, во второй 10 000 точек. Все бросаемые точки входили в агрегат. В первом случае время вычислений составило около 4 мин, во втором – около 26 мин. Как видно из приведенных цифр, время моделирования агрегата сопоставимой мощности снизилось в несколько раз.

Заметим, что для ограниченной области на сетке можно заранее подсчитать всевозможные значения коэффициентов для всех сумм  $a + b$ , таких, что  $a$  и  $b$  удалены от начала координат не далее некоторого  $N$ . Тогда при бросании новой точки для каждой из граничных точек агрегата в качестве коэффициента выбора можно взять одно из предварительно посчитанных значений, что также сокращает время вычислений. Использование алгоритма с предподсчетом при бросании 10 000 точек позволяет сократить время вычисления до 9 мин, а если не включать время предподсчета в общее время выполнения программы (т. к. вычисленные коэффициенты могут быть сохранены и использоваться многократно), – до 6 мин. Результат моделирования для 10 000 бросаний представлен на рис. 3.

Все данные о времени моделирования агрегатов по указанным алгоритмам представлены в таблице.

В различных моделях агрегации, ограниченной диффузией, вычисляют размерность полученного фрактала (как правило – емкостную). Это позволяет сравнить размерность образца, полученного в процессе диффузии, и образца, полученного с помощью модели. В работе [1], где впервые рассматривалась описанная модель, для емкостной размерности получена оценка 1,66. Во всех изученных нами оптимизациях емкостная размерность построенных агрегатов колеблется от 1,62 до 1,73. Таким образом, оптимизированные алгоритмы дают качественно близкую картину агрегата.

Предложена эффективная оптимизация классического алгоритма построения агрегата для модели агрегации, ограниченной диффузией. При различных вариантах оптимизации время вычислений сокращается в несколько десятков раз. Сравнение структуры полученных агрегатов с результатами, полученными по базовым моделям, на основании значений емкостной размерности дает близкие результаты. Можно ожидать, что подобная оптимизация возможна и в других моделях, в частности, в агрегациях кластер-кластер, которые актуальны при решении задач моделирования процессов в жидкостях и газах, а также в трехмерных моделях агрегации.



## СПИСОК ЛИТЕРАТУРЫ

1. **Pinski, A.** Diffusion-Limited Aggregation [Text] / A. Pinski // The American Physical Society. – 1981.
2. **Witten, T.A.** Diffusion-limited aggregation [Text] / T.A. Witten, L.M. Sander // The American Physical Society. – 1982.
3. **Граничин, О.Н.** Обратные связи, усреднение и рандомизация в управлении и извлечении знаний [Текст] / О.Н. Граничин // Сб. Стохастическая оптимизация в информатике. – СПб.: Изд-во СПбГУ, 2012. –Т. 8. –Вып. 2. –С. 3–49.
4. **Смирнов, Б.М.** Физика фрактальных кластеров [Текст] / Б.М. Смирнов. –М.: Наука, 1991.
5. **Исаева, В.В.** Фракталы и хаос в биологическом морфогенезе [Текст] / В.В. Исаева, Ю.А. Каретин, А.В. Чернышев, Д.Ю. Шкуратов. –Владивосток, 2004.

## REFERENCES

1. **Pinski A.** Diffusion-Limited Aggregation; The American Physical Society, 1981.
2. **Witten T.A., Sander L.M.** Diffusion-limited aggregation; The American Physical Society, 1982.
3. **Granichin O.N.** Obratnye sviazi, usrednenie i randomizatsiia v upravlenii i izvlechenii znaniia /Sb. Stokhasticheskaia optimizatsiia v informatike. – St. Petersburg: Izd-vo St. Peterburgskogo un-ta, 2012. – Т. 8. –Вып. 2. – S. 3–49. (rus)
4. **Smirnov. B.M.** Fizika fraktal'nykh klasterov. – Moscow: Nauka, 1991. (rus)
5. **Isaeva V.V., Karetin Iu.A., Chernyshev A.V., Shkuratov D.Iu.** Fraktaly i khaos v biologicheskom morfogeneze. – Vladivostok, 2004. (rus)

---

**БАТЮКОВ Александр Михайлович** – аспирант кафедры информатики математико-механического факультета Санкт-Петербургского государственного университета.

197372, Россия, Санкт-Петербург, ул. Стародеревенская, д. 21/1.

E-mail: batsun@gmail.com

**BATYUKOV Alexander M.** St. Petersburg State University.

197372, Staroderevenskaya Str. 21/1, St. Petersburg, Russia.

E-mail: batsun@gmail.com

**АМПИЛОВА Наталья Борисовна** – доцент кафедры информатики математико-механического факультета Санкт-Петербургского государственного университета, кандидат физико-математических наук.

192283, Россия, Санкт-Петербург, Загребский бул., д. 35/28.

E-mail: n.ampilova@spbu.ru

**AMPILOVA Nataly B.** St. Petersburg State University.

192283, Zagrebsky blvd. 35/28, St. Petersburg, Russia.

E-mail: n.ampilova@spbu.ru

УДК 004.415

*И.В. Никифоров, В.П. Котляров, П.Д. Дробинцев*

## **ОГРАНИЧЕНИЯ НА МНОГОПОТОЧНЫЕ КОНСТРУКЦИИ И ВРЕМЕННЫЕ ЗАДЕРЖКИ ЯЗЫКА UCM**

*I.V. Nikiforov, P.D. Drobintsev, V.P. Kotliarov*

## **RESTRICTIONS ON CONCURRENT CONSTRUCTION AND TIME DELAYS OF UCM LANGUAGE**

Описано уточнение семантики языка UCM для моделирования систем реального времени с использованием временных задержек и параллельных потоков. Изучен подход преобразования многопоточных конструкций, временных задержек языка UCM в язык базовых протоколов.

Уточнение семантики предложено исходя из того, что существующая версия стандарта UCM позволяет создавать семантически некорректные модели. Описанные расширения и ограничения на язык позволяют решить проблемы некорректности модели в промышленных проектах.

UCM. ТАЙМЕР. ЗАДЕРЖКА. СИНХРОНИЗАЦИЯ. ПОТОКИ.

The paper describes an approach to adjustment of semantics for UCM real time constructions in implementation of translator into Basic Protocols notation. The following constructions and their adjustment are described: multithreading and delays.

The main problem of such constructions is that initial version of UCM standard allows to create semantically incorrect models. Proposed extensions and restrictions of UCM semantics allowed solving of these problems for different types of projects

UCM. TIMERS. DELAYS. SYNCHRONIZATION. THREAD.

Разработка программной системы начинается с создания требований. Документы, описывающие спецификацию промышленной системы, как правило, задаются на естественном языке и состоят из сотен или тысяч пунктов. Нередко исходные спецификации содержат ошибки, связанные с противоречивостью, неполнотой и неопределенностью в поведении системы. Обнаружение и исправление ошибок, связанных с требованиями, эффективно осуществлять на ранних этапах разработки [1].

Проанализировать спецификации промышленных систем на наличие ошибок вручную, без поддерживающего инструментария, практически невозможно. Существующие системы верификации и тестирования не могут работать с неформализованными спецификациями. Поэтому актуальна задача формализации исходных текстовых описаний требований с использованием входных языков средств верифи-

кации и тестирования.

Одной из перспективных интегрированных технологий автоматизации тестирования и верификации на основе формальных моделей является технология VRS/TAT [2], в которой для высокоуровневого описания поведенческих моделей используется нотация Use Case Maps (UCM) [3, 5], а инструменты автоматизации проверки и генерации работают с моделью на языке базовых протоколов [4].

Язык спецификаций UCM стандартизован, но тем не менее содержит ряд неточностей, не позволяющих однозначно и корректным образом отобразить семантику моделируемых систем.

В статье предложены ограничения, накладываемые на разработку многопоточных моделей систем, а также уточнения семантики конструкций языка UCM, моделирующих временные задержки и прерывания.

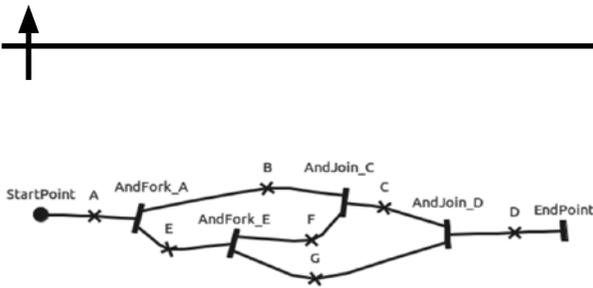


Рис. 1. Нарушение структуры параллельных потоков

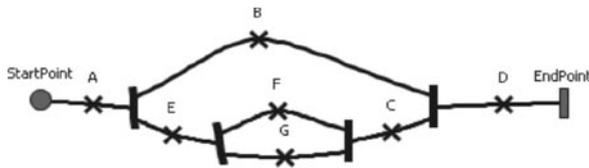


Рис. 2. Граф с правильной структурой потоков

### Ограничения на разработку многопоточных систем

Несмотря на все преимущества использования языка UCM и семантики его элементов, существующий стандарт Z.151 [3] содержит ряд неточностей, затрудняющих моделирование многопоточных систем.

**Баланс скобок в спецификации параллельных потоков.** Рассмотрим ситуацию, когда синтаксически правильные элементы порождения и синхронизации потоков на графе могут приводить к нарушению структуры параллельных потоков, а следовательно, к некорректному поведению системы.

На рис. 1 после элементов AndFork\_A и AndFork\_E происходит порождение потоков B, E и F, G соответственно, а на элементах AndJoin\_C и AndJoin\_D выполняется синхронизация потоков соответственно B, F и C, G.

Легко заметить, что синхронизация потоков, порожденных разными элементами, значительно затрудняет механизм поиска и отладки ошибок в системе, а также затрудняет возможность отслеживания связи «родитель/ребенок» в иерархии потоков. Такие связи полезны при анализе ситуаций, когда поток «ребенок» продолжает свое выполнение после завершения потока «родителя».

Граф поведения системы, который содержит корректную структуру порождения

и синхронизации потоков, изображен на рис. 2.

Анализ структуры потоков можно сравнить с анализом скобочной формы математических выражений. Если скобочная форма выражения нарушена, то оно считается синтаксически некорректным. Также и в моделировании параллельных потоков: если нарушена структура потоков, то и вся система считается синтаксически неверной.

Изучение потоков на наличие ошибок в структуре и их исправление позволяет создавать синтаксически корректные модели систем.

**Неограниченное порождение потоков.** Рассмотрим ситуацию, изображенную на рис. 3. После элемента D происходит порождение потоков B и E. Поток B завершает свое исполнение на элементе EndPoint. Поток E возвращается по циклу, не содержащему условия ограничения итераций, через элемент D и производит порождение новых потоков B' и E'. Сценарий поведения повторяется для потока E'.

Неограниченные циклы приводят к порождению неограниченного числа незавершенных потоков, что приводит к дефициту памяти или других ресурсов. Поэтому необходимо вводить количественные ограничения на использование таких конструкций в разрабатываемых моделях.

**Гонка данных при доступе параллельных процессов к общим ресурсам.** Рассмотрим ситуацию, когда на параллельных ветвях используются разделяемые ресурсы без синхронизации. На рис. 4 представлена ситуация, в которой два параллельных потока используют общий разделяемый ресурс var без синхронизации. Подобная формализация приводит к гонке при доступе к данным [6].

В модели наиболее интересны два сце-

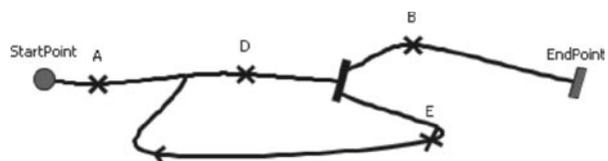


Рис. 3. Рекурсивное порождение потоков

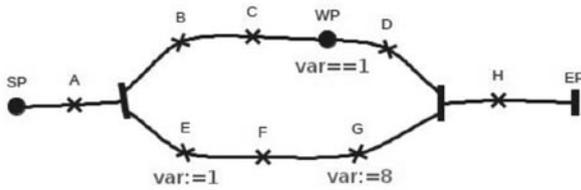


Рис. 4. Разделяемые ресурсы без синхронизации

нария исполнения:

если выполняется сценарий «E»-«F»-«G»-«WP», то элемент «D» никогда не будет применен, этот сценарий ведет в deadlock;

если выполняется сценарий «E»-«F»-«WP»-«G», то элемент «D» можно применить, и данный сценарий достигает конечной точки EP.

Устранить deadlock можно, добавив синхронизацию и тем самым исключив параллельный доступ к общему ресурсу системы (рис. 5).

**Предлагаемые ограничения на разработку многопоточных систем в UCM.** Запрещается использовать:

- параллельные конструкции с нарушением структуры потоков;
- неограниченное рекурсивное порождение потоков;
- разделяемые ресурсы без синхронизации на параллельных участках исполнения.

**Особенности моделирования временных задержек**

В промышленных системах часто встречаются требования создавать временные задержки. Следует отметить, что в рассматриваемом случае используется событийное моделирование с относительным временем — длительностью между событиями. События — это изменение значений атрибутов системы.

По стандарту [3] с элементом Timer (рис. 6) связывают два исходящих пути:



Рис. 5. Модель системы с синхронизацией

нормальный путь исполнения (regular path — RP) и путь по истечению (timeout path — TOP). Для выбора каждого из путей существуют условия CRP и STOP соответственно. Также существует триггер путь (trigger path или trigger counter), влияющий на поведение таймера и позволяющий отметить задержку.

Семантика элементов, моделирующих временные задержки, по стандарту Z.151 [3] содержит описание вариантов возможных поведений модели в зависимости от наступивших событий, но при этом не описывает, какие типы событий связаны с таймерами, и не определены типы ряда событий, характерных для спецификации телекоммуникационных приложений.

Расширим описание семантики таймера следующими событиями.

- Установка таймера: TIMER\_SET <имя таймера>. Событие происходит при достижении элемента Timer.
- Истечение таймера: TIMER\_EXPIRE <имя таймера>. Происходит после выполнения условия STOP.
- Остановка таймера: TIMER\_RESET <имя таймера>. Событие происходит после начала исполнения одного из путей RP или TOP, или наступления триггер события.

Используя семантику элемента Timer и связанных с ним событий, можно выделить три типа временных задержек:

- 1) простая задержка, особенность моделирования которой — строго зафиксированные условия исходящих путей (false) и отсутствие триггер события;
- 2) задержка с прерыванием, особенность моделирования которой — наличие триггер события;
- 3) задержка с прерыванием исполнения, особенность моделирования которой — наличие прерывания FailurePoint на таймаут пути.

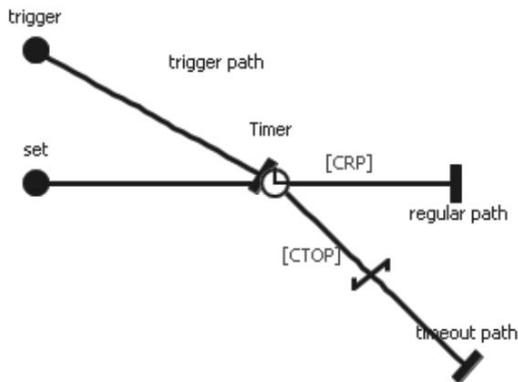


Рис. 6. UCM диаграмма с элементом Timer

Предложенное расширение семантики таймера позволило решить проблему описания задержек для телекоммуникационных проектов.

### Преобразование временных задержек в базовые протоколы

Для инструментов верификации и тестирования VRS/TAT разработан инструмент преобразования моделей на языке UCM в модели на языке базовых протоколов [8–10]. В трансляторе UCM→BP реализована концепция преобразования временных задержек и прерываний, а также реализованы проверки сформулированных ограничений на разработку многопоточных систем.

Для элемента Timer существует некоторый атрибут timer\_var, отвечающий за состояние таймера и принимающий два значения: true – если таймер установлен, false – если таймер остановлен. По умолчанию значение атрибута – false

В базовом протоколе для элемента Timer, отвечающем за установку таймера, в постусловии генерируется выражение timer\_var:=true, в поле процесса базового протокола генерируется выражение TIMER\_SET.

Для каждого исходящего пути (RP и TOP) из элемента Timer генерируется по одному базовому протоколу.

Предусловие протокола для пути RP (выражение для выбора регулярного пути) генерируется в соответствии с логической формулой, выведенной на основе [3]:

$$(timer\_var=true)\&(CRP) \vee (timer\_var=true)\&(trigger)\&(СТОП),$$

где trigger – логическое выражение для триггер события.

В поле процесса БП генерируется действие TIMER\_RESET.

В постусловии этого протокола генерируется выражение, моделирующее остановку таймера: timer\_var:=false.

Рассмотрим протокол для таймаут пути (TOP). В общем случае в предусловии генерируется выражение:

$$(timer\_var=true)\&(\sim CRP)\&[(СТОП) \vee (\sim trigger)\&(\sim СТОП)]$$

В поле процесса БП для пути TOP генерируются операции TIMER\_EXPIRE, TIMER\_RESET.

Таким образом, преобразование временных задержек сводится к генерации трех базовых протоколов с различными логическими выражениями в предусловиях.

Для каждого из рассмотренных случаев моделирования таймера можно произвести оптимизацию логических выражений, т. к. значения используемых конъюнктов и дизъюнктов заранее известно.

Рассмотренные в статье методы уточнения семантики элементов стандарта UCM, моделирующих временные задержки и прерывания, а также накладываемые ограничения на разработку многопоточных систем позволяют производить моделирование комплексных телекоммуникационных систем, при этом уменьшая возможность создания семантически неверных поведений в модели.

Методы реализованы в трансляторе UCM→BP, что позволяет сделать применение технологической цепочки VRS/TAT более удобным и эффективным на проектах средней и большой сложности.

Предложенный транслятор совместно с поддерживающим инструментарием технологии VRS/TAT был применен при разработке модулей телекоммуникационных приложений и показал существенное сокращение затрат при производстве качественного индустриального программного продукта.

### СПИСОК ЛИТЕРАТУРЫ

1. **Booch, Gr.** Object-Oriented Analysis and Design with Applications [Text] / Gr. Booch, R. Maksimchuk, M. Engel [et al.]. — 3rd ed. — Addison-Wesley Professional, 2007. — 720 p.

2. **Веселов, А.О.** Автоматизация тестирования в области телекоммуникаций [Текст] / А.О. Веселов, В.П. Котляров // Научно-технические ведомости СПбГПУ. Информатика. Телекоммуникации. Управление. — СПб.: Изд-во Политехнического ун-та, 2010. — № 4 (103). — С. 180–185.

3. Recommendation ITU-T Z.151 [Электронный ресурс] User requirements notation (URN), 11/2008.

4. **Летичевский, А.А.** Спецификация систем с помощью базовых протоколов [Текст] / А.А. Летичевский, Ю.В. Капитонова, А.А. Летичевский (мл.) [и др.] // Кибернетика и системный анализ. — 2005. — № 4. — С. 3–21.

5. **Buhr, R.J.A.** Use Case Maps for Object-Oriented Systems [Text] / R.J.A. Buhr, R.S. Casselman. — Prentice Hall, 1995.

6. **Гергель, В.П.** Высокопроизводительные вычисления для многопроцессорных многоядерных систем [Текст] / В.П. Гергель. — Нижний Новгород: Изд-во Нижегородского ун-та, 2010. — 544 с.

7. **Дробинцев, П.Д.** Автоматизация тестирования на основе покрытия пользовательских сценариев [Текст] / П.Д. Дробинцев,

В.П. Котляров, И.Г. Черноруцкий // Научно-технические ведомости СПбГПУ. Информатика. Телекоммуникации. Управление. — СПб.: Изд-во Политехнического ун-та, 2012. — № 4 (152). — С. 123–126.

8. **Никифоров, И.В.** Генерация формальной модели системы по требованиям, заданным в нотации USE CASE MAP [Текст] / И.В. Никифоров, А.В. Петров, Ю.В. Юсупов // Научно-технические ведомости СПбГПУ. Информатика. Телекоммуникации. Управление. — 2010. — № 4 (103). — С. 191–195.

9. **Ануреев, И.С.** Средства поддержки интегрированной технологии для анализа и верификации спецификаций телекоммуникационных приложений [Текст] / И.С. Ануреев, С.Н. Баранов, Д.М. Белоглазов, Е.В. Бодин, П.Д. Дробинцев, А.В. Колчин, В.П. Котляров, А.А. Летичевский, А.А. Летичевский мл., В.А. Непомнящий, И.В. Никифоров [и др.] // Тр. СПИИРАН. — 2013. — № 1. — 28 с.

10. **Никифоров, И.В.** Статический метод отладки тестовых сценариев, сгенерированных с помощью эвристик [Текст] / И.В. Никифоров, А.В. Петров, В.П. Котляров // Научно-технические ведомости СПбГПУ. Информатика. Телекоммуникации. Управление. — СПб.: Изд-во Политехнического ун-та, 2012. — № 4 (152). — С. 114–119.

### REFERENCES

1. **Booch Gr., Maksimchuk R., Engel M., Young B., Conallen J., Houston K.** Object-Oriented Analysis and Design with Applications; 3rd ed. — Addison-Wesley Professional, 2007. — 720 p.

2. **Veselov A.O., Kotliarov V.P.** Avtomatizatsiia testirovaniia v oblasti telekommunikatsii [*Testing automation of projects in telecommunication domain*] / Nauchno-tekhicheskie vedomosti SPbGPU. Informatika. Telekommunikatsii. Upravlenie [*St. Petersburg State Polytechnical University Journal. Computer Science. Telecommunications and Control Systems*]. — St. Petersburg: Izd-vo Politekhnikeskogo un-ta, 2010. — № 4 (103). — S. 180–185. (rus)

3. Recommendation ITU-T Z.151. User requirements notation (URN), 11/2008.

4. **Letichevskii A.A., Kapitonova Iu.V., Letichevskii A.A. (ml.) i dr.** Spetsifikatsiia sistem s pomoshch'iu bazovykh protokolov / Kibernetika i sistemnyi analiz. — 2005. — № 4. — S. 3–21. (rus)

5. **Buhr R.J.A., Casselman R.S.** Use Case Maps for Object-Oriented Systems. — Prentice Hall, 1995.

6. **Gergel' V.P.** Vysokoproduktivnye vychisleniia dlia mnogoprotsessornykh mnogojad-

ernykh sistem. — Nizhnii Novgorod: Izd-vo Nizhegorodskogo universiteta, 2010. — 544 s. (rus)

7. **Drobintsev P.D., Kotliarov V.P., Chernorutskii I.G.** Avtomatizatsiia testirovaniia na osnove pokrytiia pol'zovatel'skikh stsensariiev [*Approach for testing automation based on user scenarios*] / Nauchno-tekhicheskie vedomosti SPbGPU. Informatika. Telekommunikatsii. Upravlenie [*St. Petersburg State Polytechnical University Journal. Computer Science. Telecommunications and Control Systems*]. — St. Petersburg: Izd-vo Politekhnikeskogo un-ta, 2012. — № 4 (152). — S. 123–126. (rus)

8. **Nikiforov I.V., Petrov A.V., Iusupov Iu.V.** Generatsiia formal'noi modeli sistemy po trebovaniiam, zadannym v notatsii USE CASE MAP [*Generation of formal model of a system from requirements specified in use case map*] / Nauchno-tekhicheskie vedomosti SPbGPU. Informatika. Telekommunikatsii. Upravlenie [*St. Petersburg State Polytechnical University Journal. Computer Science. Telecommunications and Control Systems*]. — St. Petersburg: Izd-vo Politekhnikeskogo un-ta, 2010. — № 4 (103). — S. 191–195. (rus)

9. **Anureev I.S., Baranov S.N., Beloglazov**



D.M., Bodin E.V., Drobintsev P.D., Kolchin A.V., Kotliarov V.P., Letichevskii A.A., Letichevskii A.A. ml., Nepomniashchii V.A., Nikiforov I.V., Potienko S.V., Priima L.V., Tiutin B.V. Sredstva podderzhki integrirovannoi tekhnologii dlia analiza i verifikatsii spetsifikatsii telekommunikatsionnykh prilozhenii / Trudy SPIIRAN. – 2013. – № 1. – 28 s. (rus)

10. Nikiforov I.V., Petrov A.V., Kotliarov V.P. Statische metod otladki testovykh stsensariev,

s generirovannykh s pomoshch'iu evristik [*Static approach for debugging of test scenarios generated with usage of heuristics*] / Nauchno-tekhnicheskie vedomosti SPbGPU. Informatika. Telekommunikatsii. Upravlenie [*St. Petersburg State Polytechnical University Journal. Computer Science. Telecommunications and Control Systems*]. – St. Petersburg: Izdvo Politekhnikeskogo un-ta, 2012. – № 4 (152). – S. 114–119. (rus)

---

**НИКИФОРОВ Игорь Валерьевич** – аспирант кафедры информационных и управляющих систем Санкт-Петербургского государственного политехнического университета.

195251, Россия, Санкт-Петербург, ул. Политехническая, д. 29.

**NIKIFOROV, Igor V.** *St. Petersburg State Polytechnical University.*

195251, Politekhnikeskaya Str. 29, St. Petersburg, Russia.

**КОТЛЯРОВ Всеволод Павлович** – профессор кафедры информационных и управляющих систем Санкт-Петербургского государственного политехнического университета, кандидат технических наук.

195251, Россия, Санкт-Петербург, ул. Политехническая, д. 29.

E-mail: vpk@spbstu.ru

**KOTLYAROV, Vsevolod P.** *St. Petersburg State Polytechnical University.*

195251, Politekhnikeskaya Str. 29, St. Petersburg, Russia.

E-mail: vpk@spbstu.ru

**ДРОБИНЦЕВ Павел Дмитриевич** – доцент кафедры информационных и управляющих систем Санкт-Петербургского государственного политехнического университета, кандидат технических наук.

195251, Россия, Санкт-Петербург, ул. Политехническая, д. 29.

**DROBINTSEV, Pavel D.** *St. Petersburg State Polytechnical University.*

195251, Politekhnikeskaya Str. 29, St. Petersburg, Russia.

**НАУЧНОЕ ИЗДАНИЕ**  
**«НАУЧНО-ТЕХНИЧЕСКИЕ ВЕДОМОСТИ**  
**САНКТ-ПЕТЕРБУРГСКОГО**  
**ГОСУДАРСТВЕННОГО ПОЛИТЕХНИЧЕСКОГО УНИВЕРСИТЕТА.**  
**ИНФОРМАТИКА. ТЕЛЕКОММУНИКАЦИИ. УПРАВЛЕНИЕ»**  
**«ST. PETERSBURG STATE POLYTECHNICAL UNIVERSITY JOURNAL.**  
**COMPUTER SCIENCE. TELECOMMUNICATIONS AND CONTROL SYSTEMS»**

**№ 3 (174) 2013**

Учредитель – Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования «Санкт-Петербургский государственный политехнический университет»

Журнал зарегистрирован Федеральной службой по надзору в сфере информационных технологий и массовых коммуникаций (Роскомнадзор).  
Свидетельство о регистрации ПИ № ФС77-51457 от 19.10.2012 г.

Редакция журнала

канд. техн. наук, д-р экон. наук, профессор *А.В. Бабкин* – научный редактор  
*Е.А. Калинина* – литературный редактор, корректор  
*Г.А. Пышкина* – ответственный секретарь, выпускающий редактор

Телефон редакции (812)552-62-16, 297-18-21

E-mail: [infocom@spbstu.ru](mailto:infocom@spbstu.ru)

Компьютерная верстка *А.Н. Смирнов*

Директор Издательства Политехнического университета *А.В. Иванов*

Лицензия ЛР № 020593 от 07.08.97

---

Подписано в печать 26.06.2013. Формат 60×84 1/8. Бум. тип. № 1.  
Печать офсетная. Усл. печ. л. 18,13. Уч.-изд. л. 18,13. Тираж 1000. Заказ

---

Санкт-Петербургский государственный политехнический университет  
Издательство Политехнического университета  
член Издательско-полиграфической ассоциации университетов России  
Адрес университета и издательства: 195251, Санкт-Петербург, ул. Политехническая, д. 29.

## УСЛОВИЯ ПУБЛИКАЦИИ СТАТЕЙ

в журнале «Научно-технические ведомости Санкт-Петербургского государственного политехнического университета. Информатика. Телекоммуникации. Управление»

### 1. ОБЩИЕ ПОЛОЖЕНИЯ

Журнал «Научно-технические ведомости Санкт-Петербургского государственного политехнического университета. Информатика. Телекоммуникации. Управление» является периодическим печатным научным рецензируемым изданием. Зарегистрировано Федеральной службой по надзору в сфере информационных технологий и массовых коммуникаций (Роскомнадзор). Свидетельство о регистрации ПИ № ФС77-51457 от 19 октября 2012 г. С 2008 года выпускался в составе сериального периодического издания «Научно-технические ведомости СПбГПУ» (ISSN 1994-2354).

Издание с 2002 года входит в Перечень ведущих научных рецензируемых журналов и изданий (перечень ВАК) и принимает для печати материалы научных исследований, а также статьи для опубликования основных результатов диссертаций на соискание ученой степени доктора наук и кандидата наук по следующим основным научным направлениям: ИНФОРМАТИКА, ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА, РАДИОТЕХНИКА И СВЯЗЬ, ЭЛЕКТРОНИКА, ИЗМЕРИТЕЛЬНАЯ ТЕХНИКА, УПРАВЛЕНИЕ В СОЦИАЛЬНЫХ И ЭКОНОМИЧЕСКИХ СИСТЕМАХ. Научные направления журнала учитываются ВАК Минобрнауки РФ при защите докторских и кандидатских диссертаций в соответствии с Номенклатурой специальностей научных работников.

Сведения о публикации представлены в Реферативном журнале ВИНТИ РАН, в международной справочной системе «Ulrich's Periodical Directory».

Периодичность выхода журнала – 6 номеров в год.

### 2. ТРЕБОВАНИЯ К ПРЕДОСТАВЛЯЕМЫМ МАТЕРИАЛАМ

#### 2.1. Оформление материалов

1. Объем статей докторов наук, профессоров, докторантов, соискателей ученой степени доктора наук, как правило, 12–20 страниц формата А-4. Количество рисунков не должно превышать 5, таблиц – 4, литературных источников – 15.

2. Объем статей преподавателей, сотрудников, аспирантов, соискателей ученой степени кандидата наук, как правило, 8–15 страниц формата А-4, объем статей аспирантов – 8 страниц формата А-4. Количество рисунков не должно превышать 4, таблиц – 3, литературных источников – 10.

3. Авторы должны придерживаться следующей обобщенной структуры статьи: вводная часть (0,5–1 стр., актуальность, существующие проблемы); основная часть (постановка и описание задачи, изложение и суть основных результатов); заключительная часть (0,5–1 стр., предложения, выводы), список литературы (оформление по ГОСТ 7.1-2003).

4. Число авторов статьи не должно превышать трех человек.

5. Набор текста осуществляется в редакторе **MS Word**, формул – в редакторе **MS Equation 3.0** или **MythType**. Таблицы набираются в том же формате, что и основной текст.

6. Шрифт – TNR, размер шрифта основного текста – 14, интервал – 1,5; таблицы большого размера могут быть набраны 12 кеглем. Параметры страницы: поля слева – 3 см, сверху, снизу – 2,5 см, справа – 2 см, текст размещается без переносов. Абзацный отступ – 1 см.

## 2.2. Предоставление материалов

Вместе с материалами статьи должны быть обязательно предоставлены:

- номер УДК в соответствии с классификатором (в заголовке статьи);
- аннотация (2–3 предложения) на русском и английском языках;
- ключевые слова (5–7) на русском и английском языках;
- сведения об авторах на русском и английском языках: ФИО, место работы, должность, ученое звание, ученая степень, контактные телефоны, e-mail;
- аспиранты представляют документ отдела аспирантуры, заверенный печатью;
- рецензия на имя зам. главного редактора, подписанная специалистом, имеющим ученую степень доктора наук и/или ученое звание профессора. Рецензия должна быть **ОБЯЗАТЕЛЬНО** заверена в отделе кадров. Рецензент несет ответственность за содержание статьи, достоверность представленных материалов.

При необходимости редколлегия может потребовать представления акта экспертизы, о чем она сообщает автору (авторам).

С авторами статей заключается издательский лицензионный договор.

Предоставление всех материалов осуществляется в электронном виде через личный кабинет **ЭЛЕКТРОННОЙ РЕДАКЦИИ** по адресу <http://journals.spbstu.ru>

## 2.3. Рассмотрение материалов

Предоставленные материалы (п. 2.2) первоначально рассматриваются редакционной коллегией и передаются для рецензирования. После одобрения материалов, согласования различных вопросов с автором (при необходимости) редакционная коллегия сообщает автору решение об опубликовании статьи. В случае отказа в публикации статьи редакция направляет автору мотивированный отказ.

При отклонении материалов из-за нарушения сроков подачи, требований по оформлению или как не отвечающих тематике журнала материалы не публикуются и не возвращаются.

Редакционная коллегия не вступает в дискуссию с авторами отклоненных материалов.

Публикация материалов аспирантов очной бюджетной формы обучения осуществляется бесплатно в соответствии с очередностью.

При поступлении в редакцию значительного количества статей их прием в очередной номер может закончиться **ДОСРОЧНО**.

**Более подробную информацию можно получить по телефону/факсу редакции:**

**8(812) 552-62-16 с 10<sup>00</sup> до 18<sup>00</sup> Галина Александровна**

**или по e-mail: [infocom@spbstu.ru](mailto:infocom@spbstu.ru)**