



НАУЧНО-ТЕХНИЧЕСКИЕ ВЕДОМОСТИ

САНКТ-ПЕТЕРБУРГСКОГО ГОСУДАРСТВЕННОГО
ПОЛИТЕХНИЧЕСКОГО УНИВЕРСИТЕТА

Информатика. Телекоммуникации.
Управление

1(236) 2016

НАУЧНО-ТЕХНИЧЕСКИЕ ВЕДОМОСТИ САНКТ-ПЕТЕРБУРГСКОГО
ГОСУДАРСТВЕННОГО ПОЛИТЕХНИЧЕСКОГО УНИВЕРСИТЕТА
ИНФОРМАТИКА. ТЕЛЕКОММУНИКАЦИИ. УПРАВЛЕНИЕ

РЕДАКЦИОННЫЙ СОВЕТ ЖУРНАЛА

Председатель

Юсупов Р.М., чл.-кор. РАН;

Редакционный совет:

Абрамов С.М., чл.-кор. РАН;

Арсеньев Д.Г., д-р техн. наук, профессор;

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

Главный редактор

Коротков А.С., д-р техн. наук, профессор, Санкт-Петербургский политехнический университет Петра Великого, Россия;

Редакционная коллегия:

Бабкин А.В., д-р экон. наук, профессор, Санкт-Петербургский политехнический университет Петра Великого, Россия;

Ицыксон В.М., канд. техн. наук, доцент, Санкт-Петербургский политехнический университет Петра Великого, Россия;

Prof. Dr. Philippe Ferrari, Head of the RF and Millimeter-Wave Lab IMEP-LAHC Microelectronics, Electromagnetism and Photonic Institute, Grenoble Alpes University, France;

Клавдиев В.Е., канд. техн. наук, доцент, Санкт-Петербургский политехнический университет Петра Великого, Россия;

Prof. Dr. Wolfgang Krautschneider, Head of Nanoelectronics Institute, Hamburg University of Technology, Germany;

Кучерявый Е.А., канд. техн. наук, профессор, Tampere University of Technology, Finland.

Dr. Fa-Long Luo, Chief Scientist, Element CXI, San Jose, USA;

Макаров С.Б., д-р техн. наук, профессор, Санкт-Петербургский политехнический университет Петра Великого, Россия;

Prof. Dr. Emil Novakov, IMEP-LAHC Microelectronics, Electromagnetism and Photonic Institute, Grenoble, France;

Трифонов П.В., канд. техн. наук, доцент, Санкт-Петербургский политехнический университет Петра Великого, Россия;

Устинов С.М., д-р техн. наук, профессор, Санкт-Петербургский политехнический университет Петра Великого, Россия;

Цикин И.А., д-р техн. наук, профессор, Санкт-Петербургский политехнический университет Петра Великого, Россия;

Шкодырев В.П., д-р техн. наук, профессор, Санкт-Петербургский политехнический университет Петра Великого, Россия.

Журнал с 1995 года издается под научно-методическим руководством Российской академии наук. С 2008 года выпускается в составе серийного 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 ВИНТИ РАН, в международной справочной системе «Ulrich's Periodical Directory», в базах данных Российский индекс научного цитирования (РИНЦ), Google Scholar, EBSCO, Math-Net.Ru, ProQuest, Index Copernicus

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.

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.

Подписной индекс **47517** в объединенном каталоге «Пресса России».

© Санкт-Петербургский политехнический университет Петра Великого, 2016

THE MINISTRY OF EDUCATION AND SCIENCE OF THE RUSSIAN FEDERATION



ST. PETERSBURG STATE POLYTECHNICAL UNIVERSITY JOURNAL

Computer Science.
Telecommunications and Control Systems

1(236) 2016

Polytechnical University Publishing House
Saint Petersburg
2016

ST. PETERSBURG STATE POLYTECHNICAL UNIVERSITY JOURNAL COMPUTER SCIENCE. TELECOMMUNICATIONS AND CONTROL SYSTEMS

EDITORIAL COUNCIL

Head of the editorial council

Prof. Dr. *Rafael M. Yusupov* (corresponding member of the Russian Academy of Sciences)

Members:

Prof. Dr. *Sergey M. Abramov* (corresponding member of the Russian Academy of Sciences),

Prof. Dr. *Dmitry G. Arseniev*,

Prof. Dr. *Vladimir V. Voevodin* (corresponding member of the Russian Academy of Sciences),

Prof. Dr. *Vladimir S. Zaborovsky*,

Prof. Dr. *Vladimir N. Kozlov*,

Prof. Dr. *Alexandr E. Fotiadi*,

Prof. Dr. *Igor G. Chernorutsky*.

EDITORIAL BOARD

Editor-in-chief

Prof. Dr. *Alexander S. Korotkov*, Peter the Great St. Petersburg Polytechnic University, Russia;

Members:

Prof. Dr. *Alexandr V. Babkin*, Peter the Great St. Petersburg Polytechnic University, Russia;

Assoc. Prof. Dr. *Vladimir M. Itsykson*, Peter the Great St. Petersburg Polytechnic University, Russia;

Prof. Dr. *Philippe Ferrari*, Head of the RF and Millimeter-Wave Lab IMEP-LAHC Microelectronics, Electromagnetism and Photonic Institute, Grenoble Alpes University, France;

Assoc. Prof. Dr. *Vladimir E. Klavdiev*, Peter the Great St. Petersburg Polytechnic University, Russia;

Prof. Dr. *Yevgeni Koucheryavy*, Tampere University of Technology, Finland.

Prof. Dr. *Wolfgang Krautschneider*, Head of Nanoelectronics Institute, Hamburg University of Technology, Germany;

Dr. *Fa-Long Luo*, Chief Scientist, Element CXI, San Jose, USA;

Prof. Dr. *Sergey B. Makarov*, Peter the Great St. Petersburg Polytechnic University, Russia;

Prof. Dr. *Emil Novakov*, IMEP-LAHC Microelectronics, Electromagnetism and Photonic Institute, Grenoble, France;

Prof. Dr. *Viacheslav P. Shkodyrev*, Peter the Great St. Petersburg Polytechnic University, Russia;

Assoc. Prof. Dr. *Peter V. Trifonov*, Peter the Great St. Petersburg Polytechnic University, Russia;

Prof. Dr. *Igor A. Tsikin*, Professor, Peter the Great St. Petersburg Polytechnic University, Russia;

Prof. Dr. *Sergey M. Ustinov*, Peter the Great St. Petersburg Polytechnic University, Russia.

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 journal is indexed by Ulrich's Periodicals Directory, Google Scholar, EBSCO, ProQuest, Index Copernicus, VINITI RAS Abstract Journal (Referativnyi Zhurnal), VINITI RAS Scientific and Technical Literature Collection, Russian Science Citation Index (RSCI) database © Scientific Electronic Library and Math-Net.ru databases.

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.

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

Телекоммуникационные системы и компьютерные сети

Карганов В.В., Расчесова А.Г., Кудряшов В.А. Показатель оценки эффективности систем связи и их элементов	7
---	---

Устройства и системы передачи, приема и обработки сигналов

Малышев В.М., Никитин А.Б. Моделирование автогенераторов в системах автоматизированного проектирования	15
Бородулин Р.Ю. Метод преобразования матриц для решения двумерных электродинамических задач излучения узловым методом конечных элементов	23

Моделирование вычислительных, телекоммуникационных, управляющих и социально-экономических систем

Раевская А.П., Крылатов А.Ю. Методы оценки матрицы корреспонденций в загруженных транспортных сетях	31
--	----

Программное обеспечение вычислительных, телекоммуникационных и управляющих систем

Титов А.И., Хомоненко А.Д. Выбор программного обеспечения с помощью алгоритма Такаги-Сугено на примере систем управления проектами	41
Левченко А.В., Фёдоров С.А. Инструмент динамического двоичного анализа пространственно-временной локализации приложения	53

Интеллектуальные системы и технологии

Селиверстов Я.А., Селиверстов С.А. Использование систем класса ГАТЛОСЭМИ для упреждения причин возникновения ДТП и неблагоприятных социальных исходов в «УМНОМ ГОРОДЕ»	65
---	----

Конференция «Инструменты и методы анализа программ 2015»

Мордвинов Д.А., Литвинов Ю.В. Обзор применения формальных методов в робототехнике	84
Ермаков М.К. Проведение динамического анализа исполняемого кода формата ARM ELF на основе статического бинарного инструментирования	108

Contents

Telecommunications Systems and Computer Networks

Karganov V.V., Raschesova A.G., Kudriashov V.A. <i>The Indicator for Assessing the Effectiveness of Communication Systems and Their Elements</i>	7
---	---

Circuits and Systems for Receiving, Transmitting and Signal Processing

Malyshev V.M., Nikitin A.B. <i>An Oscillator Designing Using CAD Simulation</i>	15
Borodulin R.Yu. <i>Matrix Transform Method for Solving Two-Dimensional Electrodynamics Problems of Radiation by Nodal Finite Element Method</i>	23

Simulations of Computer, Telecommunications, Control and Social Systems

Raevskaya A.P., Krylatov A.Yu. <i>OD-Matrix Estimation for Urban Traffic Area Control</i>	31
--	----

Software of Computer, Telecommunications and Control Systems

Titov A.I., Khomonenko A.D. <i>Software Selection by Using the Takagi-Sugeno Algorithm on the Example of Project Management Systems</i>	41
Levchenko A.V., Fyodorov S.A. <i>Dynamic Binary Instrumentation Tool for Data Locality Analysis</i>	53

Intellectual Systems and Technologies

Seliverstov Ya.A., Seliverstov S.A. <i>Use of GATLOSAMI to Prevent Causes of Traffic Accidents and Adverse Social Accidents in a 'SMART CITY'</i>	65
--	----

Conference «Tools & Methods of Program Analysis – 2015»

Mordvinov D.A., Litvinov Yu.V. <i>Survey on Formal Methods in Robotics</i>	84
Ermakov M.K. <i>Dynamic Analysis of ARM ELF Executable Code Using Static Binary Instrumentation</i>	108



Телекоммуникационные системы и компьютерные сети

DOI: 10.5862/JCSTCS.236.1

УДК 621.391.28

В.В. Карганов, А.Г. Расчесова, В.А. Кудряшов

ПОКАЗАТЕЛЬ ОЦЕНКИ ЭФФЕКТИВНОСТИ СИСТЕМ СВЯЗИ И ИХ ЭЛЕМЕНТОВ

V.V. Karganov, A.G. Raschesova, V.A. Kudriashov

THE INDICATOR FOR ASSESSING THE EFFECTIVENESS OF COMMUNICATION SYSTEMS AND THEIR ELEMENTS

Предложен показатель эффективности сетей связи и их элементов, обеспечивающий возможность их объективного сравнения на этапах создания, совершенствования и эксплуатации, а также установления обоснованных тарифов. Описанная совокупность значений показателей качества функционирования определяет состояния элементов сети и сети в целом и характеризует их нагрузочную способность.

Рассмотрено понятие стратификации сети связи как сложной системы, со спецификациями реализуемых протоколов передачи и обработки и с использованием известных аналитических моделей на базе систем массового обслуживания или имитационного моделирования. Изложен метод определения показателя эффективности (Method definition indicator efficiency) и оптимальной рабочей точки функционирования сети связи и ее элементов.

СЕТЬ СВЯЗИ; КЛАССИФИКАЦИЯ ПОКАЗАТЕЛЕЙ; МЕТРИЧЕСКОЕ ПРОСТРАНСТВО; ОБЪЕМ СЕТИ; МЕТОД ОПРЕДЕЛЕНИЯ ПОКАЗАТЕЛЯ ЭФФЕКТИВНОСТИ (METHOD DEFINITION INDICATOR EFFICIENCY (MDIEF); НАГРУЗОЧНАЯ ХАРАКТЕРИСТИКА; ОПТИМАЛЬНАЯ РАБОЧАЯ ТОЧКА.

The article offers an indicator of efficiency of communication networks and their elements, providing the possibility of their objective comparison at stages of creation, perfection and operation, and also an establishment of well-founded tariffs. The considered set of values of indicators of quality of functioning defines the conditions of elements of a network and a network as a whole and characterizes their loading ability.

The concept of stratification of a communication network, as a complex system, with the specifications of implemented protocols of transfer and processing and using the known analytical models based on systems of mass service or imitating modeling is considered. A method for defining an indicator of efficiency (MDIEF) and an optimum working point of functioning of a communication network and its elements are stated.

COMMUNICATION NETWORKS (CN); CLASSIFICATION INDICATORS (CI); METRIC SPACE (NS); VOLUME NETWORK (VN); METHOD DEFINITION INDICATOR EFFICIENCY (MDIEF); LOADING CHARACTERISTIC (LC); OPTIMUM WORKING POINT.

На этапах научных исследований, проектирования, построения и эксплуатации [1, 2] стоит задача анализа и сравнения вариантов создания и применения оборудова-

ния средств связи. Для сравнения вариантов и установления обоснованных тарифов за пользование услугами связи необходим эффективный показатель эффективности [3].

Постановка задачи. Сеть связи (СС) является сложной технической системой, и цель выполняемой ею операции состоит в передаче потоков сообщений пользователей требуемого объема и с заданным качеством. Оценка ее эффективности должна осуществляться по показателям, характеризующим качество ее функционирования, которые обуславливают ее пригодность для использования по назначению. Цель настоящей статьи заключается в разработке универсального показателя для оценки эффективности СС и их элементов на соотношении результата функционирования — полезного эффекта и затрат на реализацию.

Решение. Полезный эффект СС характеризует целенаправленный процесс ее функционирования и оценивается показателями качества функционирования. Процесс ее функционирования состоит в передаче определенного количества сообщений, каждое из которых передается с определенной скоростью и степенью точности [4, 5].

На основе классификации показателей сети [6, 7] определены:

класс **A** — класс показателей качества функционирования сети;

класс **B** — класс показателей стабильности качества функционирования;

класс **E** — класс показателей эффективности.

В отдельные классы выделены показатели, определяющие значения показателей качества функционирования СС, — класс **C**; показатели системы эксплуатации, определяющие стабильность показателей качества функционирования, — класс **D**; стоимостные показатели — класс **F**, влияющие на показатели эффективности.

В класс **A** входят производительность P , скорость V и точность передачи информации T , что отвечает положениям рекомендации МСЭ I.350. Показатели взаимосвязаны и значения одного или двух из них недостаточно для оценки сети. Вероятность своевременной доставки Q является интегральным показателем качества функционирования. Показатели качества функционирования являются компонентами вектора $\Phi = (P, V, T)$ в пространстве R^3 (рис. 1). По совокупностям значений этих компо-

нентов определяются состояния элементов сети и сети в целом.

Метрическое пространство. Совокупность значений показателей качества функционирования определяет состояния элементов сети и сети в целом и характеризует их нагрузочную способность (НС) H_c .

Для наибольшей НС СС H_c (имеет статус паспортной характеристики), определяемой наибольшими возможными значениями, имеем показатели: пропускная способность сети C ; максимальная скорость доставки $V_{\text{дост}}$ как обратная величина времени доставки; наибольшая точность передачи T_c — среднее количество сообщений, передаваемых в единицу времени точно (без искажений, потерь и засылок не по адресу).

Состояние НС получателя информации также характеризуется вектором $H_{\text{пол}}$ в пространстве R^3 . Пространство показателей качества функционирования сети Φ ограничивается НС H_c [8, 14].

Процесс функционирования Φ СС и любого из ее элементов может происходить только в рамках их НС, определяемых максимально возможными значениями показателей. НС СС H_c и ее элементов в процессе использования по назначению может быть вычислена аналитически, получена экспериментально путем измерений или сбора и обработки статистических данных.

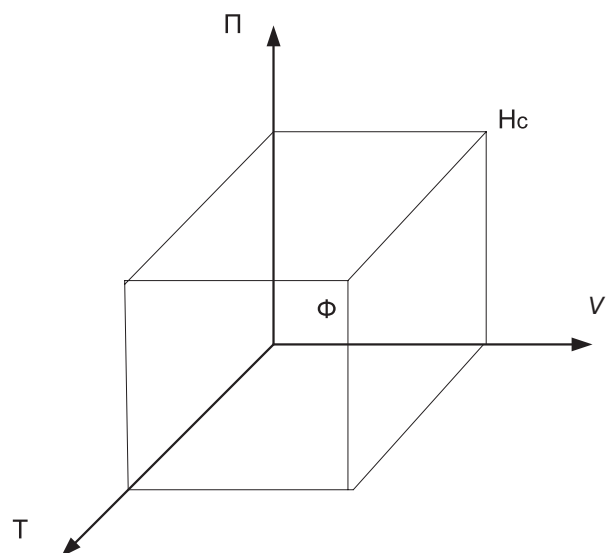


Рис. 1. Пространство Φ показателей качества функционирования



Для некоторой системы подмножеств пространства показателей качества функционирования Φ вводится отображение $W: H_{ci} \rightarrow R$, такое что $W_i = P_i \times V_i \times T_i$, где W_i является неотрицательной, счетно-аддитивной функцией, определяющей объем в (сообщ/с)³ и представляющей собой меру на пространстве показателей качества функционирования сети. Пространство Φ является метрическим: каждой паре $H_{ci}, H_{cj} \in \Phi$ соотнесено вещественное число $\rho(H_{ci}, H_{cj})$, представляющее собой расстояние между элементами пространства, удовлетворяющее условиям теории меры. Данные оценки особенно важны для конвергентных систем связи с целью организации их эффективного управления, установления объективных тарифов, взаиморасчетов между подсетями и т. д.

Метрика на Φ может быть введена не единственным образом. При данном определении она представляет собой трехмерное Евклидово пространство. Метрика W введена через показатели качества функционирования системы (взаимосвязанные с показателями, характеризующими различные свойства системы) и позволяет определить удельные затраты, приходящиеся на единицу объема:

$$y = \frac{C \text{ руб.}}{W} \Big/ \left(\frac{\text{сообщ}}{c} \right)^3.$$

Здесь C — затраты на организацию и эксплуатацию системы связи.

Показатели качества функционирования определяются в количестве сообщений в единицу времени, Y — удельная себестоимость, имеющая размерность руб./(сообщ/с)³ и названная *Ливн* в честь известного советского ученого В.Н. Листова, [9].

Определение показателя эффективности Ливн. Рассматриваем модель СС как «черный ящик»: СС — это S-система определенного поколения, целенаправленно созданная для выполнения глобальной операции A , имеющая единственную цель и некоторые затраты на ее реализацию, с потоками $P_{вх}$ на входе и $P_{вых}$ на выходе системы, с внутренними свойствами и возможными воздействиями внешней среды.

Эта глобальная операция A СС состоит из ряда составных операций подсистем СС таких, что каждая из них направлена на выполнение целевой функции глобальной операции. Глобальная операция A СС как сложной системы декомпозируется на отдельные составляющие операции, например, выделяемые плоскостями пользователя, сигнализации и управления как подсистемами, выполняющие свои целевые операции $A1, A2, A3$, необходимые для выполнения глобальной операции: $A \supset A1, A2, A3$.

При рассмотрении дерева декомпозиции «вглубь», каждая из плоскостей, как система, декомпозируется на подсистемы — уровни и т. д.

В СС передача сообщений пользователей производится по некоторому сквозному тракту. Оценка показателей качества обслуживания осуществляется в соответствии со стратификацией СС как сложной системы [10, 14], со спецификациями реализуемых протоколов передачи и обработки и с использованием известных аналитических моделей на базе систем массового обслуживания (СМО) или имитационного моделирования [11, 15].

В соответствии с рекомендацией [10] основными объектами сквозных трактов передачи являются маршрутизаторы и канальные соединения транспортной сети переноса данных.

Для аналитического описания СС строится концептуальная модель сети. Элементами модели являются участки абонентской $P_{са}$ и магистральной $P_{см}$ сетей, функционирование которых можно описать соответствующими СМО. Для сети передачи данных (ПД) коммутации пакетов это объекты транспортного и сетевого уровней в терминальном оборудовании, объекты сетевого уровня в узлах коммутации (УК) магистральной сети и объекты канального соединения между смежными элементами сети. Обычно принимается, что поступающие в эти объекты потоки пакетов или сообщений являются пуассоновскими, а законы распределения их времени обслуживания — детерминированными для пакетов и показательными для сообщений.

Для элемента сети, описываемого как СМО типа М/М/1 с неограниченной очередью и прямым порядком обслуживания, выражение для нагрузочной характеристики (НХ) имеет вид уравнения прямой в отрезках:

$$\frac{\lambda}{\mu} + \frac{1}{\mu T_v} = 1.$$

Нагрузочная характеристика СМО типа М/Д/1 определяется для каждого ρ соответствующими парами значений:

$$\lambda_{\text{обсл}} = \rho\mu \text{ и } T_v^{-1} = \frac{2 - \rho}{2\mu(1 - \rho)}.$$

Она является выпуклой вверх, а ее предельные точки совпадают с предельными точками НХ СМО типа М/М/1 [12, 13]. В случае когда требование, поступившее в занятую или свободную СМО при отказах обслуживающего прибора, ожидает его восстановления, сколько бы раз он не отказывал в процессе обслуживания, имеем следующее выражение для СМО типа М/М/1:

$$\frac{\lambda_{\text{обсл}}}{\mu K_r} + \frac{T_v^{-1}}{\mu K_r / (1 + \mu K_r K_n T_n)} = 1,$$

где K_r — коэффициент готовности обслуживающего прибора; K_n — коэффициент простоя; T_n — среднее время простоя.

В силу свойства прямолинейности НХ просто определяются требования к показателям надежности обслуживающего прибора. Время исправной работы T_n и время простоя T_n объектов предполагается экспоненциально распределенным.

Нагрузочная характеристика сети ПД определяется по НХ элементов аналитической модели:

$$P_c = \min(P_{CA \text{ вх}}, P_{CM}, P_{CA \text{ вых}});$$

$$T_{vC} = T_{vCA \text{ вх}} + T_{vCM} + T_{vCA \text{ вых}}.$$

Участок абонентской сети P_{CA} , входной и выходной, можно рассматривать как трехфазную сеть массового обслуживания, включающую СМО для описания свойств объектов транспортного и сетевого уровней в терминале, канального соединения и объекта сетевого уровня в УК, соединенных

последовательно. НХ P_{CA} определяется парами значений $\lambda_{\text{обсл}}$ и T_v^{-1} для каждого $\lambda_{\text{вх}} \in [0, \text{Sup}\lambda_{j,ck}]$.

При этом для отдельного участка P_{CA} $T_v^{-1} = \left(\sum_{i=1}^3 T_{vi} \right)^{-1}$, здесь i — объект P_{CA} . Средние значения $\lambda_{\text{обсл}}$ и T_v^{-1} определяют усреднением по всем входным и выходным участкам абонентской сети.

Магистральная сеть включает объекты сетевого уровня во входном, выходном и транзитных УК, а также объекты канальных соединений между ними, рассматриваемые как СМО типа М/М/С или М/Д/С (С — количество обслуживающих приборов). НХ магистральной сети определяется парами значений:

$\lambda_{\text{обсл CM}}$ для каждого $\lambda_{\text{вх CM}} \in [0, \text{Sup}\lambda_{\text{обсл CA вх}}]$;

T_{vCM}^{-1} , соответствующее каждому $\lambda_{\text{вх CM}}$.

Для дейтаграммной сети среднее время пребывания пакета в магистральной сети может быть определено с использованием критерия качества сети $T_{vкр}$:

$$T_{vCM} = \phi T_{vкр} + \sum_j \frac{\lambda_{\text{обсл } j}}{\lambda_{\text{обсл CM}}} T_{vУК j}.$$

Здесь ϕ — среднее число УК, проходимых пакетами в магистральной сети. Второе слагаемое представляет собой среднее время пребывания пакетов в выходном УК, где $T_{vкр}$ определяет усредненное в соответствии с частью обрабатываемого потока значение среднего времени пребывания пакета во входном УК и в магистральной сети до выходного УК.

Определение оптимальной рабочей точки функционирования СС и ее элементов. Система связи является одноцелевой, и ее операция состоит в передаче потока сообщений требуемого объема и с заданными показателями качества. Глобальная операция системы связи состоит из ряда составных операций ее подсистем, каждая из которых направлена на выполнение целевой функции глобальной операции. Эффективность реализации каждой из отдельных операций также измеряется соотношением результата ее функционирования — полезного эффекта и затрат на реализацию данной операции.

Наибольшему значению показателя объема системы связи W при одних и тех же затратах соответствует и наибольшее значение показателя эффективности Ливн. Максимальное значение $W_{\max}$ соответствует оптимальной рабочей точке функционирования системы связи и ее элементов (точке «золотого сечения»), в которой «сбалансированы» отдельные показатели качества функционирования. Поддержание процесса функционирования СС в оптимальной рабочей точке является одной из основных задач в процессе использования сети по назначению.

«Сбалансированность» показателей производительности системы связи и скорости передачи наглядно представима на НХ. Она определена [3, 9, 14] как зависимость между интенсивностью обслуженного потока сообщений и скоростью доставки T сообщений. НХ для элемента, некоторого участка

и СС в целом может быть вычислена аналитически, например, с использованием теории СМО, получена на имитационной модели и экспериментально путем измерений или сбора и обработки статистических данных.

В частном случае, в плоскости $\Pi \times V$, НХ, устанавливающая зависимость между показателями производительности — Π и скорости — V , обеспечивает наглядность определения оптимальной рабочей точки функционирования, характеризующейся максимальной эффективностью вследствие сбалансированности отдельных показателей. Предполагается, что требуемая точность передачи обеспечивается методами помехоустойчивого кодирования.

На рис. 2 представлено определение оптимальной рабочей точки по НХ для элемента системы связи, описываемого СМО типа М/М/1.

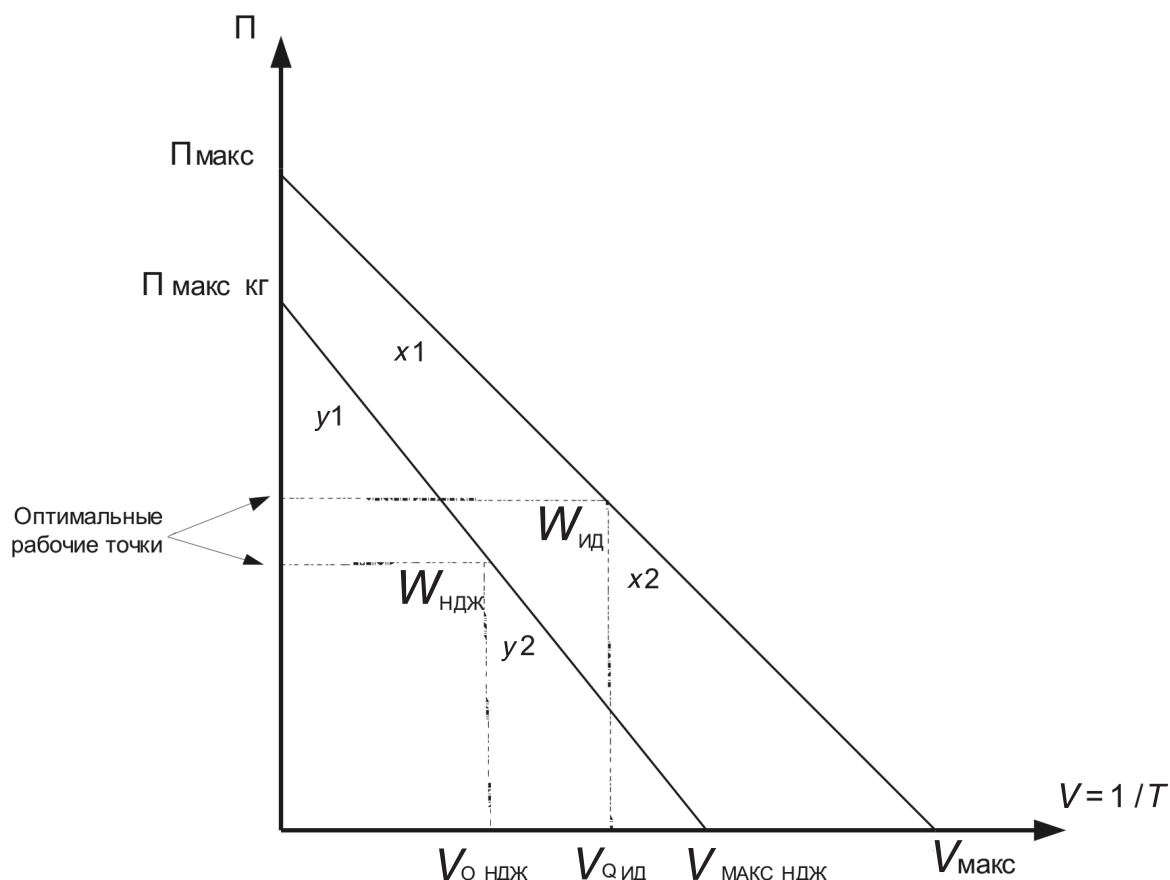


Рис. 2. Определение оптимальных рабочих точек по НХ

Приведены две характеристики с идеальным обслуживающим прибором и с ненадежным прибором, при стратегии обслуживания такой, что требование, поступившее при отказах обслуживающего прибора в занятую или свободную систему, ожидает его восстановления, сколько бы раз он не отказывал в процессе обслуживания.

Показаны оптимальные рабочие точки и выделены прямоугольники, площади которых равны максимальному объему идеальной СМО и с неидеальным обслуживающим прибором. Оптимальные рабочие точки просто определяются на графике НХ в силу равенства подобных треугольников x_1 и x_2 , а также y_1 и y_2 и соответствуют коэффициенту загрузки $\rho = 0,5$.

Для получения оптимального значения объема $W_{\max}$ средняя интенсивность входного потока λ^* должна выбираться соответствующей оптимальной рабочей точке Π^* с параметрами $\Pi^* = \Pi(\lambda^*) = \Pi_{\max} / 2$, а также $T(\Pi^*) = 2 / \Pi_{\max} = 2T(0)$. В оптимальной рабочей точке производительность и время передачи сбалансированы.

Предложен универсальный показатель для оценки эффективности СС и их эле-

ментов, основанный на соотношении полезного эффекта функционирования СС и затрат на нее. Универсальность обеспечивается построением метрического пространства показателей качества функционирования, вычислением обобщенного показателя — объема сети W и затем показателя Ливн, представляющего собой удельную себестоимость и позволяющего осуществлять объективную сравнительную оценку различных СС и их элементов. Предложена НХ, отличающаяся наглядностью оптимальной рабочей точки функционирования СС и ее элементов, при которой обеспечивается наивысшая эффективность сети.

Рассмотренный показатель эффективности может использоваться при создании новых и совершенствовании существующих СС, которые могут быть построены на любых сетевых технологиях: с коммутацией сообщений, пакетов и каналов, сетей интегрального обслуживания. Он может также применяться и для оценки отдельных элементов СС. Объективность сравнительной оценки сетей по показателю эффективности позволяет положить его в основу тарификации.

СПИСОК ЛИТЕРАТУРЫ

1. Воробьев С.П., Давыдов А.Е., Курносое В.И. Жизненный цикл инфокоммуникационных сетей: Монография. М.: Наука, 2011. 303 с.
2. Карпов М.А., Абоелазм М.М., Зимин В.Г., Егорова Е.В., Мамаева О.Ю., Цуников А.Ю. Многоканальные волоконно-оптические телекоммуникационные системы // Успехи современной радиоэлектроники. 2014. № 1. С. 55–59.
3. Пирмагомедов Р.Я. О повышении эффективности процесса обслуживания сетей связи // Современные научные исследования и инновации. 2013. № 10 [Электронный ресурс] / URL: <http://web.snauka.ru/issues/2013/10/28072> (Дата обращения: 04.03.2016).
4. Межуев А.М. Тензорные методы в теории оценки информационной эффективности и анализа элементов цифровых радиосетей: Монография. Тамбов: Интеграция, 2008. 262 с.
5. Воробьев С.П., Хейстонен П. Эффективность жизненного цикла инфокоммуникационных сетей // Инфокоммуникационные технологии. 2009. Вып. 1.
6. Шварцман В.О. Качество услуг сетей следующего поколения // Электросвязь. 2006. № 3. С. 6–10.
7. Барабаш П.А., Воробьев С.П., Курносое В.И., Советов Б.Я. Инфокоммуникационные технологии в Глобальной информационной инфраструктуре. СПб.: Наука, 2008. 552 с.
8. Кулешов И.А., Расчесова А.Г. Концепции оценки эффективности современных систем связи. Моделирование развития информационно-телекоммуникационных систем. СПб.: Изд-во «Синтез-Бук», 2009. С. 146–190.
9. Кудряшов В.А., Красовский А.Е. сост. Владимир Николаевич Листов. Ученый, педагог, человек: Сб. статей и воспоминаний. СПб.: Информ. центр «Выбор», 2002. 136 с.
10. ITU-T RECOMMENDATION I.380- internet protocol data communication service — IP packet transfer and availability performance parameters.
11. Клейнрок Л. On Flow Control in Computer Networks. Computer Science Department University of California, Los Angeles, California, 2002.
12. Лохмотко В.В. Модели и методы оптими-



зации структуры телекоммуникационных сетей: Дисс. ... д-ра техн. наук. СПб.: СПбГУТ, 2009.

13. **Моига Н.Н.** Модели, методы и алгоритмы анализа процессов функционирования инфокоммуникационных транспортных систем:

Дисс. ... д-ра техн. наук. СПб.: СПбГУТ, 2009.

14. **Giessler A., Koenig A., Haenle J., Pade E.** Deadlock-Free Packet Networks. 2007.

15. **Zeigler B.P.** Theory of modelling and simulation. Wiley, 2006. P. 211

REFERENCES

1. **Vorobyev S.P., Davydov A.Ye., Kurnosov V.I.** *Zhiznennyy tsikl infokommunikatsionnykh setey* [The life cycle of info-communication networks]. Moscow: Nauka Publ., 2011, 303 p. (rus)

2. **Karpov M.A., Aboyelazm M.M., Zimin V.G., Yegorova Ye.V., Mamayeva O.Yu., Tsunikov A.Yu.** *Mnogokanalnyye volokonno-opticheskiye telekommunikatsionnyye sistemy* [Multi-channel fiber-optic telecommunication systems]. *Uspekhi sovremennoy radioelektroniki* [Achievements of Modern Radioelectronics], 2014, No. 1, Pp. 55–59. (rus)

3. **Pirmagomedov R.Ya.** *O povyshenii effektivnosti protsessa obsluzhivaniya setey svyazi* [Increasing efficiency of communication networks service]. *Sovremennyye nauchnyye issledovaniya i innovatsii* [Modern scientific researches and innovations], 2013, No. 10. Available: <http://web.snauka.ru/issues/2013/10/28072> (Accessed: 04.03.2016). (rus)

4. **Mezhuyev A.M.** *Tenzornyye metody v teorii otsenki informatsionnoy effektivnosti i analiza elementov tsifrovyykh radiosetey* [Tensor methods in the theory of estimation of information efficiency and analysis of elements of digital radio]. Tambov: Integratsiya Publ., 2008, 262 p. (rus)

5. **Vorobyev S.P., Kheystonen P.** *Effektivnost zhiznennogo tsikla infokommunikatsionnykh setey* [The effectiveness of the infocommunication network lifecycle]. *Infokommunikatsionnyye tekhnologii* [Information and Communication Technologies], 2009, Vol. 1. (rus)

6. **Shvartsman V.O.** *Kachestvo uslug setey sleduyushchego pokoleniya* [The quality of next-generation network services]. *Elektrosvyaz* [Electric Communication], 2006, No. 3, Pp. 6–10. (rus)

7. **Barabash P.A., Vorobyev S.P., Kurnosov V.I., Sovetov B.Ya.** *Infokommunikatsionnyye tekhnologii v Globalnoy informatsionnoy infrastrukture* [Information and Communication Technologies in the Global Information Infrastructure]. St. Petersburg: Nauka

Publ., 2008, 552 p. (rus)

8. **Kuleshov I.A., Raschesova A.G.** *Kontseptsii otsenki effektivnosti sovremennykh sistem svyazi Modelirovaniye razvitiya informatsionno-telekommunikatsionnykh sistem* [Concept evaluation of the effectiveness of modern communication systems. Modelling of development of information and telecommunication systems]. St. Petersburg: Sintez-Buk Publ., 2009, Pp. 146–190. (rus)

9. **Kudryashov V.A., Kraskovskiy A.Ye. comp.** *Vladimir Nikolayevich Listov. Uchenyy, pedagog, chelovek. Sbornik statey i vospominaniy* [Vladimir Nikolayevich Listov. Scientist, teacher, man. Collection of articles and memoirs]. St. Petersburg: Informatsionnyy tsentr «Vybor» Publ., 2002, 136 p. (rus)

10. **ITU-T RECOMMENDATION I.380**-internet protocol data communication service – IP packet transfer and availability performance parameters.

11. **Kleinrock L.** *On Flow Control in Computer Networks*. Computer Science Department University of California, Los Angeles, California, 2002.

12. **Lokhmotko V.V.** *Modeli i metody optimizatsii struktury telekommunikatsionnykh setey: Dissertatsiya na soiskaniye uchenoy stepeni doktora tekhnicheskikh nauk* [Models and methods of optimization of the structure of telecommunication networks]. St. Petersburg: SPbGUT Publ., 2009. (rus)

13. **Moigak N.N.** *Modeli, metody i algoritmy analiza protsessov funktsionirovaniya infokommunikatsionnykh transportnykh sistem: Dissertatsiya na soiskaniye uchenoy stepeni doktora tekhnicheskikh nauk* [Models, methods and algorithms for the analysis of processes of functioning of info-communication transport systems]. St. Petersburg: SPbGUT Publ., 2009. (rus)

14. **Giessler A., Koenig A., Haenle J., Pade E.** *Deadlock-Free Packet Networks*, 2007.

15. **Zeigler B.P.** *Theory of modelling and simulation*. Wiley, 2006, P. 211.

КАРГАНОВ Виталий Вячеславович — старший научный сотрудник Военной академии связи имени С.М. Буденного, кандидат технических наук.

194064, Россия, Санкт-Петербург, Тихорецкий пр., д. 3.

E-mail: vitalik210277@mail.ru

KARGANOV Vitaly V. *Military Academy of communications named after S.M. Budyonny.*

194064, Tikhoretsky Ave. 3, St. Petersburg, Russia.

E-mail: vitalik210277@mail.ru

РАСЧЕСОВА Антонина Григорьевна — научный сотрудник Военной академии связи имени С.М. Буденного, кандидат технических наук.

194064, Россия, Санкт-Петербург, Тихорецкий пр., д. 3.

E-mail: tshlagra@mail.ru

RASCHESOVA Antonina G. *Military Academy of communications named after S.M. Budyonny.*

194064, Tikhoretsky Ave. 3, St. Petersburg, Russia.

E-mail: tshlagra@mail.ru

КУДРЯШОВ Владимир Александрович — профессор кафедры электрической связи Петербургского государственного университета путей сообщения Императора Александра I, кандидат технических наук.

190031, Россия, Санкт-Петербург, Московский пр., д. 9.

E-mail: Kudriashov37@mail.ru

KUDRIASHOV Vladimir A. *Emperor Alexander I St. Petersburg State Transport University.*

190031, Moskovsky Ave. 9, St. Petersburg, Russia.

E-mail: Kudriashov37@mail.ru

МОДЕЛИРОВАНИЕ АВТОГЕНЕРАТОРОВ В СИСТЕМАХ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ

V.M. Malyshev, A.B. Nikitin

AN OSCILLATOR DESIGNING USING CAD SIMULATION

Рассмотрены особенности моделирования твердотельных автогенераторов в системе автоматизированного проектирования СВЧ-устройств Microwave Office. Описаны основные причины, не позволяющие получить на практике достоверные характеристики при анализе стационарного режима автогенератора. На основе ряда примеров моделей генераторов даны практические рекомендации, позволяющие повысить достоверность результатов, получаемых при моделировании автогенераторов в системе Microwave Office.

АВТОГЕНЕРАТОР; ТРАНЗИСТОР; ОТРИЦАТЕЛЬНОЕ СОПРОТИВЛЕНИЕ; СВЧ.

The CAD simulation of solid-state oscillators is considered in this article. A negative resistance concept is frequently utilized to design series feedback microwave transistor oscillator. The article discusses using nonlinear design tools for negative resistance oscillator simulation. It describes the main reasons of simulation failure for microwave oscillator design: the position of the oscillator probe in the circuit, the values of the oscillator probe parameters. As illustrated by a number of examples of oscillator models, small-signal negative resistance simulation results can be used for correct placing of oscillator probe in the circuit. These results can be used for enhancing the validity of steady-state oscillation detection using CAD simulation of solid-state oscillators.

OSCILLATOR; TRANSISTOR; NEGATIVE RESISTANCE; MICROWAVES.

В диапазоне СВЧ в качестве твердотельных источников колебаний в настоящее время широко используются автогенераторы на транзисторах различных типов [1–4]. Разработка таких генераторов требует проведения моделирования с использованием сложных эквивалентных схем активных элементов, учета «паразитных» параметров других компонентов устройства. Данные факторы, а также специфика частотного диапазона предполагают использование при проектировании таких автогенераторов методов компьютерного моделирования.

Многие современные системы автоматизированного проектирования СВЧ-устройств имеют в своем составе средства, позволяющие проводить анализ режима автоколебаний [5–7]. Одна из наиболее широко используемых и популярных у разработчиков систем такого рода – пакет Microwave Office (MWO) от National Instruments [5]. В процессе разработки с помощью MWO появляется возможность провести анализ характеристик стационарного режима автоколебаний при их наличии или же скорректировать надлежащим образом схему генератора в случае диагностирова-

ния автоматизированной системой отсутствия условий существования генерации. Однако, как показывает практика работы в *Microwave Office*, при моделировании автогенераторов нередки ситуации, когда диагностирование системой отсутствия автоколебаний вызвано не действительной невозможностью выполнения исследуемой схемой функций автогенератора (например, из-за отсутствия решений уравнений стационарного режима, нарушения условий устойчивости и т. п.), а связано с нарушением условий сходимости используемого симулятором MWO алгоритма при наличии на самом деле в схеме условий существования генерации. Это обстоятельство требует дополнительных исследований и затрудняет процесс разработки твердотельных автогенераторов.

В данной статье рассматриваются особенности моделирования транзисторных СВЧ-генераторов в системе MWO, предлагаются подходы, позволяющие повысить надежность получаемых результатов.

Для нахождения характеристик стационарного режима автогенератора при его моделировании в *Microwave Office* используется специальный инструмент — так называемый *генераторный зонд* (Oscaprobe). Этот генераторный зонд представляет собой специальный двухполюсник, который необходимо поместить в схему проектируемого устройства. При этом, как следует из описания системы *Microwave Office*, чтобы диагностировать наличие автоколебаний в системе, рекомендуется подключать генераторный зонд к узлу ветви, соединяющей активный элемент и резонансную нагрузочную цепь автогенератора [5]. На рис. 1

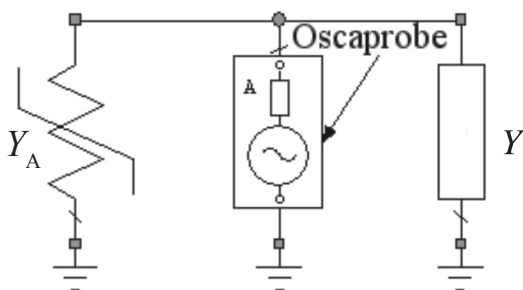


Рис. 1. Схема включения генераторного зонда

приведена упрощенная эквивалентная схема автогенератора на двухполюснике с отрицательным сопротивлением, иллюстрирующая работу генераторного зонда.

На схеме обозначены: Y_A — нелинейная отрицательная проводимость двухполюсного активного элемента (АЭ); Y — полная проводимость линейной части схемы автогенератора, включающей в себя колебательную систему и нагрузку; Oscaprobe — генераторный зонд, используемый в MWO для анализа стационарного режима автоколебаний.

Зонд Oscaprobe включается между АЭ и линейной частью схемы. Он позволяет осуществлять поиск результирующей частоты генерации, определяет выходную мощность, позволяет анализировать спектральные характеристики автогенератора. Этот инструмент включает в себя источник синусоидального напряжения. Причем внутреннее сопротивление $Z(\omega)$ этого источника таково, что оно выполняет короткое замыкание на частоте генерации ω_0 и размыкает цепь на всех других частотах [5]:

$$Z(\omega) = 0 \text{ при } \omega = \omega_0;$$

$$Z(\omega) = \infty \text{ при } \omega \neq \omega_0.$$

Такая комбинация источника вместе с его изменяемым внутренним сопротивлением, отмеченная на рис. 1, и представляет собой в первом приближении генераторный зонд. Если напряжение зонда точно равно комплексной амплитуде установившегося режима в узле подключения, то в этом случае через зонд на данной частоте ω_0 никакие токи протекать не будут. Следовательно, для нахождения характеристик стационарного режима автогенератора с помощью Oscaprobe нужно сделать следующее:

соединить генераторный зонд с подходящим узлом цепи;

найти амплитуду и частоту источника колебаний, для которых результирующий ток, протекающий через зонд, равен нулю [5]. В MWO специальный алгоритм варьирует параметры зонда с целью нахождения точного решения на основе заданного начального приближения.

На процесс нахождения характеристик стационарного режима автогенератора существенное влияние оказывает выбор точки подключения зонда, рациональное задание значений его параметров. Как уже было отмечено выше, в описании к автоматизированной системе MWO рекомендуется включать генераторный зонд между активным элементом и колебательной системой автогенератора [5]. Если для простых схем, выполненных на основе двухполюсных активных элементов с отрицательным сопротивлением (например, генераторных диодах), такая рекомендация однозначно определяет узел для включения измерительного зонда (рис. 1), то применительно к автогенераторам на транзисторах разделить в схеме активный элемент и колебательную систему часто оказывается не так просто.

В научно-технической литературе приводятся модели генераторов, в которых измерительный зонд подключается к различным выводам транзисторного активного элемента. Так, например, в [5] представлена схема автогенератора с зондом, подключаемым к затвору полевого транзистора, а в [8] — к базе биполярного транзистора. В то же время в работе [9] при рассмотрении примера проектирования в среде MWO перестраиваемого СВЧ-генератора на биполярном транзисторе генераторный зонд подключается к эмиттерному выводу активного элемента.

Отсутствие научного обоснования и строгих технических рекомендаций по способу включения измерительного зонда в схему транзисторного автогенератора затрудняет процедуру его анализа и разработки. Данное обстоятельство может усугубляться, например, в широкополосных электрически перестраиваемых генераторах наличием нескольких управляющих частотой элементов, а также усложнением эквивалентной схемы устройства из-за необходимости учета на СВЧ дополнительных реактивных элементов, возникающих в местах соединений.

Прежде чем приступить к исследованию особенностей использования измерительного зонда в системе MWO, рассмотрим основные принципы работы транзисторных СВЧ-автогенераторов на основе метода, базирующегося на анализе двухполюсников с

отрицательным сопротивлением. При таком подходе к одному из двух портов транзистора подключается цепь, обеспечивающая появление на другом (нагрузочном) порту отрицательного сопротивления на требуемой частоте [1–3]. В этом случае нагрузочный двухполюсник проектируется таким образом, чтобы были выполнены условия генерации. На рис. 2 приведены упрощенные эквивалентные схемы автогенераторов на двухполюсниках с отрицательной проводимостью (двухполюсник, управляемый напряжением) (рис. 2 а) и отрицательным сопротивлением (двухполюсник, управляемый током) (рис. 2 б).

На схемах использованы следующие обозначения:

$$Y_A(U_1, \omega) = G_A(U_1, \omega) + jB_A(U_1, \omega),$$

$$Z_A(I_1, \omega) = R_A(I_1, \omega) + jX_A(I_1, \omega),$$

$$Y(\omega) = G(\omega) + jB(\omega), \quad Z(\omega) = R(\omega) + jX(\omega),$$

где $Y_A(U_1, \omega)$ и $Z_A(I_1, \omega)$ — соответственно, полные проводимость и сопротивление активного двухполюсника, а $Y(\omega)$ и $Z(\omega)$ — проводимость и сопротивление линейной части схемы; U_1, I_1 — амплитуды напряжения и тока первой гармоники (рис. 2).

Условия существования стационарного режима в схемах на рис. 2 запишем в следующем виде [1, 3, 10]:

$$Y_A(U_1, \omega) + Y(\omega) = 0, \quad (1)$$

$$Z_A(I_1, \omega) + Z(\omega) = 0. \quad (2)$$

Наряду с условиями (1), (2) в автоколебательных системах должны быть выполнены условия самовозбуждения $G_{A0} + G < 0$ или $R_{A0} + R < 0$, где G_{A0} и R_{A0} — дифференциальные значения вещественной составляющей иммитанса активного двухполюсника в рабочей точке.

Решение уравнений (1), (2) позволяет найти амплитуду и частоту первой гармоники автоколебаний.

Как известно, для устойчивости найденных решений необходимо, чтобы на частоте автоколебаний были справедливы следующие неравенства:

$$\frac{\partial G_A}{\partial U_1} \cdot \frac{\partial(B_A + B)}{\partial \omega} - \frac{\partial(G_A + G)}{\partial \omega} \cdot \frac{\partial B_A}{\partial U_1} > 0, \quad (3)$$

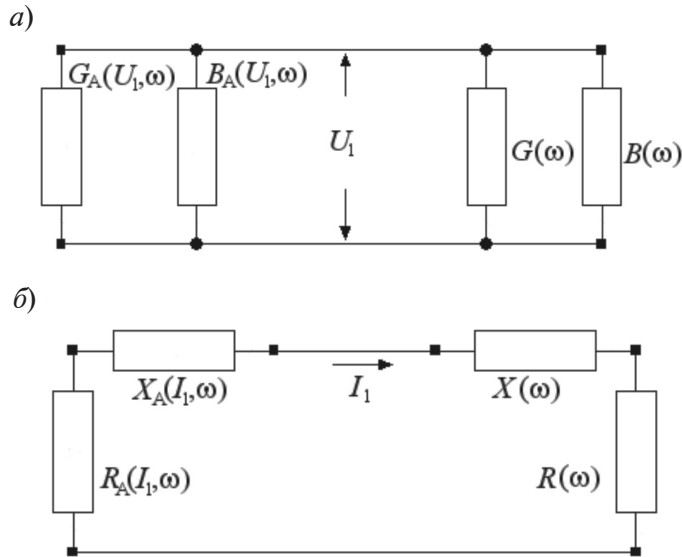


Рис. 2. Эквивалентные схемы автогенераторов на двухполюсниках с отрицательными проводимостью и сопротивлением

$$\frac{\partial R_A}{\partial I_1} \cdot \frac{\partial (X_A + X)}{\partial \omega} - \frac{\partial (R_A + R)}{\partial \omega} \cdot \frac{\partial X_A}{\partial I_1} > 0, \quad (4)$$

где производные рассчитываются для значений частоты и амплитуды, являющихся решениями систем (1) и (2) [1, 3, 10]. В случае резистивного нелинейного элемента ($B_A = 0, X_A = 0$) или слабой зависимости от частоты вещественных составляющих суммарного иммитанса $G(\omega) + G_A(\omega)$ и $R(\omega) + R_A(\omega)$ вторыми слагаемыми в левых частях неравенств (3), (4) можно пренебречь:

$$\begin{aligned} \frac{\partial G_A}{\partial U_1} \cdot \frac{\partial (B + B_A)}{\partial \omega} &> 0, \\ \frac{\partial R_A}{\partial I_1} \cdot \frac{\partial (X + X_A)}{\partial \omega} &> 0. \end{aligned} \quad (5)$$

Зависимости $G_A(U_1)$, $R_A(I_1)$ определяются механизмом ограничения амплитуды нелинейным АЭ. Для двухполюсников, управляемых напряжением (с динамической вольт-амперной характеристикой N -типа), должно выполняться неравенство $\frac{\partial G_A}{\partial U_1} > 0$, а для двухполюсников, управляемых током (с динамической вольт-амперной характеристикой S типа), — неравенство $\frac{\partial R_A}{\partial I_1} > 0$ [3, 10]. В этом случае из (5) следуют неравенства:

$$\frac{\partial (B + B_A)}{\partial \omega} > 0, \quad \frac{\partial (X + X_A)}{\partial \omega} > 0. \quad (6)$$

Как правило, частотной зависимостью реактивностей АЭ $B_A(\omega)$ и $X_A(\omega)$ можно пренебречь по сравнению с аналогичными зависимостями реактивных составляющих иммитанса колебательной системы генератора $B_A(\omega)$ и $X_A(\omega)$. Для резистивного нелинейного элемента $B_A(\omega)$ и $X_A(\omega)$ равны нулю. В результате неравенства (6) для АЭ с нелинейностями, соответственно, N и S типа упрощаются:

$$\frac{\partial B}{\partial \omega} > 0 \quad \text{и} \quad \frac{\partial X}{\partial \omega} > 0.$$

Это означает, что для автогенераторов на двухполюснике с отрицательной проводимостью (N типа) необходима колебательная система с резонансом типа параллельного контура (рис. 2 а), а на двухполюснике с отрицательным сопротивлением (S типа) — колебательная система с резонансом типа последовательного контура (рис. 2 б).

На рис. 3 представлена упрощенная эквивалентная схема транзисторного автогенератора с так называемый *последовательной обратной связью* [1–3].

На схеме Z_1, Z_2, Z_3 обозначают двухполюсные элементы, подключаемые, соответственно, к базовому, коллекторному и

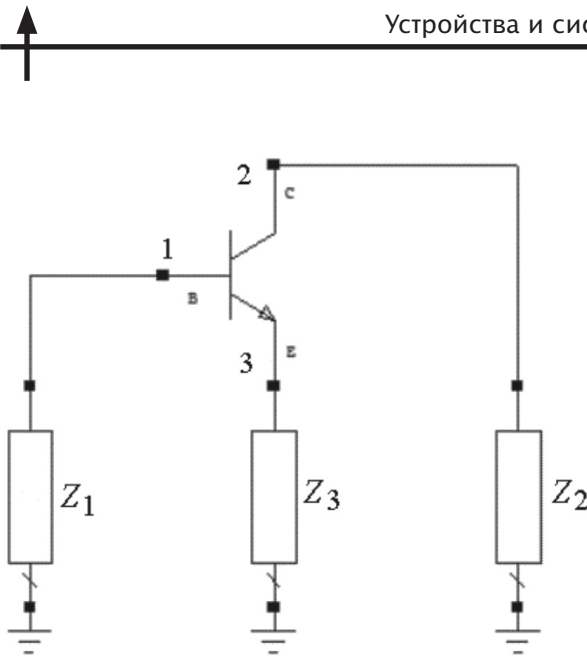


Рис. 3. Упрощенная эквивалентная схема транзисторного автогенератора

эмиттерному выводам транзистора. Эти двухполюсники включают в себя реактивности, необходимые для обеспечения условий генерации. Кроме того, один из них, например Z_2 , еще и сопротивление нагрузки.

При использовании концепции нелинейного двухполюсника с отрицательным сопротивлением, разработку транзисторного автогенератора часто осуществляют в два этапа: анализ в линейном и нелинейном режимах. На первом этапе исследуются импедансные характеристики активного элемента генератора (часть схемы, включающей в себя транзистор) в линейном режиме с целью создания схемы, обеспечивающей отрицательную вещественную составляющую входного иммитанса АЭ. На втором этапе (нелинейный анализ) для полученной схемы автогенератора исследуются характеристики его стационарного режима. На рис. 4 а приведены частотные зависимости вещественной $\text{Re } Z_C$ и мнимой составляющих $\text{Im } Z_C$ входного сопротивления АЭ автогенератора на сверхвысокочастотном гетеропереходном SiGe-транзисторе со стороны его коллекторного вывода в линейном режиме.

Как видно из представленных зависимостей, при соответствующем выборе резистивной нагрузки (двухполюсник Z_2 на рис. 3) можно ожидать установления режима автоколебаний на частоте порядка 5 ГГц,

поскольку на данной частоте $\text{Re } Z_C < 0$, $\text{Im } Z_C = 0$. При этом на частоте возможных автоколебаний ($f = 5$ ГГц) справедливо неравенство $\partial(\text{Im } Z_C) / \partial \omega > 0$, что позволяет ожидать выполнения условия устойчивости для генератора на двухполюснике с отрицательным сопротивлением (6). Однако, как показывают результаты моделирования такой схемы в нелинейном режиме, при подключении генераторного зонда к коллекторному выводу транзистора (узел 2 на схеме рис. 3) система Microwave Office не диагностирует существование автоколебательного режима ни при каких значениях параметров зонда.

Вместе с тем подключение измерительного зонда к эмиттерному выводу (узел 3 на схеме рис. 3) приводит к нахождению в процессе моделирования режима автоколебаний на частоте $f = 5,08$ ГГц с выходной мощностью 5,6 мВт. На рис. 4 б приведены частотные зависимости вещественной $\text{Re } Y_E$ и мнимой составляющих $\text{Im } Y_E$ входной проводимости АЭ со стороны эмиттерного вывода транзистора в линейном режиме. На частоте, близкой к 5 ГГц, справедливы условия $\text{Re } Y_E < 0$, $\text{Im } Y_E = 0$, причем $\partial(\text{Im } Y_E) / \partial \omega > 0$, что соответствует условиям генерации для АЭ с отрицательной проводимостью, управляемой напряжением ((1), (5), (6), рис. 2 а).

Это обстоятельство, а также другие многочисленные примеры моделирования автогенераторов различных типов указывают на то, что реализованный в системе Microwave Office алгоритм поиска решений уравнений стационарного режима предполагает подключение генераторного зонда к такому узлу АЭ, на входе которого в линейном режиме иммитансные характеристики соответствуют следующим условиям:

$$\text{Re } Y_A < 0, \text{Im } Y_A = 0, \partial(\text{Im } Y_A) / \partial \omega > 0, (7)$$

где Y_A — входная проводимость активного элемента на выбранном для линейного анализа порту, а все значения вычисляются на ожидаемой частоте генерации.

Как показывает практика моделирования одностранзисторных генераторов различного типа, для корректного диагностирования в исследуемой схеме автоколебательного

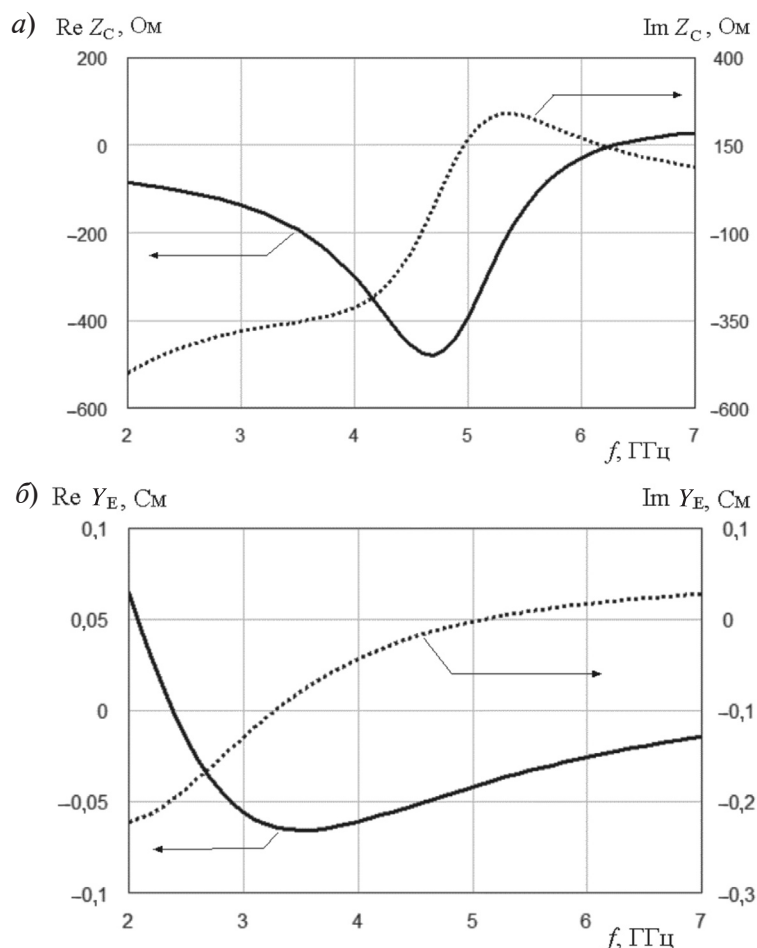


Рис. 4. Иммитансные характеристики активного элемента в линейном режиме

режима генераторный зонд следует подключать в большинстве случаев к эмиттерному выводу транзистора. Впрочем, в каждом конкретном случае этот вопрос решается опытным путем. Так, например, при разработке сверхширокополосных (октава и более) перестраиваемых варикапами генераторов сантиметрового диапазона было установлено, что в полосе перестройки может меняться характер нелинейности активного элемента. В качестве иллюстрации на рис. 5 приведены иммитансные характеристики активного элемента сверхширокополосного генератора, перестраиваемого в диапазоне 6...12 ГГц.

На графиках представлены частотные зависимости вещественной и мнимой составляющих проводимости Y_C (рис. 5 а) и сопротивления Z_C (рис. 5 б) со стороны коллекторного вывода, полученные в ре-

зультате линейного анализа для двух значений управляющего напряжения U на варикапе микрополоскового генератора, выполненного на основе SiGe-транзистора. Как следует из приведенных зависимостей, при $U = 12$ В характеристики активного элемента в области резонансной частоты f_2 соответствуют условиям (7). В результате для указанного выше значения управляющего напряжения подключение генераторного зонда к коллекторному узлу позволяет осуществлять нелинейный анализ в отличие от значения управляющего $U = 2$ В, для которого отсутствие необходимых условий в области резонансной частоты f_1 (рис. 5 а) требует для корректного нахождения характеристик режима автоколебаний использования других точек подключения измерительного зонда.

При моделировании стабилизированных

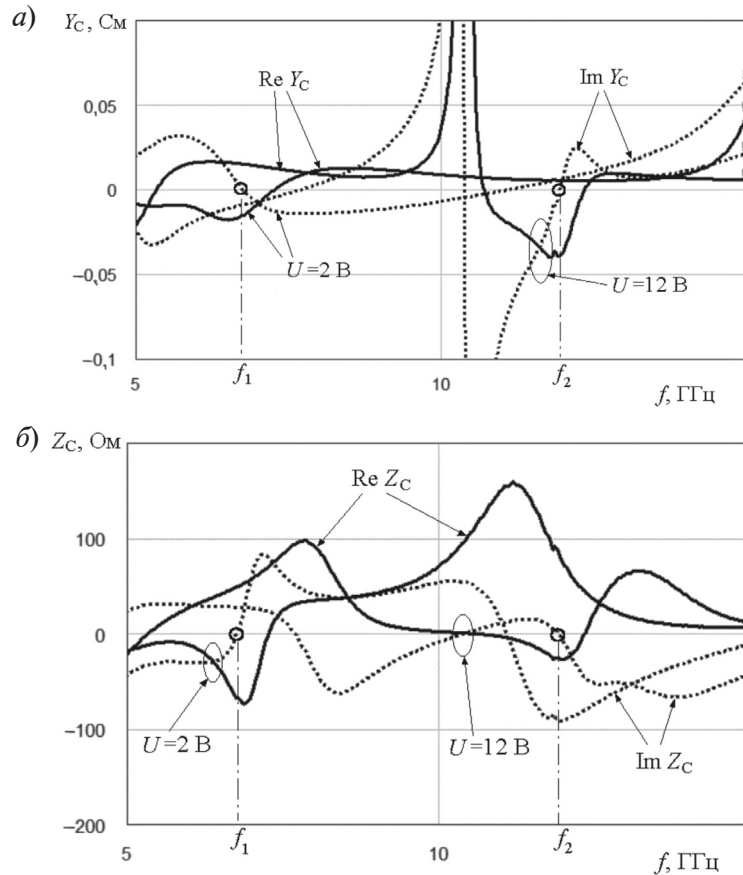


Рис. 5. Иммитансные характеристики активного элемента перестраиваемого генератора для двух значений управляющего напряжения на варикапе

транзисторных автогенераторов, построенных на высокочастотных диэлектрических резонаторах, нередко ситуации, когда автоматизированной системе не удастся диагностировать режим автоколебаний даже при правильном выборе узла подключения генераторного зонда. Для устранения этой проблемы при проектировании необходимо корректировать параметры зонда, определяющие частотный диапазон, в котором осуществляется поиск решения. С этой целью на основе результатов линейного анализа целесообразно определить ожидаемую частоту автоколебаний, на которой выполняются условия (7). А при проведении нелинейного анализа частотный диапазон генераторного зонда следует локализовать в районе ожидаемой частоты генерации с одновременным уменьшением шага поиска. Так, например, при проектировании генератора трехсантиметрового диапазона

волн, стабилизированного диэлектрическим резонатором добротностью 10 000 и более, для уверенного определения автоколебательного режима требовалось уменьшать полосу частот зонда до единиц процентов и менее.

Таким образом, проведенное исследование особенностей моделирования в системе автоматизированного проектирования СВЧ-устройств Microwave Office твердотельных автогенераторов позволило сформировать практические рекомендации, повышающие эффективность процесса их разработки. В частности показано, что при определении узла подключения генераторного зонда, с помощью которого осуществляется расчет характеристик стационарного режима генератора, целесообразно проводить анализ его АЭ в линейном режиме с целью проверки выполнения условий (7) на ожидаемой частоте

автоколебаний. На достоверность получаемых результатов также влияет рациональный выбор значений параметров зонда, определяющих частотный диапазон поиска решений.

Полученные результаты могут использоваться при разработке СВЧ-генераторов на основе моделирования в системах автоматизированного проектирования.

СПИСОК ЛИТЕРАТУРЫ

1. **Gonzalez G.** Foundations of Oscillator Circuit Design. Artech House, Inc., 2007. 422 p.
2. **Grebennikov A.** RF and Microwave Transmitter Design. John Wiley & Sons, Inc., 2011. 816 p.
3. **Grebennikov A.** RF and Microwave Transistor Oscillator Design. John Wiley & Sons Ltd, 2007. 441 p.
4. **Khanna A.P.S.** State of the Art in Microwave VCOs // Microwave Journal. 2015. No. 5. Pp. 23–40.
5. AWRDE Simulation and Analysis Guide [Электронный ресурс] // URL: <http://www.awrcorp.com/products/microwave-office> (Дата обращения: 12.12.2015).
6. Advanced Design System [Электронный ре-

сурс] // URL: <http://www.keysight.com> (Дата обращения: 12.12.2015).

7. ANSYS Designer. [Электронный ресурс] // URL: <http://www.ansys.com> (Дата обращения 12.12.2015).

8. **Maas S.** Nonlinear microwave and RF circuits. Artech House, Inc., 2003. 582 p.

9. **Gilmore R., Besser L.** Active circuits and systems // Practical RF circuit design for modern wireless systems. Vol. 2. Artech House, Inc., 2003. 569 p.

10. **Петров Б.Е., Романюк В.А.** Радиопередающие устройства на полупроводниковых приборах. М.: Высш. шк., 1989. 282 с.

REFERENCES

1. **Gonzalez G.** *Foundations of Oscillator Circuit Design*. Artech House, Inc., 2007, 422 p.
2. **Grebennikov A.** *RF and Microwave Transmitter Design*. John Wiley & Sons, Inc., 2011, 816 p.
3. **Grebennikov A.** *RF and Microwave Transistor Oscillator Design*. John Wiley & Sons Ltd, 2007, 441 p.
4. **Khanna A.P.S.** State of the Art in Microwave VCOs. *Microwave Journal*, 2015, No. 5, Pp. 23–40.
5. *AWRDE Simulation and Analysis Guide*. Available: <http://www.awrcorp.com/products/microwave-office> (Accessed: 12.12.2015).
6. *Advanced Design System*. Available: [http://](http://www.keysight.com)

www.keysight.com (Accessed: 12.12.2015).

7. *ANSYS Designer*. Available: <http://www.ansys.com> (Accessed: 12.12.2015).

8. **Maas S.** *Nonlinear microwave and RF circuits*. Artech House, Inc., 2003, 582 p.

9. **Gilmore R., Besser L.** Active circuits and systems. *Practical RF circuit design for modern wireless systems*. Vol. 2. Artech House, Inc., 2003, 569 p.

10. **Petrov B.Ye., Romanyuk V.A.** *Radio-peredayushchiye ustroystva na poluprovodnikovyykh priborakh* [Radio transmitters semiconductor devices]. Moscow: Vysshaya shkola Publ., 1989, 282 p. (rus)

МАЛЫШЕВ Виктор Михайлович — доцент кафедры радиотехнических и телекоммуникационных систем Института физики, нанотехнологий и телекоммуникаций Санкт-Петербургского политехнического университета Петра Великого, кандидат физико-математических наук.

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

E-mail: uhmal@mail.ru

MALYSHEV Victor M. *Peter the Great St. Petersburg Polytechnic University.*

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

E-mail: uhmal@mail.ru

НИКИТИН Александр Борисович — доцент кафедры радиотехнических и телекоммуникационных систем Института физики, нанотехнологий и телекоммуникаций Санкт-Петербургского политехнического университета Петра Великого, кандидат технических наук.

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

E-mail: nikitin@mail.spbstu.ru

NIKITIN Aleksandr B. *Peter the Great St. Petersburg Polytechnic University.*

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

E-mail: nikitin@mail.spbstu.ru



DOI: 10.5862/JCSTCS.236.3

УДК 519.612

*Р.Ю. Бородулин***МЕТОД ПРЕОБРАЗОВАНИЯ МАТРИЦ ДЛЯ РЕШЕНИЯ ДВУМЕРНЫХ ЭЛЕКТРОДИНАМИЧЕСКИХ ЗАДАЧ ИЗЛУЧЕНИЯ УЗЛОВЫМ МЕТОДОМ КОНЕЧНЫХ ЭЛЕМЕНТОВ***R.Yu. Borodulin***MATRIX TRANSFORM METHOD FOR SOLVING TWO-DIMENSIONAL ELECTRODYNAMIC PROBLEMS OF RADIATION BY NODAL FINITE ELEMENT METHOD**

Представлен математический аппарат решения задач излучения узловым методом конечных элементов в двумерной области. Получены итоговые выражения для элементов локальной матрицы и методика вывода функциональной зависимости возбуждения точечного источника (для формирования вектора правой части). Данным методом выполнен тестовый расчет излучения усовершенствованной модели элементарного электрического диполя в виде плоской рамки с током. Достоверность результатов подтверждена высоким совпадением с аналитическим решением задачи излучения элементарного электрического диполя. Представленные результаты могут использоваться при расчетах любых осесимметричных антенн, возбуждаемых точечным источником, в том числе в задачах конструктивного синтеза осесимметричных излучателей различного назначения.

АКСИАЛЬНО-СИММЕТРИЧНОЕ ПОЛЕ ИЗЛУЧЕНИЯ; ОСЕСИММЕТРИЧНАЯ АНТЕННА; ТОЧЕЧНЫЙ ИСТОЧНИК; ЭЛЕМЕНТАРНЫЙ ЭЛЕКТРИЧЕСКИЙ ДИПОЛЬ; ЛОКАЛЬНАЯ КОНЕЧНО-ЭЛЕМЕНТНАЯ МАТРИЦА; ФУНКЦИЯ ВОЗБУЖДЕНИЯ; КОНСТРУКЦИОННЫЙ СИНТЕЗ.

The article presents a mathematical apparatus for solving the radiation problems by the nodal finite elements method in a two-dimensional field. We obtain the final expression for the elements of the local matrix and the methodology for deriving the functional dependence of the excitation field of a point source (for forming the right side of the vector). The calculation test performed by this method improved the model of elementary electric dipole in the form of a flat frame with electric current. The reliability of the results confirmed a high coincidence with the analytical solution of the radiation problem of the elementary electric dipole. The results of the article can be used in the calculations of any axially symmetric antenna excited by a point source, including the problems of structural synthesis of axially symmetric emitters for different purposes.

AXIALLY SYMMETRICAL RADIATION FIELD; AXIALLY SYMMETRIC ANTENNA; POINT SOURCE; ELEMENTARY ELECTRIC DIPOLE; LOCAL FINITE ELEMENT MATRIX; EXCITATION FUNCTION; STRUCTURAL SYNTHESIS.

Осесимметричные (аксиально-симметричные) антенны составляют большой класс излучателей. Широкое применение таких антенн в технике радиосвязи выявило ряд проблем, касающихся их массогабаритных показателей, решаемых при помощи теории конструктивного синтеза (КС), требующей быстроты электродинамического анализа синтезируемых излучателей [1].

Оптимизационные методы, применяемые в КС, имеют большое число итераций, и чем меньше время будет потрачено на одну такую итерацию, тем быстрее будет найдена оптимальная конструкция антенны.

Любую аксиально-симметричную антенну можно представить в двумерной области в виде половины ее сечения. Это позволяет существенно экономить ресур-

сы ЭВМ, затрачиваемые на ее анализ, и вместо векторных уравнений использовать скалярные, имеющие меньшую размерность.

Существуют проблемы решения задач излучения узловым методом конечных элементов (МКЭ). Они заключаются в некорректности работы методик, разработанных для стационарных аксиально-симметричных полей применительно к уравнению Гельмгольца, описывающему распространяющуюся аксиально-симметричную волну, т. к. решение, получаемое МКЭ, не совпадает с решением, получаемым строгими методами для элементарного магнитного диполя (ЭМД) (рис. 1 а, б) [2–4].

Данный факт приводит к необходимости разработки метода повышения точности решения неоднородного уравнения Гельмгольца в двумерной области для аксиально-симметричных волн:

$$\frac{\partial}{\partial r} \left(\frac{1}{r\mu_r} \frac{\partial}{\partial r} (rE_\varphi) \right) + \frac{\partial}{\partial z} \left(\frac{1}{r\mu_r} \left(\frac{\partial(rE_\varphi)}{\partial z} \right) \right) - \frac{k_0^2 \varepsilon_{rk}}{r} (rE_\varphi) = -M^z, \quad (1)$$

где r – расстояние от оси симметрии до точки наблюдения; μ_r – относительная магнитная проницаемость среды распространения; E_φ – азимутальная компонента

электрического поля; ε_{rk} – относительная комплексная диэлектрическая проницаемость; M^z – источник, порождающий волну.

Вывод «матрицы жесткости» конечного элемента для решения осесимметричных задач

Аналитическое решение задачи излучения элементарного магнитного диполя (ЭМД) для азимутальной компоненты вектора электрического поля в ближней зоне выглядит как

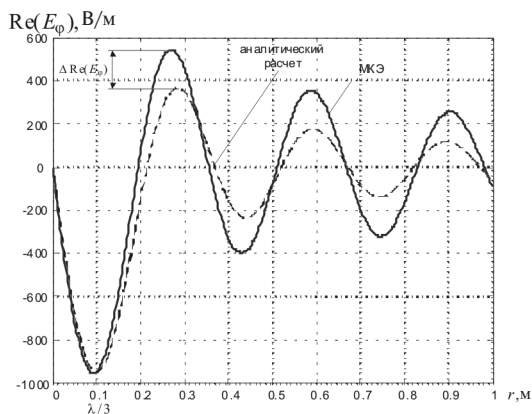
$$\dot{E}_m = \frac{\dot{m}_m \omega}{4\pi} \left(\frac{-i}{r^2} + \frac{k}{r} \right) e^{-ikr} \sin \theta, \quad (2)$$

где $\dot{m}_m$ – магнитный момент диполя; ω – круговая частота; θ – меридиональный угол [3].

Преобразуем «матрицу жесткости» конечного элемента (термин взят из теории упругости) к виду, соответствующему уравнению (1):

$$[K^e] = 2\pi \int_r \int_z r_c \left\{ \frac{\partial H_1}{\partial r} \quad \frac{\partial H_2}{\partial r} \quad \frac{\partial H_3}{\partial r} \right\} \left\{ \frac{\partial H_1}{\partial r} \quad \frac{\partial H_2}{\partial r} \quad \frac{\partial H_3}{\partial r} \right\}^T +$$

а)



б)

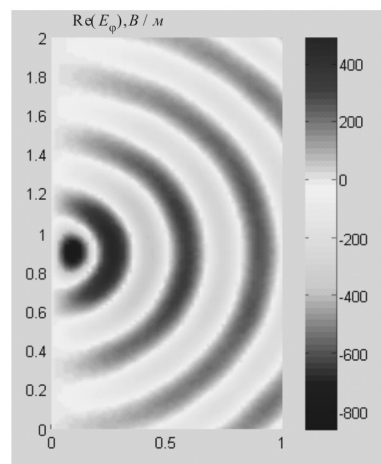


Рис. 1. Результаты расчета действительной компоненты электрического поля круглой рамки радиусом $a = 0,05$ м, имитирующей ЭМД, классическим двумерным МКЭ:
а – в направлении максимума излучения; б – в виде амплитуды E_φ , распределенной в расчетной области

$$+ \left\{ \frac{\partial H_1}{\partial z} \quad \frac{\partial H_2}{\partial z} \quad \frac{\partial H_3}{\partial z} \right\} \left\{ \frac{\partial H_1}{\partial z} \right. \\ \left. \frac{\partial H_2}{\partial z} \right. \\ \left. \frac{\partial H_3}{\partial z} \right\} dr dz. \quad (3)$$

К слагаемым подынтегральной функции добавим $\frac{k_0^2 \varepsilon_{\text{рк}}}{r} (r E_\phi)$ из (1). Функция формы треугольного элемента H_i (для узлов с локальными номерами $i = 1, 2, 3$), показанная в данном уравнении, может описываться полиномами различных степеней. Так, если функция формы содержит полиномы первой степени вида $p = c_1 + c_2 x_1 + c_3 x_2$, где в нашем случае $x_1 = r$, $x_2 = z$, элемент принято называть P_1 -элементом. Если функция формы содержит полиномы второй степени вида $p = c_1 + c_2 x_1 + c_3 x_2 + c_4 x_1^2 + c_5 x_1 x_2 + c_6 x_2^2$, такой элемент принято называть P_2 -элементом.

Если рассматривать P_1 -элементы, то полученная нами функция формы H_i примет следующий вид:

$$H_1 = \frac{1}{2A} [(r_2 z_3 - r_3 z_2) + (z_2 - z_3)r + (r_3 - r_2)z], \quad (4)$$

$$H_2 = \frac{1}{2A} [(r_3 z_1 - r_1 z_3) + (z_3 - z_1)r + (r_1 - r_3)z], \quad (5)$$

$$H_3 = \frac{1}{2A} [(r_1 z_2 - r_2 z_1) + (z_1 - z_2)r + (r_2 - r_1)z], \quad (6)$$

то есть будет являться по сути матрицей базисных функций единичного элемента. Здесь (r_i, z_i) — координаты соответствующих вершин элемента.

Данная функция формы удовлетворяет условию

$$H_i(r_j, z_j) = \delta_{ij}, \quad (7)$$

то есть

$$\sum_{i=1}^3 H_i = 1. \quad (8)$$

Здесь δ_{ij} — дельта Кронекера. Она будет равна

$$\delta_{ij} = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}. \quad (9)$$

После подстановки выражений (4)–(6)

в уравнение (3) получим «матрицу жесткости» элемента K^e вида

$$[K^e] = \begin{bmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \\ k_{31} & k_{32} & k_{33} \end{bmatrix}, \quad (10)$$

элементы которой получены путем дифференцирования H_i :

$$k_{11} = \frac{1}{4A} \left[\frac{r_1(z_2 - z_3)^2}{r_c} + (r_3 - r_2)^2 \right], \quad (11)$$

$$k_{12} = \frac{1}{4A} \left[\frac{r_2(z_2 - z_3)(z_3 - z_1)}{r_c} + (r_3 - r_2)(r_1 - r_3) \right], \quad (12)$$

$$k_{13} = \frac{1}{4A} \left[\frac{r_3(z_2 - z_3)(z_1 - z_2)}{r_c} + (r_3 - r_2)(r_2 - r_1) \right], \quad (13)$$

$$k_{21} = \frac{1}{4A} \left[\frac{r_1(z_3 - z_1)(z_2 - z_3)}{r_c} + (r_1 - r_3)(r_3 - r_2) \right], \quad (14)$$

$$k_{22} = \frac{1}{4A} \left[\frac{r_2(z_3 - z_1)^2}{r_c} + (r_1 - r_3)^2 \right], \quad (15)$$

$$k_{23} = \frac{1}{4A} \left[\frac{r_3(z_3 - z_1)(z_1 - z_2)}{r_c} + (r_1 - r_3)(r_2 - r_1) \right], \quad (16)$$

$$k_{31} = \frac{1}{4A} \left[\frac{r_1(z_1 - z_2)(z_2 - z_3)}{r_c} + (r_2 - r_1)(r_3 - r_2) \right], \quad (17)$$

$$k_{32} = \frac{1}{4A} \left[\frac{r_2(z_1 - z_2)(z_3 - z_1)}{r_c} + (r_2 - r_1)(r_1 - r_3) \right], \quad (18)$$

$$k_{33} = \frac{1}{4A} \left[\frac{r_3(z_1 - z_2)^2}{r_c} + (r_2 - r_1)^2 \right], \quad (19)$$

где

$$A = 0,5 \det \begin{bmatrix} 1 & r_1 & z_1 \\ 1 & r_2 & z_2 \\ 1 & r_3 & z_3 \end{bmatrix} \quad (20)$$

является площадью треугольного элемента, выраженной через узловые координаты его вершин.

Формирование матрицы **H** и векторов **F** и **G** производится аналогично классическому узловому МКЭ для декартовых координат.

Как ни странно, выражения (11)–(19) именно в такой интерпретации в литературе не встречались, поэтому будем считать, что они получены нами впервые путем прямой подстановки функций формы в исходное уравнение Гельмгольца в цилиндрических координатах. Именно такое представление матрицы жесткости приводит к правильному снижению комплексных амплитуд поля при удалении от оси симметрии и практически полному совпадению фазы волны с фазой точного решения, что в свою очередь позволяет существенно снизить относительную погрешность действительных и мнимых составляющих E_φ , что будет показано ниже.

Методика вывода функциональной зависимости возбуждения точечного источника для решения осесимметричных задач излучения

Ненулевые элементы правой части **b** решаемого СЛАУ уже рассматривались ранее, где они были равны $-i\delta \dot{I}_m^{\text{ст}} \neq 0$. Таким образом, за счет априорного задания данной величины, можно получить начальное приближение.

Представление функции возбуждения начнем с рассмотрения неоднородного уравнения Гельмгольца (1). Для данного уравнения правая часть (функция возбуждения) записывается как

$$\bar{M}^3 = -i\omega\mu_a \bar{j}^{\text{э.ст}} + \frac{1}{i\omega\epsilon_{\text{ак}}} \text{graddiv} \bar{j}^{\text{э.ст}} - \text{rot} \bar{j}^{\text{м.ст}}, \quad (21)$$

где $\bar{j}^{\text{э.ст}}$, $\bar{j}^{\text{м.ст}}$ – мгновенные значения век-

торов объемной плотности электрического и магнитного тока; $\epsilon_{\text{ак}}$, μ_a – абсолютные значения комплексной диэлектрической и магнитной проницаемостей среды. Так как МКЭ решается скалярное уравнение, то векторные слагаемые лишаются смысла.

Показанное выше представление функции возбуждения в виде точки нецелесообразно, т. к. в силу зависимости составляющих вектора **F** от величины площади элемента *A* при уменьшении последней в ходе учащения сетки разбиения будут уменьшаться и полученные комплексные амплитуды.

Тогда, в силу сложности вывода строгого соотношения для правой части исходного скалярного уравнения Гельмгольца, выражение (21) можно переписать с точностью до постоянного множителя $\gamma(ah)$ как

$$M_{\text{МКЭ}} = -i\omega\mu_0\mu_r j^{\text{э.ст}} + \frac{1}{i\omega\epsilon_{\text{ак}}} \text{graddiv} \bar{j}^{\text{э.ст}} = \quad (22)$$

$$= -i\omega\mu_0\mu_r \dot{I}_m^{\text{ст}} \gamma(ah),$$

где $\gamma(ah)$ – скалярная величина, пропорциональная плотности стороннего тока (зависящая от положения источника относительно оси симметрии (радиуса рамки *a*) и ширины полоски *h*, образующей рамку).

Таким образом, в двумерном аксиально-симметричном представлении рамка будет представлена нитью постоянного тока, параллельной оси симметрии, длина которой равна ширине полоски *h* (рис. 2).

Такое представление модели возбуждения позволяет избежать зависимости полученных комплексных амплитуд поля от размера элементов.

На практике задание токовой нити в СЛАУ $[\mathbf{K} + \mathbf{H}] \cdot \mathbf{x} = \mathbf{F} + \mathbf{G}$ программируется в векторе **G** в виде дополнительной границы с ГУ вида (6), где $\kappa = 0$, $g = -i\omega\mu_0\mu_r \dot{I}_m^{\text{ст}} \gamma(ah)$.

В выражении (22) величина функции возбуждения заменяется на $\dot{I}_m^{\text{ст}} \gamma(a, h)$, так как сторонний источник в задаче представляется в виде бесконечно тонкой плоской круглой рамки. Из анализа выражения (2) видно, что величина $\dot{E}_m$ пропорциональна магнитному моменту, т. е. прямо пропорциональна стороннему току $\dot{I}_m^{\text{ст}}$, радиусу рамки и обратно пропорциональна длине волны λ . Тогда выражение (22) можно

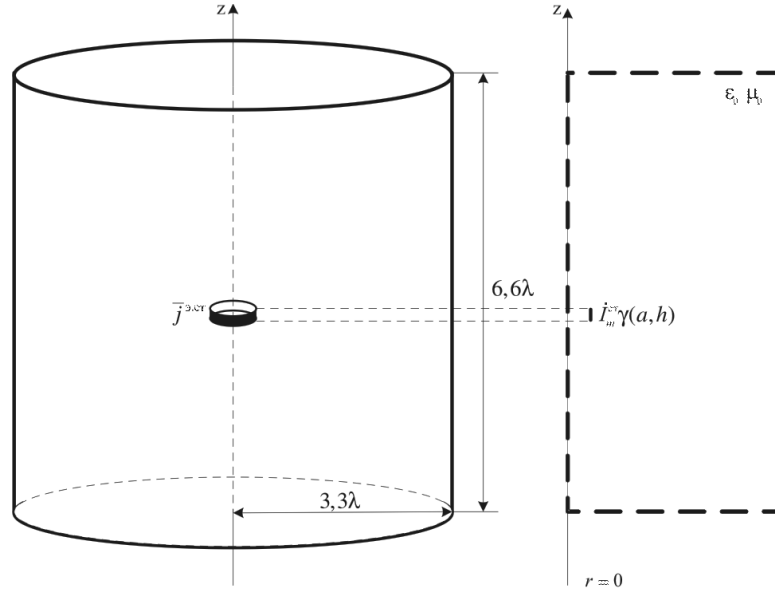


Рис. 2. Усовершенствованная модель ЭМД в виде плоской рамки с током

с точностью до постоянного множителя $\gamma(a, h)$ можно переписать как

$$M_{\text{МКЭ}} = \frac{-i2\pi c \mu_0 \mu_r \dot{J}_m^{\text{ct}}}{\lambda} \gamma(a, h), \quad (23)$$

где c — фазовая скорость волны в среде рас-

пространения (в нашем случае — скорость света).

Поиск значения величины $\gamma(a, h)$ произведем методом перебора с допуском 2 % погрешности, путем минимизации следующего функционала:

$$\Delta = \text{mean} \left| \frac{\text{Re}(E_{\phi}^{\text{расч}}(r, z=1)) - \text{Re}(E_{\phi}^{\text{ан}}(r, z=1))}{\text{Re}(E_{\phi}^{\text{ан}}(r, z=1))} \right| \rightarrow \min, \quad (24)$$

где Δ — относительная погрешность; $E_{\phi}^{\text{расч}}(r, z=1)$ — рассчитанные МКЭ комплексные амплитуды вектора электрического поля; $E_{\phi}^{\text{ан}}(r, z=1)$ — комплексные амплитуды поля, полученные аналитически согласно выражению (2).

Результаты поиска представлены в таблице.

Необходимо оговориться, что для поиска минимума функционала (24) можно использовать любой метод оптимизации. Однако, учитывая, что аппроксимация множества полученных точек некоторой непрерывной функцией все равно приведет к некоторой погрешности, в данной ситуации метод перебора вполне приемлем.

Полученные результаты лишней раз подтверждают тот факт, что с увеличением радиуса рамки при фиксированной толщи-

не полосы плотность тока, распределенная по полоске, определяемая выражением $\dot{J}_m^{\text{ct}} \gamma(a, h)$, закономерно убывает.

Интерполяцию дискретных значений функции $\gamma(a, h)$ для нахождения непрерывной зависимости произведем полиномом Лагранжа.

Аппроксимирующий полином Лагранжа, пригодный для практических алгоритмов, получаемый путем интерполяции $\gamma(a, h = \text{const})$, в зависимости от a имеет вид

$$L_n(a/\lambda) = \sum_{k=1}^{n+1} \gamma_k \prod_{\substack{j=1,2,\dots,n+1 \\ j \neq k}} \frac{a/\lambda - (a/\lambda)_j}{(a/\lambda)_k - (a/\lambda)_j}.$$

На рис. 3 видно, что полученную из таблицы кривую достаточно удобно аппроксимировать полиномом вида $y = C_1 e^{-x} + C_2 x^{-1}$. Итоговое выражение для

Результаты минимизации функционала (24) методом перебора при фиксированной $h = 0,02$ м

a , м	0,01	0,015	0,02	0,025	0,03	0,035	0,04	0,045	0,05	0,055	0,06
$\gamma(a, h)$	940	615	480	375	330	283	252	229,5	215	198	186
Δ , %	1,9	1,9	1,9	2,0	2,0	2,0	2,1	2,2	2,2	2,3	2,4

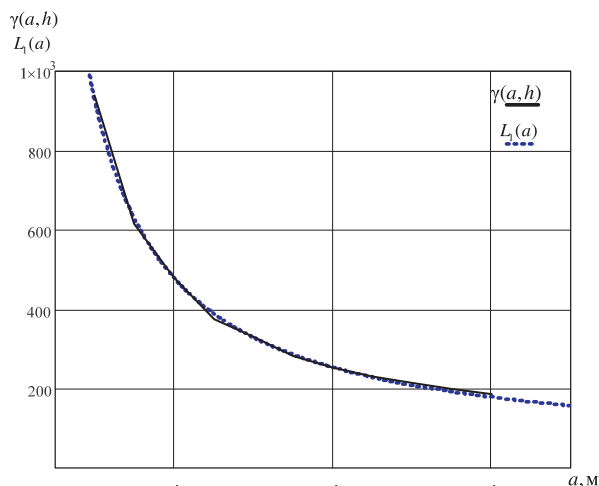
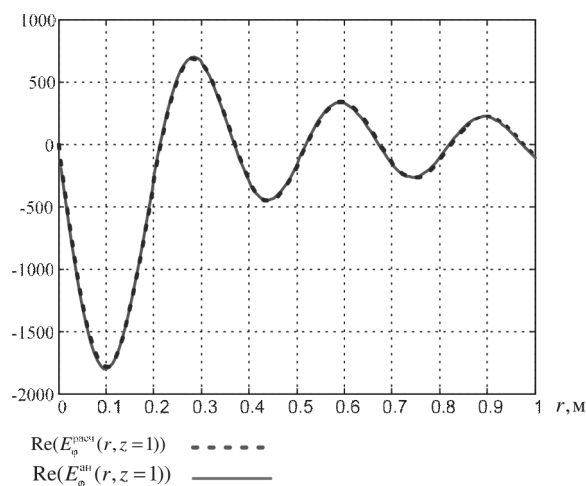


Рис. 3. Зависимость $L_1(a)$ и γ от a/λ

трех членов, пригодное для расчета величины $\gamma(a, h = 0,02)$, полученное методом наименьших квадратов, приняло следующий вид:

$$\gamma(a, h = 0,02) \approx L_1(a) = 30,763 \exp(-a) + 8,977(a^{-1}), \quad (25)$$

где a — радиус витка.



На рис. 4 и 5 показаны результаты расчета поля излучения рамки радиусом $a = 0,03$ м с применением модифицированной матрицы жесткости (3) для реальной и мнимой частей комплексной амплитуды в сравнении с результатами, полученными согласно выражению (2). При таком радиусе рамки относительная погрешность МКЭ составляет 2 %.

Из представленных рисунков видно практически полное совпадение полученного при помощи выведенной функции $\gamma(a, h = 0,02)$ МКЭ-решения в сравнении со значениями комплексных амплитуд, полученными аналитически. Если увеличивать радиус рамки, погрешность может достигнуть значения 2,3 % и более, что является вполне закономерным.

Выведенные выражения для элементов «матрицы жесткости» позволяют производить электродинамический анализ осесимметричных излучателей с высокой точностью.

Таким образом, полученная нами методика вывода функциональной зависимости возбуждения точечного источника для ре-

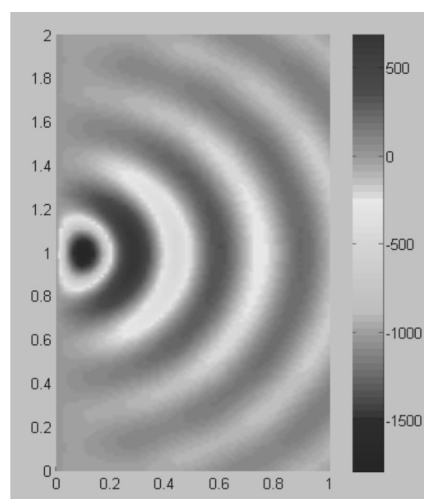


Рис. 4. Результаты численного расчета действительной составляющей комплексной амплитуды вектора $E_\phi^{\text{расч}}$ в сравнении с $E_\phi^{\text{ан}}$, полученными строгим методом рамки радиусом 0,03 м

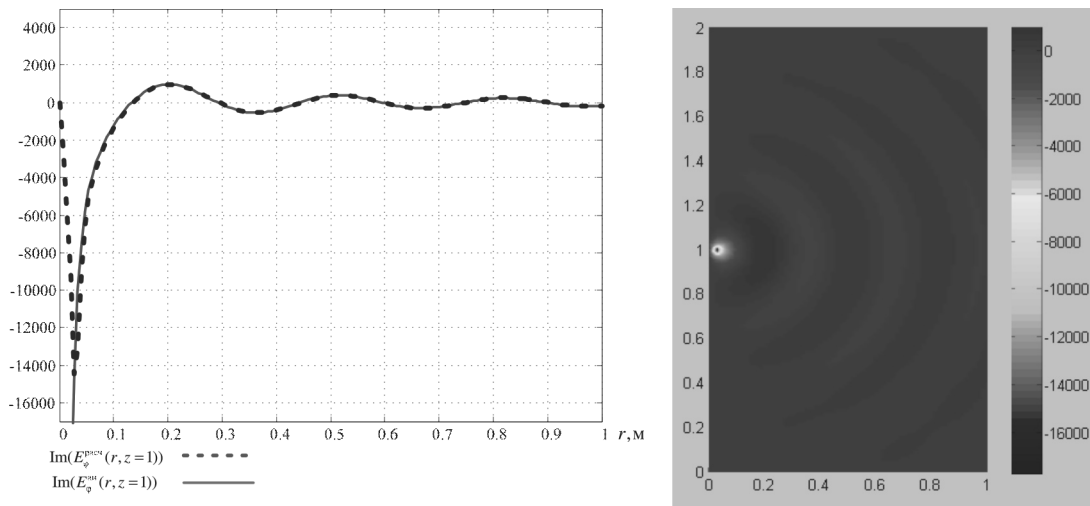


Рис. 5. Результаты численного расчета мнимой составляющей комплексной амплитуды вектора $E_{\phi}^{\text{расч}}$ в сравнении с $E_{\phi}^{\text{ан}}$, полученными строгим методом рамки радиусом 0,03 м

шения задач излучения в цилиндрических координатах будет содержать следующие этапы:

фиксация ширины полосы h , желатель- но симметрично точке возбуждения;

получение дискретных значений функ- ции $\gamma(a, h = \text{const})$ в зависимости от радиу- сов витков;

нахождение C_1 и C_2 полинома Лагран- жа вида $y = C_1 e^{-x} + C_2 x^{-1}$ для получения не- прерывной функциональной зависимости функции возбуждения от радиуса витка путем минимизации функционала (24) с оценкой относительной погрешности лю- бым оптимизационным методом;

оценка полученного результата путем

сравнения МКЭ-решения с решением, по- лученным любым строгим методом.

Выведенные выражения для элементов «матрицы жесткости» позволяют произво- дить электродинамический анализ (кон- струкционный синтез) осесимметричных излучателей с высокой точностью.

Описанная методика получения функ- ции возбуждения позволяет производить корректное моделирование источника воз- буждения осесимметричных тел. Ее можно легко расширить на предмет анализа осе- симметричных антенн, существенно умень- шив при этом размерность задачи и полу- чив выигрыш во времени.

СПИСОК ЛИТЕРАТУРЫ

1. Сосунов Б.В., Бородулин Р.Ю. Конструк- ционный синтез элементов фазированных антен- ных решеток // Научно-технические ведомости СПбГПУ. Информатика. Телекоммуникации. Управление. СПб., 2013. № 2(169). С. 47–54.
2. Kwon Young W. The finite element method using MATLAB (The mechanical engineering

series). New York: CRC Press LLC, 1997.

3. Jianming Jin. The finite element method in electromagnetics. 2nd ed. New York: John Wiley and Sons Inc., 2002.

4. Никольский В.В., Никольская Т.И. Элек- тродинамика и распространение радиоволн. М.: Наука, 1989.

REFERENCES

1. Sosunov B.V., Borodulin R.Yu. Konstruktsionnyy sintez elementov fazirovannykh anten- nykh reshetok [Constructural synthesis of element of a fased array of antennas]. Nauchno-tehnicheskkiye vedomosti SPbGPU. Informatika. Telekommunikatsii. Upravleniye

[St. Petersburg State Polytechnical University Journal. Computer Science. Telecommunication and Control]. St. Petersburg, 2013, No. 2(169), Pp. 47–54. (rus)

2. Kwon Young W. The finite element method using MATLAB. The mechanical engineering series,

New York: CRC Press LLC, 1997.

3. **Jianming Jin.** *The finite element method in electromagnetics*, 2nd ed., New York: John Wiley and Sons Inc., 2002.

4. **Nikolskiy V.V., Nikolskaya T.I.** *Elektrodinamika i rasprostraneniye radiovoln [Electrodynamics and distribution of radio waves]*. Moscow: Nauka Publ., 1989. (rus)

БОРОДУЛИН Роман Юрьевич — докторант Военной академии связи имени С.М. Буденного, кандидат технических наук.

194064, Россия, Санкт-Петербург, Тихорецкий пр., д. 3.

E-mail: borodulroman@yandex.ru

BORODULIN Roman Yu. *Military academy of communications named after S.M. Budyonny.*

194064, Tikhoretsky Ave. 3, St. Petersburg, Russia.

E-mail: borodulroman@yandex.ru



Моделирование вычислительных, телекоммуникационных, управляющих и социально-экономических систем

DOI: 10.5862/JCSTCS.236.4

УДК 51-74

А.П. Раевская, А.Ю. Крылатов

МЕТОДЫ ОЦЕНКИ МАТРИЦЫ КОРРЕСПОНДЕНЦИЙ В ЗАГРУЖЕННЫХ ТРАНСПОРТНЫХ СЕТЯХ

A.P. Raevskaya, A.Yu. Krylatov

OD-MATRIX ESTIMATION FOR URBAN TRAFFIC AREA CONTROL

Статья посвящена вопросам оценки и восстановления матрицы корреспонденций (МК) в загруженных транспортных сетях. Проведен анализ зарубежной и отечественной литературы, выделены наиболее эффективные методы оптимального расположения датчиков регистрации номерных знаков транспортных средств на сети для получения наиболее полной информации, необходимой при восстановлении МК. С целью повышения эффективности метода, базирующегося на двухуровневой модели оценки МК, разработана оригинальная процедура оптимального расположения датчиков фиксации номерных знаков транспортных потоков на сети произвольной топологии, основанная на решении некоторой оптимизационной задачи. Разработанная процедура позволяет максимизировать вероятность фиксации наиболее значимых транспортных потоков на протяжении всего маршрута следования. Проведено имитационное моделирование с экспериментальными данными на транспортной сети Санкт-Петербурга.

МАТРИЦА КОРРЕСПОНДЕНЦИЙ; ТРАНСПОРТНЫЕ ПОТОКИ; ДАТЧИКИ, СКАНИРУЮЩИЕ НОМЕРНЫЕ ЗНАКИ.

The paper is dedicated to the problem of OD-matrix estimation and reconstruction. The authors carried out a review of the literature of foreign and Russian authors on the subject of the estimation and reconstruction of a trip matrix. The most effective method of optimal plate scanning sensors location on the road network is noted. To improve this method, the new optimizational model is developed for the road network of general topology. The developed model allows to maximize the probability of fixation of the most significant traffic flows throughout the whole route. Simulation with experimental data is carried out on the transportation network of St. Petersburg.

TRIP MATRIX; OD-MATRIX; TRAFFIC FLOW; PLATE-SCANNING SENSORS.

Задача оценки и восстановления матрицы корреспонденций — крайне сложная и актуальная проблема в области транспортных исследований. Работа над ней ведется с 1960-х гг. Задача оценки и задача восстановления матрицы корреспонденций — это две совершенно разные задачи, и их решения могут сильно различаться [1]. Когда речь идет о восстановлении матрицы

корреспонденций, предполагается наличие возможности однозначного определения межрайонных корреспонденций и их распределение по имеющимся маршрутам. Однако такая возможность существует не всегда, т. к. требует наблюдаемости всех элементов транспортной сети. На самом деле не все дуги и узлы являются наблюдаемыми; на ненаблюдаемых элементах

потоки также необходимо оценивать. В таком случае встает задача оценки матрицы корреспонденций. В этом смысле наиболее наглядной моделью оценки матриц корреспонденций является гравитационная модель [2]. Авторами [3] предложен мощный метод Байеса решения задачи предсказания, оценки и восстановления матрицы корреспонденций.

Одна из первых моделей оценки матриц корреспонденций сформулирована как двухуровневая модель и разработана в конце XX в. [4]. Задача верхнего уровня заключается в нахождении целевой OD-матрицы по заданному набору наблюдаемых на дугах потоков, которая минимизировала бы функцию расстояния между априорными и наблюдаемыми потоками и априорной и наблюдаемой матрицей корреспонденций. При этом требованием нижнего уровня является такое распределение потоков по сети, которое удовлетворяет условию равновесия Вардропа.

В настоящее время в России широко внедряются различные системы мониторинга дорожного движения. Данные таких систем в принципе могут применяться для построения матриц корреспонденций между узлами улично-дорожной сети, использование которых в процессе математического моделирования транспортных потоков поможет повысить эффективность инфраструктурных преобразований транспортных систем [5, 6]. Несмотря на многочисленные публикации, проблема оценки и восстановления матрицы корреспонденций остается актуальной и требует дальнейшего научного исследования.

Стоит отметить статью [7], авторы которой предлагают минимизировать затраты на сбор данных при помощи комбинации информации, полученной со счетчиков автомобилей и с датчиков, сканирующих номерные знаки. В этом, в принципе, и состоит основная идея — использовать все возможные данные для оценки потоков и матрицы корреспонденций. Но, к сожалению, не каждая модель может использовать все типы данных. Поэтому столь важным становится вопрос адекватной дезагрегации потоков. В статье [8] рассмотрен вопрос

надежности установки датчиков на транспортную сеть для фиксации транспортных потоков. Эвристический подход выбора местоположения для датчиков, фиксирующих транспортные потоки, предложен в [9].

В [10] произведен детальный сравнительный анализ трех методов оценки матрицы корреспонденций: метода линейного программирования, Байесовского подхода, метода изменяющейся во времени томографии сети. В связи с тем, что реальные размеры транспортных сетей очень велики, формулируются задачи минимизации количества датчиков, обеспечивающих достаточную наблюдаемость потоков на них [11, 12].

Следует еще раз особо отметить работу [7], в которой используется информация о номерных знаках, полученная с датчиков сканирования. Авторами предложен эффективный метод оценки матрицы корреспонденций. Однако вопрос оптимального расположения датчиков с точки зрения максимального покрытия основных транспортных потоков ими не рассматривается [13]. В то же время для эффективной работы предлагаемого метода требуется высококачественный сбор данных. В настоящей работе ставится задача устранить данный пробел и предложить модель расположения датчиков фиксации номерных знаков, обеспечивающую эффективное применение метода Кастилло [7].

Двухуровневая модель оценки матрицы корреспонденций

Будем рассматривать транспортную сеть, представленную графом $G = \langle N, A \rangle$, где N — множество вершин, A — множество дуг, $\bar{A} \subset A$ — подмножество наблюдаемых дуг, на которых установлены датчики.

Будем считать, что $R \subset N$ — множество районов отправления, $r \in R$, а $S \subset N$ — множество районов прибытия, $s \in S$; $R \cap S = \emptyset$.

Введем следующие обозначения: K^{rs} — множество маршрутов из r в s ; A_k — множество дуг, из которых состоит маршрут $k \in K^{rs}$, $A_k \subset A$; F^{rs} — транспортный спрос между районами отправления и прибытия r - s ; $F = \{F^{rs}\}^{r,s}$; $\bar{F}^{rs}$ — апри-



орный транспортный спрос (ранее оцененный) между районами отправления и прибытия $r-s$; $\bar{F} = \{\bar{F}^{rs}\}^{r,s}$; f_k — поток по k -му маршруту, $k \in K^{rs}$, $f = \{f_k\}_{k \in K^{rs}}$; $\bar{f}_k$ — априорный поток по k -му маршруту, $k \in K^{rs}$, $f = \{f_k\}_{k \in K^{rs}}$; x_a — поток по дуге $a \in A$, $x = (\dots, x_a, \dots)$; $\bar{x}_a$ — априорный поток по дуге $a \in A$, $\bar{x} = (\dots, \bar{x}_a, \dots)$; $t_a(x_a)$ — время движения потока x_a по дуге $a \in A$; U , V — весовые матрицы; $\delta_{a,k}^{rs}$ — индикатор: единица, если дуга a содержится в пути $k \in K^{rs}$, ноль в противном случае.

Двухуровневая модель оценки матрицы корреспонденций получается из комбинации условия минимизации суммы квадратов между наблюдаемыми и оцениваемыми значениями транспортного спроса и потоками на дугах и условия конкурентного равновесия Вардропы [14, 15]:

$$\min_F Z(F, x) = \min_F (\bar{F} - F)^T V^{-1} (\bar{F} - F) + (\bar{x} - x)^T U^{-1} (\bar{x} - x),$$

где $x \geq 0$ и удовлетворяет условию конкурентного равновесия

$$\min_x \sum_{a \in A} \int_0^{x_a} t_a(x) dx$$

при

$$\sum_{k \in K^{rs}} f_k = F^{rs}, f_k \geq 0, \quad \forall k \in K^{rs},$$

где

$$x_a = \sum_{r \in R} \sum_{s \in S} \sum_{k \in K^{rs}} f_k \delta_{a,k}^{rs}.$$

Целевой функционал верхнего уровня может быть интерпретирован как мера расстояния от оцениваемой матрицы корреспонденций F до ранее полученной $\bar{F}$ и от оцениваемого потока x до полученного со счетчиков $\bar{x}$. Следует отметить, что после того как датчики расставлены на сети, мы точно фиксируем потоки на соответствующих дугах, а на дугах без датчиков данные о потоках необходимо аппроксимировать. Второе слагаемое в целевой функции верхнего уровня имеет именно такой смысл.

Модель Кастилло оценки матрицы корреспонденций

Кастилло [7] модернизировал двухуровневую модель при помощи использования

датчиков, сканирующих номерные знаки. Одно из введенных в его модели ограничений отменяет необходимость во втором слагаемом в целевом функционале верхнего уровня двухуровневой модели. Дополнительно им вводятся ограничения, связанные с датчиками сканирования номерных знаков. Рассмотрим модель, которая опирается на квадратичную целевую функцию с взвешенной суммой разницы между априорными и оцениваемыми потоками. Введем дополнительные обозначения: $v_k \in V$ — вес маршрута k ; $v \in W$ — номер последовательности дуг с датчиками, сканирующими номерные знаки; η_{vk}^{rs} — индикатор: единица, если путь k между районами отправления-прибытия $r-s$ содержит все дуги из последовательности с номером v , ноль — в противном случае; $\bar{w}_v$ — количество пользователей, зафиксированных на v -й последовательности дуг. Таким образом, имеем

$$\min_f Z = \min_f \sum_{r \in R} \sum_{s \in S} \sum_{k_1, k_2 \in K^{rs}} (f_{k_1} - \bar{f}_{k_1}) \times v_k (f_{k_2} - \bar{f}_{k_2})$$

при условии

$$\bar{w}_v = \sum_{r \in R} \sum_{s \in S} \sum_{k \in K^{rs}} f_k \eta_{vk}^{rs}, \quad \forall v \in W, \quad (1)$$

$$\bar{x}_a = \sum_{r \in R} \sum_{s \in S} \sum_{k \in K^{rs}} f_k \delta_{a,k}^{rs}, \quad \forall a \in \bar{A},$$

$$f_k \geq 0 \quad \forall k \in K^{rs}, \quad \forall r, s.$$

Эффективность работы модели напрямую зависит от расположения датчиков, сканирующих номерные знаки транспортных средств, о чем свидетельствует ограничение (1). Значит, имеет смысл ставить вопрос об эффективном расположении датчиков для восстановления матрицы корреспонденций методом Кастилло.

Математическая модель расположения датчиков на транспортной сети с учетом многополосности для оценки матрицы корреспонденций

Рассмотрим вопрос оптимального расположения датчиков регистрации номерных знаков транспортных средств на многополосных транспортных сетях для эффективной реализации метода Кастилло.

Будем считать, что датчик может фиксировать данные об автомобилях, движущихся по той полосе, на которой он установлен. Введем дополнительные обозначения: q_a и c_a — количество датчиков и полос на дуге $a \in A$ соответственно; $q = (\dots, q_a, \dots)$.

Поскольку датчик однозначно идентифицирует автомобиль лишь на одной из полос, а автомобили могут перестраиваться с одной полосы на другую (рис. 1), то вероятность того, что автомобиль будет зафиксирован на c_a -полосной дороге с q_a датчиками равна $\frac{q_a}{c_a}, \forall a \in A$. Задача состоит в том, чтобы максимизировать вероятность охвата максимального транспортного потока на всей сети.

Будем считать, что произвольные события фиксации номерного знака автомобиля на последовательности дуг независимы. В таком случае оптимизационный функционал представляет собой сумму произведений вероятностей фиксации автомобилей датчиками, сканирующими номерные знаки, на протяжении всего пути следования, которая домножена на априорные транспортные потоки по соответствующим маршрутам:

$$\max_q z(q) = \max_q \sum_{r \in R} \sum_{s \in S} \sum_{k \in K^r} \left(\prod_{a \in A_k} \frac{q_a}{c_a} \right) \bar{f}_k. \quad (2)$$

Таким образом мы максимизируем вероятность фиксации транспортных потоков на протяжении всего маршрута следования. При этом условие того, что хотя бы одна дуга на каждом пути является сканируемой, задается неравенством

$$\sum_{a \in A_k} q_a \geq 1, \forall k \in K^r. \quad (3)$$

Существование как минимум одной сканируемой дуги, которая находится на пути k_1 и при этом не принадлежит пути k_2 , гарантируется следующим ограничением:

$$\sum_{a \in A} q_a \delta_a^{k_1 k_2} \geq 1, \forall k_1, k_2 \in K^r : k_1 \neq k_2 \quad (4)$$

$$\delta_a^{k_1 k_2} = \begin{cases} 1, & \text{если } a \in k_1, a \notin k_2, \\ 0, & \text{иначе.} \end{cases} \quad (5)$$

Введем бюджетное ограничение, где Q — доступное количество детекторов:

$$\sum_{a \in A} q_a \leq Q. \quad (6)$$

Количество датчиков на дуге не должно



Рис. 1. Маршрут, состоящий из многополосных дуг

превышать количества полос:

$$0 \leq q_a \leq c_a, \forall a \in A. \quad (7)$$

Таким образом, мы сформулировали задачу целочисленного программирования на ограниченном множестве решений. Следовательно, решение у данной задачи существует, когда множество допустимых решений не пусто. Множество допустимых решений сформулированной оптимизационной программы может оказаться пустым при слишком сильных бюджетных ограничениях, когда количество датчиков недостаточно для охвата сети. Для решения этой проблемы необходимо либо увеличить бюджет, либо сократить количество второстепенных маршрутов, потоки по которым незначительны.

Эксперимент на транспортной сети Санкт-Петербурга

Рассмотрим многополосную транспортную сеть центральной части Санкт-

Петербурга. Выберем девять районов отправления и прибытия (рис. 2), зададим девять маршрутов между районами отправления и прибытия и разделим все маршруты на дуги, которых всего на транспортной сети получится 21 (рис. 3).

Зададим количество полос c_a на каждой из дуг. Присвоим произвольным образом каждой из дуг значение от 1 до 3 — количество полос в одном из направлений дуги. Аналогичным образом зададим априорные потоки $\bar{f}_k$ по маршрутам. Затем рассчитаем оптимальное количество и местоположение датчиков для различных значений бюджетных ограничений от нуля до 50.

Реализацию данной модели произведем в программной среде MatLab. Перед нами стоит оптимизационная задача с нелинейным функционалом и линейными ограничениями. Такого рода задачи в программной среде MatLab решаются при помощи пакета Optimisation Toolbox. Используем функцию *fmincon* поиска минимума не-

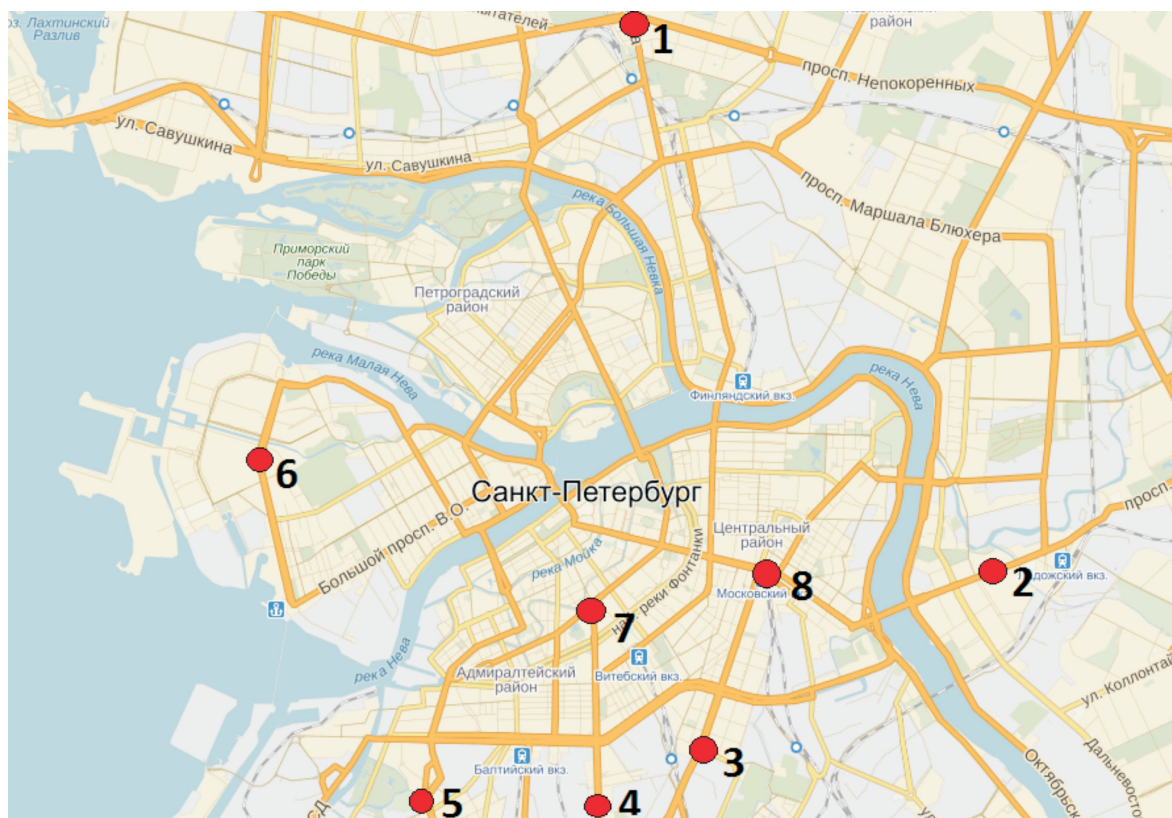


Рис. 2. Выбор районов отправления и прибытия

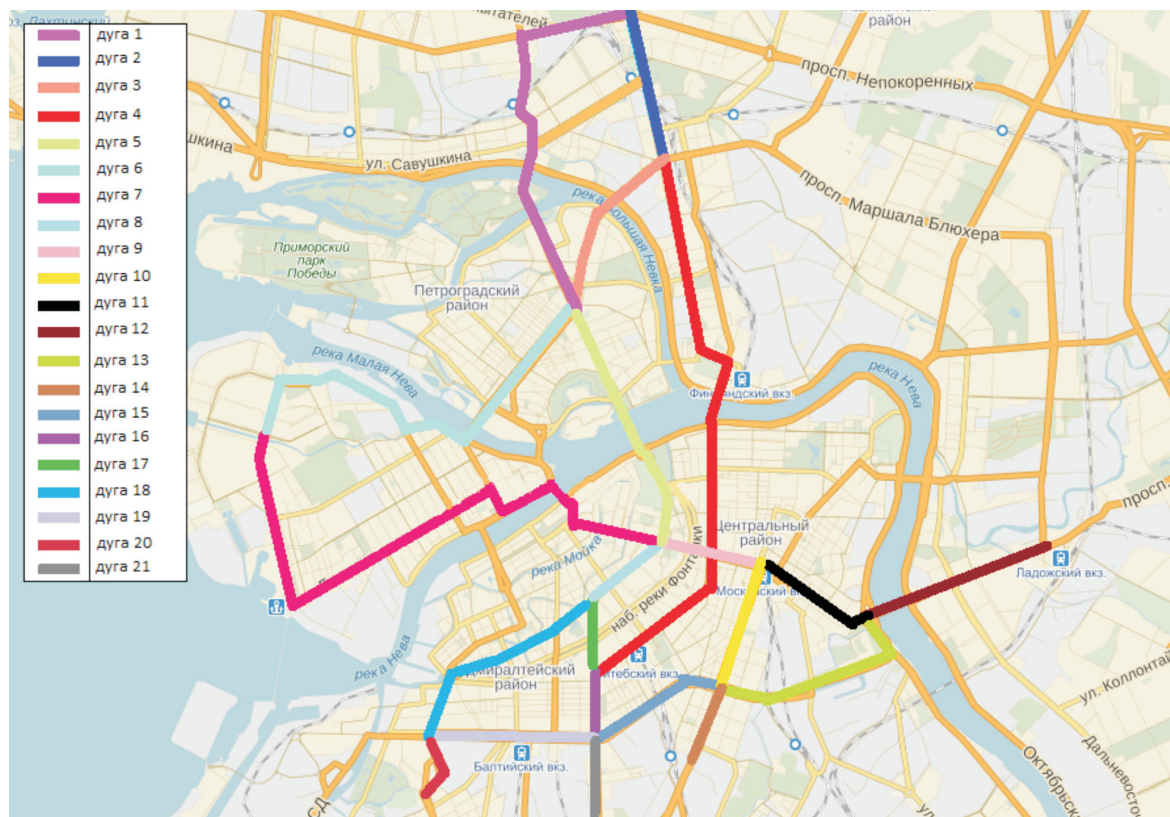


Рис. 3. Дуги, составляющие маршруты

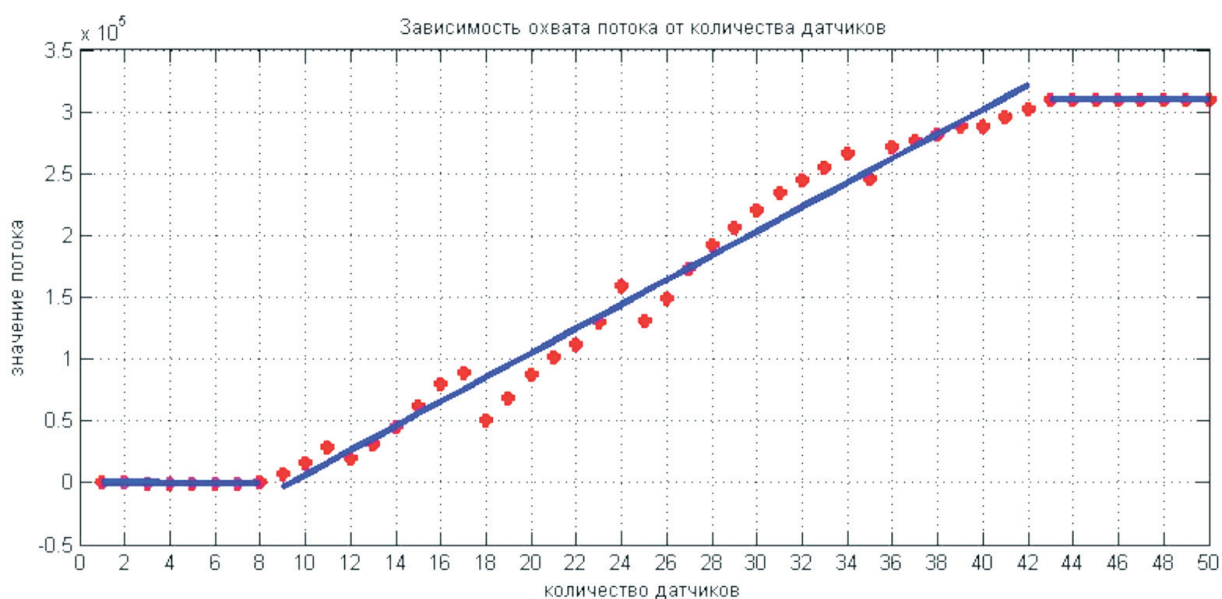


Рис. 4. Аппроксимирующая прямая зависимости значения потоков от количества датчиков на сети

линейной функции с ограничениями, которая решает задачу на основе алгоритма SQP (последовательного квадратичного программирования). На графике (рис. 4) показано, как изменяются потоки на каждом из маршрутов в зависимости от количества датчиков. По горизонтали указано количество датчиков, по вертикали — значения потоков.

Рассмотрено решение задачи для разных комбинаций ограничений:

- отдельно для бюджетного (6);
- для бюджетного (6) и для ограничения (3);
- для всех ограничений (3)–(7).

Благодаря такому разбиению можно произвести анализ чувствительности разработанной модели. Из полученных данных видно, что значение общего наблюдаемого потока в зависимости от количества датчиков на рассматриваемой транспортной сети при разных видах ограничений практически не меняется. В связи с этим был проведен анализ изменения потоков по каждому из маршрутов в зависимости от количества датчиков на сети и различных комбинаций ограничений. Такой анализ показал, что в первую очередь начинают наблюдаться маршруты, по которым проходят значительные потоки и при этом составляющие их

число дуг наименьшее. Как видим (рис. 5), при увеличении количества ограничений большие потоки на маршрутах с большим количеством датчиков начинают фиксироваться значительно позднее.

Практическое применение разработанной модели

Модель, описанная в статье, является статической. Как правило, в статических моделях рассматривается средний поток пользователей и усредненные характеристики движения. При реализации данной модели на практике следует учитывать, что величины потоков изменяются в зависимости от времени суток, времени года и других внешних факторов. Поэтому при расстановке датчиков следует ориентироваться именно на величины потоков в «часы пик», поскольку, удовлетворив спрос в самые загруженные часы для самых больших потоков, в остальные часы сеть точно будет удовлетворять спросу на перемещения. Ситуация на дорогах очень зависит от времени года и погодных условий, которые тяжело предсказать. Поэтому необходимо в течение первого апробационного периода длительностью год накопить статистические данные, позволяющие учитывать сезонные изменения

Количество датчиков на сети	Доля наблюдаемого потока $f(1)$. Поток = 47516. Дуги = 7. Полосы = 15. Бюджетное ограничение.	Доля наблюдаемого потока $f(1)$. Поток = 47516. Дуги = 7. Полосы = 15. Два ограничения.	Доля наблюдаемого потока $f(1)$. Поток = 47516. Дуги = 7. Полосы = 15. Все ограничения.
5	0	0	0
10	0	0	0
15	1,00	0	0
20	1,00	1,00	0
25	1,00	1,00	1,00
30	1,00	1,00	1,00
35	1,00	1,00	1,00
40	1,00	1,00	1,00
45	1,00	1,00	1,00
50	1,00	1,00	1,00

Рис. 5. Фиксация потоков в зависимости от ограничений

в модели. Благодаря тому, что предполагается использование мобильных датчиков, которые можно легко перемещать, можно производить калибровку модели, меняя местоположения датчиков и анализируя полученные результаты.

Статья посвящена проблеме оптимального расположения датчиков фиксации номерных знаков транспортных потоков на сети с целью сбора информации для восстановления матрицы корреспонденций. В качестве метода восстановления матриц корреспонденций выбран метод Кастилло, базирующийся на классическом двухуровневом методе. С целью повышения

эффективности метода оценки матрицы корреспонденций при помощи фиксации номерных знаков, предложенного Кастилло, получены следующие теоретические и практические результаты:

разработана детерминированная модель оптимального расположения датчиков фиксации номерных знаков транспортных потоков на сети произвольной топологии;

разработан алгоритм решения детерминированной задачи расположения датчиков;

реализован алгоритм решения детерминированной задачи расположения датчиков на примере транспортной сети Санкт-Петербурга в программной среде MatLab.

СПИСОК ЛИТЕРАТУРЫ

1. Hazelton M. Inference for origin-destination matrices: estimation, prediction and reconstruction // *Transportation Research. Part B*. 2001. No. 35. Pp. 667–676.
2. Швецов В.И. Математическое моделирование транспортных потоков // *Автоматика и телемеханика*. 2003. № 11. С. 3–46.
3. Ehlert A., Bell M., Grosso S. The optimisation of traffic count locations in road networks // *Transportation Research. Part B*. 2006. No. 40. Pp. 460–479.
4. Yang H., Sasaki T., Iida Y., Asakura Y. Estimation of origin-destination matrices from link traffic counts on congested networks // *Transportation Research. Part B*. 1992. No. 26(6). Pp. 417–434.
5. Захаров В.В., Крылатов А.Ю. Современные проблемы использования интеллектуальной базы математического моделирования при борьбе с заторами в крупных городах России // *Транспорт Российской Федерации*. 2014. № 4(53). С. 69–73.
6. Захаров В.В., Крылатов А.Ю. Системное равновесие транспортных потоков в мегаполисе и стратегии навигаторов: теоретико-игровой подход // *Математическая теория игр и ее приложения*. 2012. Т. 4. № 4. С. 23–44.
7. Castillo E., Menedez J.M., Jimenez P. Trip matrix and path flow reconstruction and estimation based on plate scanning and link observations // *Transportation Research. Part B*. 2008. No. 42. Pp. 455–481.
8. Li X., Ouyang Y. Reliable sensor deployment for network traffic surveillance // *Transportation Research. Part B*. 2011. No. 45. Pp. 218–231.
9. Yim P., Lam W. Evaluation of count location selection of OD matrices // *J. of Transport Engineering*. 1998. Pp. 376–383.
10. Medina A., Taft N., Salamatian K., Bhattacharyya S., Diot C. Traffic matrix estimation: existing techniques and new directions // *Computer Communication Review. Proc. of the SIGCOMM Conf.* 2002. No. 32. Pp. 161–174.
11. Bianco L., Cerrone C., Cerulli R., Gentili M. Locating sensors to observe network arc flows: exact and heuristic approaches // *Computers and Operation Research*. 2014. No. 46. Pp. 12–22.
12. Minguez R., Sanchez-Cambronero S., Castillo E., Jimenez P. Optimal traffic plate scanning location for OD trip matrix and route estimation in road networks // *Transportation Research. Part B*. 2010. No. 44. Pp. 282–298.
13. Крылатов А.Ю., Раевская А.П. Оптимальное расположение датчиков на транспортной сети для оценки матрицы корреспонденций // *Процессы управления и устойчивость*. 2015. Т. 2. № 18. С. 629–634.
14. Захаров В.В., Крылатов А.Ю. Конкурентная маршрутизация транспортных потоков поставщиками услуг навигации // *Управление большими системами: Сб. трудов*. 2014. № 49. С. 129–147.
15. Захаров В.В., Крылатов А.Ю. Конкурентное равновесие Вардропы на транспортной сети из параллельных неоднородных маршрутов // *Процессы управления и устойчивость*. 2014. Т. 1. № 17. С. 476–481.

REFERENCES

1. Hazelton M. Inference for origin-destination matrices: estimation, prediction and reconstruction. *Transportation Research, Part B*, 2001, No. 35, Pp. 667–676.
2. Shvetsov V.I. Matematicheskoye modelirovaniye transportnykh potokov [Mathematical Modeling of Traffic Flows]. *Avtomatika i telemekhanika [Automation and Remote Control]*, 2003, Vol. 11, Pp. 3–46. (rus)
3. Ehlert A., Bell M., Grosso S. The optimisation of traffic count locations in road networks. *Transportation Research, Part B*, 2006, No. 40, Pp. 460–479.
4. Yang H., Sasaki T., Iida Y., Asakura Y. Estimation of origin-destination matrices from link traffic counts on congested networks. *Transportation Research, Part B*, 1992, No. 26(6), Pp. 417–434.
5. Zakharov V.V., Krylatov A.Yu. Sovremennyye problemy ispolzovaniya intellektualnoy bazy matematicheskogo modelirovaniya pri borbe s zatorami v krupnykh gorodakh Rossii [Current problems in using simulation-analysis intellectual database in the fight against traffic jams in Russia's major cities]. *Transport Rossiyskoy Federatsii [Transport of the Russian Federation]*, 2014, No. 4(53), Pp. 69–73. (rus)
6. Zakharov V.V., Krylatov A.Yu. Sistemnoye ravnovesiye transportnykh potokov v megapolise i strategii navigatorov: teoretiko-igrovoy podkhod [Traffic flows' system equilibrium in megapolis and navigators' strategies: game theory approach]. *Matematicheskaya teoriya igr i yeye prilozheniya [Automation and Remote Control]*, 2012, Vol. 4, No. 4, Pp. 23–44. (rus)
7. Castillo E., Menedez J.M., Jimenez P. Trip matrix and path flow reconstruction and estimation based on plate scanning and link observations. *Transportation Research, Part B*, 2008, No. 42, Pp. 455–481.
8. Li X., Ouyang Y. Reliable sensor deployment for network traffic surveillance. *Transportation Research, Part B*, 2011, No. 45, Pp. 218–231.
9. Yim P., Lam W. Evaluation of count location selection of OD matrices. *Journal of transport engineering*, 1998, Pp. 376–383.
10. Medina A., Taft N., Salamatian K., Bhattacharyya S., Diot C. Traffic matrix estimation: existing techniques and new directions. *Computer Communication Review, Proceedings of the SIGCOMM Conference*, 2002, No. 32, Pp. 161–174.
11. Bianco L., Cerrone C., Cerulli R., Gentili M. Locating sensors to observe network arc flows: exact and heuristic approaches. *Computers and Operation Research*, 2014, No. 46, Pp. 12–22.
12. Minguez R., Sanchez-Cambronero S., Castillo E., Jimenez P. Optimal traffic plate scanning location for OD trip matrix and route estimation in road networks. *Transportation Research, Part B*, 2010, No. 44, Pp. 282–298.
13. Krylatov A.Yu., Rayevskaya A.P. Optimalnoye raspolozheniye datchikov na transportnoy seti dlya otsenki matritsy korrespondentsiy [Optimal traffic plate scanning location for trip matrix estimation on the road networks]. *Protsessy upravleniya i ustoychivost [Control Processes and Stability]*, 2015, Vol. 2, No. 18, Pp. 629–634. (rus)
14. Zakharov V.V., Krylatov A.Yu. Konkurentnaya marshrutizatsiya transportnykh potokov postavshchikami uslug navigatsii [Competitive routing of traffic navigation systems]. *Upravleniye bolshimi sistemami [Control of large systems]*, 2014, No. 49, Pp. 129–147. (rus)
15. Zakharov V.V., Krylatov A.Yu. Konkurentnoye ravnovesiye Vardropa na transportnoy seti iz parallelnykh neodnorodnykh marshrutov [Wardrop user equilibrium on the transportation network of parallel inhomogeneous]. *Protsessy upravleniya i ustoychivost [Control Processes and Stability]*, 2014, Vol. 1, No. 17, Pp. 476–481. (rus)

РАЕВСКАЯ Анастасия Павловна — студентка кафедры моделирования энергетических систем факультета прикладной математики — процессов управления Санкт-Петербургского государственного университета.

199034, Россия, Санкт-Петербург, Университетская наб., д. 7-9.

E-mail: a.p.raevskaya@yandex.ru

RAEVSKAYA Anastasiya P. St. Petersburg State University.

199034, Universitetskaya emb., 7-9, St. Petersburg, Russia.

E-mail: a.p.raevskaya@yandex.ru

КРЫЛАТОВ Александр Юрьевич — ассистент кафедры моделирования энергетических систем факультета прикладной математики — процессов управления Санкт-Петербургского государственного университета, кандидат физико-математических наук.

199034, Россия, Санкт-Петербург, Университетская наб., д. 7-9.

E-mail: aykrylatov@yandex.ru

KRYLATOV Alexander Yu. *St. Petersburg State University.*

199034, Universitetskaya emb., 7-9, St. Petersburg, Russia.

E-mail: aykrylatov@yandex.ru



Программное обеспечение вычислительных, телекоммуникационных и управляющих систем

DOI: 10.5862/JCSTCS.236.5

УДК 004.89 +004.94

А.И. Титов, А.Д. Хомоненко

ВЫБОР ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ С ПОМОЩЬЮ АЛГОРИТМА ТАКАГИ-СУГЕНО НА ПРИМЕРЕ СИСТЕМ УПРАВЛЕНИЯ ПРОЕКТАМИ

A.I. Titov, A.D. Khomonenko

SOFTWARE SELECTION BY USING THE TAKAGI-SUGENO ALGORITHM ON THE EXAMPLE OF PROJECT MANAGEMENT SYSTEMS

Проведен анализ современных подходов к выбору программного обеспечения. Обоснована целесообразность применения алгоритма нечеткого вывода Такаги-Сугено для выбора программного обеспечения. Рассмотрен вариант модификации алгоритма, используемый для выбора программного обеспечения систем управления проектами (СУП). Предложена база правил для применяемого алгоритма нечеткого вывода. Приведен численный пример сравнения систем управления проектами Spider Project и Oracle Primavera

ПРИНЯТИЕ РЕШЕНИЙ; ВЫБОР ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ; НЕЧЕТКИЙ ВЫВОД; АЛГОРИТМ ТАКАГИ-СУГЕНО; СИСТЕМЫ УПРАВЛЕНИЯ ПРОЕКТАМИ.

This article analyzes the current approaches to selecting software and the expediency of applying the algorithm of fuzzy Takagi-Sugeno output for software selection. The main stages of fuzzy inference using the Takagi-Sugeno algorithm have been described, along with a variant modification of the algorithm used to select the software project management systems (PMS). A rule base for the fuzzy inference algorithm using fuzzy membership function for all key indicators has been established, with numerical examples of comparison provided (Spider Project and Oracle Primavera project management systems). In addition, this article describes the main characteristics of the indicators of project management systems and selected indicators for evaluation experts.

DECISION THEORY; SOFTWARE SELECTION; FUZZY LOGIC; TAKAGI-SUGENO ALGORITHM; PROJECT MANAGEMENT SYSTEM.

При разработке и использовании информационных систем важную роль играет правильный выбор программного обеспечения, такого как системы управления проектами, системы программирования, системы управления базами данных и др. При этом используются различные подходы к принятию решения по выбору программного обеспечения. В настоящее время это направление интенсивно развивается, но пока не найдено исчерпывающее решение.

Например, для систем управления проектами (СУП) крайне важны вопросы ин-

теграции с другими системами. Для проектов в сфере информационных технологий важна интеграция с системами управления версиями кода. Также часто возникает необходимость интеграции СУП с системами автоматизации бухгалтерского учета, причем сценарии подобной интеграции могут сильно различаться. Влияние на требования к интеграции оказывают многие факторы: сфера деятельности компании, совокупность используемых программных средств, объем данных.

Выбор программного обеспечения (ПО)

часто осложняется тем, что в каждой конкретной задаче сравнения необходимо выбрать показатели и указать, насколько они значимы в данной ситуации.

Выбор может проводиться как на основе оценок одного эксперта, так и на основе согласования оценок группы экспертов.

Групповое обсуждение специалистами компании может ускорить принятие решения, однако это трудоемкий процесс, и присутствие всех специалистов на общем обсуждении не всегда возможно. Кроме того, при организации группового обсуждения можно столкнуться с тем, что мнение одного эксперта или группы экспертов сильно отличается от мнения большинства.

Во многих случаях более продуктивным и менее трудоемким является процесс проведения сравнений методом индивидуального анкетирования. При этом необходимо предусмотреть вычисление и согласование общей оценки на основе полученных от специалистов данных.

При оценке большого числа вариантов ПО не все специалисты могут иметь компетенции по каждому варианту. В условиях недостаточной уверенности в оценке показателя большинство специалистов склоняется к усредненным оценкам, что может помешать выбрать оптимальное по качеству решение. В такой ситуации следует предусмотреть возможность обработки оценок по определенному сценарию, например, повышению значимости оценок с наиболее высокой или низкой оценкой.

Таким образом, метод сравнения ПО должен удовлетворять следующим требованиям:

- независимость методики расчета от количества и состава показателей;
- корректная обработка субъективных оценок экспертов;
- возможность использования весов показателей;
- возможность использования дополнительного параметра расчета для определения веса оценки эксперта;
- высокая наглядность и умеренная трудоемкость вычислений.

Среди существующих методов выбора наибольший интерес представляют мето-

ды принятия решения (парных сравнений Саати, продукционные системы с четкими правилами, алгоритмы нечеткого вывода, и др.) [1–3]. В настоящей статье с учетом отмеченных соображений выполняется обоснование использования алгоритма нечеткого вывода Такаги-Сугено [4, 5] для решения задач выбора программного обеспечения. Предлагается вариант модификации названного алгоритма, состоящий в учете весов показателей важности сравниваемых программных продуктов. С его помощью проводится сравнение систем управления проектами.

Характеристика подходов к выбору программного обеспечения

Задачи выбора ПО рассматривались в относительно небольшом количестве работ. В частности, в статье [6] дан сравнительный анализ программного инструментария для анализа и оценки рисков информационной безопасности, проведенной на уровне (есть/нет). В [7] предложен визуальный подход к сравнительной оценке математических пакетов (MathLab, Scientific WorkPlace, Maple и Mathcad) по ряду показателей, представленных в виде лучевых диаграмм. В докладе [8] на основе названного подхода предложен метод выбора программного обеспечения систем управления версиями. Указанные подходы обладают невысокой точностью при сравнении программных продуктов.

В диссертационной работе А.В. Ахаева [9] применительно к модулям программной системы 1С бухгалтерия исследованы вопросы выбора программных продуктов на основе онтологий, нечеткой меры и интеграла Шоке. Отмеченный подход обеспечивает получение более точных интегральных оценок сравниваемых программных продуктов, но характеризуется высокой трудоемкостью вычислений.

В работе [10] проведен выбор пакета программного обеспечения для имитационного моделирования с помощью нечеткого метода анализа иерархий. В [11] метод анализа иерархий предложено использовать для выбора одной из двух систем ERP (Enterprise Resource Planning). В статье



[12] на основе методологии анализа среды функционирования решена задача выбора программного обеспечения системы маршрутизации. При этом выполнено описание характеристик программных продуктов в ранговых шкалах.

В настоящее время разработано большое число методов принятия решения при нескольких критериях. Наиболее широкое применение получили интерактивные методы. Методы этого класса включают совокупность итераций, каждая из которых включает в себя этап анализа. Среди интерактивных методов решения многокритериальных задач перспективными являются методы, основанные на оценках решений. Интерактивные методы можно разделить на две группы: основанные на оценках функции предпочтений и основанные на парном сравнении решений (в частности, метод анализа иерархий Т. Саати).

Применительно к задаче выбора ПО использование метода анализа иерархий [3] имеет ряд существенных недостатков, в том числе необходимость построения большого числа матриц парных сравнений. Кроме того, метод позволяет получить жесткие оценки альтернатив и не допускает возможности принятия решения при нечетких данных. В работе [13] предложено обобщение метода анализа иерархий для принятия решения при неточных сравнениях с использованием теории Демпстера–Шейфера [14], позволяющее обрабатывать неполные и неточные предпочтения.

Несмотря на полученные результаты, указанные подходы к обоснованию выбора ПО, на наш взгляд, нуждаются в дальнейшем совершенствовании в направлении учета нечеткости данных показателей сравниваемых продуктов, снижения трудоемкости вычислений и повышения наглядности решений.

Характеристика показателей систем управления проектами

При разработке программных проектов в области информационных технологий важную роль играет правильный выбор СУП. Систематическое рассмотрение систем управления программными проектами,

решаемых задач и принципов построения содержится в [15, 16].

К системам управления проектами предъявляется большое количество требований, которые могут сильно различаться в зависимости от объема проекта, выбранной методологии и многого другого. При сравнении систем управления проектами (см. сайт www.capterra.com) выделяют следующие функциональные возможности:

- 1) управление бюджетом (Budget Management);
- 2) трассировка ошибок (Bug Tracking);
- 3) сотрудничество (Collaboration);
- 4) интеграция электронной почты (Email Integration);
- 5) общий доступ к файлам (File Sharing);
- 6) диаграммы Ганта (Gantt Charts);
- 7) целевое управление (Idea Management);
- 8) управление выпусками (Issue Management);
- 9) отслеживание этапов (Milestone Tracking);
- 10) отслеживание процента завершений (Percent-Complete Tracking);
- 11) управление портфелем (Portfolio Management);
- 12) планирование проекта (Project Planning);
- 13) управление требованиями (Requirements Management);
- 14) управление ресурсами (Resource Management);
- 15) отслеживание состояния (Status Tracking);
- 16) управление задачами (Task Management);
- 17) тестирование / управления качеством (Testing / QA Management);
- 18) отслеживание времени и расходов (Time & Expense Tracking).

Более обобщенно можно сказать, что современные системы управления проектами должны включать в себя приложения для решения следующих задач: планирования задач; составления расписания; управления бюджетом; распределения ресурсов; общения между участниками проекта; документирования.

Как видим, выбор системы управления проектами основывается на большом количестве показателей, которые могут основываться на важнейших параметрах системы, например, таких как поддержка мобильных платформ, возможность интеграции со сторонними системами и др. Кроме того, здесь могут учитываться субъективные оценки экспертов, например, стоимость внедрения, быстрота обучения и т. д.

Итоговая стоимость ПО основывается на различных факторах, таких как:

- модель использования;
- количество необходимых лицензий;
- комплект поставки.

Оценка эксперта по этим факторам может существенно различаться в зависимости от потребностей проектной команды и существующих условий. Например, если внутренняя инфраструктура компании не позволяет развернуть ПО и возникает потребность в облачной модели использования, то предпочтения эксперта будут отданы варианту с более высокой ценой за одну лицензию, но с нужной моделью использования. Оценивание функциональности может включать еще большее число факторов, с еще большим количеством внешних условий. Поэтому для сравнения подходов к оценке все факторы, наиболее влияющие на оценку эксперта, были разделены на несколько основных групп показателей. Каждая группа показателей может получить оценку либо по сумме оценок всех показателей, либо как субъективную общую оценку по всей группе. Полученные группы показателей представлены в табл. 1.

На практике, кроме этих показателей,

выбор системы управления проектами может основываться на других показателях, например, информации о поставщике решения. В этом случае требуется анализ по критериям, учитывающим следующие показатели:

- устойчивость компании на рынке;
- уровень продаж;
- качество технической поддержки;
- быстрота и качество поставки продукта.

Поскольку целью настоящей статьи является получение методики сравнения именно конечных продуктов, то при сравнении СУП нами не учитываются показатели, задаваемые поставщиками программных систем управления проектами.

По данным сайта www.capterra.com, в число наиболее популярных СУП входят следующие системы: MS Project, Atlassian, Wrike, Podio, Basecamp, Teamwork и др. При этом отбор наиболее популярных программ управления проектами проводился с различных позиций и при различных параметрах (со стороны пользователей, с учетом разного числа пользователей, со стороны разработчиков и др.).

В задаче выбора СУП можно выделить две категории показателей: четкие и нечеткие. Четкая часть показателей, например, количество пользователей в СУП и т. п., может разрешаться с помощью МАИ. Нечеткие показатели СУП, например, удобство использования, функциональность и т. п., целесообразно учитывать на основе одного из алгоритмов нечеткого вывода, Мамдани, Такаги-Сугено, Ларсена и др. [17]. Сравнительный анализ алгоритмов показывает, что в нашем случае целесообразно использовать алгоритм нечеткого вывода Такаги-Сугено,

Таблица 1

Группы показателей систем управления проектами

Показатель	Описание	Обозначение
Стоимость	Оценка стоимости приобретения и сопровождения	A_1
Скорость обучения	Время обучения работе с системой	A_2
Функциональность	Функциональная глубина	A_3
Интегрируемость	Возможности интеграции с внешними системами	A_4

отличающийся невысокой трудоемкостью и приемлемой точностью.

Характеристика алгоритма Такаги-Сугено

Алгоритм Такаги-Сугено описывается следующим образом [4, 5, 17]: для нечеткого множества A используется обозначение функции принадлежности $A(x)$, $x \in X$. Предполагается, что правило нечеткого вывода (импликация) R имеет следующий формат:

R : ЕСЛИ $f(x_1 \text{ есть } A_1, \dots, x_k \text{ есть } A_k)$,
ТО $y = g(x_1, \dots, x_k)$,

где y — переменная заключения, значение которой выводится; $x_1 - x_k$ — переменные предпосылки, значения которых также присутствуют в заключении; f — логическая функция, связывающая предпосылки в условии; g — функция, определяющая значение y при условии, что $x_1 - x_k$ удовлетворяют исходным условиям правила вывода.

Формирование базы правил систем нечеткого вывода и фаззификация входных переменных. В базе правил используются только правила нечетких продукций (предположим, что база состоит из двух правил с двумя входами и одним выходом) в форме:

Правило 1 : ЕСЛИ x_1 есть A_{11} И x_2
есть A_{12} , ТО $y = c_{11}x_1 + c_{12}x_2 + c_{10}$,

Правило 2 : ЕСЛИ x_1 есть A_{21} И x_2
есть A_{22} , ТО $y = c_{21}x_1 + c_{22}x_2 + c_{20}$.

Здесь c_{ij} — весовые коэффициенты компонентов вектора, c_{10} — смещение. При этом значение выходной переменной y в заключении определяется как действительное число.

Агрегирование подусловий в нечетких правилах продукций. Для нахождения степени истинности условий всех правил нечетких продукций, как правило, используется логическая операция $\min$ -конъюнкции:

$$\alpha_1 = \min\{\mu_{A_{11}}(x_1'), \mu_{A_{12}}(x_2')\},$$

$$\alpha_2 = \min\{\mu_{A_{21}}(x_1'), \mu_{A_{22}}(x_2')\}.$$

Для выполнения агрегирования могут использоваться и другие логические опера-

ции. Правила, степень истинности условий которых отлична от нуля, считаются *активными* и используются для дальнейших расчетов.

Активизация подзаключений в нечетких правилах продукций. Во-первых, с использованием $\min$ -активизации, как и в алгоритме Мамдани, находятся значения степеней истинности всех заключений правил нечетких продукций. Во-вторых, осуществляется расчет обычных (не нечетких) значений выходных переменных каждого правила. Это выполняется с использованием формул для заключения:

$$y_1' = c_{11}x_1' + c_{12}x_2' + c_{10},$$

$$y_2' = c_{21}x_1' + c_{22}x_2' + c_{20}.$$

Здесь вместо x_1 и x_2 подставляются значения входных переменных до этапа фаззификации.

Аккумуляция заключений нечетких правил продукций. Фактически отсутствует, поскольку расчеты осуществляются с обычными действительными числами y_j .

Дефаззификация выходных переменных. Используется модифицированный вариант в форме метода центра тяжести для одноточечных множеств:

$$y' = (\alpha_1 y_1' + \alpha_2 y_2') / (\alpha_1 + \alpha_2).$$

При этом не требуется проведения предварительного аккумуляирования активизированных заключений отдельных правил.

Реализация алгоритма Такаги-Сугено для выбора СУП

При одинаковой значимости различных показателей сравниваемых программных средств (в нашем случае СУП) общая оценка i -й СУП при использовании алгоритма Такаги-Сугено вычисляется по формуле [4, 17]:

$$y_i = \sum_{j=1}^n x_{ij} \cdot c(x_{ij}).$$

На практике состав и значимость различных показателей существенно зависит от категории (типа) сравниваемых программных средств (операционных систем, систем программирования, СУП и др.). При этом

вполне очевидным решением представляется необходимость для заданной категории сравниваемых программных средств выполнить экспертную оценку значимости отдельных показателей (z_j) и нормировку так, чтобы выполнялось условие вида:

$$\sum_{j=1}^n z_j = 1.$$

В результате общая оценка i -й СУП будет вычисляться по формуле вида:

$$y_i = \sum_{j=1}^n x_{ij} \cdot c(x_{ij}) z_j.$$

Таким образом, правило оценки i -го программного продукта можно представить в общем виде следующим образом:

ЕСЛИ $c_{i1}(x_{i1}) = A_{i1}(x_{i1})$ И $c_{i2}(x_{i2}) = A_{i2}(x_{i2})$... И $c_{in}(x_{in}) = A_{in}(x_{in})$,

$$\text{ТО } y_i = \sum_{j=1}^n x_{ij} \cdot c(x_{ij}) z_j.$$

Функции принадлежности выбраны таким образом, что наименьший вес имеют средние значения. Вид функций принадлежности для весов показателей при условии использования двух вариантов оценки показателей (A_{ij} высокое (High) и низкое (Low)) приведен на рис. 1.

Вычисляя значение функции $c(x)$ для значения x_{ij} показателя по каждой СУП, получаем значение веса показателя c_{ij} .

Пример правил и обоснования выбора СУП

С учетом вида функций принадлежно-

сти для показателей СУП, приведенных на рис. 1, правила для выбора с четырьмя показателями имеют вид:

$$R^1 : \text{ЕСЛИ } x_{i1} = \text{низкое И ... И } x_{i4} = \text{низкое, ТО } y_i^1 = \sum_{j=1}^4 c_{ij}(x_{ij}) z_j.$$

$$R^2 : \text{ЕСЛИ } x_{i1} = \text{низкое И ... И } x_{i4} = \text{высокое, ТО } y_i^2 = \sum_{j=1}^4 c_{ij}(x_{ij}) z_j.$$

...

$$R^{16} : \text{ЕСЛИ } x_{i1} = \text{высокое И ... И } x_{i4} = \text{высокое, ТО } y_i^{16} = \sum_{j=1}^4 c_{ij}(x_{ij}) z_j.$$

Схематично перебор посылок — термов низкое (Low) и высокое (High) в приведенных правилах для трех показателей сравниваемых СУП может быть представлен, как показано в табл. 2.

Легко показать, что количество правил в базе правил для одного сравниваемого программного продукта (СУП) есть c^n , где n — число показателей, используемых при выборе. В частности, при четырех показателях мы имеем 16 правил.

Для демонстрации рассматриваемого подхода ограничимся сравнением показателей для Oracle Primavera и Spider Project.

Расчет итоговой оценки по каждому продукту проводится при помощи модифицированного метода Такаги-Сугено. На первом этапе формируется база правил для модифицированного алгоритма.

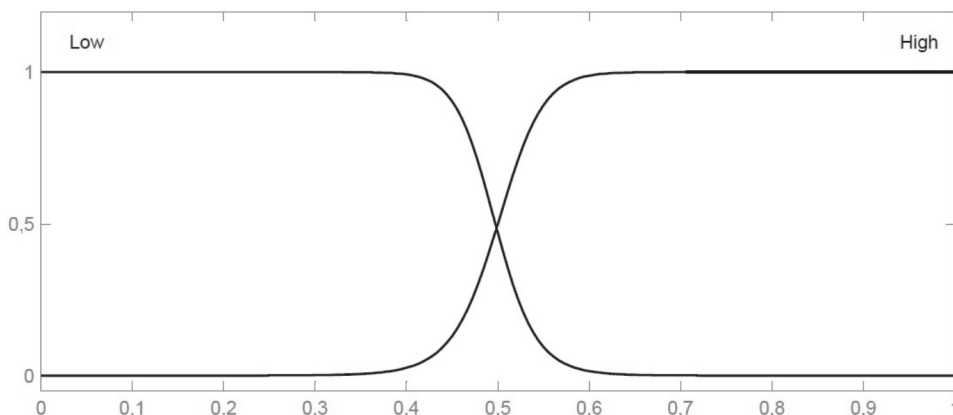


Рис. 1. Функции принадлежности для весов показателей

Таблица 2

Схема перебора посылок термов в правилах

	A ₁	A ₂	A ₃	A ₄
1	Low	Low	Low	Low
2				High
3			High	Low
4				High
5		High	Low	Low
6				High
7			High	Low
8				High
9	High	Low	Low	Low
10				High
11			High	Low
12				High
13		High	Low	Low
14				High
15			High	Low
16				High

Формула вычисления оценки по i -му правилу в общем случае имеет вид:

$$y_i = \sum_{j=1}^n x_{ij} \cdot c(x_{ij})z_j,$$

где n – число показателей, используемых при выборе; y_i – суммарная оценка качества продукта по i -му правилу; x_{ij} – оценка j -го показателя i -го правила; c_{ij} – вес j -го показателя i -го правила; z_j – значимость j -го показателя для всех типов СУП.

После вычисления оценки по всем правилам определяется итоговая оценка:

$$y = \frac{\sum_{i=1}^m c_{\min}(y_i)y_i}{\sum_{i=1}^m c_{\min}(y_i)},$$

где $c_{\min}(y_i)$ определяется как результат выполнения логической операции min-конъюнкции всех показателей в i -м правиле.

В табл. 3 приведены значения значимостей показателей СУП, определенные на основе экспертной оценки. Наиболее значимыми показателями в нашем примере являются стоимость и функциональность.

В общем случае набор значений значимости показателей должен быть нормированным к единице:

$$\sum_{j=1}^n z_j = 1.$$

После расчета всех параметров вычисляется оценка качества для каждой альтернативы. Согласованные оценки экспертов по выбранным показателям представлены в табл. 4.

Таблица 3

Значимости показателей

j	A ₁	A ₂	A ₃	A ₄
z_j	0,3	0,2	0,3	0,2

Таблица 4

Оценки программных продуктов по выбранным показателям

Продукт	A ₁	A ₂	A ₃	A ₄
Oracle Primavera	0,4	0,5	0,9	0,5
Spider Project	0,6	0,6	0,6	0,4

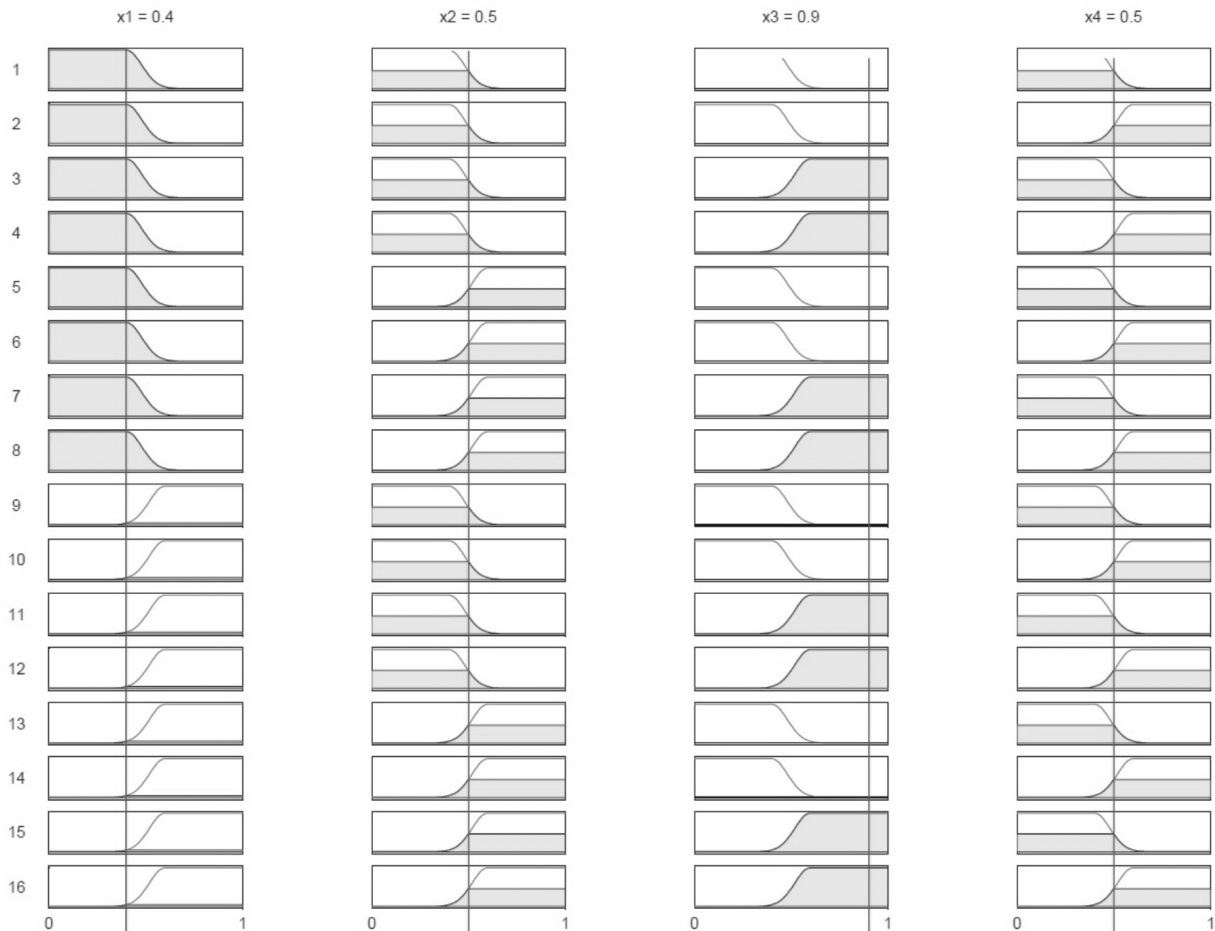


Рис. 2. Значения функций принадлежности для оценок Oracle Primavera

На основе оценок по показателям вычисляются значения функций принадлежности для каждого правила. Пример вывода значений функций принадлежности для Oracle Primavera представлен на рис. 2.

Для примера приведем результат расчета по первому правилу для первого продукта:

$$y_{\text{primavera}} = \sum_{j=1}^4 x_{ij} \cdot c(x_{ij})z_j = 0,4 \cdot 0,9886 \cdot 0,3 + \\ + 0,5 \cdot 0,5 \cdot 0,2 + 0,9 \cdot 0,0000078 \cdot 0,3 + \\ + 0,5 \cdot 0,5 \cdot 0,2 = 0,218634.$$

Результаты вычислений по всей базе правил для Oracle Primavera представлены в табл. 5.

После вычислений по всем правилам для каждого продукта производится вычисление подусловий и дефаззификация выходной переменной:

$$y_{\text{primavera}} = \frac{\sum_{i=1}^m c_{\min}(y_i)y_i}{\sum_{i=1}^m c_{\min}(y_i)} = \\ = \frac{1,015176}{2,101663} = 0,483035.$$

$$y_{\text{spider}} = \frac{\sum_{i=1}^m c_{\min}(y_i)y_i}{\sum_{i=1}^m c_{\min}(y_i)} = \\ = \frac{0,606417}{1,176} = 0,515661.$$

Из приведенных результирующих оценок сравниваемых СУП можно сделать вывод, что предпочтение следует отдать СУП Spider Project. Итоговая оценка по выбранным показателям для этого продук-

Таблица 5
Результаты вычисления оценок

	y	$c(y)_{\min}$
1	0,218634	0,00000784
2	0,218634	0,00000784
3	0,488632	0,5
4	0,488632	0,5
5	0,218634	0,00000784
6	0,218634	0,00000784
7	0,488632	0,5
8	0,488632	0,5
9	0,10305	0,00000784
10	0,10305	0,00000784
11	0,373048	0,0254
12	0,373048	0,0254
13	0,10305	0,00000784
14	0,10305	0,00000784
15	0,373048	0,0254
16	0,373048	0,0254

та оказалась выше.

Для сравнения приведем пример расчета без использования функции весов показателей, где входными параметрами для расчета являются оценки эксперта и значимость показателя:

$$y_{\text{primavera}} = \sum_{j=1}^4 x_{ij} \cdot z_j = 0,4 \cdot 0,3 + 0,5 \cdot 0,2 + 0,9 \cdot 0,3 + 0,5 \cdot 0,2 = 0,59.$$

$$y_{\text{spider}} = \sum_{j=1}^4 x_{ij} \cdot z_j = 0,6 \cdot 0,3 + 0,6 \cdot 0,2 + 0,6 \cdot 0,3 + 0,4 \cdot 0,2 = 0,56.$$

Как видим, при отсутствии использования функции весов показателей сравниваемых СУП итоговая оценка заметно отличается. Более высокую оценку в данном варианте расчета получает Oracle Primavera.

На основе использования нечеткого вывода достигнута итоговая оценка, в которой больший вес имеют значения показателей с более выраженной оценкой, т. е. выше

или ниже среднего значения. Этот эффект можно объяснить выбором функций принадлежности (рис. 1) для показателей. Если использовать функции принадлежности с максимумом в средних значениях диапазона значений, то показатели со средними оценками имеют более выраженное влияние на итоговый результат.

Чтобы подтвердить влияние средних оценок на результат вычисления с использованием функций принадлежности, выполним оценку по измененному набору значений для показателей. В частности, для продукта Spider Project вместо экспертной оценки установлены средние значения для двух показателей, а у Oracle Primavera одно из средних значений уменьшено (табл. 6).

При таких входных значениях итоговые оценки без использования нечеткого вывода определяются как

$$y_{\text{primavera}} = \sum_{j=1}^4 x_{ij} \cdot z_j = 0,4 \cdot 0,3 + 0,4 \cdot 0,2 + 0,8 \cdot 0,3 + 0,5 \cdot 0,2 = 0,54.$$

$$y_{\text{spider}} = \sum_{j=1}^4 x_{ij} \cdot z_j = 0,5 \cdot 0,3 + 0,5 \cdot 0,2 + 0,7 \cdot 0,3 + 0,4 \cdot 0,2 = 0,54.$$

Результат вычисления с использованием нечеткого вывода имеет вид:

$$y_{\text{primavera}} = \frac{\sum_{i=1}^m c_{\min}(y_i) y_i}{\sum_{i=1}^m c_{\min}(y_i)} = \frac{0,596609}{1,167062} = 0,511206.$$

$$y_{\text{spider}} = \frac{\sum_{i=1}^m c_{\min}(y_i) y_i}{\sum_{i=1}^m c_{\min}(y_i)} = \frac{0,86401}{2,1112} = 0,409251.$$

Таблица 6
Измененные оценки программных продуктов
по выбранным показателям

Продукт	A_1	A_2	A_3	A_4
Oracle Primavera	0,4	0,4	0,8	0,5
Spider Project	0,5	0,5	0,7	0,4

Обсуждение результатов и трудоемкости подхода

По результатам численных экспериментов можно отметить, что с использованием нечеткого вывода итоговая оценка больше соответствует требованиям к методу обоснования выбора продуктов. Предложенный подход является лучшим, поскольку в нем учитывается разность весов оценки показателей сравниваемых программных продуктов.

Для наглядности нами рассмотрен пример обоснования выбора из двух вариантов СУП с использованием четырех показателей. При этом отметим, что в статьях [7–12] выбор ПО для каждого класса продуктов выполнялся также с использованием относительно небольшого числа показателей (6–8).

Применительно к СУП число используемых показателей может быть заметно большим. При этом возникает дилемма между стремлением к полноте учета всех показателей выбора СУП и желанием избежать излишней трудоемкости вычислений. Напомним, что число правил, используемых в предлагаемом подходе, определяется величиной $r = 2^n$, где n — число показателей.

При принятии решения по выбору СУП достаточно большое время, на наш взгляд, затрачивается на сбор и согласование мнений экспертов. Что касается затрат времени на вычисления с помощью алгоритма Такаги-Сугено, то для примера приведем зависимость числа правил r и времени вычисления t от числа показателей (см. табл. 7). Расчет проводился с помощью разработанного авторами комплекса программ на компьютере с процессором Intel Core i5 с тактовой частотой 2,79 ГГц.

Как видно из данных табл. 7, при увеличении числа сравниваемых показателей наблюдается существенный рост времени вычисления t . В общем случае в практических ситуациях обоснования выбора СУП целесообразно использовать некоторое обосно-

Таблица 7

Пример зависимости r и t от n

n	4	10	20	30
r	16	1024	1048576	1073741824
t , с	0,001	0,002	0,769	1403,387

ванное число правил. Задача определения этого числа правил требует отдельного рассмотрения.

Предложенный подход (несмотря на наличие ряда фирменных сайтов, предлагающих готовые решения СУП), на наш взгляд, может быть полезен для широкого круга разработчиков программных проектов, у которых нет полюбившегося набора инструментальных средств, находящихся в длительном применении.

Выбор программного обеспечения систем управления проектами на основе применения предложенного модифицированного алгоритма Такаги-Сугено, на наш взгляд:

по сравнению с методом анализа иерархий Саати и его обобщением на основе теории Демпстера-Шейфера [13, 14] позволяет снизить трудоемкость вычислений;

по сравнению с методом предпочтений [8] дает возможность в явной мере учесть нечеткость исходных данных;

по сравнению с подходом, предложенным в [9] на основе онтологий, нечеткой меры и интеграла Шоке, достигается большая наглядность получаемого вывода.

Дальнейшие исследования целесообразно продолжить в направлении практического применения предложенного подхода для выбора других программных продуктов; выработки предложений по формализованному проведению экспертной оценки ряда сложно определяемых параметров, например «Функциональность», «Стоимость» и др.; обоснования состава и числа правил, используемых при выборе программного обеспечения.

СПИСОК ЛИТЕРАТУРЫ

1. Шварц Д.Т. Интерактивные методы решения задачи многокритериальной оптимизации.

Обзор // Наука и образование. МГТУ им. Н.Э. Баумана. 2013. С. 245–264. [Электрон-

ный ресурс] // URL: <http://technomag.bmstu.ru/doc/547747.html>.

2. **Лотов А.В., Поспелова И.И.** Многокритериальные задачи принятия решений: Учеб. пособие. М.: МАКС Пресс, 2008. 197 с.

3. **Saaty Th.L.** The Hierarchon: A Dictionary of Hierarchies. Pittsburgh, Pennsylvania: RWS Publications, 1992. 496 p.

4. **Takagi T., Sugeno M.** Fuzzy Identification of Systems and Its Applications to Modeling and Control // IEEE Trans. Systems, Man, and Cybernetics. 1985. Vol. 15. No. 1. Pp. 116–132.

5. **Ma J., Lu J., Zhang G.Q.** Decider: A fuzzy multi-criteria group decision support system // Knowledge-Based Systems. 2010. Vol. 23. Pp. 23–31.

6. **Баранова Е.К., Чернова М.В.** Сравнительный анализ программного инструментария для анализа и оценки рисков информационной безопасности // Проблемы информационной безопасности. Компьютерные системы. 2014. № 4. С. 160–168.

7. **Брякалов Г.А., Войцеховский С.В., Воробьев Е.Г., Гофман В.Э., Гридин В.В., Дрюков Ю.П., Замула А.А., Захаров А.И., Компаниец Р.И., Липецких А.Г., Рыжиков Ю.И., Хомоненко А.Д., Цыганков В.М.** Основы современных компьютерных технологий: Учебник. СПб.: Корона принт, 2005. 672 с.

8. **Горшков А.В., Хомоненко А.Д.** Метод выбора программного обеспечения систем управления версиями // Анализ и прогнозирование систем управления в промышленности и на транспорте. Матер. XVI междунар. науч.-практ. конф. молодых ученых, студентов и аспирантов. СПб., 2015.

9. **Ахаев А.В.** Методика, модели и алгорит-

мы выбора программных продуктов на основе онтологии и нечеткой меры: Дисс. ... канд. техн. наук. Томск, 2014. 19 с.

10. **Azadeh A., Shirkouhi S.N., Rezaie K.** A robust decision-making methodology for evaluation and selection of simulation software package // Internat. J. of Advanced Manufacturing Technology. 2010. Vol. 47. Pp. 381–393.

11. **Karaarslan N., Gundogar E.** An application for modular capability-based ERP software selection using AHP method // Internat. J. of Advanced Manufacturing Technology. 2009. Vol. 42. Pp. 1025–1033.

12. **Smirlis Y.G., Zeimpekis V., Kaimakamis G.** Data envelopment analysis models to support the selection of vehicle routing software for city logistics operations // Operational Research. 2012. Vol. 12. Pp. 399–420.

13. **Уткин Л.В., Симанова Н.В.** Обобщение метода анализа иерархий для принятия решения при неточных сравнениях с использованием теории Демпстера-Шейфера // Изв. ОрелГТУ. Информационные системы и технологии. 2007. № 4/268(535). С. 223–227.

14. **Dempster A.P.** Upper and Lower Probabilities Induced by a Multi-Valued Mapping // Annales of Mathematical Statistics, 1967. Vol. 38. Pp. 325–339.

15. A Guide to the Project Management Body of Knowledge (PMBOK Guide). 5th ed. 2014. 589 p.

16. **Фатрелл Р.Т., Шафер Д.Ф., Шафер Л.И.** Управление программными проектами. Пер. с англ. М., СПб., Киев: ИД «Вильямс», 2004. 1136 с.

17. **Борисов В.В., Круглов В.В., Федулов А.С.** Нечеткие модели и сети. М.: Горячая линия-Телеком, 2007. 284 с.

REFERENCES

1. **Shvarts D.T.** Interaktivnyye metody resheniya zadachi mnogokriterialnoy optimizatsii. Obzor [Interactive methods for solving multi-objective optimization problem. Review]. *Nauka i obrazovaniye. MGTU imeni N.E. Bauman* [Science and Education of Bauman MSTU], 2013, Pp. 245–264. Available: <http://technomag.bmstu.ru/doc/547747.html>. (rus)

2. **Lotov A.V., Pospelova I.I.** *Mnogokriterialnyye zadachi prinyatiya resheniy* [Multicriteria decision making problems], Moscow: MAKS Press Publ., 2008, 197 p. (rus)

3. **Saaty Th.L.** The Hierarchon: A Dictionary of Hierarchies. Pittsburgh, Pennsylvania: RWS Publications, 1992, 496 p.

4. **Takagi T., Sugeno M.** Fuzzy Identification of Systems and Its Applications to Modeling and Control. *IEEE Trans. Systems, Man, and Cybernetics*, 1985, Vol. 15, No. 1, Pp. 116–132.

5. **Ma J., Lu J., Zhang G.Q.** Decider: A fuzzy multi-criteria group decision support system. *Knowledge-Based Systems*, 2010, No. 23, Pp. 23–31.

6. **Baranova Ye.K., Chernova M.V.** Sravnitelnyy analiz programmnoy instrumentariya dlya analiza i otsenki riskov informatsionnoy bezopasnosti [Comparative analysis of programming tools for cybersecurity risk assessment]. *Problemy informatsionnoy bezopasnosti. Kompyuternyye sistemy* [Information security problems. Computer Systems], 2014, No. 4, Pp. 160–168. (rus)

7. **Bryakalov G.A., Voytsekhovskiy S.V., Vorobyev Ye.G., Gofman V.E., Gridin V.V., Dryukov Yu.P., Zamula A.A., Zakharov A.I., Kompaniyets R.I., Lipetskikh A.G., Ryzhikov Yu.I., Khomonenko A.D., Tsygankov V.M.** *Osnovy sovremennykh kompyuternykh tekhnologiy* [Fundamentals of modern computer technologies]. St. Petersburg: Korona print

Publ., 2005, 672 p. (rus)

8. **Gorshkov A.V., Khomonenko A.D.** Metod vybora programmnoho obespecheniya sistem upravleniya versiyami [The method of selection of software version control systems]. *Materialy XVI Mezhdunarodnoy nauchno-prakticheskoy konferentsii molodykh uchenykh, studentov i aspirantov Analiz i prognozirovaniye sistem upravleniya v promyshlennosti i na transporte [XVI Materials International scientific-practical conference of young scientists, students and graduate students Analysis and Forecasting Management Systems in Industry and Transport]*. St. Petersburg, 2015. (rus)

9. **Akhayev A.V.** Metodika, modeli i algoritmy vybora programmnykh produktov na osnove ontologii i nechetkoy mery [Abstract of dissertation for the degree of candidate of technical sciences, made the topic: *Methods, models and algorithms for the selection of software products based on ontology and fuzzy measures*], Tomsk, 2014, 19 p. (rus)

10. **Azadeh A., Shirkouhi S.N., Rezaie K.** A robust decision-making methodology for evaluation and selection of simulation software package, *International Journal of Advanced Manufacturing Technology*, 2010, Vol. 47, Pp. 381–393.

11. **Karaarslan N., Gundogar E.** An application for modular capability-based ERP software selection using AHP method, *International Journal of Advanced Manufacturing Technology*, 2009, Vol. 42, Pp. 1025–1033.

12. **Smirlis Y.G., Zeimpekis V., Kaimakamis G.** Data envelopment analysis models to support the selection of vehicle routing software for city logistics operations, *Operational Research*, 2012, Vol. 12, Pp. 399–420.

13. **Utkin L.V., Simanova N.V.** Obobshcheniye metoda analiza iyerarkhiy dlya prinyatiya resheniya pri netochnykh sravneniyakh s ispolzovaniyem teorii Dempstera-Sheyfera [The generalization of the analytic hierarchy process for deciding when inaccurate comparisons using the Dempster-Shafer theory]. *Izvestiya OrelGTU. Informatsionnyye sistemy i tekhnologii [Izvestiya OrelSTU. Information systems and technologies]*, 2007, No. 4/268(535), Pp. 223–227. (rus)

14. **Dempster A.P.** Upper and Lower Probabilities Induced by a Multi-Valued Mapping, *Annales of Mathematical Statistics*, 1967, Vol. 38, Pp. 325–339.

15. *A Guide to the Project Management Body of Knowledge (PMBOK Guide)*, 5th ed., 2014, 589 p.

16. **Fatrell R.T., Shafer D.F., Shafer L.I.** *Upravleniye programmnyimi proyektami. Dostizheniye optimalnogo kachestva pri minimume zatrat* [Quality Software Project Management]. Moscow-St. Petersburg-Kiev: Publishing House “Vilyams”, 2004, 1136 p. (rus)

17. **Borisov V.V., Kruglov V.V., Fedulov A.S.** *Nechetkiye modeli i seti* [Fuzzy models and networks]. Moscow: Goryachaya liniya-Telekom Publ., 2007, 284 p. (rus)

ТИТОВ Александр Игоревич — аспирант кафедры информационных и вычислительных систем факультета автоматизации и интеллектуальных технологий Петербургского государственного университета путей сообщения Императора Александра I.

190031, Россия, Санкт-Петербург, Московский пр., д. 9.

E-mail: titovvvv@rambler.ru

ТИТОВ Aleksandr I. *Petersburg State Transport University.*

190031, Moskovsky Ave. 9, St. Petersburg, Russia.

E-mail: titovvvv@rambler.ru

ХОМОНЕНКО Анатолий Дмитриевич — заведующий кафедрой информационных и вычислительных систем факультета автоматизации и интеллектуальных технологий Петербургского государственного университета путей сообщения Императора Александра I, доктор технических наук.

190031, Россия, Санкт-Петербург, Московский пр., д. 9.

E-mail: khomon@mail.ru

KHOMONENKO Anatoly D. *Petersburg State Transport University.*

190031, Moskovsky Ave. 9, St. Petersburg, Russia.

E-mail: khomon@mail.ru



DOI: 10.5862/JCSTCS.236.6

УДК 004.4`233:004.382.2

А.В. Левченко, С.А. Фёдоров

ИНСТРУМЕНТ ДИНАМИЧЕСКОГО ДВОИЧНОГО АНАЛИЗА ПРОСТРАНСТВЕННО-ВРЕМЕННОЙ ЛОКАЛИЗАЦИИ ПРИЛОЖЕНИЯ

A.V. Levchenko, S.A. Fyodorov

DYNAMIC BINARY INSTRUMENTATION TOOL FOR DATA LOCALITY ANALYSIS

Наличие проблемы стены памяти (Memory Wall Problem) приводит к простоям процессора в ожидании данных из памяти в результате неудовлетворительной пространственно-временной локализации приложения. Средства динамического двоичного анализа важны для оценки пространственно-временного поведения приложения с целью смягчения последствий этой проблемы. Для исследования эффективности приложения необходимо получить исчерпывающий профиль доступа приложения к памяти, позволяющий дать численную и визуальную оценку локализации доступа. Процесс профилирования имеет существенные издержки, возникающие при низкоуровневом инструментировании и сохранении данных профиля, способные исказить оценку реальной локализации. Описан программный инструмент для динамического двоичного анализа пространственно-временной локализации приложения, ориентированный на практическое применение методологии Арех для глубокого анализа приложений. Инструмент профилирования *tslmap* разработан на основе платформы Valgrind, доступной на большинстве современных высокопроизводительных платформ. Показано, что разработанный инструмент позволяет корректно оценить локализацию приложения, с минимальными издержками профилирования получить и визуализировать профили на архитектурах с распределенной памятью.

ВРЕМЕННАЯ ЛОКАЛИЗАЦИЯ; ВЫСОКОПРОИЗВОДИТЕЛЬНЫЕ ВЫЧИСЛЕНИЯ; ДИНАМИЧЕСКИЙ ДВОИЧНЫЙ АНАЛИЗ; ПРОБЛЕМА СТЕНЫ ПАМЯТИ; ПРОСТРАНСТВЕННАЯ ЛОКАЛИЗАЦИЯ; ИНСТРУМЕНТ ПРОФИЛИРОВАНИЯ ПРИЛОЖЕНИЙ.

This work addresses the fundamental challenges covered in a number of papers on high performance computing. One of the most important issues is the Memory Wall problem. Standard memory access patterns can cause sparse temporal and spatial locality and thus performance degradation. It is important to get the profile of an application's memory access to analyze its performance. The main purpose of this study is the dynamic binary analysis tool for data locality analysis. The *tslmap* profiler was developed for analyzing the performance behavior of applications. It is based on the Valgrind platform intended for creating tools for binary analysis of applications. The *tslmap* tool allows to get and visualize the profiles helping to measure the metrics of temporal and spatial locality by dynamic analysis of a program's memory references. Experiments were performed on cluster architectures for plotting the performance of client application as a function of its spatial and temporal scores for certain systems.

APEX-MAP; DYNAMIC BINARY ANALYSIS; HIGH PERFORMANCE COMPUTING; MEMORY WALL PROBLEM; PERFORMANCE SURFACE; PROFILER; SPATIAL LOCALITY; TEMPORAL LOCALITY; VALGRIND.

Важнейшим вопросом современных высокопроизводительных вычислений с использованием архитектуры с распределенной памятью является низкая реальная производительность системы по сравнению

с ее пиковой производительностью. Доминирующей причиной деградации производительности остается *проблема стены памяти* (Memory Wall Problem), которая заключается в недостаточно быстром вы-

полнении операций в оперативной памяти по сравнению со временем выполнения арифметико-логических операций процессорами. Масштабы этой проблемы растут за счет применения современных высокопроизводительных машин с глубокой иерархией памяти. В результате непонимание особенностей пространственно-временного поведения приложения приводит к непредсказуемым издержкам операций с памятью. Одним из способов анализа проблемы стены памяти является оценка пространственной и временной локализации (ПВЛ) обращений анализируемого приложения к памяти. К сожалению, на сегодняшний день отсутствует свободное программное обеспечение для корректной оценки и визуализации ПВЛ, а издержки такого анализа на высокопроизводительных системах не определены.

На основании требований, сформулированных в настоящей работе, разработан профилировщик ПВЛ *tslmap*. Этот инструмент осуществляет динамический двоичный анализ (Dynamic Binary Analysis – DBA) операций доступа к оперативной памяти посредством функций низкоуровневого инструментария, предоставляемых ядром платформы Valgrind, доступной на подавляющем большинстве современных суперкомпьютеров.

Проведено сравнение издержек профилирования для современных средств DBA и показано, как получение данных, относящихся исключительно к работе анализируемого приложения, позволяет оценить его ПВЛ точнее и добиться существенного снижения доминирующих издержек профилирования.

Предпосылки разработки *tslmap*

Профилировщик *tslmap*, который является целью данной статьи, основан на практическом применении методологии Apex [1, 2], используемой сегодня для анализа производительности приложений современных высокопроизводительных систем. Для построения профиля и оценки ПВЛ приложения в настоящей работе используется подход, предложенный авторами Apex. *Пространственная локализация* (Spatial Locality – SL) определяется, как тенденция приложений обращаться к тем адресам в памяти, которые находятся рядом с другими, недавно использовавшимися адресами. *Временная локализация* (Temporal Locality – TL) определяется, как тенденция приложения обращаться к тем же адресам в памяти, к которым это приложение обращалось в последнее время [3–5].

В качестве примера на рис. 1 представлен профиль региона памяти кластера, ха-

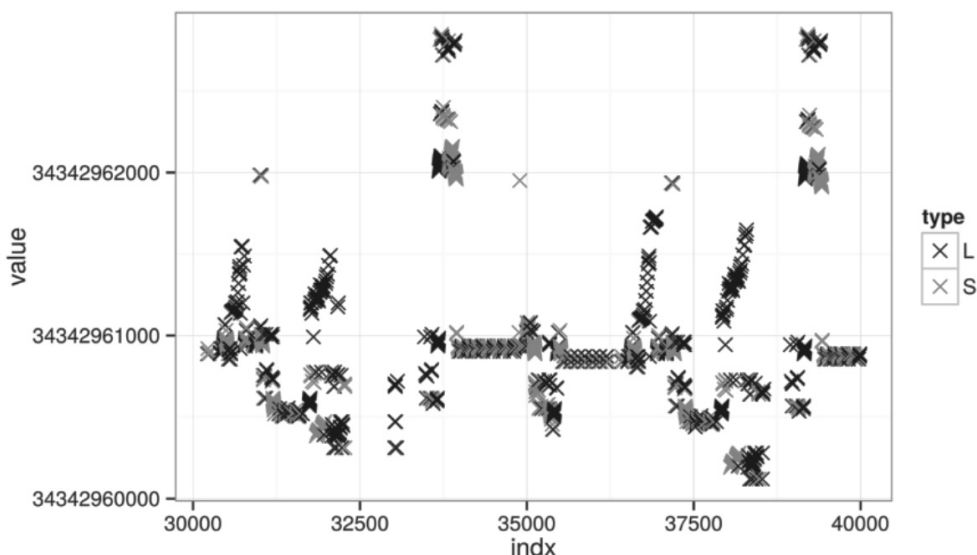


Рис. 1. Профиль региона памяти с неудовлетворительной локализацией



рактеризующийся неудовлетворительной ПВЛ обращений к памяти (этот профиль приложения получен с помощью профилировщика *tslmap*). Здесь обращение к регионам памяти происходит непоследовательно. При таком многократном обращении к данным приложения могут чаще наступать кэш-промахи, что будет приводить к снижению производительности и являться причиной неэффективного использования оборудования [6].

Для получения подобной картины работы приложения требуются средства профилирования программ, учитывающие особенности конкретной операционной системы, чтобы получить адекватный (лишенный избыточных данных) профиль приложения, в котором собраны обращения к памяти только конкретного приложения. На практике подбор подходящего инструмента профилирования представляет проблему. В [1] в качестве профилировщика использован разработанный в 2001 г. программный инструмент *MetaSim*, позволявший получить динамический профиль доступа приложений к памяти. Позднее в работе [7], посвященной издержкам динамического инструментирования на примере *MetaSim*, была проанализирована проблема больших временных затрат на сбор данных и проблема сохранения и анализа недопустимо большого формата выходного файла.

В настоящее время использовать *MetaSim* не представляется возможным по причине ряда существенных отличий современных высокопроизводительных систем от компьютеров, использовавшихся разработчиками *Арех*. В связи с этим необходимо было разработать профилировщик. Такая программа уже была ранее разработана на основе *Dyninst API* в 2013 г. в Санкт-Петербургском государственном политехническом университете в группе С.А. Фёдорова [5]. В данной работе были получены профили и дана оценка ПВЛ ряда приложений. К сожалению, не был подтвержден опыт исследования [7], уже показавший недостатки *Dyninst API* с точки зрения разработки подобных профилировщиков в связи с избыточным количеством данных при инструментировании, которые

искажают реальную оценку ПВЛ и порождают неадекватные (гигантские по размеру) профили для относительно небольших приложений. Таким образом, предварительные исследования показали отсутствие доступного свободного программного обеспечения для анализа ПВЛ, и было решено создать профилировщик, способный выполнить эту задачу. Важнейшие требования, предъявляемые к такому профилировщику, сформулированы ниже.

Функциональные требования

Задача оценки пространственно-временного поведения исследуемого приложения требует построения профиля в виде журнала адресов, по которым это приложение осуществляет доступ в оперативную память. На основании этого были определены следующие требования к программному инструменту:

1. Построение профиля ПВЛ непосредственно во время выполнения приложения: как для всего приложения, так и для отдельных регионов памяти на основе символической информации. Вывод должен содержать исчерпывающий журнал отслеживания обращений к памяти для отдельной функции или всего приложения.

2. Оценка показателей SL и TL «на лету», используя полученную информацию о доступе к памяти.

Входным форматом данных должен быть файл клиентского приложения, имеющий исполняемый и компоновочный форматы (ELF), используемый в современных UNIX-подобных операционных системах. Выходной формат данных должен содержать полный журнал доступа клиентского приложения к оперативной памяти с маркировкой типа доступа для каждой ячейки: сохранение (*load*) либо загрузка (*store*).

Для наблюдения за естественной работой приложения наиболее применимым подходом является динамический двоичный анализ, подразумевающий инструментирование клиентского приложения в процессе его выполнения, т. е. внесение анализирующего кода в двоичные данные исполняемого ELF-файла. Анализирующий код включается в процесс выполнения про-

граммы, не вызывая изменений в обычном поведении программы, но выполняя сторонние функции. Динамическое двоичное инструментирование сложно в реализации, поэтому было решено разработать профилировщик на основе одной из платформ (библиотек) для создания инструментов динамического двоичного анализа.

Наиболее развитыми API для создания средств динамического двоичного анализа для операционных систем Linux являются PerfSuite, PIN, Dyninst и Valgrind. В табл. 1 обобщены рассмотренные платформы и критерии выбора. Первое, что необходимо отметить: без счетчика адресов затруднительно посчитать обращения к памяти, поэтому сразу был отброшен вариант разработки инструмента профилирования на основе PerfSuite. Следующий вызов — это издержки инструментирования, избыточные данные в профиле, не относящиеся непосредственно к естественному поведению профилируемого приложения. Избыточные данные искажают реальные результаты пространственной и временной локализации, существенно замедляют профилирование и порождают профиль избыточного размера.

Авторы работы [7], сопоставив эти издержки для PIN и Dyninst API, показали, что объем аналитического кода при инструментировании с использованием Dyninst API примерно в два раза превышает показатели PIN и делает использование Dyninst менее выгодным для профилирования больших задач.

Мы частично используем результаты работы [5], в которой применялся доступный профилировщик на основе Dyninst API, как наиболее актуальные и воспроизводимые из имеющихся результатов. Для снижения объемов производимого профиля и временных издержек анализа было предложено профилировать не только приложение целиком, но и отдельные функции на основе символьной информации, что не было предусмотрено при разработке профилировщика в работе [5]. По указанным выше причинам, а также по причинам, связанным с особенностями лицензирования (которые не были показаны в таблице), предпочтение было отдано платформе Valgrind.

Valgrind — это платформа динамического двоичного инструментирования, предназначенная для создания инструментов динамического двоичного анализа. Значительным отличием Valgrind является то, что в ходе анализа инструментруется каждая инструкция, и информация о состоянии программы передается между частями инструментлирующего кода на низком, локальном уровне. Valgrind широко используется в области высокопроизводительных вычислений в последние годы, главным образом для анализа использования памяти MPI-приложениями [8, 9].

Цель настоящей работы — программный инструмент динамического двоичного анализа на базе Valgrind для оценки пространственно-временной локализации приложения. На основе сформулирован-

Таблица 1

Сравнение API для разработки средств динамического двоичного анализа

Возможности	Valgrind	PerfSuite	Dyninst	PIN
Поддержка POSIX	Да	Да	Да	Да
Многопоточность	Да	Да	Да	Да
Поддержка MPI	Да	Да	Да	Да
Счетчик адресов	Да	Нет	Да	Да
Поддержка C ABI	Да	Да	Нет	Да
JIT компилятор	Да	Нет	Нет	Да
Независимость от glibc	Да	Нет	Нет	Да
Профилирование на основе символьной информации	Нет	Нет	Нет	Да

ных требований разработан профилировщик. Особенности работы профилировщика приведены далее.

Введение в tslmap

Программный инструмент **tslmap** (**temporal and spatial locality memory access probe**) — это разработанный на основе Valgrind профилировщик Linux-приложений, созданный для исследования пространственной и временной локализации доступа к оперативной памяти в соответствии с методологией Арех. Профилировщик позволяет построить профиль ПВЛ как для всего приложения, так и для отдельных регионов памяти на основе символьной информации и вычислить показатели пространственной и временной локализации для анализируемого приложения. Вывод осуществляется в текстовый лог или на сетевой порт, что дает существенный выигрыш в скорости сохранения профилей при работе с высокопроизводительной базой данных. Преимущества tslmap как инструмента на основе Valgrind состоят в следующем:

1. В лог трассировки не попадают системные вызовы и другие действия ядра Linux (игнорируются операции планирования и обработки сигналов); ядро Valgrind заменяет небольшую часть кода собствен-

ным кодом в целях инструментирования. Этот код также не попадает в профиль приложения, что значительно снижает издержки хранения больших профилей.

2. Фиксируются почти все операции обращения к памяти, выполняемые непосредственно изучаемым приложением. Достигается высокая детализация информации о доступе клиентского приложения к памяти. Визуализация тестового журнала отслеживания осуществляется с помощью сценария на языке R. На рис. 2 показан пример полученного профиля приложения.

Профилировщик является свободным программным обеспечением, что гарантирует его высокую доступность и расширяемость.

Данные возможности позволяют существенно сократить время профилирования, визуализации и сохранения профиля, получая только необходимую информацию о работе приложения.

Архитектура tslmap

Программный инструмент **tslmap** является модулем, взаимодействующим с ядром платформы Valgrind — VEX IR. Ядро предоставляет базовый механизм запуска, управления и инструментирования клиентских приложений.

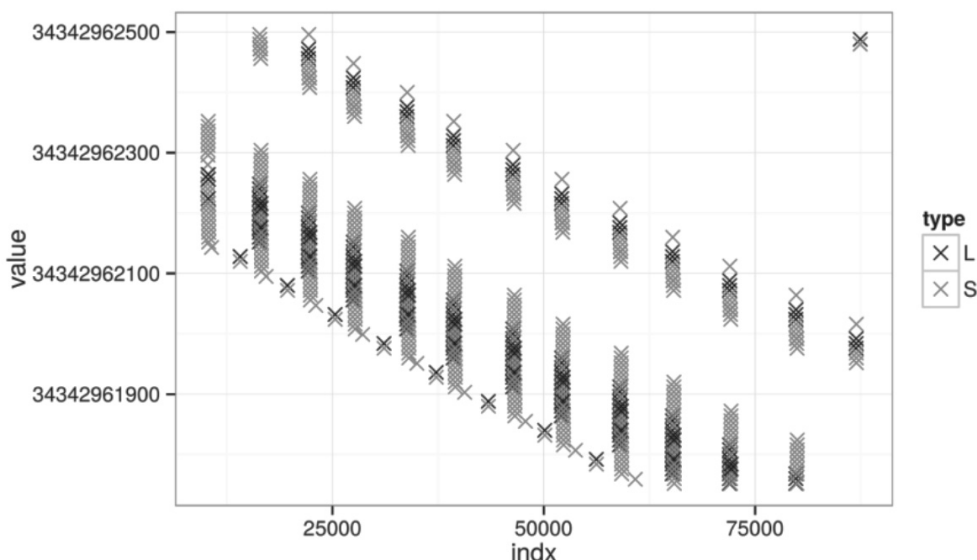


Рис. 2. Профиль приложения перемножения массивов

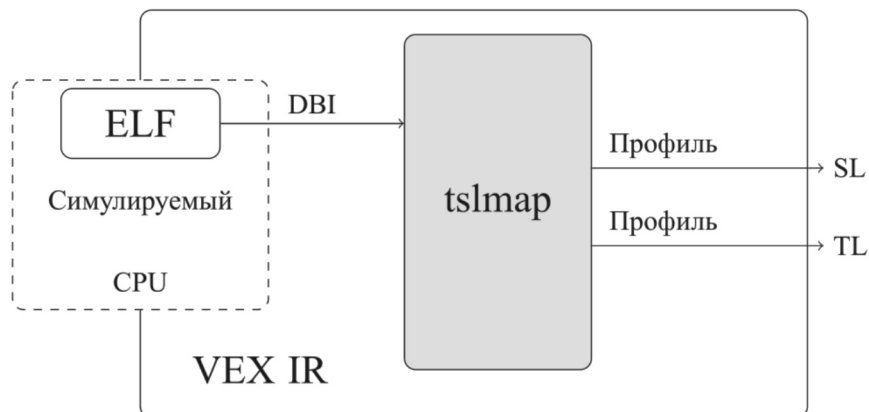


Рис. 3. Выполнение tslmap

Плагин `tslmap` является динамической библиотекой, инициализируемой ядром `VEX IR` при старте `Valgrind` посредством введения команд в командной строке `Linux`. Процесс выполнения `tslmap` представлен на рис. 3. Клиентское приложение выполняется на симулируемом `VEX IR` процессоре. `Valgrind` использует динамическую компиляцию: `tslmap` внедряется в процесс клиентского приложения в процессе JIT компиляции, включающем в себя дизассемблирование двоичного кода в язык промежуточного выполнения, который инструментируется с помощью `tslmap` и затем компилируется обратно. После динамического двоичного инструментария (`DBI`) строится профиль клиентского приложения, и осуществляется вычисление `SL` и `TL`.

Преимущество такой архитектуры состоит в том, что динамическая компиляция и инструментаривание ядром `VEX IR` являются высокоскоростными операциями и не оказывают существенного влияния на работу анализируемого приложения, что позволяет анализировать ПВЛ больших приложений, получая только ту информацию об их работе, которая необходима в конкретном случае. Это, в свою очередь, дает возможность исследовать приложения, работающие в нормальном режиме на высоконагруженных системах, не производя отдельных запусков.

Код плагина включает в себя базовые

функции модуля, а как инструмент профилирования – вычислительные функции для оценки ПВЛ. Ниже описана работа двух основных функций в терминах методологии `Apex`.

Оценка ПВЛ

Оценку пространственной локализации можно представить в виде суммы, где $stride_i$ обозначает долю от общего объема операций динамической памяти, имеющих шаг i . Особенность такого подхода состоит в генерации нормированной оценки в диапазоне $[0, 1]$, что обратно пропорционально средней длине шага. Приложение, которое выполняет только один шаг, получает один балл. Приложение, которое выполняет только два шага, получает оценку пять, а приложение без шагов получает оценку ноль. С целью уменьшения времени выполнения алгоритма индекс суммирования в формуле ограничивают некоторой величиной $i \max$. Выбирать очень большое значение этой величины не имеет особого смысла, т. к. каждое последующее слагаемое стремительно уменьшается [1].

В качестве входных данных используется текущий адрес доступа к памяти, размер окна адресов $W = 32$, к которым доступ уже был осуществлен, $i \max$ – ограничение шага (массив шагов размером $i \max$).

Дистанция между текущим адресом и адресами окна вычисляется для всех адресов доступа, и если дистанция больше или

равна i тах, элемент в массиве шагов с индексом полученного расстояния увеличивается на единицу. После обработки всех адресов вычисляется значение пространственной локализации (SL) по формуле:

$$SL = \sum_{i=1}^{\infty} \frac{stride_i}{i},$$

где $stride_i$ — доля обращений к памяти с длиной шага i [1].

Временная локализация — это тенденция приложения осуществлять доступ к регионам в памяти, к которым оно уже недавно обращалось. Дистанция повторного использования (*reuse distance*) для адреса в памяти A — это число уникальных номеров в памяти, к которыми приложение обращалось с момента последнего доступа к A . Временная локализация (TL) приложения вычисляется по формуле:

$$TL = \sum_{i=0}^{\log(N)-1} \frac{((reuse_{2^{i+1}} - reuse_{2^i}) \cdot (\log_2(N) - i))}{\log_2(N)},$$

где $reuse_i$ обозначает долю операций с динамической памятью с дистанцией повторного использования меньшей, либо равной i [1].

В качестве входных данных используется текущий адрес доступа к памяти и размер максимально рассматриваемого расстояния повторного использования N . Вывод профилировщика содержит трассировку доступа к памяти и оценку ПВЛ клиентского приложения.

Экспериментальные запуски

Цель и задачи экспериментальных запусков. Цель экспериментальных запусков состоит в проверке выполнения профилировщиком *tslmap* требований, изложенных выше в данной статье. Задачи экспериментальных запусков:

1. Построить и визуализировать профиль ПВЛ MPI-приложения перемножения матриц, выполняющегося на вычислительном кластере. Вычислить показатели ПВЛ приложений SL и TL и сопоставить их с результатами существующих исследований. На данном этапе показатели ПВЛ (SL и TL) будут сопоставлены с результатами имею-

щегося аналогичного профилировщика на основе Dyninst API, ранее представленного в работе [5], — единственного доступного профилировщика на основе методологии Арех.

2. Оценить избыточность данных в профилях всего приложения и отдельной функции, полученных с помощью *tslmap* и аналогичного профилировщика.

На разных стадиях экспериментальных запусков использовались разные анализируемые приложения. Выбор приложения для первой задачи (оценка ПВЛ) обусловлен наличием ранее полученных в [5] результатов SL и TL для инструмента на основе Dyninst API, что дает возможность сравнить их с результатами данной работы. С другой стороны, результаты работы [7] свидетельствуют о существенном снижении данных в профиле, полученном с помощью PIN, по сравнению с Dyninst API при анализе тестов NAS Parallel Benchmark. Поэтому на второй стадии экспериментальных запусков (оценка избыточности данных в профиле) для сравнения используется PIN и задача NAS Parallel Benchmark. Таким образом, предполагается подтверждение преимуществ *tslmap* в двух задачах экспериментальных запусков.

Оценка ПВЛ. Актуальность первой задачи состоит в исследовании пространственно-временного поведения MPI-программы на кластере: визуализации профиля и оценки локализации. Параллельные программы могут иметь ошибки параллелизма, включающие в себя множественный доступ на множестве взаимосвязанных переменных, что в сложных случаях приводит к непредсказуемому пространственно-временному поведению приложения и к неадекватному снижению производительности. Подобные ошибки могут существенно исказить данные, полученные при профилировании по причине непредсказуемости локализации доступа к памяти [10].

Для поддержки эксперимента использовался кластер «Триптих-2», включающий в себя вычислительные узлы IBM SYSTEM x3300 M4 и IBM SYSTEM x3650 M4 под управлением контроллера кластера.

Перемножение элементов матриц и векторов — одна из наиболее часто встречающихся задач в вычислительных приложениях. Можно предположить, что, поскольку в программе происходит обращение к элементам матрицы по столбцам, пространственная локализация (SL) будет, по крайней мере, ниже 0,5, как известно из работы [5]. Временная локализация (TL) будет выше среднего, т. к. происходит многократное обращение к одним и тем же элементам. Визуализация профиля данного приложения представлена на рис. 4.

В результате профилирования данного приложения с помощью *tslmap* получена следующая оценка локализации: $SL = 0,25$, $TL = 0,65$. В данном случае пространственная локализация оказалась несколько лучше результатов, полученных ранее с помощью инструмента на основе *Dyninst API* (ранее в [5] было получено $SL = 0,336$, $TL = 0,68$). В первую очередь это объясняется разницей содержимого построенных профилей: профиль *Dyninst API* был насыщен избыточным количеством не относящихся непосредственно к работе приложения служебных операций (инструкций инструментирования). Разреженный характер операций обращения к

памяти с ненулевым шагом объясняет искажение реальной картины пространственной локализации. Потенциально в случае с разработанным ранее на основе *Dyninst API* профилировщиком такие лишние данные могли достигать до 50 % содержимого получаемого профиля для больших приложений в связи с особенностями инструментирования *Dyninst API*.

В случае с *tslmap* избыточные данные в профиле удастся отсечь. Это становится возможным за счет особенностей низкоуровневого инструментирования, которые предоставляет ядро *VEX IR* платформы *Valgrind*, а также за счет маркировки каждого обращения к памяти с последующим удалением инструкций в выводе плагина. В результате в полученном с помощью *tslmap* профиле содержатся только обращения к памяти типа *S (store)* и *L (load)*, сделанные именно анализируемым клиентским приложением. Значение *SL*, полученное на основе уже «очищенного» профиля, является примером наиболее близкой оценки пространственного поведения данного приложения при выполнении на данном экспериментальном макете. Значение *SL* может несколько улучшаться по мере отключения узлов кластера, на которых за-

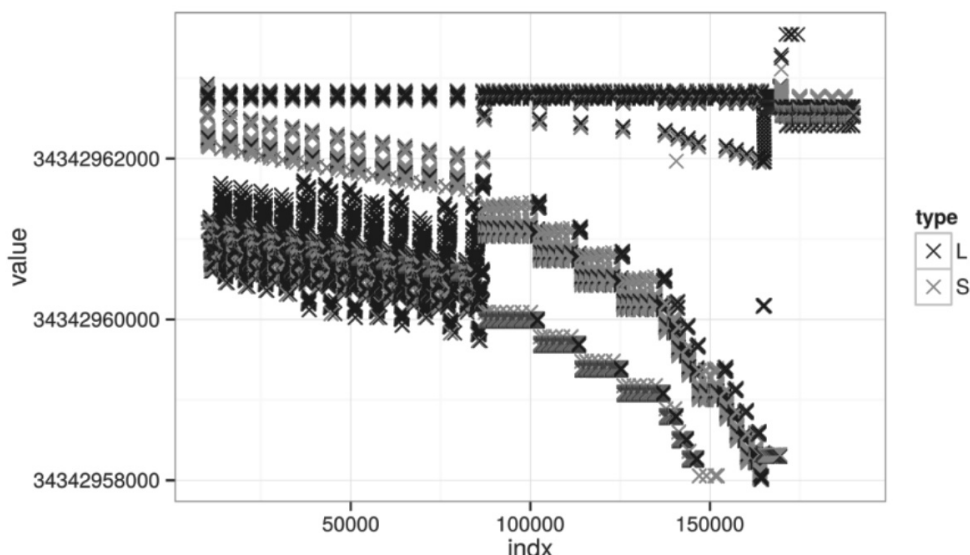


Рис. 4. Приложение перемножения матриц, выполняемое на трех узлах опытного кластера

пускается анализируемое приложение. При запуске лишь на одном узле полученное значение SL приближается к 0,15, что объясняется, в первую очередь, особенностями производительности сети кластера при использовании нескольких узлов, а также ранее отмеченным значительным количеством инструкций инструментирования при использовании Dyninst API.

Оценка избыточности данных в профилях.

Первая стадия экспериментальных запусков показала, что значения SL и TL близки к предполагаемым теоретическим результатам и к ранее полученным результатам для профилировщика на основе Dyninst API [5]. На второй стадии экспериментальных запусков необходимо проанализировать полученный профиль. Поскольку оценка ПВЛ весьма чувствительна к наличию в профиле посторонней информации о доступе к памяти, который мог быть осуществлен не только анализируемой программой, важно было сопоставить показатели отношения строк профиля к количеству строк кода для одной и той же функции, одного и того же приложения, но для разных профилировщиков.

Для сравнения характеристик профилей, полученных с помощью tslmap и PIN, применять профилировщик на основе Dyninst API нецелесообразно. Во-первых, данный профилировщик не поддерживает профилирование отдельно взятых функций на основе символьной информации, что не позволяет выделить в профиле только обращения, связанные с работой данной функции. Во-вторых, как было сказано выше, ранее в работе [7] уже были выявлены преимущества PIN с точки зрения компактности полученного профиля по

сравнению с Dyninst API. И если на первой стадии исследования нам были важны только его результаты оценки ПВЛ, то на второй стадии потребуется другой инструмент для сравнения.

Сравнение характеристик профилирования проводилось с использованием пакета тестов NAS Parallel Benchmark версии NPB3.3-MPI, который используется для комплексной оценки производительности вычислительных систем и для экспериментальных запусков профилирования [5, 11]. Для экспериментального запуска использовался тест IS (Integer Sort, сортировка целых чисел) с минимальным классом вычислительной сложности A для мультипроцессорных систем, что объясняется отсутствием необходимости проводить комплексную оценку производительности. Исследование размерности профиля имеет две основные задачи:

1. Сравнение размеров профиля, полученных посредством tslmap и PIN, для определения наличия избыточных данных в профиле.

2. Сравнение отношения количества строк профиля (KLoP) к количеству строк исходного кода (LoC) как для всего приложения (IS), так и для отдельной его функции (*find_my_seed*) на основе символьной информации, с целью определить достаточность данных в профиле.

В табл. 2 представлены результаты сравнения.

Сравнительный подход при оценке показателя KLoP/LoC для отдельной функции и для всей программы для каждого из профилировщиков основан на идее, что избыточное количество внедряемого при инструментировании аналитического кода

Таблица 2

Сравнение параметров профиля всего приложения и отдельной функции

Тип профиля	tslmap/Valgrind		PIN	
	IS (<i>find_my_seed</i>)	IS (<i>full</i>)	IS (<i>find_my_seed</i>)	IS (<i>full</i>)
LoC	43	1085	43	1085
KLoP	103 729	2 829 680	147 409	3 819 200
KLoP/LoC	2 412	2 608	3 428	3 520
GboP	1,4	38,8	2,0	52,3

повлияет на общий объем профиля. В данном случае пропорции между показателем KLoP/LoC отдельной функции и всей программы близки для сравниваемых профилировщиков. Однако в случае с `tslmap` удалось ограничить попадание избыточных данных инструментирования в профиль как за счет особенностей инструментирования, предоставляемых ядром VEX IR по умолчанию, так и за счет удаления маркированных обращений, связанных с инструментировавшими инструкциями, на этапе вывода профиля. В результате общий показатель размера для профиля (Gigabyte of Profile — GboP), полученного с помощью `tslmap`, составил 38,8 против 52,3 для PIN.

Важно отметить, что при профилировании больших приложений в первую очередь будет необходимо получать профиль конкретных функций. Получение профиля отдельной функции существенно снижает временные издержки на построение профиля и его визуализацию, когда это необходимо. А при планируемом дальнейшем массовом автоматизированном анализе приложений можно будет достичь существенной экономии при хранении в базе данных профилей значительного размера.

Оценка временных издержек профилирования с помощью `tslmap`. Оценка временных издержек профилирования производилась с помощью стандартной утилиты GNU `time`. Выполнение приложения перемножения матриц заняло 2,4 с без профилирования с помощью `tslmap` и 4,9 с с профилированием. Для некоторых задач пакета NAS Parallel Benchmark продолжительность выполнения при профилировании увеличивалась в три раза. Тем не менее полученные значения временных издержек являются сравнительно невысокими для инструмента на основе Valgrind. Так, ранее для стандартных инструментов на основе Valgrind (`Memcheck`, `Cachegrind`) продолжительность анализа возрастала в 10 раз и выше для ряда задач пакета SPEC CPU2000. В случае `tslmap` незначительное количество связанных с расчетом SL и TL операций обуславливает сравнительно низкие временные затраты анализа. Возможность профилирования отдельных регионов памяти на основе сим-

вольной информации существенно снижает продолжительность исследования.

Выводы по экспериментальным запускам. Получены следующие результаты экспериментальных запусков:

1. Программный инструмент `tslmap` позволяет с приемлемыми временными издержками построить профиль ПВЛ клиентского MPI-приложения, выполняемого на кластере, как для всего приложения, так и для отдельного региона памяти на основании символьной информации.

2. Полученные результаты ПВЛ ряда приложений соответствуют ожидаемым результатам и результатам более ранних исследований.

3. Экспериментальный профиль содержит полную информацию о доступе к памяти при сравнительно небольшом размере выходного файла.

Использование `tslmap` можно считать успешным, т. к. профилировщик позволяет получить более точную оценку ПВЛ по сравнению с известными аналогами. Важным итогом можно считать то, что удалось снизить издержки, связанные со временем профилирования, за счет профилирования отдельных регионов памяти, и снизить размер полученного профиля за счет маркировки и удаления обращений, не имеющих отношения к работе анализируемой программы. Значительное снижение размеров профиля расширяет спектр приложений, которые можно анализировать. Указанные особенности являются существенными преимуществами `tslmap` по сравнению с доступным профилировщиком ПВЛ [5].

Полученные в работе результаты позволяют проводить дальнейшие исследования на основе методологии Арех. Результаты оценки ПВЛ простых приложений показали корректность работы профилировщика. После этого можно проводить тестирование всех задач пакета NAS Parallel Benchmark. Задачи пакета NPВ являются задачами-ядрами большинства современных приложений. Знание ПВЛ таких задач и соответствующих им точек на поверхности Арех для различных вычислительных систем позволяет прогнозировать производитель-

ность приложений и подбирать наиболее подходящие конфигурации высокопроизводительных систем для конкретных классов задач. В работе [5] с помощью профилировщика на основе Dyninst API была получена ПВЛ только отдельных задач класса W: IS, CG, FT, MG и EP. Остальные задачи рассмотреть тогда не удалось по причине избыточного размера генерируемого разработанным профилировщиком журнала.

Планируется проведение экспериментальных запусков tsmap для построения поверхности быстрогодействия гибридной вычислительной системы с использовани-

ем всех задач пакета NPВ. На основе Valgrind разрабатывается набор утилит для автоматизированного поиска интересующих регионов профиля. Предполагается расширить понятия показателей ПВЛ на многопоточные приложения с целью адаптации и исследования особенностей оценки ПВЛ для случая множества процессов.

Поскольку исследования ПВЛ приложений уже имеют значительную теоретическую базу, дальнейшие исследования должны учитывать существующий опыт экспериментального применения методологии Apex.

СПИСОК ЛИТЕРАТУРЫ

1. Weinberg J., McCracken M.O., Strohmaier E., Snavey A. Quantifying Locality In The Memory Access Patterns of HPC Applications // Proc. of the 2005 ACM/IEEE Conf. on Supercomputing. Washington, DC, USA: IEEE Computer Society, 2005. P. 50. [Электронный ресурс] // URL: <http://dx.doi.org/10.1109/SC.2005.59>
2. Meuer H.W., Strohmaier E., Dongarra J., Simon H.D. The TOP500: History, Trends, and Future Directions in High Performance Computing. Chapman & Hall/CRC, 2014.
3. Ibrahim Khaled Z., Strohmaier Erich. Characterizing the Relation Between Apex-Map Synthetic Probes and Reuse Distance Distributions // Proc. of the 2010 39th Internat. Conf. on Parallel Processing. Washington, DC, USA: IEEE Computer Society, 2010, Pp. 353–362. [Электронный ресурс] // URL: <http://dx.doi.org/10.1109/ICPP.2010.43>
4. Семенов А.С. Разработка и исследование архитектуры глобально адресуемой памяти мультитредово-поточкового суперкомпьютера: Дисс. ... канд. техн. наук. Москва, 2010. 224 с.
5. Максимов В.И. Реализация методики измерения пространственно-временной локализации приложения и ее отображения на синтезируемую APEX-поверхность используемого оборудования: Дисс. ... магистра. СПб., 2013. 72 с.
6. Shan Hongzhang, Strohmaier Erich. A Multi-dimensional Micro-benchmark for Performance Study and Prediction // Proc. of the 2010 3rd Internat. Joint Conf. on Computational Science and Optimization. Washington, DC, USA: IEEE Computer Society, 2010. Pp. 156–160. [Электрон-

ный ресурс] // URL: <http://dx.doi.org/10.1109/CSO.2010.134>

7. Xiaofeng Gao, Laurenzano M., Simon B., Snavey A. Reducing overheads for acquiring dynamic memory traces // Workload Characterization Symp., 2005. Proc. of the IEEE Internat. Washington, DC, USA: IEEE Computer Society, 2005. Pp. 46–55. [Электронный ресурс] // URL: <http://dx.doi.org/10.1109/IISWC.2005.1526000>
8. Fan Shiqing, Keller Rainer, Resch Michael. Advanced Memory Checking Frameworks for MPI Parallel Applications in Open MPI // Tools for High Performance Computing 2011. Springer Berlin Heidelberg, 2012. Pp. 63–78. [Электронный ресурс] // URL: http://dx.doi.org/10.1007/978-3-642-31476-6_6
9. Baumann Th., Gracia J. Cudagrind: Memory-Usage Checking for CUDA // Tools for High Performance Computing 2013. Springer International Publishing, 2014. Pp. 67–78. [Электронный ресурс] // URL: http://dx.doi.org/10.1007/978-3-319-08144-1_6
10. Jannesari A. Detection of High-Level Synchronization Anomalies in Parallel Programs // Int. J. Parallel Program. 2015. Vol. 43. No. 4. Pp. 656–678. [Электронный ресурс] // URL: <http://dx.doi.org/10.1007/s10766-014-0313-x>
11. Wu Xingfu, Taylor Valerie. Performance Characteristics of Hybrid MPI/OpenMP Implementations of NAS Parallel Benchmarks SP and BT on Large-scale Multicore Supercomputers // SIGMETRICS Perform. Eval. Rev. 2011. Vol. 38. No. 4. Pp. 56–62. [Электронный ресурс] // URL: <http://doi.acm.org/10.1145/1964218.1964228>

REFERENCES

1. Weinberg J., McCracken M.O., Strohmaier E., Snavey A. Quantifying Locality In The Memory Access Patterns of HPC Applications. *Proceedings of the 2005 ACM/IEEE Conference on Supercomputing*.

Washington, DC, USA: IEEE Computer Society, 2005, P. 50. Available: <http://dx.doi.org/10.1109/SC.2005.59>

2. Meuer H.W., Strohmaier E., Dongarra J.,

Simon H.D. *The TOP500: History, Trends, and Future Directions in High Performance Computing*. Chapman & Hall/CRC, 2014.

3. **Ibrahim Khaled Z., Strohmaier Erich.** Characterizing the Relation Between Apex-Map Synthetic Probes and Reuse Distance Distributions, *Proceedings of the 2010 39th International Conference on Parallel Processing*. Washington, DC, USA: IEEE Computer Society, 2010, Pp. 353–362. Available: <http://dx.doi.org/10.1109/ICPP.2010.43>

4. **Semenov A.S.** *Razrabotka i issledovaniye arkhitektury globalno adresuyemoy pamyati multi-tredovo-potokovogo superkompyutera [Development and research of globally addressable memory architecture multithreaded streaming supercomputer: Thesis for the degree of candidate of technical sciences]*. Moscow, 2010, 224 p. (rus)

5. **Maksimov V.I.** *Realizatsiya metodiki izmereniya prostranstvenno-vremennoy lokalizatsii prilozheniya i yeye otobrazheniya na sinteziruyemyu APEX-poverkhnost ispolzuyemogo oborudovaniya [Implementation methodology for measuring spatial-temporal localization of the application and displayed on the surface of the synthesized APEX-used equipment: Thesis for a master's degree]*. St. Petersburg, 2013, 72 p. (rus)

6. **Shan Hongzhang, Strohmaier Erich.** A Multi-dimensional Micro-benchmark for Performance Study and Prediction, *Proceedings of the 2010 3rd International Joint Conference on Computational Science and Optimization*. Washington, DC, USA: IEEE Computer Society, 2010, Pp. 156–160.

Available: <http://dx.doi.org/10.1109/CSO.2010.134>

7. **Xiaofeng Gao, Laurenzano M., Simon B., Snively A.** Reducing overheads for acquiring dynamic memory traces, *Workload Characterization Symposium, 2005, Proceedings of the IEEE International*. Washington, DC, USA: IEEE Computer Society, 2005, Pp. 46–55. Available: <http://dx.doi.org/10.1109/IISWC.2005.1526000>

8. **Fan Shiqing, Keller Rainer, Resch Michael.** Advanced Memory Checking Frameworks for MPI Parallel Applications in Open MPI, *Tools for High Performance Computing 2011*. Springer Berlin Heidelberg, 2012, Pp. 63–78. Available: http://dx.doi.org/10.1007/978-3-642-31476-6_6

9. **Baumann Th., Gracia J.** Cudagrind: Memory-Usage Checking for CUDA, *Tools for High Performance Computing 2013*. Springer International Publishing, 2014, Pp. 67–78. Available: http://dx.doi.org/10.1007/978-3-319-08144-1_6

10. **Jannesari A.** Detection of High-Level Synchronization Anomalies in Parallel Programs, *Int. Journal Parallel Program*, 2015, Vol. 43, No. 4, Pp. 656–678. Available: <http://dx.doi.org/10.1007/s10766-014-0313-x>

11. **Wu Xingfu, Taylor Valerie.** Performance Characteristics of Hybrid MPI/OpenMP Implementations of NAS Parallel Benchmarks SP and BT on Large-scale Multicore Supercomputers, *SIGMETRICS Perform. Eval. Rev.*, 2011, Vol. 38, No. 4, Pp. 56–62. Available: <http://doi.acm.org/10.1145/1964218.1964228>

ЛЕВЧЕНКО Алексей Викторович — ассистент кафедры информационных и управляющих систем Института компьютерных наук и технологий Санкт-Петербургского политехнического университета Петра Великого.

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

E-mail: 2@expk.org

LEVCHENKO Aleksei V. *Peter the Great St. Petersburg Polytechnic University.*

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

E-mail: 2@expk.org

ФЁДОРОВ Станислав Алексеевич — старший преподаватель кафедры информационных и управляющих систем Института компьютерных наук и технологий Санкт-Петербургского политехнического университета Петра Великого.

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

E-mail: stanislav.fyodorov@ya.ru

FYODOROV Stanislav A. *Peter the Great St. Petersburg Polytechnic University.*

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

E-mail: stanislav.fyodorov@ya.ru

DOI: 10.5862/JCSTCS.236.7

УДК 656, 007; 004.81, 614.8; 007; 51-7, 351; 351.81; 351.78

Я.А. Селиверстов, С.А. Селиверстов

ИСПОЛЬЗОВАНИЕ СИСТЕМ КЛАССА ГАТЛОСЭМИ ДЛЯ УПРЕЖДЕНИЯ ПРИЧИН ВОЗНИКНОВЕНИЯ ДТП И НЕБЛАГОПРИЯТНЫХ СОЦИАЛЬНЫХ ИСХОДОВ В «УМНОМ ГОРОДЕ»

Ya.A. Seliverstov, S.A. Seliverstov

USE OF GATLOSAMI TO PREVENT CAUSES OF TRAFFIC ACCIDENTS AND ADVERSE SOCIAL ACCIDENTS IN A 'SMART CITY'

Произведено формальное построение модели интеллектуального городского анализатора транспортно-логистической и социально-экономической мобильности, посредством которого осуществляется переход к персонифицированному рекомендательному управлению элементами городской системы. Введено понятие директивный элемент городской системы. Поставлена проблема построения интеллектуальных систем профилактики причин возникновения ДТП и неблагоприятных социальных исходов в городской среде. Поиск причин наступления неблагоприятных социальных исходов разрешен процедурой абдуктивного вывода. Формирование кодифицированных библиотек потенциально-опасных шаблонов произведено с помощью процедуры индуктивного вывода. Рассмотрены практические приложения и даны рекомендации по дальнейшему функциональному развитию системы.

ИНТЕЛЛЕКТУАЛЬНЫЕ ТРАНСПОРТНЫЕ СИСТЕМЫ; УМНАЯ ГОРОДСКАЯ ЛОГИСТИКА; МОНИТОРИНГ ГОРОДСКИХ ТРАНСПОРТНО-ЛОГИСТИЧЕСКИХ И СОЦИАЛЬНО-ЭКОНОМИЧЕСКИХ ПРОЦЕССОВ; СИСТЕМЫ ПЕРСОНАЛЬНОГО РЕКОМЕНДАТЕЛЬНОГО УПРАВЛЕНИЯ; СИСТЕМЫ ПРОФИЛАКТИКИ ПРИЧИН ДТП.

The model of an intelligent data analyzer of city logistics, transportation, social and economic behavior of people and the state objects of urban systems has been developed in this paper. Personalized recommendation control elements of the urban system have been described. The concept of a directive element of the urban system was given and defined. The problem of prevention of causes of accidents has been detailed. Searching for causes of accidents was solved through an abductive inference process. The Codified Template Library of Potentially Dangerous Patterns of Behavior was created with an inductive inference process. Some examples and recommendations for improvement of the GATLOSEMI model were given.

INTELLIGENT TRANSPORT SYSTEMS; SMART URBAN LOGISTICS; MONITORING OF LOGISTICS AND SOCIO-ECONOMIC PROCESSES; PERSONAL RECOMMENDER SYSTEMS; ACCIDENT CAUSE PREVENTION SYSTEMS.

Постановка проблемы и анализ предметной области. Прогрессивное устойчивое развитие густонаселенных городских агломераций и мегаполисов [1] в существующих реалиях становится невозможным под

воздействием нарастающих факторов региональной логистической дестабилизации.

Традиционное городское и региональное управление требует ускоренного перехода к качественно новым системам класса

«Умный город» или Smart City, способным интеллектуализировать городскую среду [2].

Работа современных транспортных автоматизированных информационно-аналитических систем [3, 4] строится на вероятностных данных. Как правило, такие системы выполняют функцию низковероятностного прогноза [5] и измерения реальной транспортной ситуации [6] в рамках существующих показателей [7, 8]. Они также в состоянии разглядеть результат неблагоприятных дорожно-транспортных исходов, но оперативно распознать и своевременно предупредить процессы формирования городских социально-экономических, транспортных и логистических конфликтов в границах существующего программно-алгоритмического и технологического базисов для них является недостижимым шагом.

Класс систем, способных производить интеллектуальный анализ городских транспортно-логистических процессов [9, 10] и упреждать возникновение неблагоприятных социальных исходов посредством информационного уведомления, будем называть *городскими анализаторами транспортно-логистической и социально-экономической мобильности интеллектуальными* (ГАТЛОСЭМИ).

Среди последних научно-исследовательских инициатив, направленных на конструирование систем анализа городских транспортно-логистических данных, можно выделить следующие: в работах [11, 12] решены задачи анализа городских данных применительно к транспортным системам; в [13] произведено построение формальной модели классификации межагентных отношений социально-экономического поведения городского населения, а в [14] формально раскрыта структурная схема системы городского транспортно-логистического мониторинга; в [15] рассмотрены существующие междисциплинарные подходы и предложена концепция «умных городов»; в [16] обозначены перспективы их будущего развития.

В настоящей статье рассматриваются следующие проблемы: создания модели ГАТЛОСЭМИ для рекомендательно-управления городскими транспортно-

логистическими процессами; интеграции разобщенной информации городских ситуационных дата-центров: управления дорожным движением, сотовых операторов, информационных центров МЧС и др.; обеспечения координации городских ведомственных служб: МЧС, ГИБДД, Роспотребнадзора, скорой медицинской помощи, информационной безопасности, федеральной безопасности, статистики, санитарно-эпидемиологического, налогового, миграционного надзоров и др.; профилактики неблагоприятных социальных исходов посредством информационного персонального упреждения населения.

Классификация и анализ социально-экономического поведения элементов городской системы должны осуществляться в границах существующих систем уникальной идентификации [13], Единой системы классификации и кодирования технико-экономической и социальной информации [14], Единой адресной системы, Анатомо-терапевтическо-химической классификации, Медицинской классификации болезней десятого пересмотра (МКБ-10). Работа с данными должна осуществляться в рамках Федерального закона № 152-ФЗ «О персональных данных». Эксплуатация системы предполагает использование вычислительной инфраструктуры ПАО «Ростелеком» и электронного правительства РФ, которая включает 78 серверов IBM, 29 серверов HP, девять серверов Sun, четыре сервера Kraftway, одиннадцать систем хранения данных (СХД) Hitachi, три СХД IBM и три СХД Sun.

Функциональное наполнение системы ГАТЛОСЭМИ опирается на аналитически выведенные ранее в [14] *условия достоверной мобильности*, обуславливающие наличие функций идентификации, аутентификации, местоопределения и распознавания состояний подвижных и стационарных объектов городской транспортной системы (ГТС).

Условия достоверной мобильности применительно к городской системе (ГС) будем трактовать в следующей интерпретации: 1) каждый объект, входящий в ГС или находящийся внутри нее, должен иметь



идентификатор; 2) состояния объектов в ГС должны быть распознаваемы; 3) процесс идентификации объектов в ГС должен быть самоорганизован.

Выполнение условий достоверной мобильности в ГС реализуется процедурой директивизации в границах третьего условия (рис. 1).

Под директивизацией будем понимать процесс связывания совокупного множества элементов городской системы с директивными элементами (ДЭ) ГАТЛОСЭМИ, согласно выражению:

$$F_T^\Gamma : \mathcal{E}_c^s \oplus_{f_\delta} \{\mathcal{E}_d^g\} = \mathcal{E}_c^\Gamma, \quad (1)$$

где $\mathcal{E}^s = \{\mathcal{E}_c^s, c = 1, \dots, N_c\}$ – множество элементов городской системы; $\mathcal{E}^g = \{\mathcal{E}_d^g, d = 1, \dots, N_d\}$ – множество директивных элементов системы ГАТЛОСЭМИ; $\mathcal{E}^\Gamma = \{\mathcal{E}_c^\Gamma, c = 1, \dots, N_c\}$ – множество директивизированных элементов ГС; $\oplus_{f_\delta}$ – операция директивизации элементов ГТС.

Зададим формальную модель системы ГАТЛОСЭМИ набором функционалов, согласно:

$$\Phi^\Gamma = \left\langle \mathcal{E}^\Gamma; \hat{F}_N; \hat{F}_C; \hat{F}_T; \hat{F}_R; \hat{F}_\Psi; \right. \\ \left. \hat{F}_A; \hat{F}_\Xi; \hat{R}; C_{\mathcal{E}^\Gamma}; U^\Gamma \right\rangle_T, \quad (2)$$

где $\hat{F}_N : \mathcal{E}^\Gamma \rightarrow N$ – функционал идентификации элементного множества $\mathcal{E}^\Gamma$, который каждому объекту из $\mathcal{E}^\Gamma$ ставит в соответствие идентификатор/набор идентификаторов из множества N ; $\hat{F}_C : \mathcal{E}^\Gamma \times I^\Gamma \rightarrow CH$ – функционал характеристической параметризации элементного множества $\mathcal{E}^\Gamma$, ко-

торый каждому объекту из $\mathcal{E}^\Gamma$ с уникальным номером из I^Γ ставит в соответствие наборы характеристик, присущих данному объекту, таких что $CH = \{ch_\pi^{\mathcal{E}^\Gamma}, \pi = 1, \dots, N_\pi\}$ – множество характеристик элементов $\mathcal{E}^\Gamma$, причем $ch_\pi^{\mathcal{E}^\Gamma}$ задается кортежем $ch_\pi^{\mathcal{E}^\Gamma} = \langle name, \{value\} \rangle$, где $name$ – имя π -й характеристики, $\{value\}$ – область допустимых значений, которая задается перечислением этих значений, интервалом или функционально, с помощью правил вычисления и оценки; $\hat{F}_T : \mathcal{E}^\Gamma \rightarrow [\tau_T^{def} \vee (\tau_T; \tau_{T+1})]$ – временной оператор, который каждому объекту из $\mathcal{E}^\Gamma$ ставит в соответствие момент или интервал времени из множества T , на котором задано отношение строгого порядка, то есть $\tau_1 < \tau_2, \dots, \tau_T < \tau_{T+1}$; $\tau_T^{def} \in T$ – определенный момент времени, а $(\tau_T; \tau_{T+1}) \in T$ – определенный интервал времени; $\hat{F}_A : (\mathcal{E}^\Gamma; I^\Gamma) \rightarrow A$ – агентное отображение, которое каждому ДЭ ГС $\mathcal{E}_c^\Gamma \in \mathcal{E}^\Gamma$ со множеством идентификаторов $I_{\mathcal{E}_c^\Gamma}^\Gamma$ ставит в соответствие агента из $a_i \in A$; $\hat{F}_\Xi$ – функционал интеллектуального анализа графов социально-экономического поведения элементов ГС $\Gamma_{\mathcal{E}_c^\Gamma}^\Gamma(V_{\hat{F}_C}; T)$; $\hat{R} = OCH \cup ASCH \cup FCH$ – регламентирующая область социально-экономической и транспортно-логистической деятельности ДЭ ГС, включающая область оптимально-устойчивых OCH , допустимо-устойчивых $ASCH$ и неустойчивых значений FCH характеристик ДЭ ГС CH и их поведение $B_i(\mathcal{E}_i^\Gamma)$; $C_{\mathcal{E}^\Gamma} = (C_A \vee C_F) \vee \wedge C_{\mathcal{E}^\Gamma}^{HSB}$ – регулирующие воздействия ГАТЛОСЭМИ, состоящие из воздействий C_A рекомендательного,

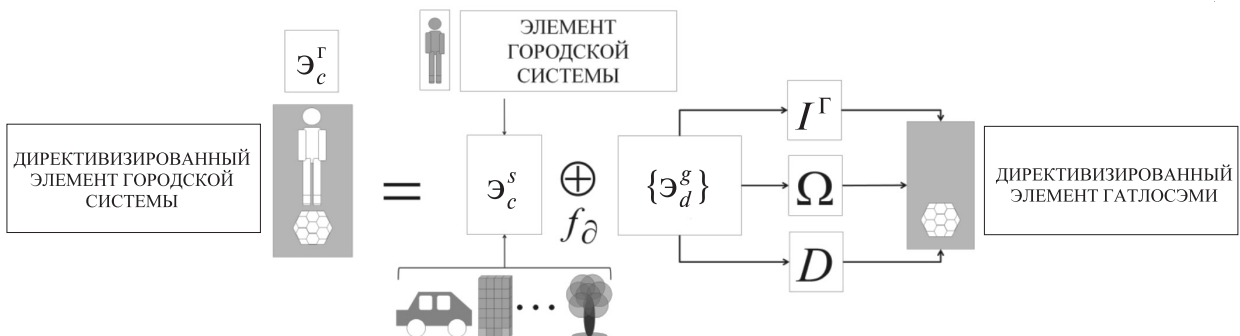


Рис. 1. Директивизация элементов городской системы

C_F принудительного и C_{Ξ}^{HSB} девиантного* управлений; $U^\Gamma = U^\Gamma(CH_U^{\Xi_i}) = \{u_{\Xi_i}(\{ch_\pi^u\})\}$ – целевая функция управления ДЭ ГС (или функция полезности), зависящая от субъ-ективных характеристик управления $\{u_{\Xi_i}(\{ch_\pi^u\})\}$ каждого ДЭ ГС; T – множество моментов или интервалов времени.

Тогда целевой функционал ГАТЛОСЭМИ Φ^C примет вид:

$$\begin{aligned} \Phi_{\text{ГАТЛОСЭМИ}}^C &= \lim_{\substack{B_i(\Xi_i^\Gamma) \rightarrow B_i^{\text{Pr ef}}(\Xi_i^\Gamma) \\ C_{\Xi_i^\Gamma} : OCH \leq (CH^\Gamma) \leq ACH}} U^\Gamma = \\ &= \max U^\Gamma = U_{\text{Pr ef}}^\Gamma, \end{aligned} \quad (3)$$

где $U_{\text{Pr ef}}^\Gamma$ – функция оптимального или предпочтительного управления, максими-зирующая полезность ДЭ ГС; $B_i^{\text{Pr ef}}(\Xi_i^\Gamma)$ – предпочтительное поведение ДЭ ГС.

Профилактика причин возникновения ДТП требует расширения существующих систем диагностики социально-экономической и потребительской деятельности. Послед-нее делает необходимым введение в систе-му ГАТЛОСЭМИ новых информационных функционалов и операторов, позволяющих связать существующие и новые системы классификации.

Таковым является функционал контро-ля ДЭ ГС:

$$\hat{\Pi} = \{p_x\}, \quad (4)$$

где $\{p_x\}$ – множество операторов контроля характеристик ДЭ ГС.

Функционал контроля множеству коди-фицированных социальных и иных откло-нений $CL_{dev} = \{CL_d\}$ и множеству кодифи-цированных объектов $CL_n = \{CL_k^\Xi\}$ ставит в соответствие множество кодифициро-ванных причин отклонений или кодифи-цированных библиотек противопоказаний $L = \{L_x\}$, в соответствии с выражением:

$$\hat{\Pi} : \{CL_d\} \cap \{CL_k^\Xi\} \rightarrow \{L_x\}. \quad (5)$$

Рассмотрим в составе функционала кон-троля ДЭ ГС два оператора, ответственных

за надлежащее физиологическое состояние городского населения. Выражение (5) пере-пишем в следующем виде:

$$\hat{\Pi} = \{p_g^h; p_n^h; \dots\}, \quad (6)$$

где p_g^h – оператор контроля товарного потребления, который множеству коди-фицированных медицинских диагнозов международной классификации болез-ней десятого пересмотра [МКБ-10] = $\{dm_\mu\}$ ставит в соответствие кодифицированную библиотеку товарных противопоказаний из множества существующих систем то-варной и продуктовой классификации [ОКП, АТХ, ...] в соответствии с выраже-нием:

$$p_g^h : [\text{МКБ-10}] \cap [\text{ОКП, АТХ, ...}] \rightarrow \text{LGC}, \quad (7)$$

где LGC – кодифицированная библиотека товарных противопоказаний.

p_n^h – оператор контроля потребитель-ской деятельности, который, сравнивая мно-жество кодифицированных медицинских диагнозов [МКБ-10] = $\{dm_\mu\}$ с системами классификации социально-экономической деятельности и социальных потребностей [ОКПД, ОКВЭД, LHN...], выводит соот-ветствующую кодифицированную библио-теку противопоказанных видов потреби-тельской деятельности LNC в соответствии с выражением:

$$\begin{aligned} p_n^h : [\text{МКБ-10}] \cap \\ \cap [\text{ОКПД, ОКВЭД, LHN...}] \rightarrow \text{LNC}, \end{aligned} \quad (8)$$

где LHN = $\{need_\alpha\}$ – кодифицированная библиотека человеческих потребностей.

Введение новых систем потребитель-ского контроля согласно (7), (8) потребует уточнений стандартов кодификации това-ров, таких как EAN-13.

Интеллектуальный анализ. Рассмотрим более подробно структуру функционала интеллектуального анализа:

$$\hat{F}_\Xi = \{\Xi_A^L; \Xi_I^L; \Xi_P^L \dots\}, \quad (9)$$

где Ξ_A^L – процедура абдуктивного вывода; Ξ_I^L – процедура индуктивного вывода; Ξ_P^L – процедура вероятностного анализа.

Последовательно рассмотрим каждую из процедур (9).

Ξ_A^L – процедура абдуктивного выво-

* Информационное воздействие ГАТЛОСЭМИ, упреждающее приближение индивидуального поведения ДЭ ГС к потенциально-опасному шаблону.



да [17, 18], каждому состоянию перехода $S_{\mathfrak{A}_{ic}}^{\tau_s}(V_{\hat{F}_c}; T)$ директивного элемента $\mathfrak{A}_{ic}^r$ из области оптимально-устойчивых состояний в область допустимо-устойчивых или неустойчивых состояний на интервале времени $\tau_s \in T_S$ ставит в соответствие множество его порождающих временных гипотетических графов $\Gamma_h^{\mathfrak{A}_{ic}}(V_{\hat{F}_c}; T)$ на временном горизонте T_S .

Состояние перехода $S_{\mathfrak{A}_{ic}}^{\tau_s}(V_{\hat{F}_c}; T)$ в общем случае будем называть ситуацией, а применительно к рассматриваемой проблеме – «неблагоприятным социальным исходом», вида:

$$S_{\mathfrak{A}_{ic}}^{\tau_s}(V_{\hat{F}_c}; T) \xrightarrow{\Xi_d^L} \bigvee_{\lambda=1}^{\lambda=(2^h-1)} \left(\bigwedge_{h=1}^{h=N_h} \Gamma_h^{\mathfrak{A}_{ic}}(V_{\hat{F}_c}; T) \right)_{\lambda} = C_{\lambda=1}^S \vee \dots \vee C_{\lambda=2^h-1}^S, \quad (10)$$

где $h = 1, \dots, N_h$ – число гипотез, квазипорождающих неблагоприятный социальный исход $S_{\mathfrak{A}_{ic}}^{\tau_s}(V_{\hat{F}_c}; T)$; λ – количество наборов гипотетических графов C_{λ}^S , представляющих собой совместное сочетание выявленных причин; S – индекс-указатель неблагоприятного социального исхода.

Процедуру Ξ_p^L вероятностного анализа причин возникновения ДТП и неблагоприятных социальных исходов представим в виде:

$$S_{\mathfrak{A}_{ic}}^{\tau_s}(V_{\hat{F}_c}; T) \xrightarrow{\Xi_p^L} p(C_{\delta=1}^S) \geq \dots \geq p(C_{\delta=2^h-1}^S), \quad (11)$$

где $p(C_{\delta=1}^S), \dots, (C_{\delta=2^h-1}^S)$ – вероятности совместных сочетаний гипотетических графов, на которых задан строгий порядок.

Процедура вероятностного анализа Ξ_p^L строится на хорошо проработанном статистическом аппарате [19].

Пояснение процедуры абдуктивного вывода (10) в графовой интерпретации дано на рис. 2.

Гипотетические графы (10) формируются в процессе двух этапов.

Этап 1. Сравнение суточного социально-экономического поведения ДЭ ГС $\Gamma_{\mathfrak{A}_{ic}}^{date}(V_{\hat{F}_c}; T)$, предшествующего неблагоприятному социальному исходу $S_{\mathfrak{A}_{ic}}^{\tau_s}(V_{\hat{F}_c}; T)$, со следующими эталонными наборами графов социально-экономического поведения ДЭ ГС, находящихся в оптимально-устойчивой области:

- наиболее вероятным в интервале одного месяца социально-экономическим поведением ДЭ ГС с учетом дня недели:

$$h = 1 : h_{\Gamma=1} = 1$$

$$V_{h=1}^S = V_{\mathfrak{A}_{ic}}^{date} \setminus (V_{\mathfrak{A}_{ic}}^{date} \cap V_{\mathfrak{A}_{ic}}^{e.date}) \rightarrow \Gamma_{h=1}^{\mathfrak{A}_{ic}}(V_{\hat{F}_c}; T), \quad (12)$$

где V_h^S – множество ДЭ ГС (вершин), входящих в вершины $V_{\mathfrak{A}_{ic}}^{date}$ графа $\Gamma_{\mathfrak{A}_{ic}}^{date}(V_{\hat{F}_c}; T)$ и не входящих в вершины $V_{\mathfrak{A}_{ic}}^{e.date}$ графа $\Gamma_{\mathfrak{A}_{ic}}^{e.date}(V_{\hat{F}_c}; T)$;

- наиболее вероятным в интервале одного месяца социально-экономическим

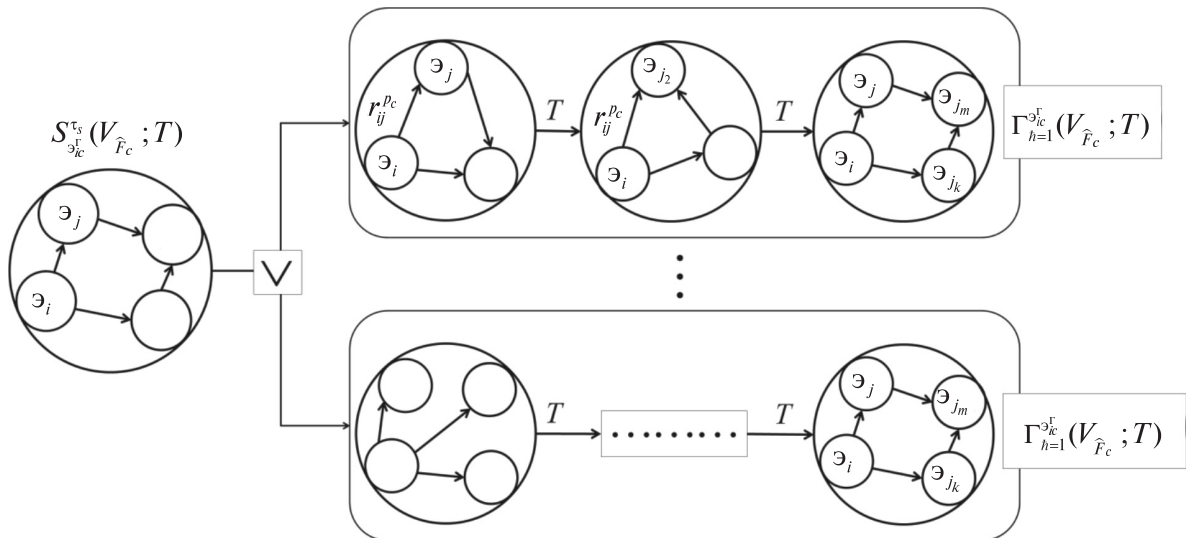


Рис. 2. Графовое представление процедуры абдуктивного вывода

поведением ДЭ ГС:

$$\hbar = 2 : \hbar_2 = 2$$

$$V_{\hbar=2}^s = V_{\mathfrak{E}_{ic}^f}^{date} \setminus (V_{\mathfrak{E}_{ic}^f}^{date} \cap V_{\mathfrak{E}_{ic}^f}^{month}) \rightarrow \Gamma_{\hbar=2}^{\mathfrak{E}_{ic}^f}(V_{\hat{F}_c}; T), \quad (13)$$

где $V_{\hbar}^s$ — множество ДЭ ГС (вершин), входящих в вершины $V_{\mathfrak{E}_{ic}^f}^{date}$ графа $\Gamma_{\hbar}^{date}(V_{\hat{F}_c}; T)$ и не входящих в вершины $V_{\mathfrak{E}_{ic}^f}^{month}$ графа $\Gamma_{\mathfrak{E}_{ic}^f}^{month}(V_{\hat{F}_c}; T)$;

• наиболее вероятным социально-экономическим поведением ДЭ ГС в интервале недели, предшествующей неблагоприятному социальному исходу:

$$\hbar = 3 : \hbar_{\Gamma=N_{\Gamma}} = 3$$

$$V_{\hbar}^s = V_{\mathfrak{E}_{ic}^f}^{date} \setminus (V_{\mathfrak{E}_{ic}^f}^{date} \cap V_{\mathfrak{E}_{ic}^f}^{week}) \rightarrow \Gamma_{\hbar=3}^{\mathfrak{E}_{ic}^f}(V_{\hat{F}_c}; T), \quad (14)$$

где $V_{\hbar}^s$ — множество ДЭ ГС (вершин), входящих в вершины $V_{\mathfrak{E}_{ic}^f}^{date}$ графа $\Gamma_{\hbar}^{date}(V_{\hat{F}_c}; T)$ и не входящих в вершины $V_{\mathfrak{E}_{ic}^f}^{week}$ графа $\Gamma_{\mathfrak{E}_{ic}^f}^{week}(V_{\hat{F}_c}; T)$.

Параметр времени T при этом определяет горизонт исследуемого события, регулирование которого позволяет включать и исключать из рассмотрения акты социально-экономического взаимодействия между ДЭ ГС. Наиболее вероятным для исследования является интервал времени $T = [\tau_0; \tau_s]$ от начала τ_0 до свершения события τ_s исследуемого события.

Таким образом, на первом этапе в процессе выполнения процедур (12)–(14) выделяются отличные от эталонных акты социально-экономической деятельности и формируются гипотетические графы, которые могли явиться причиной возникновения исследуемого неблагоприятного социального или иного исхода.

Этап 2. Проверка оператором $C_{\mathfrak{E}_{\Gamma}}$ состояний всех ДЭ ГС, входящих в граф суточного социально-экономического поведения ДЭ ГС $\Gamma_{\mathfrak{E}_{ic}^f}^{date}(V_{\hat{F}_c}; T)$, предшествующего неблагоприятному социальному исходу $S_{\mathfrak{E}_{ic}^f}^{\tau_s}(V_{\hat{F}_c}; T)$, через кодифицированные базы, сформированные операторами контроля характеристик ДЭ ГС $\hat{\Pi} = \{\pi_x^{\mathfrak{E}_{ic}^f}\}$.

В гипотетический граф включаются ДЭ ГС, чье социально-экономическое поведение протекает с нарушением кодифицированных баз операторов контроля, согласно условиям:

$$\begin{aligned} \Gamma_{\mathfrak{E}_{ic}^f}^{date}(V_{\hat{F}_c}; T) &\rightarrow (V_{\hat{F}_c})_{\mathfrak{E}_{ic}^f}^{date} \overset{C_{\mathfrak{E}_{\Gamma}}}{\cap} L_x^{\mathfrak{E}_{ic}^f} = \\ &= \Gamma_{\hbar_x}^{\mathfrak{E}_{ic}^f}(V_{\hat{F}_c}^x; T), \quad x = \text{п}, \dots, \text{п}_x \end{aligned} \quad (15)$$

$$\begin{aligned} \{\Gamma_{\mathfrak{E}_{ic}^f}^{date}(V_{\hat{F}_c}; T)\} &\rightarrow \{(V_{\hat{F}_c})_{\mathfrak{E}_{ic}^f}^{date}\} \overset{C_{\mathfrak{E}_{\Gamma}}}{\cap} \{L_x^{\mathfrak{E}_{ic}^f}\} = \\ &= \{\Gamma_{\hbar_x}^{\mathfrak{E}_{ic}^f}(V_{\hat{F}_c}^x; T)\}, \quad x = \text{п}, \dots, \text{п}_x, \end{aligned}$$

где $\mathfrak{E}_{ic}^f$ — ДЭ ГС, чей суточный граф социально-экономического поведения проверяется операторами функционала контроля; $\{\mathfrak{E}_{ic}^f\}$ — множество ДЭ ГС, взаимодействующих с $\mathfrak{E}_{ic}^f$ и входящими в его суточный граф социально-экономического поведения $\Gamma_{\mathfrak{E}_{ic}^f}^{date}(V_{\hat{F}_c}; T)$; $L_x^{\mathfrak{E}_{ic}^f}$ — кодифицированная библиотека для ДЭ ГС $\mathfrak{E}_{ic}^f$, сформированная оператором контроля п_x ; $\{\Gamma_{\hbar_x}^{\mathfrak{E}_{ic}^f}(V_{\hat{F}_c}^x; T)\}$ — множество гипотетических графов актов социально-экономического поведения ДЭ ГС, входящих в граф $\Gamma_{\mathfrak{E}_{ic}^f}^{date}(V_{\hat{F}_c}; T)$, в отношении которых выявлено нарушение кодифицированных библиотек, сформированных оператором контроля; $\{\Gamma_{\mathfrak{E}_{ic}^f}^{date}(V_{\hat{F}_c}^x; T)\}$ — социально-экономическое поведение множества ДЭ ГС $\{\mathfrak{E}_{ic}^f\}$, взаимодействующих с ДЭ ГС $\mathfrak{E}_{ic}^f$.

На втором этапе в процессе процедур (15) формируются гипотетические графы актов ДЭ ГС, в отношении которых выявлены нарушения кодифицированных баз, сформированных операторами контроля $\{\text{п}_x\}$, и которые могли бы явиться причиной возникновения неблагоприятного социального исхода $S_{\mathfrak{E}_{ic}^f}^{\tau_s}(V_{\hat{F}_c}; T)$. Таким образом, число гипотетических графов (12)–(15) вычисляется в соответствии с выражением:

$$\hbar = \hbar_{\Gamma} + \hbar_x. \quad (16)$$

Процедура абдуктивного вывода (10) запускается в момент зарегистрированного перехода ДЭ из области оптимально-устойчивых состояний в область допустимо-устойчивых или неустойчивых состояний.

Так как ситуация $S_{\mathfrak{E}_{ic}^f}^{\tau_s}(V_{\hat{F}_c}; T)$ свидетельствует о переходе директивного элемента в допустимо-устойчивое или неустойчивое состояние, то гипотетический граф ее порождения $\Gamma_{\hbar}^{\mathfrak{E}_{ic}^f}(V_{\hat{F}_c}^x; T)$ будем называть потенциально-опасным шаблоном и обозначать $\Gamma_{\hbar}^{\mathfrak{E}_{ic}^f}(V_{\hat{F}_c}^x; T)$.

Ξ_{Γ}^L — процедура индуктивного вывода



[20], которая из множества межэлементных отношений $\{\mathcal{E}_{ic}^r \times \mathcal{E}_{jc}^r\} = \{\Gamma(v; r_{ij}^{p_c})\}$, событий $\Gamma_h^{\mathcal{E}_{ic}^r}(V_{\hat{F}_c}^x; T)$ и фактов $CH_x^{\Gamma_h}(\mathcal{E}^S)$ за определенный период $\Gamma_{\mathcal{E}_{ic}^r}^{period}(V_{\hat{F}_c}^x; T)$, породивших

неблагоприятный социальный исход (ситуацию) $S_{\mathcal{E}_{ic}^r}^{\tau_s}(V_{\hat{F}_c}^x; T)$, выводит правило Π_S порождения данной ситуации $\Gamma_s^{\mathcal{E}_{ic}^r}(V_{\hat{F}_c}^x; T)$, согласно

$$\Xi_I^L : \frac{\left. \begin{aligned} &CH_x^{\Gamma}(\mathcal{E}_{\Gamma}^{date}) \cap (CH_x^{\Gamma}(\mathcal{E}_{\Gamma}^{d_i}), CH_x^{\Gamma}(\mathcal{E}_{\Gamma}^S); L_x; LPDP) \rightarrow CH_x^{\Gamma_h}(\mathcal{E}^S) \\ &\Gamma_{\mathcal{E}_{ic}^r}^{date}(V_{\hat{F}_c}; T) \cap \{\Gamma_{\mathcal{E}_{ic}^r}^{d_i}(V_{\hat{F}_c}; T)\} \cap S_{\mathcal{E}_{ic}^r}^{\tau_s}(V_{\hat{F}_c}; T) \rightarrow \Gamma_h^{\mathcal{E}_{ic}^r}(V_{\hat{F}_c}; T) \\ &\{\Gamma(v; r_{ij}^{p_c})\} \in \Gamma_{\mathcal{E}_{ic}^r}^{d_i}(V_{\hat{F}_c}; T), i = 1, N_i^s, j = 1, N_j^s, p_c = 1, N_{p_c}^s \end{aligned} \right\} \rightarrow S_{\mathcal{E}_{ic}^r}^{\tau_s}(V_{\hat{F}_c}; T)}{X : S_{\mathcal{E}_{ic}^r}^{\tau_s}(V_{\hat{F}_c}; T) \rightarrow \Gamma_s^{\mathcal{E}_{ic}^r}(V_{\hat{F}_c}; T), CH_x^{\Gamma_h}(\mathcal{E}^S) \mid \Rightarrow X = \Pi_S}, \quad (17)$$

где $CH_x^{\Gamma}(\mathcal{E}_{\Gamma}^{date}), CH_x^{\Gamma}(\mathcal{E}_{\Gamma}^{d_i})$ — характеристики множества ДЭ ГС графа $\Gamma_{\mathcal{E}_{ic}^r}^{date}(V_{\hat{F}_c}; T)$ и эталонных графов $\{\Gamma_{\mathcal{E}_{ic}^r}^{d_i}(V_{\hat{F}_c}; T)\}$ на анализируемом интервале времени; $S_{\mathcal{E}_{ic}^r}^{\tau_s}(v; r_{ij}^{p_c})$ — множество зарегистрированных параметров функциональных отношений, предшествовавших неблагоприятному социальному исходу; $CH_x^{\Gamma}(\mathcal{E}_{\Gamma}^S)$ — характеристики ДЭ ГС, ставших участниками неблагоприятного социального исхода; $CH_x^{\Gamma_h}(\mathcal{E}^S)$ — характеристики ДЭ ГС в гипотетических графах

порождения неблагоприятного социального исхода.

Пояснение процедуры индуктивного вывода (17) в графовой интерпретации дано на рис. 3.

В основу процедуры индуктивного вывода и формирования правила Π_S порождения ситуаций положен *принцип ситуационного подобия*. Сформулируем его следующим образом:

Если в отношении ДЭ ГС $\mathcal{E}_{\alpha}$ с характери-

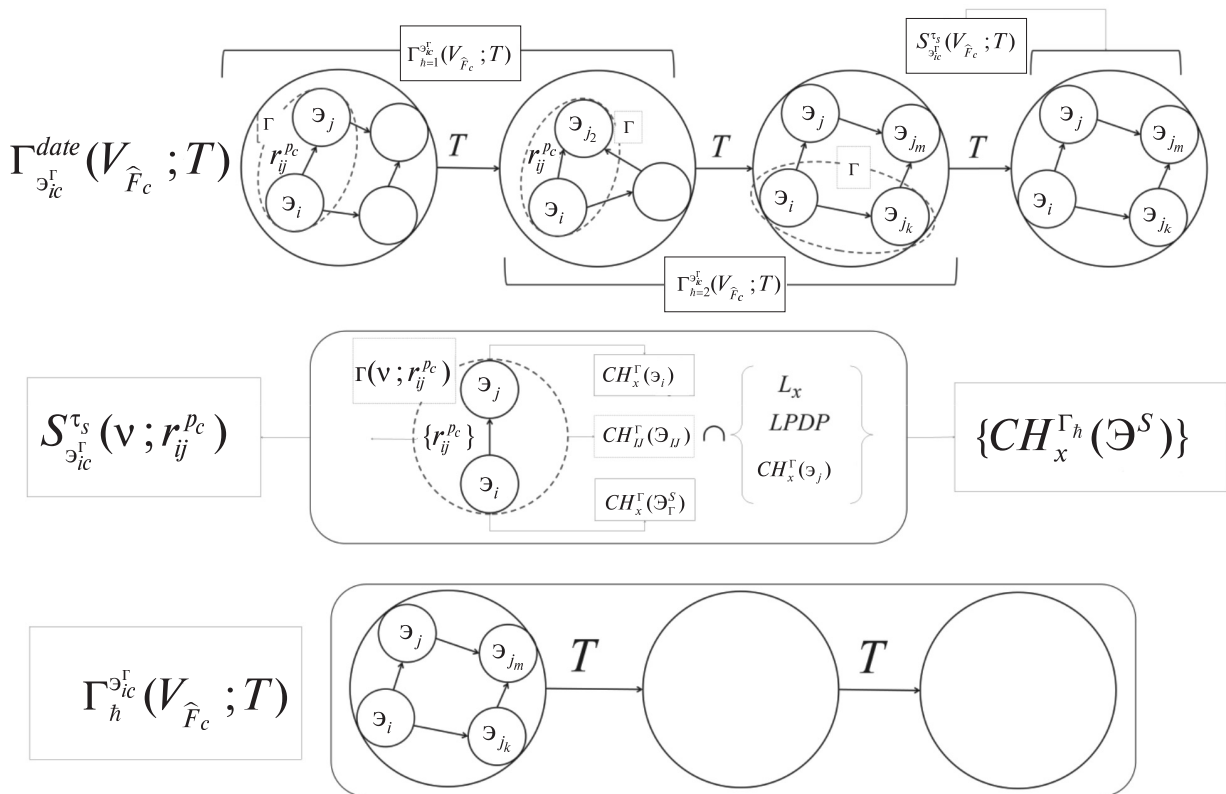


Рис. 3. Процедура индуктивного вывода

стиками CH_{α} и социально-экономическим поведением $\Gamma_{\alpha}(V_{\hat{F}_c}; T)$ по причине $\Gamma_h(V_{\hat{F}_c}; T)$ наступил неблагоприятный социальный исход $S_{\phi} = S_{\alpha}(V_{\hat{F}_c}; T)$, то ДЭ ГС α , имеющий схожие параметры $CH_{\alpha} \sim CH_{\beta}$; $\Gamma_{\alpha}(V_{\hat{F}_c}; T) \sim \Gamma_{\beta}(V_{\hat{F}_c}; T)$ по той же причине $\Gamma_h(V_{\hat{F}_c}; T)$ может попасть в схожий неблагоприятный социальный исход S_{ϕ} , то есть:

$$\Gamma_{\alpha}^{def}(V_{\hat{F}_c}; T) \oplus \Pi_{S_{\phi}} = \Gamma_{\beta}(V_{\hat{F}_c}; T). \quad (18)$$

Пояснение принципа ситуационного подобия (18) в графовой интерпретации дано на рис. 4.

Граф порождения ситуации $S_{\alpha}(V_{\hat{F}_c}; T)$ при этом представляет собой потенциально опасный шаблон, то есть $\Gamma_{\alpha}^{def}(V_{\hat{F}_c}; T) = HSB_{\alpha}^s(V_{\hat{F}_c}; T)$, который регистрируется в кодифицированной библиотеке потенциально-опасных шаблонов социально-экономической деятельности ДЭ ГС LPDP [21].

Рассмотрим работу ГАТЛОСЭМИ упрощенно на примере интеллектуального анализа причин возникновения ДТП с участием городского жителя.

Пример. Социально-экономическое поведение жителя Санкт-Петербурга $\Gamma_{h_c}^{date}(V_{\hat{F}_c}; T) = \Gamma_{id_1^h}^{04.11.2015}(V_{\hat{F}_c}; [18:00 - 00:00])$ с ИНН = id_1^h , повлекшее 4 ноября 2015 года ДТП $S_{\alpha}(V_{\hat{F}_c}; T)$, представлено на рис. 6. Эталонное социально-экономическое поведение жителя Санкт-Петербурга с ИНН = id_1^h за октябрь 2015 года, день недели среда $\Gamma_{h_c}^{date}(V_{\hat{F}_c}; T) = \Gamma_{id_1^h}^{10.2015(wend)}(V_{\hat{F}_c}; [18:00 - 00:00])$ в интервале времени с 18:00 по 00:00 представлено на рис. 5.

На рис. 5 и 6 нами использованы следующие обозначения:

$$\text{ИНН (жителя)} = id_1^h,$$

$$\text{ИНН} \begin{pmatrix} \text{друга} \\ \text{жителя} \end{pmatrix} = id_2^h,$$

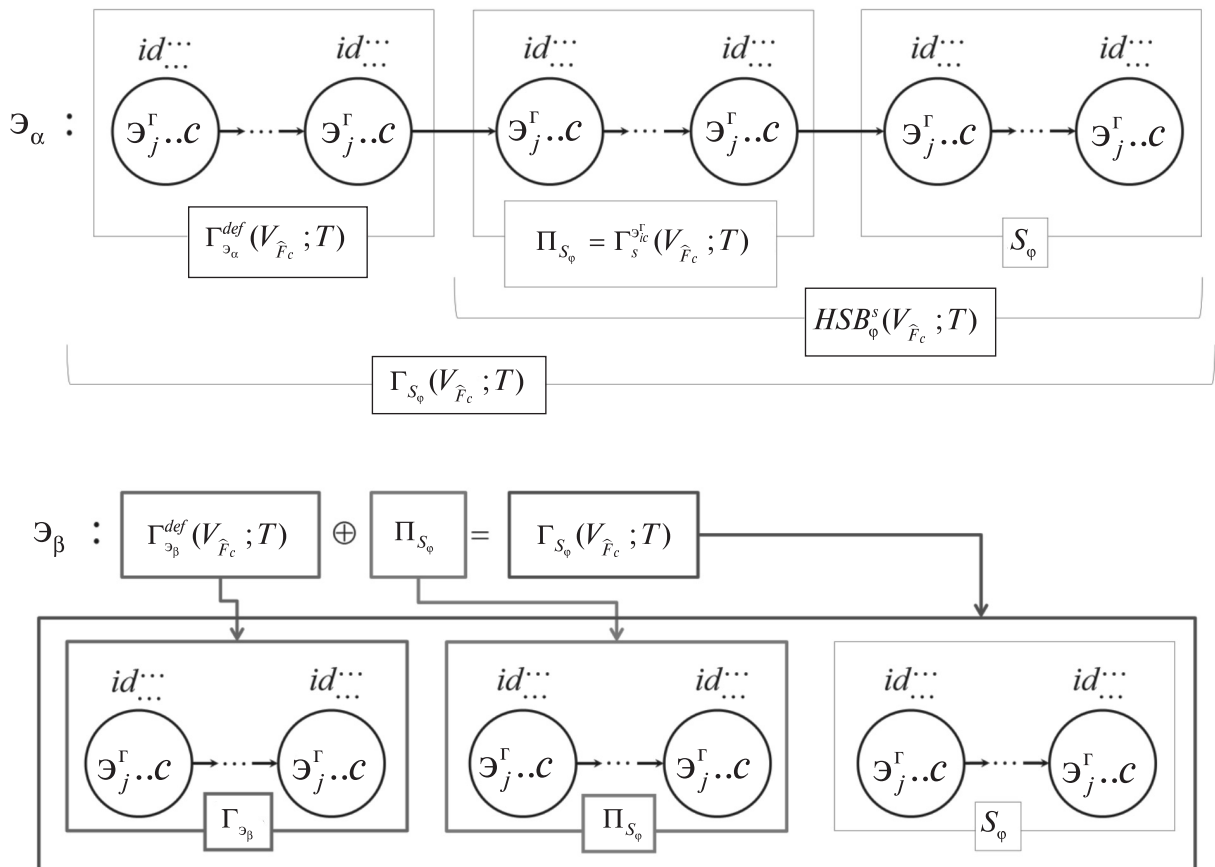


Рис. 4. Графовое представление принципа ситуационного подобия



ИНН (потерпевшего) = id_3^h ,

ИНН $\begin{pmatrix} \text{сотрудника} \\ \text{ГИБДД} \end{pmatrix} = id_4^h$,

ИНН (фельдшера) = id_5^h ,

ИНН (бармена) = id_6^h ; id_1^o , id_2^o , id_3^o , id_4^o — уникальные кадастровые номера (УКН) места жительства, работы, бара и больницы соответственно; id_1^{rp} , id_2^{rp} — УКН пешеходных перегонов, примыкающих к работе и к бару соответственно; id_1^{rr} , id_2^{rr} , id_3^{rr} , id_4^{rr} — УКН улично-дорожного перегона от автостоянки у места работы до бара, улично-дорожного перегона, на котором произошло ДТП, улично-дорожного перегона на пути в больницу и улично-дорожного перегона от автостоянки у места работы до дома соответственно; id_1^{rs} , id_2^{rs} , id_3^{rs} , id_4^{rs} — УКН автостоянки у места работы, парковки у бара, парковочного пространства приемного отделения скорой помощи и дома соответственно; id_1^{tr} , id_2^{tr} , id_3^{tr} , id_4^{tr} — государственные номерные знаки транспортных средств жителя, потерпевшего, сотрудника ГИБДД и скорой помощи; $id_{1,2}^{gd}$ — уникальный идентификатор товаров (EAN-13) листового чая и курительного табака; id_1^{ins} — уникальный идентификатор бланка ДТП; id_1^{ofn} — уникальный идентификатор начисления штрафа ГИБДД; id_1^{med} — уникальный номер электронной ме-

дицинской карты; id_1^w — уникальный номер погодной карты района в день ДТП; id_2^w — уникальный номер погодной карты района в день, принятый за эталонный; id_1^{ea} , id_2^{ea} , id_3^{ea} , id_4^{ea} , id_5^{ea} , id_6^{ea} — ИНН управляющей организации многоквартирным домом, в котором проживает житель; компании, в которой работает житель id_1^h ; бара; больницы; эксплуатационной службы, обслуживающей УДС; компании, которая владеет автопарковками соответственно; id_1^{tc} , id_2^{tc} , id_4^{tc} — уникальный сетевой адрес электронной проходной работы и парковок у работы и бара; id_3^{tc} , id_5^{tc} — уникальные номера элементов системы транспортного мониторинга соответственно; id_1^{ch} — идентификатор счета в баре (чек контрольно-кассовой машины).

Пусть также известны следующие данные о жителе с $ИНН = id_1^h$:

житель имеет нестабильную стенокардию (I20.0 по МКБ) и хронический бронхит неуточненный (J42 по МКБ); библиотека товарных противопоказаний для жителя id_1^h имеет вид: $LGC (МКБ-10 \cap ОКП) = ОКП 919316$ (Табак соусированный и ароматизированный); а библиотека противопоказанных видов потребительской деятельности для него: $LNC (МКБ-10 \cap LHN) = f_{sm}$ «употребление табака»; процесс курения кальяна жителем id_1^h осуществлялся

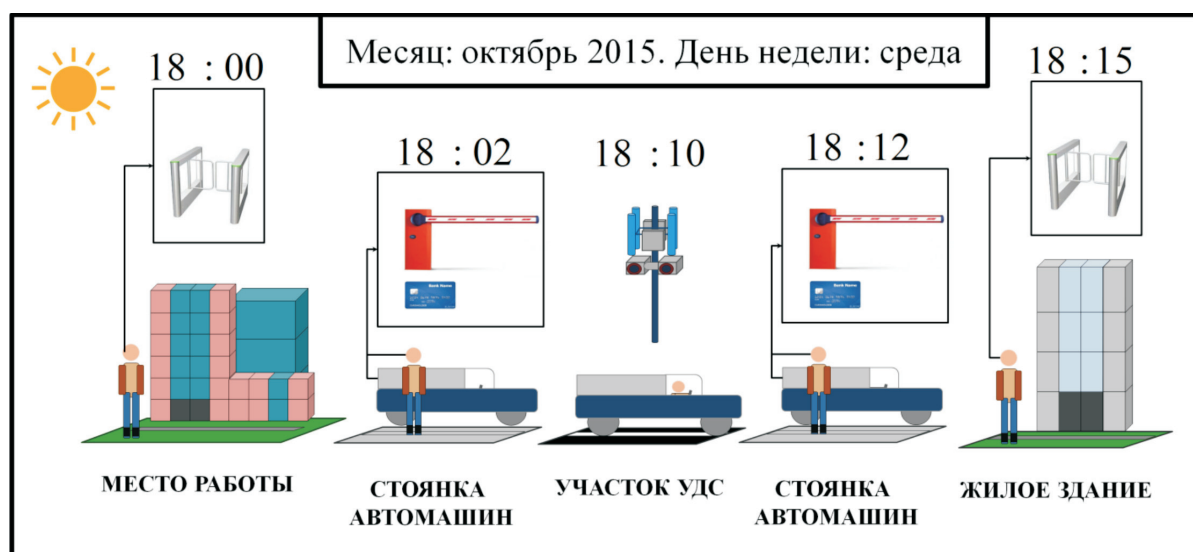


Рис. 6. Эталонное социально-экономическое поведение жителя с id_1^h



с 19:00–20:00.

Жители с $ИНН = id_2^h$ и $ИНН = id_3^h$ на момент ДТП зарегистрированных отклонений по здоровью не имеют.

Сотрудник ГИБДД id_4^h , прибывший на место аварии, установил причину ДТП, классифицировав ее в соответствии с Кодексом административных правонарушений как 12.14-1 – «Невыполнение требований ПДД подать сигнал перед началом движения, перестроением, поворотом, разворотом или остановкой».

В результате ДТП фельдшером id_5^h были классифицированы по МКБ-10 следующие травмы: S00 (поверхностная травма головы), S80 (поверхностная травма голени), S20 (поверхностная травма грудной клетки). А также установлена причина травмы V43.4 (водитель, пострадавший в результате дорожного несчастного случая на автостраде).

Со слов потерпевшего, фельдшером были установлены и другие неуточненные эффекты воздействия внешних причин, такие как T65.2 отравление табаком и никотином.

Также известно, что эксплуатационное состояние транспортных средств, участков УДС и технического обеспечения УДС в момент ДТП находилось в пределах нормы.

Требуется произвести построение графа ДТП с указанием установленных ответственными лицами фактов в порядке следования и формально описать процесс выявления причин ДТП системой

ГАТЛОСЭМИ с использованием процедуры абдуктивного вывода на временном интервале 18:00–20:45.

Решение. На основе исходных данных построим граф $S_{id_1^h}^{ts}(V_{\hat{F}_c}; T)$ ДТП с указанием установленных инспектором ГИБДД и фельдшером фактов.

Образное представление графа ДТП показано на рис. 7.

Далее перейдем к формализации процедуры абдуктивного вывода.

Гипотетические графы причин ДТП сформируем согласно процедуре (12) первого этапа.

Для этого выделим из графа суточного социально-экономического поведения пользователя $\Gamma_{id_1^h}^{04.11.2015}(V_{\hat{F}_c}; [18:00 - 00:00])$, предшествующего неблагоприятному социальному исходу $S_{id_1^h}^{ts}(V_{\hat{F}_c}; T)$, отличные от эталонных акты социально-экономической деятельности, представляющие собой вершины и заданные графом $\Gamma_{id_1^h}^{10.2015(wend)}(V_{\hat{F}_c}; [18:00 - 00:00])$. В итоге, опираясь на рис. 8, получим:

1) для интервала времени с 18:00-00:00 (рис. 8)

$$V_{id_1^h}^{date} \setminus (V_{id_1^h}^{date} \cap V_{id_1^h}^{\varepsilon, date}) \Big|_{18:00}^{00:00} = \left\{ \begin{array}{l} id_1^w; id_2^{ph}; id_1^{rr}; id_2^{rs}; id_3^o; id_2^h; id_6^h; id_{1,2}^{gd}; id_2^{mp}; \\ id_3^h; id_2^{tr}; id_4^{ph}; id_4^h; id_4^{tr}; id_5^{ph}; id_1^{ins}; id_1^{ofn}; id_1^{ins}; \\ id_5^h; id_3^{tr}; id_1^{med}; id_3^{rs}; id_4^{rs}; id_4^o \end{array} \right\};$$

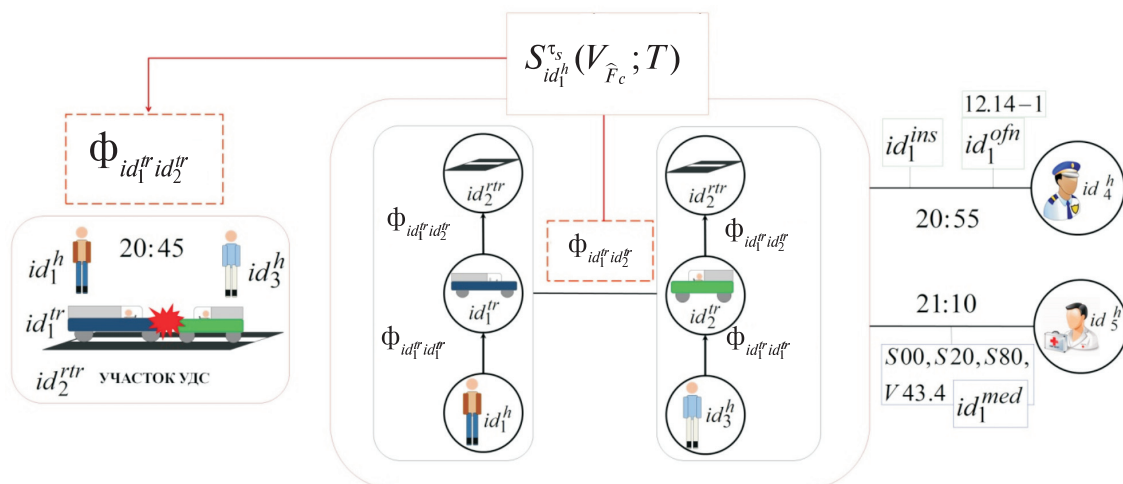


Рис. 7. Формальное представление графа ДТП

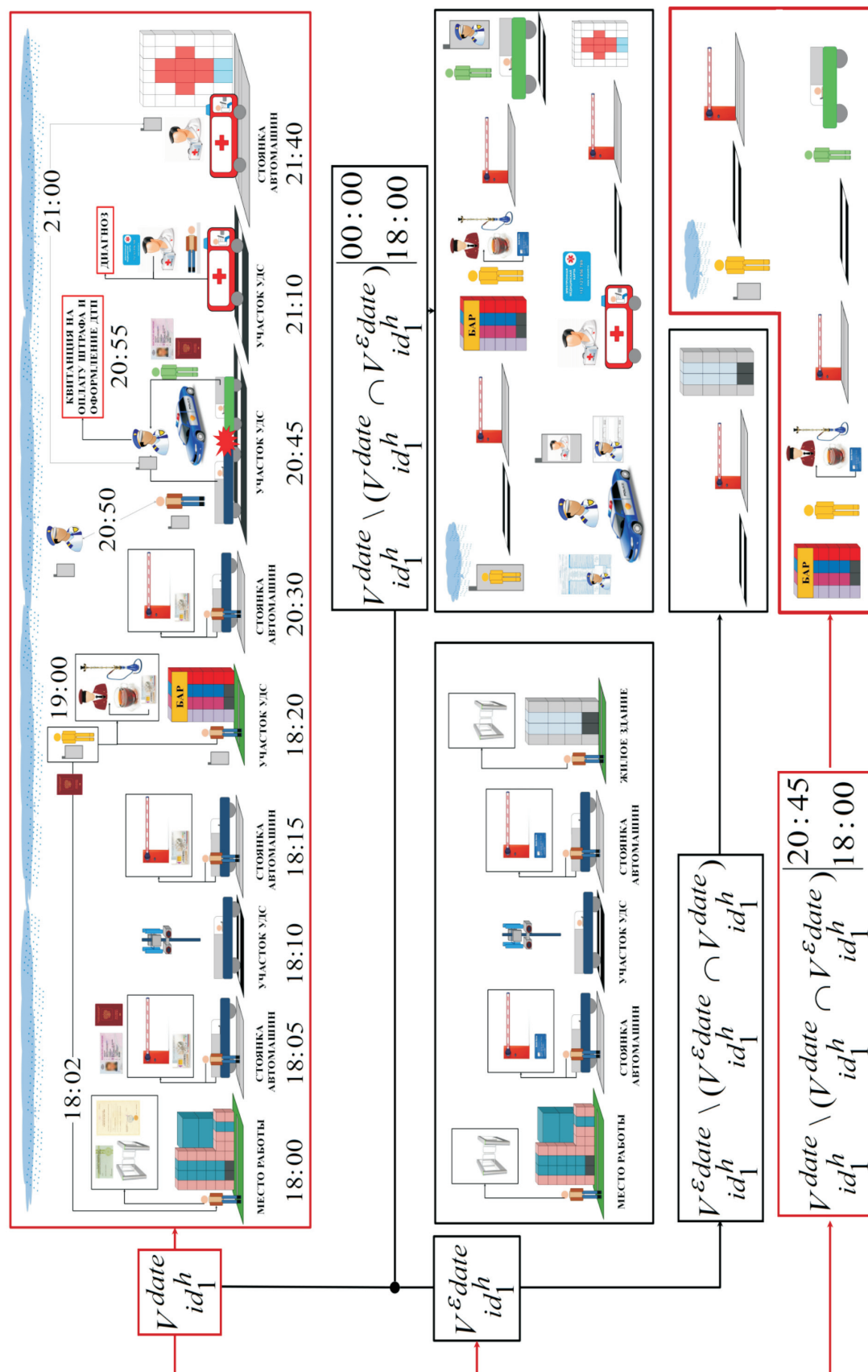


Рис. 8. Формирование гипотез процедурой абдуктивного вывода

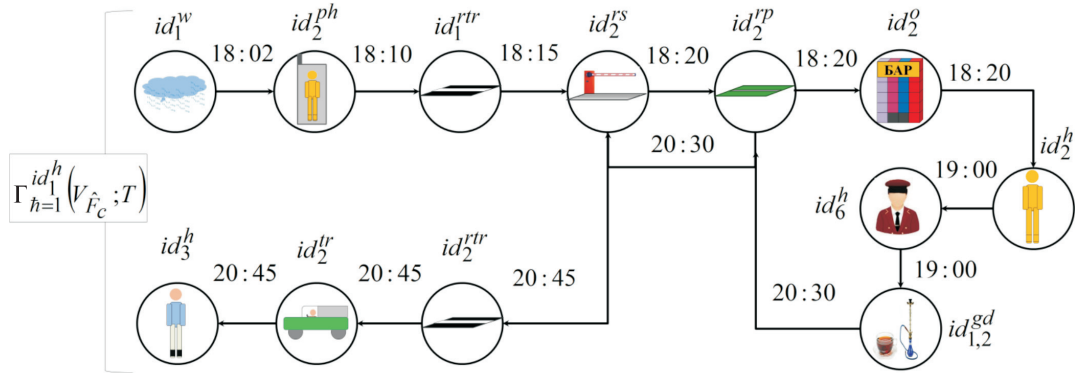


Рис. 9. Гипотетический граф ДТП

2) для требуемого интервала времени с 18:00–20:45 (рис. 8)

$$V_{h=1}^s = V_{id_1^h}^{date} \setminus (V_{id_1^h}^{date} \cap V_{id_1^h}^{e_{date}}) \Big|_{18:00}^{20:45} = \left\{ id_1^w; id_2^{ph}; id_1^{rtr}; id_2^{rs}; id_3^o; id_2^h; id_6^h; id_{1,2}^{gd}; id_2^{rs}; id_2^{rtr}; id_3^h; id_2^{tr} \right\}.$$

Таким образом, пользователь с номером id_1^h не реализовал следующие эталонные акты социально-экономической деятельности (рис. 8):

3)

$$V_{id_1^h}^{e_{date}} \setminus (V_{id_1^h}^{e_{date}} \cap V_{id_1^h}^{date}) \Big|_{18:00}^{00:00} = \{ id_4^{rtr}; id_4^{rs}; id_1^{rp}; id_1^o \}.$$

Граф $\Gamma_{h=1}^{id_1^h}(V_{Fc}; T)$, построенный на вершинах множества $V_{h=1}^s$, ГАТЛОСЭМИ отнесет в набор гипотетических графов.

Вид данного графа представим на рис. 9.

Определив гипотетический граф с использованием процедуры первого этапа абдуктивного вывода, перейдем к опре-

делению причин ДТП с использованием процедуры второго этапа абдуктивного вывода.

Для этого произведем анализ суточного социально-экономического поведения пользователя с номером id_1^h , предшествующего неблагоприятному социальному исходу $S_{id_1^h}^{rs}(V_{Fc}; T)$, на предмет его нарушения кодифицированным базам операторов контроля, согласно выражениям (7) и (8).

Для этого еще раз запишем выражение (15), подставив в него исходные данные в параметрическом виде, в итоге получим:

$$\Gamma_{id_1^h}^{date}(V_{Fc}; T) \rightarrow (V_{Fc})_{id_1^h}^{date} \cap LGC \rightarrow \rightarrow \Gamma_{h, x=n_g^h}^{id_1^h}(V_{Fc}^{x=n_g^h}; T).$$

В представленное параметрическое выражение подставим заданные количественные значения, дополнив его для лучшего понимания связью МКБ-10 с LNG, в итоге получим:

$$id_1^h : \underbrace{\{I20, J42, T65.2\}}_{\text{МКБ-10}} \xrightarrow{n_g^h} \underbrace{\{\text{ОКП 919316}\}}_{\text{LGC}} \cap \underbrace{\{id_2^{gd}\}_{id_1^h}}_{\underbrace{(V_{Fc})_{id_1^h}^{date}}_{\Gamma_{id_1^h}^{date}(V_{Fc}; T)}} \stackrel{C_{\Gamma}}{\neq 0} \rightarrow \Gamma_{h, n_g^h}^{id_1^h}(V_{Fc}^{n_g^h}; T).$$

Аналогично определим второй гипотетический граф, подставив в выражение (15) исходные данные. В итоге получим:

$$id_1^h : \underbrace{\{I20, J42, T65.2\}}_{\text{МКБ-10}} \xrightarrow{n_n^h} \underbrace{\left\{ \begin{array}{l} \text{употребление} \\ \text{табака} \end{array} \right\}}_{\text{LNC}} \cap \underbrace{\{f_{sm}\}_{id_1^h}}_{\underbrace{(V_{Fc})_{id_1^h}^{date}}_{\Gamma_{id_1^h}^{date}(V_{Fc}; T)}} \stackrel{C_{\Gamma}}{\neq 0} \rightarrow \Gamma_{h, n_n^h}^{id_1^h}(V_{Fc}^{n_n^h}; T).$$

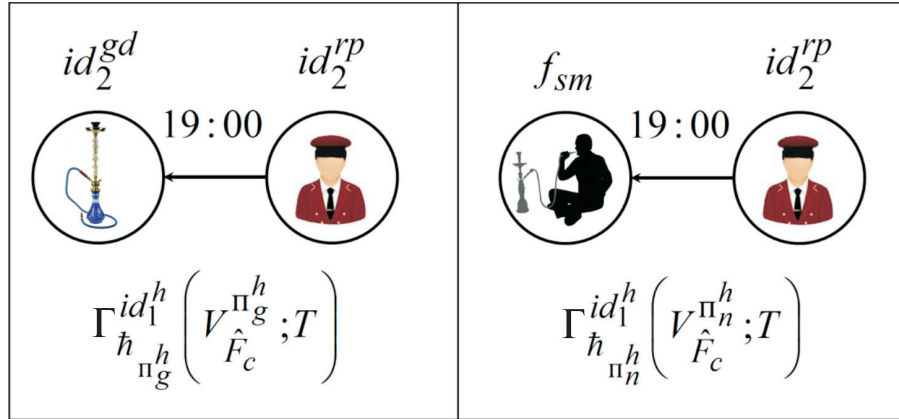


Рис. 10. Гипотетические графы второго этапа абдуктивного вывода

Произведем построение гипотетических графов $\Gamma_{\pi_g}^{id_1^h}(V_{\hat{F}_c}^{\pi_g^h}; T)$ и $\Gamma_{\pi_n}^{id_1^h}(V_{\hat{F}_c}^{\pi_n^h}; T)$, выявленных на втором этапе процедуры абдуктивного вывода. Вид данных графов представим на рис. 10.

Таким образом, в процессе процедуры абдуктивного вывода системой ГАТЛОСЭМИ будет сформирован набор гипотез согласно (10). Перепишем его для наглядности в следующем виде:

$$S_{\pi_g}^{\tau_s}(V_{\hat{F}_c}; T) \xrightarrow{\Xi_A^L} \bigvee_{\lambda=1}^{\lambda=(2^h-1)} \left(\bigwedge_{h=1}^{h=N_h} \Gamma_{\pi_g}^{\tau_s}(V_{\hat{F}_c}; T) \right)_{\lambda}, \\ h = 1, N_h.$$

Подставив в вышестоящее выражение выявленные в процессе абдуктивного анализа гипотетические графы, получим:

$$S_{id_1^h}^{\tau_s}(V_{\hat{F}_c}; T) \xrightarrow{\Xi_A^L} \bigvee_{\lambda=1}^{\lambda=5} \left(\bigwedge_{h=1}^{h=3} \Gamma_{id_1^h}^{\tau_s}(V_{\hat{F}_c}; T) \right)_{\lambda} = \\ = \Gamma_{h=1}^{id_1^h}(V_{\hat{F}_c}; T) \vee \Gamma_{h_{(n_g)}=2}^{id_1^h}(V_{\hat{F}_c}; T) \wedge \\ \wedge \Gamma_{h_{(n_g)}=3}^{id_1^h}(V_{\hat{F}_c}; T) \vee (\Gamma_{h=1}^{id_1^h}(V_{\hat{F}_c}; T) \wedge \\ \wedge \Gamma_{h_{(n_g)}=2}^{id_1^h}(V_{\hat{F}_c}; T)) \vee \\ \vee \left(\Gamma_{h=1}^{id_1^h}(V_{\hat{F}_c}; T) \wedge \Gamma_{h_{(n_g)}=3}^{id_1^h}(V_{\hat{F}_c}; T) \right) \vee \\ \vee \left(\Gamma_{h=1}^{id_1^h}(V_{\hat{F}_c}; T) \wedge \Gamma_{h_{(n_g)}=2}^{id_1^h}(V_{\hat{F}_c}; T) \wedge \right. \\ \left. \wedge \Gamma_{h_{(n_g)}=3}^{id_1^h}(V_{\hat{F}_c}; T) \right), h = 1, N_h.$$

Данный пример наглядно раскрывает работу системы класса ГАТЛОСЭМИ по

автоматическому выявлению причин произошедшего неблагоприятного социального исхода.

В настоящей статье рассмотрено технологическое обеспечение «Умных городов». Произведено формальное построение интеллектуальных городских анализаторов транспортно-логистической и социально-экономической мобильности.

Реализация рекомендательного управления в границах профилактики неблагоприятных социальных исходов позволила не только расширить набор управляющих функционалов, дополнив его функционалами контроля и интеллектуального анализа данных, но и увязать существующие российские и международные классификаторы недостающими кодифицированными библиотеками.

Последнее продемонстрировано на примере построения рекомендательного управления потребительской деятельности городского жителя при помощи Кодифицированной библиотеки товарных противопоказаний. Непосредственное формирование Кодифицированной библиотеки товарных противопоказаний осуществляется оператором контроля потребительской деятельности в ходе анализа Международной классификации болезней десятого пересмотра, Единой системы классификации и кодирования технико-экономической и социальной информации и Кодифицированной библиотеки человеческих потребностей.



Описанное выше, в контексте «Умных городов» (Smart City), позволяет говорить о возможности разрешения системных проблем рациональной организации и устойчивости городских транспортно-логистических процессов.

Согласованное использование ГАТЛОСЭМИ с интеллектуальной систе-

мой организации ГТС [22] позволит качественно повысить эффективность транспортного и логистического обслуживания современного мегаполиса.

Работа выполнена при поддержке РФФИ № 16-31-00306 в рамках проекта «Построение модели интеллектуального управления городскими транспортными потоками».

СПИСОК ЛИТЕРАТУРЫ

1. **Баринова Л.Д., Забалканская Л.Э.** Комплексный подход к управлению экологически устойчивым развитием городской транспортной системы // *Транспорт: наука, техника, управление*. 2013. № 10. С. 24–27.
2. **Paroutis S., Bennett M., Heracleous L.** A strategic view on smart city technology: The case of IBM Smarter Cities during a recession // *Technological Forecasting & Social Change*. 2013. Pp. 1–11.
3. **Каримов Т.Н., Симонова Л.А.** Интеллектуальная система поддержки принятия решения для оперативного управления транспортными потоками // *Научно-технические ведомости СПбГПУ. Информатика. Телекоммуникации. Управление*. 2011. № 1(115). С. 37–41.
4. **Малыгин И.Г., Сильников М.В.** Интеллектуальные системы транспортной безопасности // *Проблемы управления рисками в техносфере*. 2014. № 1(29). С. 1–13.
5. **Нурминский Е.А., Пугачев И.Н., Шамрай Н.Б.** Определение пассажиропотоков в региональной транспортной системе на основе модифицированных гравитационных моделей // *Наука и техника*. 2015. № 5. С. 39–45.
6. **Селиверстов Я.А.** Методы и модели построения матриц транспортных корреспонденций // *Научно-технические ведомости СПбГПУ. Информатика. Телекоммуникации. Управление*. 2015. № 2-3(217-222). С. 49–70.
7. **Селиверстов С.А., Селиверстов Я.А.** Разработка показателей интегрального развития транспортной системы мегаполиса // *Науковедение*. 2015. Т. 7. № 5. С. 1–24. DOI 10.5862/JCSTCS/5
8. **Селиверстов С.А., Селиверстов Я.А.** О методе оценки эффективности организации процесса дорожного движения мегаполиса // *Вестник транспорта Поволжья*. 2015. № 2(50). С. 91–96.
9. **Daganzo C.F.** *Logistics Systems Analysis*. Springer, 2005. 310 p.
10. **Селиверстов Я.А.** О методе анализа структуры смешанного транспортного потока на городской улично-дорожной сети // *Известия ПГУПС*. 2015. № 3. С. 104–111.
11. **Silva C.T.** Bridging Theory with Practice: An Exploratory Study of Visualization Use and Design for Climate Model Comparison // *IEEE Transaction on Visualization and Computer Graphics*. 2015. Pp. 1–16.
12. **Silva C.T.** Visual Exploration of Big Spatio-Temporal Urban Data: A Study of New York City Taxi Trips // *IEEE Transactions on Visualization and Computer Graphics*. 2013. Vol. 19(12). Pp. 2149–2158.
13. **Селиверстов Я.А.** О построении модели классификации межагентных отношений социально-экономического поведения городского населения в системах управления транспортными потоками мегаполиса // *Науковедение*. 2014. № 5. С. 188.
14. **Селиверстов Я.А., Селиверстов С.А., Стариценков А.Л.** Особенности построения системы городского транспортно-логистического мониторинга // *Известия Санкт-Петербургского государственного электротехнического университета ЛЭТИ*. 2015. Т. 1. С. 29–36.
15. **Hafedh Chourabi et al.** Understanding Smart Cities: An Integrative Framework // *45th Hawaii International Conf. on System Sciences*. 2012. Pp. 2289–2297.
16. **Batty M., Axhausen K.W.** Smart cities of the future // *The European Physical Journal Special Topics*. 2012. Vol. 214. Iss. 1. Pp. 481–518.
17. **Kakas A.C., Kowalski R., Toni F.** Abductive logic programming // *Journal of Logic and Computation*. 1992. Vol. 2(6). Pp. 719–770.
18. **Strabykin D.A.** Logical Method for Predicting Situation Development Based on Abductive Inference // *J. of Computer and Systems Sciences International*. 2013. Vol. 52(5). Pp. 759–763.
19. **Таранцев А.А.** Случайные величины и работа с ними: Учеб.-метод. пособие. СПб., 2011. 160 с.

20. **Wei Li.** Inductive Inference // *Mathematical Logic. Ser. Progress in Computer Science and Applied Logic.* 2014. Vol. 25. Pp. 235–255.

21. **Селиверстов Я.А., Селиверстов С.А.** Формальное построение цепочек транспортной активности городского населения // *Научно-технические ведомости СПбГПУ. Информатика. Телекоммуникации. Управление.* 2015.

№ 4(224). С. 91–104. DOI 10.5862/JCSTCS.224.9

22. **Селиверстов С.А., Селиверстов Я.А.** О построении интеллектуальной системы организации и развития транспортной системы мегаполиса // *Научно-технические ведомости СПбГПУ. Информатика. Телекоммуникации. Управление.* 2015. № 2-3(217-222). С. 139–161. DOI 10.5862/JCSTCS/12

REFERENCES

1. **Barinova L.D., Zabalkanskaya L.E.** Kompleksnyy podkhod k upravleniyu ekologicheskoy ustoychivym razvitiyem gorodskoy transportnoy sistemy [Integrated Approach to the Management of Environmentally Stable Development of the Urban Transportation System]. *Transport: nauka, tekhnika, upravleniye* [Transport: Science, Technology, Management], 2013, No. 10, Pp. 24–27. (rus)

2. **Paroutis S., Bennett M., Heracleous L.** A strategic view on smart city technology: The case of IBM Smarter Cities during a recession, *Technological Forecasting & Social Change*, 2013, Pp. 1–11.

3. **Karimov T.N., Simonova L.A.** Intellektualnaya sistema podderzhki prinyatiya resheniya dlya operativnogo upravleniya transportnymi potokami [Intelligent decision support system for operational management of traffic flows], *Nauchno-tehnicheskiye vedomosti SPbGPU. Informatika. Telekommunikatsii. Upravleniye* [St. Petersburg State Polytechnical University Journal. Computer Science. Telecommunication and Control]. St. Petersburg, 2011, No. 1(115), Pp. 37–41. (rus)

4. **Malygin I.G., Silnikov M.V.** Intellektualnyye sistemy transportnoy bezopasnosti [Intelligent transport safety systems]. *Problemy upravleniya riskami v tekhnosfere* [Problems of technosphere risk management], 2014, No. 1 (29), Pp. 1–13. (rus)

5. **Nurminskiy Ye.A., Pugachev I.N., Shamray N.B.** Opredeleniye passazhiropotokov v regionalnoy transportnoy sisteme na osnove modifitsirovannykh gravitatsionnykh modeley [Determination of passenger-traffic flows in regional transport system on the basis of modified gravity models]. *Nauka i tekhnika* [Science & Technology], 2015, No. 5, Pp. 39–45. (rus)

6. **Seliverstov S.A., Seliverstov Ya.A.** Metody i modeli postroyeniya matrits transportnykh korrespondentsiy [Methods and models of the construction of transport correspondence matrix]. *Nauchno-tehnicheskiye vedomosti SPbGPU. Informatika. Telekommunikatsii. Upravleniye* [St. Petersburg State Polytechnical University Journal. Computer Science.

Telecommunication and Control]. St. Petersburg, 2015, No. 2-3 (217-222), Pp. 49–70. (rus)

7. **Seliverstov S.A., Seliverstov Ya.A.** Razrabotka pokazateley integralnogo razvitiya transportnoy sistemy megapolisa [The development of indicators for integrated development of the transport system of the metropolis]. *Naukovedeniye* [Internet magazine Naukovedeniye], 2015, Vol. 7, No. 5, Pp. 1–24. (rus)

8. **Seliverstov S.A., Seliverstov Ya.A.** O metode otsenki effektivnosti organizatsii protsessa dorozhnogo dvizheniya megapolisa [A method of evaluating the effectiveness of the organization of traffic metropolis]. *Vestnik transporta Povolzhya*, 2015, No. 2 (50), Pp. 91–96. (rus)

9. **Daganzo C.F.** *Logistics Systems Analysis*. Springer, 2005, 310 p.

10. **Seliverstov Ya.A.** O metode analiza struktury smeshannogo transportnogo potoka na gorodskoy ulichno-dorozhnoy seti [On the method of structure analysis of mixed traffic flow on urban road network]. *Izvestiya PGUPS*, 2015, No. 3, Pp. 104–111. (rus)

11. **Silva C.T.** Bridging Theory with Practice: An Exploratory Study of Visualization Use and Design for Climate Model Comparison. *IEEE Transactions on Visualization and Computer Graphics*, 2015, Pp. 1–16.

12. **Silva C.T.** Visual Exploration of Big Spatio-Temporal Urban Data: A Study of New York City Taxi Trips. *IEEE Transactions on Visualization and Computer Graphics*, 2013, Vol. 19 (12), Pp. 2149–2158.

13. **Seliverstov Ya.A.** O postroyenii modeli klassifikatsii mezhagentnykh otnosheniy sotsialno-ekonomicheskogo povedeniya gorodskogo naseleniya v sistemakh upravleniya transportnymi potokami megapolisa [On the construction of classification models mezhagentnykh relations of social and economic behavior of the urban population in traffic management systems metropolis]. *Naukovedeniye* [Internet magazine Naukovedeniye], 2014, No. 5, P. 188. (rus)



14. Seliverstov Ya.A., Seliverstov S.A., Stari-chenkov A.L. Osobennosti postroyeniya sistemy gorodskogo transportno-logisticheskogo monitorin-ga [Features of construction of urban transport and logistics monitoring]. *Izvestiya SPbGETU «LETI»*, 2015, No. 1, Pp. 29–36. (rus)
15. Hafedh Chourabi et al. Understanding Smart Cities: An Integrative Framework. *45th Hawaii International Conference on System Sciences*, 2012, Pp. 2289–2297.
16. Batty M., Axhausen K.W. Smart cities of the future. *The European Physical Journal Special Top-ics*, 2012, Vol. 214, Iss. 1, Pp. 481–518.
17. Kakas A.C., Kowalski R., Toni F. Abductive logic programming, *Journal of Logic and Computa-tion*, 1992, Vol. 2(6), Pp. 719–770.
18. Strabykin D.A. Logical Method for Predict-ing Situation Development Based on Abductive In-ference. *Journal of Computer and Systems Sciences International*, 2013, Vol. 52(5), Pp. 759–763.
19. Tarantsev A.A. *Sluchaynyye velichiny i rabota s nimi* [Random variables and work with them]. St. Petersburg, 2011, 160 p. (rus)
20. Wei Li. Inductive Inference, *Mathematical Logic*. 2014. Vol. 25 of the series Progress in Com-puter Science and Applied Logic, Pp 235–255.
21. Seliverstov Ya.A., Seliverstov S.A. Formal-noye postroyeniye tsepochek transportnoy aktivnosti gorodskogo naseleniya [The Formal Development of Patterns of Daily Travel Activity Chains of the Urban Population]. *Nauchno-tehnicheskiye ve-domosti SPbGPU. Informatika. Telekommunikatsii. Upravleniye* [St. Petersburg State Polytechnical Uni-versity Journal. Computer Science. Telecommunica-tion and Control]. St. Petersburg, 2015, No. 4 (224), Pp. 91–104. (rus)
22. Seliverstov S.A., Seliverstov Ya.A. O postroy-enii intellektualnoy sistemy organizatsii i razvitiya transportnoy sistemy megapolisa [Construction of Intelligent System for the Organization and Devel-opment of Transport System Metropolis]. *Nauch-no-tehnicheskiye vedomosti SPbGPU. Informatika. Telekommunikatsii. Upravleniye* [St. Petersburg State Polytechnical University Journal. Computer Science. Telecommunication and Control]. St. Petersburg, 2015, No. 2-3 (217-222), Pp. 139–161. (rus)

СЕЛИВЕРСТОВ Ярослав Александрович — научный сотрудник Института проблем транспорта имени Н.С. Соломенко РАН.

199178, Россия, Санкт-Петербург, 12-я линия ВО, д. 13.

E-mail: maxwell_8-8@mail.ru

SELIVERSTOV Yaroslav A. *Solomenko Institute of Transport Problems of the Russian Academy of Sciences.*

199178, 12th line of Vasilievsky Island, 13, St. Petersburg, Russia.

E-mail: maxwell_8-8@mail.ru

СЕЛИВЕРСТОВ Святослав Александрович — научный сотрудник Института проблем транспорта имени Н.С. Соломенко РАН.

199178, Россия, Санкт-Петербург, 12-я линия ВО, д. 13.

E-mail: amuanator@rambler.ru

SELIVERSTOV Sviatoslav A. *Solomenko Institute of Transport Problems of the Russian Academy of Sciences.*

199178, 12th line of Vasilievsky Island, 13, St. Petersburg, Russia.

E-mail: amuanator@rambler.ru

**ЮБИЛЕЙНАЯ XV САНКТ-ПЕТЕРБУРГСКАЯ МЕЖДУНАРОДНАЯ КОНФЕРЕНЦИЯ
«РЕГИОНАЛЬНАЯ ИНФОРМАТИКА (РИ-2016)»**

САНКТ-ПЕТЕРБУРГ, 26-28 ОКТЯБРЯ 2016

26-28 октября 2016 года состоится XV Санкт-Петербургская международная конференция «Региональная информатика (РИ-2016)».

Учредители конференции «РИ-2016» Правительство и Законодательное собрание Санкт-Петербурга, Правительство Ленинградской области, Министерство образования и науки, Министерство связи и массовых коммуникаций.

Организатор конференции Санкт-Петербургский институт информатики и автоматизации РАН.

Соорганизатор конференции Санкт-Петербургский политехнический университет Петра Великого.

Основные даты конференции «РИ-2016»

1. Начало регистрации на сайте «РИ-2016» (<http://spoisu.ru/conf/ri2016>) и в Оргкомитете (e-mail: spoisu@mail.ru) – 15.02.2016.
2. Прием заявок на участие, докладов и статей для опубликования в трудах конференции до 21.10.2016.
3. Открытие конференции 26.10.2016, работа секций 26.10–28.10.2016, закрытие конференции и отъезд участников 28.10.2016.

Рабочий язык конференции – русский.

Основные темы секций конференции, статей и докладов для публикации в трудах «РИ-2016» включают около 30 наименований, в том числе следующие:

1. Телекоммуникационные системы, технологии и оборудование.
2. Информационная безопасность и информационные технологии.
3. Техническая медицина, телемедицина, оборудование, приборы и технологии.
4. Системы и технические средства безопасности и физической защиты.

Организационный взнос за участие в конференции 3000 руб.

Информация на сайте конференции <http://spoisu.ru/conf/ri2016>

Контактное лицо от СПИИРАН – старший научный сотрудник В.В. Касаткин, тел. +7 (911) 9919094.

Конференция «Инструменты и методы анализа программ 2015»



ИНСТРУМЕНТЫ И МЕТОДЫ АНАЛИЗА ПРОГРАММ
САНКТ-ПЕТЕРБУРГ, СПбПУ, 12-14, НОЯБРЬ



С 12 по 14 ноября 2015 года в Санкт-Петербургском политехническом университете Петра Великого прошла третья ежегодная научная конференция по программной инженерии «Инструменты и методы анализа программ» (Tools & Methods of Program Analysis, TMPA-2015). В программе конференции были приглашенные выступления, конкурсные доклады и семинар по системному программированию OS Day.

С приглашенными докладами на конференции выступили ведущие ученые в области программной инженерии. Николя Гюльфи (Nicolas Guelfi) из Университета Люксембурга выступил с докладом «Software Engineering Education: The Messir Approach», в котором осветил особенности подхода к обучению управлению требованиями на основе инструментария Messir. Михаил Глухих из компании JetBrains представил доклад «Язык Котлин: от разадресации нуля до умных преобразований типов», посвященный особенностям безопасного программирования с помощью нового языка программирования Котлин, разработанного в России. Бертран Мейер (Bertrand Meyer) из Высшей технической школы Цюриха (ETH) представил доклад «A theory of programs», в котором предложил простой математический аппарат для описания программ. Николай Пакулин из Института системного программирования РАН (Москва) представил доклад «Стандарты и стандартизация в программной инженерии. Какое отношение это имеет к вам?», в котором дал обзор стандартов, используемых в программной инженерии. Антон Семенченко из DPI Solutions (Минск) посвятил свое выступление «Автоматизированное тестирование вчера, сегодня, завтра — векторы развития» актуальным вопросам автоматизации тестирования программного обеспечения. Матей Михалко (Matej Michalko, DECENT, Швейцария) рассказал о новых защищенных технологиях в докладе «Blockchain Technologies: The Future of Publishing Will Have No Middleman».

В рамках основной части конференции в программный комитет было подано 35 конкурсных статей, 22 из которых были рекомендованы рецензентами для участия в виде пятнадцати полных и семи коротких докладов. В ходе представления работ на конференции шли интересные научные дискуссии. Лучшие доклады были рекомендованы программным комитетом для дальнейшей доработки и публикации в нескольких российских рецензируемых научных журналах. В данном разделе нашего журнала представлены две статьи, рекомендованные для публикации программным комитетом конференции и прошедшие дополнительное рецензирование в соответствии с правилами журнала.

Четвертая конференция «Инструменты и методы анализа программ — 2016» (Tools & Methods of Program Analysis, TMPA-2016) пройдет в ноябре 2016 года в Москве. Вся информация о программе конференции, сроках приема докладов, о приглашенных докладчиках будет оперативно отражаться на сайте конференции <http://tmpaconf.org>.

*Сопредседатель программного комитета
конференции TMPA-2015
В.М. Ицыксон*

DOI: 10.5862/JCSTCS.236.8

УДК 51-74

Д.А. Мордвинов, Ю.В. Литвинов

ОБЗОР ПРИМЕНЕНИЯ ФОРМАЛЬНЫХ МЕТОДОВ В РОБОТОТЕХНИКЕ

D.A. Mordvinov, Yu.V. Litvinov

SURVEY ON FORMAL METHODS IN ROBOTICS

Представлен обзор применения формальных методов в контексте робототехники. Рассмотрены недавние работы, посвященные спецификациям поведения роботов в терминах темпоральных логик, применению идей подхода model checking к таким системам. Также рассмотрено применение формальных методов анализа сетей Петри и моделирования поведения робототехнических систем с их помощью. Отдельное внимание уделено верификации гибридных систем, применению алгебр процессов для спецификации поведения параллельных систем, а также использованию других подходов для верификации и синтеза программ поведения роботов.

ФОРМАЛЬНЫЕ МЕТОДЫ; РОБОТОТЕХНИКА; ТЕМПОРАЛЬНЫЕ ЛОГИКИ; ФОРМАЛЬНАЯ ВЕРИФИКАЦИЯ; СИНТЕЗ ФОРМАЛЬНЫХ СИСТЕМ.

This paper is a survey of applying formal methods in the robotics field. We consider a number of recent works on robotic behavior specification in terms of temporal logics and using the model checking approach. Formal analysis techniques for Petri nets and robotics systems modeling using those methods are also considered. Verification of hybrid systems, application of process algebras for concurrent systems and other approaches for synthesis and verification of robotics controllers are described. We survey both fundamental papers that lay a foundation for the entire branches of research and recent papers from the top conferences of the last five years hoping to cover most of the actively developed research topics.

FORMAL METHODS; ROBOTICS; TEMPORAL LOGICS; FORMAL VERIFICATION; SYNTHESIS OF FORMAL SYSTEMS.

Разговор о формальных методах в большинстве случаев сопряжен с понятием опасности отказа системы или некорректного ее поведения, особенно когда речь идет о дорогостоящих системах и ошибках, приводящих к причинению вреда людям. Чаще всего формальные методы находят применение в военных, космических, медицинских, промышленных технологиях, протоколах общения узлов в компьютерных сетях, о чем написано немало литературы.

Естественным образом аналогичные проблемы появляются и в области робототехники: отказ дорогостоящих робототехнических систем может привести к весьма неприятным последствиям (особенно если речь идет о боевых, медицинских или кос-

мических роботах). Неудивительно, что применению формальных методов посвящены целые секции на крупнейших конференциях по робототехнике (таких как ICRA¹ и IROS²), на каждой из которых ежегодно публикуются десятки работ. Однако стоит отметить, что далеко не всегда они касаются тематики безопасности поведения роботов. Проблемы, решаемые авторами таких статей, чаще относятся к методам

¹ IEEE International Conference on Robotics and Automation, URL: <http://icra2015.org> (дата обращения: 30.09.2015).

² IEEE/RSJ International Conference on Intelligent Robots and Systems, URL: <http://www.iros2015.org> (дата обращения: 30.09.2015).



планирования действий и рассуждений, симуляции, управления роботами, а также их групповой координации.

Многие формализмы, построенные в контексте применения формальных методов в робототехнике, выходят за рамки апробации на компьютерных симуляторах, их воплощения работают на дорогостоящих роботах. Например, Verifiable Robotics Research Group³, авторы одного из наиболее обсуждаемых в настоящее время подходов к планированию действий робота для задачи, поставленной в терминах темпоральных логик, применяют результаты своих работ на одном из крупнейших робототехнических состязаний DARPA Challenge⁴ в составе команды Vigir⁵. Известно, что различные методы формальной верификации были применены для марсохода Curiosity, причем, как выяснилось в 2012 г., весьма успешно [1]. Существует и будет рассмотрено применение формальных методов в реабилитационной робототехнике, к созданию автономных промышленных робототехнических систем, робофутболу и т. д.

Задача данной статьи — обзор и классификация применения формальных методов к проблемам робототехники. Будут рассмотрены как самые известные работы середины 90-х гг., положившие начало целым областям исследований, так и работы с последних робототехнических конференций вплоть до 2015 г. Мы не претендуем на полноту обзора как со стороны формальных методов, так и со стороны задач робототехники, тем не менее, многие из самых обсуждаемых идей последних годов, а также самые разработанные за два десятилетия области будут здесь рассмотрены.

Статья представляет собой обзор сразу двух научных областей: формальных методов разработки и верификации, и робототехники. Структура работы отдает приори-

тет формальным методам, т. е. одна глава соответствует применению конкретного класса формальных методов во всей области робототехники. Рассматриваемые методы можно разделить на два класса: статические и динамические. Статические методы используются для решения двух задач: верификации и синтеза формальных систем. В свою очередь, существует великое множество формальных систем и исчислений, для которых применимы эти методы, из них в работе будут рассмотрены шесть больших классов: конечные системы переходов, сети Петри, алгебры процессов, гибридные системы, марковские модели и формальные логики.

Темпоральные логики и синтез конечных систем переходов

В 1995 г. в статье [2] был предложен новаторский подход к синтезу управления шагающей системой. В работе строится модель искусственной ноги робота в виде конечного автомата с начальным состоянием (start) и состояниями «под нагрузкой» (load), «толчок» (drive), «без нагрузки» (unload), «переносится вперед» (recover) и «соскользнула» (slipped). Далее, рассматривая *полное произведение* (shuffle product) четырех автоматов, авторы получают модель четырехногой шагающей системы (состоящей из 1296 состояний и 5184 переходов). Очевидно, что пространство состояний результирующей системы содержит множество некорректных состояний (например, для двуногой системы не может быть состояния, когда обе ноги переносят свой вес в одно и то же время, или во время переноса одной ноги вторая не может перейти в состояние «без нагрузки»). Для спецификации таких инвариантов (свойств живучести системы, если выражаться в терминологии [3]) авторы используют темпоральную логику ветвящегося времени (CTL, [4]). Исторически это первый пример успешного использования темпоральной логики в контексте робототехнических систем. Техники проверки модели (model checking) применены в работе в своем «стандартном» виде: по заданной модели поведения системы и ее

³ URL: <http://verifiablerobotics.com> (дата обращения: 30.09.2015).

⁴ URL: <http://www.theroboticschallenge.org> (дата обращения: 30.09.2015).

⁵ URL: <http://www.teamvigir.org> (дата обращения: 30.09.2015).

инварианту проведена формальная верификация соответствия поведения системы требованиям.

Читателю рекомендуется ознакомиться с [5] или с [6] для понимания материала данного раздела. Введем понятия и обозначения, используемые далее в статье. Для спецификации формул темпоральной логики мы будем использовать стандартную нотацию: $\Box\phi$ для оператора *next*, $\Box\phi$ для оператора *always*, $\Diamond\phi$ для оператора *eventually* и $\phi\mathcal{U}\psi$ для оператора *until*. Множество формул темпоральной логики линейного времени (LTL, [7]) вида $(\Box\Diamond p_1 \wedge \dots \wedge \Box\Diamond p_m) \Rightarrow (\Box\Diamond q_1 \wedge \dots \wedge \Box\Diamond q_n)$ будем называть, в соответствии с [8], классом *General Reactivity* (1), или *GR*(1). Класс *GR*(1) часто используется в контексте решения задачи синтеза, цель которой — построение программного формализма, удовлетворяющего какой-либо формальной спецификации (в нашем случае, спецификации в терминах темпоральной логики) [9]. По представителю класса *GR*(1) ϕ можно синтезировать автомат (при условии его существования), все траектории которого удовлетворяют ϕ , за $O(n^3)$, где n — количество вхождений атомарных предикатов в ϕ (размер формулы), в то время как для общего случая известны лишь алгоритмы, делающие это с двойной экспоненциальной сложностью [9]. Структурой Крипке [10] будем называть пятерку (S, s_0, R, AP, L) , где S — непустое конечное множество состояний, $s_0 \in S$ — начальное состояние, $R \subseteq S \times S$ — тотальное отношение на S , AP — конечное множество атомарных предикатов, $L: S \rightarrow 2^{AP}$ — функция пометок.

В 2005, 2007 и 2009 гг. авторы из лаборатории GRASP Пенсильванского университета представили работы, определившие текущее состояние области применения формальных методов в робототехнике. За последние три года (с 2013 по 2015) более половины работ, публикуемых на секциях формальных методов крупнейших робототехнических конференций (таких как ICRA и IROS), так или иначе затрагивают обсуждаемую ниже область, дополняя ее теоретическую базу или проводя эксперименты с применением теоретических наработок этой области.

В статье [11] 2005 г. обсуждается задача планирования траектории движения мобильного робота, формально удовлетворяющей спецификации LTL. Предполагается, что робот представляет собой точку в плоской области, ограниченной многоугольником — окружении робота, внешней среде. Окружение разбивается на n многоугольников (ячеек, регионов), некоторые из которых считаются *интересными*. Интересные регионы могут символизировать комнаты в здании, либо препятствия, с которыми робот не должен сталкиваться. Разбиение геометрического пространства на регионы превращает непрерывную задачу планирования движения в дискретную (*дискретизирует* ее). Каждый регион получает в соответствие свой номер, а также вводится множество предикатов $\Pi = \{\pi_1, \pi_2, \dots, \pi_n\}$, где π_i означает, что «робот находится внутри региона $i \in \{1..n\}$ ». «Карта» окружения в таком случае будет представлять собой структуру Крипке $D = (S, s_0, R, \Pi, L)$, где $S = \{s_1, s_2, \dots, s_n\}$ — множество состояний, соответствующих регионам разбиения, s_0 — состояние, соответствующее региону, в котором робот находится в начальный момент времени, R — отношение перехода, $(s_i, s_j) \in R$ тогда и только тогда, когда регион i геометрически смежен региону j , $L(s_i) := \pi_i$.

Основная идея работы — постановка задачи роботу декларативно в виде LTL-спецификации. Например, формула $\phi = \Diamond(\pi_2 \wedge \Diamond(\pi_3 \wedge \Diamond(\pi_4 \wedge (\neg\pi_2 \wedge \neg\pi_3) \cup \pi_1)))$ для стартового состояния π_1 может означать «посетить регион π_2 , затем π_3 , затем π_4 и, наконец, вернуться в регион π_1 , избегая регионов π_2 и π_3 ». Для обнаружения траектории, удовлетворяющей спецификации ϕ на структуре Крипке D окружения, достаточно построить контрпример для свойства $\neg\phi$ на модели D . В статье проводятся эксперименты с использованием верификаторов NuSMV [12] и SPIN [13]. Полученная дискретная траектория робота затем преобразуется в непрерывное управление при помощи методов теории управления, описанных в [14].

Подход, описанный в [11], может показаться непрактичным и чересчур усложнен-



ным для решаемой задачи, однако идеи, на основе которых он строится, легли в основу действительно полезного метода. Речь идет о статье 2007 г. [15]. Развитие подхода, предложенное в ней, в основном, происходит в двух направлениях: работа с датчиками и мультиагентность. Множество атомарных предикатов теперь имеет вид $AP = \Pi \cup X$, где Π — предикаты принадлежности робота региону, а $X = \{x_1, \dots, x_m\}$ — конечное множество предикатов, отражающих информацию с датчика о внешнем мире. В такой модели становится возможным построение поведения с реакцией на внешний мир, например, такая неформальная спецификация: «Стартуя в регионе 1, проверять, не плачет ли ребенок в регионах 2 или 3. Если плачущий ребенок обнаружен, необходимо отыскать родителей в регионах 4, 5 или 6». Расширение модели на мультиагентные системы происходит естественным образом, т. к. каждый робот представляет собой часть окружения другого.

Очевидно, что дискретное управление системой в случае такой модели уже не описывается одной траекторией на «карте» окружения, а представляет собой автомат, учитывающий информацию с датчиков и описывающий все возможные траектории робота, удовлетворяющие LTL-спецификации. Авторы предлагают проводить синтез автомата, описывающего всевозможные поведения робота, удовлетворяющие LTL-спецификации из класса $GR(1)$. Структура Крипке, описывающая «карту» окружения, в данном случае становится не нужна и заменяется описанием в терминах LTL: в общую спецификацию системы добавляются утверждения о геометрически смежных регионах вида $\Box(\pi_1 \Rightarrow (\bigcirc \pi_2 \wedge \bigcirc \pi_3))$, а также утверждения о состоятельности окружения, например, что в один момент времени робот должен находиться точно в одном регионе. Общий вид LTL-требования имеет вид $\varphi = \varphi_e \Rightarrow \varphi_s$, где φ_e — требования к поведению окружения, φ_s — требования к системе (группе роботов). Говоря неформально, такая спецификация подразумевает, что, если среда «играет не по правилам», т. е. в случае нарушения ожидаемых условий окружения, система не

обязана вести себя в соответствии с требованиями к ней.

Интересна трактовка синтеза дискретного управления как игры в математическом смысле между средой и системой роботов. Начиная в стартовом состоянии, среда и система по очереди делают ходы. Среда делает ход первой. Задача среды — опровергнуть спецификацию φ (поэтому она должна «играть по правилам» для того, чтобы импликация была ложной), задача системы — удовлетворять спецификации вне зависимости от того, что делает среда. Если система выигрывает, то синтез требуемого дискретного управления может быть произведен, в случае если выигрывает среда — полностью корректной стратегии не существует.

В статье 2009 г. [16] те же авторы производят синтез непрерывного управления роботом с обратной связью по дискретной модели (т. е. с учетом реакции системы на значения датчиков) с использованием подхода, описанного в [17]. Другой результат, полученный в [16], — формализация действий робота в общем виде. Множество атомарных предикатов расширяется множеством $A = \{a_1, \dots, a_k\}$ действий, которые могут выполняться роботом. Каждый из них принимает значение истины в определенном состоянии тогда и только тогда, когда робот выполняет действие, находясь в этом состоянии. Передвижение робота из региона в регион является теперь элементом множества A вместе с такими действиями, как, например, захват предмета клешней или включение/выключение камеры.

Достоинства описанного подхода очевидны: по декларативной спецификации требований к системе в выразительном формализме LTL автоматически синтезируется математически корректное «по построению» непрерывное управление группой роботов, которое гарантированно приведет к решению задачи. Тем не менее ограничений у такого подхода достаточно много. Во-первых, система может функционировать только в досконально известной среде, которая ведет себя только так, как ожидается при постановке требований. Во-вторых, хоть сложность синтеза полиноми-

альна от размера пространства состояний, сам размер пространства состояний может зависеть экспоненциально от количества входов и выходов (датчиков и действий). Однако в результате получается готовый алгоритм поведения системы, т. е. синтез можно произвести лишь один раз.

Шаги к решению как минимум второй проблемы были сделаны в работе [18] 2013 г., главный вклад которой заключается в описании подхода синтезирования дискретной стратегии робота в онлайн-режиме, «на лету». Получая на вход все ту же LTL-спецификацию в форме $GR(1)$, описывающую как «карту» окружения, так и ограничения и цели системы, алгоритм выдает на выходе не автомат, описывающий корректное поведение робота, а величину *горизонта*. Интуитивно *горизонт* — это глубина, на которую нужно просчитывать всевозможные состояния, получаемые из текущего, иначе говоря, это высота дерева игры, количество ходов, которые нужно просчитать в контексте математической игры между средой и роботом для того, чтобы локальные решения, принятые по такому неполному обсчету, гарантированно приводили бы к глобально корректному поведению. В худшем случае трудоемкость принятия решения на определенном шаге сравнится с трудоемкостью алгоритма построения автомата синтеза полной стратегии, применяемого ранее. Работоспособность такого подхода объясняется следующим фактом. В статье [8] построение автомата по LTL-спецификации производилось за два шага: итеративное построение множества «выигрышных» для системы состояний (вычисление неподвижной точки состояний с инвариантом «выигрышности»), и далее сам синтез автомата по результатам каждого шага вычисления неподвижной точки. Подсчет неподвижной точки оперирует пространствами состояний, допуская применение символических структур данных, которые показали себя на практике очень лаконично представляющими множества состояний. Интуитивно обсуждаемый алгоритм принятия решения в реальном времени использует только вычисление неподвижной точки, не проводя самого синтеза

автомата, что и дает выигрыш во времени.

Схема работы системы теперь принимает следующий вид. В каждый момент времени робот производит наблюдение за поведением среды, т. е. определяет «ход» оппонента. Это может быть сдвиг передвигающегося препятствия или сигнал «батарея почти разряжена». Среди всех «выигрышных» состояний, полученных в результате вычисления неподвижной точки, выбирается наиболее близкое в смысле пути на графе состояний с одной эвристикой исключения уже посещенных состояний, чтобы не случилось «зацикливания» системы в среде. Такая эвристика должна быть проигнорирована, если среда «нарушает правила», это повлечет за собой повторные попытки выполнения задачи роботом.

Кроме возможности планирования поведения робота в режиме реального времени, такой подход позволяет системе принять полностью реактивный характер. Другими словами, система получает возможность функционирования в изменяющемся со временем окружении (но только если характер изменений среды полностью известен на момент синтеза управления). В частности, становится возможным объезд движущегося препятствия, что ранее было слабо затронуто в статьях на тему планирования поведения по LTL-спецификации. Работа [8] содержит результаты экспериментов по синтезу поведения робота при наличии подвижного препятствия.

Существует множество статей с экспериментами, поставленными на теоретической основе обсуждаемого подхода. Например, в [19] рассказывается о применении LTL-спецификаций для планирования движений манипулятора робота-бармена за прилавком суши-ресторана. Применение именно LTL-подхода дало значительный прирост в «интеллектуальности» системы. Например, робот «понимал», что предметы, которые лежат на месте других или не дают перенести банку на другое место, должны быть временно отодвинуты в другую сторону, что раньше достигалось только методом *бэктрекинга*, описанном в статье [20]. Однако бэктрекинг работал не во всех ситуациях и был крайне неэффективен



во временном и ресурсном отношении). Работа [21] описывает планирование для мультиагентной системы раздельного сбора отходов. В [22] обсуждается планирование поведения робота-спасателя, исследующего место катастрофы с целью тушения пожаров и предоставления первой помощи выжившим.

Для формальных спецификаций поведения робота используются и другие фрагменты логики LTL. Например, в [23] предлагается использование специфичного подкласса LTL для эффективного синтеза стратегии поведения робота. Главная особенность предложенного фрагмента LTL состоит в том, что он позволяет проводить синтез стратегии прямо на изначальной модели Крипке, без использования теоретико-автоматного подхода, что может сильно уменьшить время синтеза, т. к. отпадает необходимость рассматривать композицию с автоматами для LTL-ограничения, значительно увеличивающую размер итоговой системы. Например, в тексте статьи описана ситуация, когда подход с использованием предложенного фрагмента LTL синтезирует стратегию за время порядка минуты на системе из 300 000 состояний, в то время как реализация подхода с использованием $GR(1)$ -фрагмента не закончила работу на системе из 100 000 состояний. Однако сравнение проводилось с реализацией, не использующей символические методы представления пространства состояний $GR(1)$ -системы, что является сильной стороной $GR(1)$ -подхода.

Существуют и работы, в которых для спецификации поведения роботов используются другие темпоральные логики. Например, в [24] предложен язык MASL, расширяющий LTL временными ограничениями и кванторами коалиций, что полезно для спецификации поведения мультиагентных систем. Там же можно найти подробный обзор применения различных темпоральных логик в теории мультиагентных систем.

Следует отметить, что количество опубликованных работ на тему планирования поведения робота при помощи спецификаций в темпоральных логиках на момент

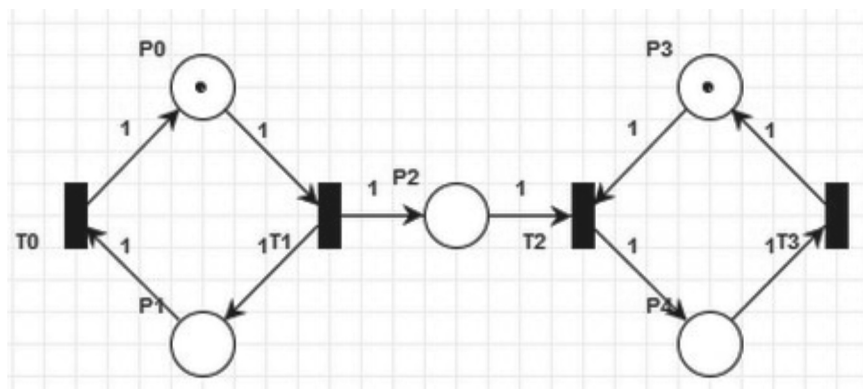
2015 г. имеет порядок сотен или даже тысяч, поэтому мы считаем уместным ограничиться здесь лишь рассмотренными статьями, т. к. они дают общую характеристику подхода. Актуальные статьи на данную тему можно найти, в частности, в списках докладов секций формальных методов конференций ICRA и IROS.

Сети Петри

Один из самых старых и некогда один из самых популярных методов формального анализа программ — сети Петри. Впервые предложенные Карлом Петри в 1962 г., они предназначались прежде всего для моделирования и анализа параллельных и распределенных систем. Они интересны тем, что имеют наглядную графическую нотацию, и многие их свойства, имеющие важное практическое значение, можно установить аналитически, зная топологию и начальное состояние сети. Сети Петри быстро стали популярны: например, в статье [25], опубликованной в 1991 г., приводится библиография из 4099 статей, рассматривающих сети Петри или их применение. Мы здесь рассмотрим лишь некоторое применение сетей Петри в робототехнике.

Сетью Петри мы, следуя [26], будем называть тройку (P, T, ϕ) , где P — конечное множество позиций, T — конечное множество переходов, а $\phi : (P \times T) \cup (T \times P) \rightarrow \mathbb{N}$ — функция потока. Маркировкой называется отображение $\mu : P \rightarrow \mathbb{N}$, ставящее в соответствие позиции количество маркеров, находящихся в данной позиции. Графически сеть Петри представляется как двудольный граф, в котором позиции обозначаются кругами (и маркеры — точками в позициях), переходы — прямоугольниками, функция потока рисуется как ребра графа (каждое ребро может быть входным или выходным и иметь вес, который пишется над ребром, см. рисунок).

Переход $t \in T$ называется активным при маркировке μ , если $\forall p \in P \phi(p, t) \leq \mu(p)$. Любой активный переход сети может сработать, в этом случае из каждой входной позиции p перехода удаляются $\phi(p, t)$ маркеров и для каждой выходной позиции перехода p' в нее добавляются $\phi(t, p')$



Визуализация сети Петри

маркеров. Можно записать срабатывание перехода как $\mu \xrightarrow{t} \mu'$, где $\mu'(p) = \mu(p) - \varphi(p, t) + \varphi(t, p) \forall p \in P$.

Практический смысл приведенных определений можно представить себе так: позиция — это состояние системы, либо предусловие для выполнения некоторого действия. Переход — это действие, выполняемое системой. Функция потока показывает поток управления или поток данных в системе, маркировка показывает состояние, в котором в определенный момент времени находится система (при этом маркер можно понимать как указатель на текущую исполняемую инструкцию, хотя, конечно, это далеко не всегда так), активный переход — действие, которое может быть исполнено, тот факт, что может сработать любой активный переход в сети — неопределенность, возникающую при параллельном исполнении (гонки), срабатывание перехода — исполнение действия и передачу управления (или данных) дальше.

Про сеть Петри с заданной топологией и маркировкой можно аналитически установить следующие свойства:

Достижимость. Существует ли последовательность срабатываний переходов из маркировки μ в маркировку μ' . Практический смысл — может ли система в принципе решить поставленную задачу, или наоборот, оказаться в нежелательном состоянии.

Ограниченность. Существует ли наперед заданное число k такое, что в любой достижимой маркировке сети в каждой позиции

находится не более k маркеров. Практический смысл — в системе никогда не произойдет переполнения буфера.

Безопасность. В каждой позиции сети в каждой достижимой маркировке находится не более одного маркера. Практический смысл — возможность не использовать буферы вообще, проверка корректности моделирования (например, движения собираемой детали по конвейеру).

Живость. Достижимость из заданной маркировки такой маркировки, в которой заданный переход может сработать. Живость вводится как для конкретного перехода, так и для сети в целом, при этом определяется шесть уровней живости, от нулевого (переход не может сработать никогда) до пятого (в любой маркировке, достижимой из заданной, существует последовательность переходов, в которой переход сработает хотя бы один раз). Практический смысл — анализ системы на возможность тупиковых ситуаций, поиск мертвого кода.

Существует несколько аналитических методов анализа, которые будут лишь упомянуты здесь в силу ограниченности объема статьи:

Анализ уравнения состояния. Сеть Петри представляется в виде матрицы инцидентности, тогда текущее состояние выражается в виде линейного уравнения, решения которого показывают количество срабатываний переходов, необходимых для перевода сети из начального состояния в некоторое заданное. Если решений нет, заданное состояние недостижимо, но если решения



есть, это еще не значит, что заданное состояние достижимо. Методы, связанные с уравнением состояния, имеют низкую вычислительную сложность, но некоторые важные свойства установить не могут.

Анализ графа (или дерева) достижимости. Вершинами такого графа являются достижимые состояния сети, ребрами — срабатывающие переходы. Такие методы позволяют установить любое из перечисленных свойств сети, но для неограниченных сетей граф будет бесконечным. Для анализа неограниченных сетей вводится граф покрытия, в котором потенциально бесконечное количество маркеров обозначается специальным символом ω , но в таком случае теряется информация о переходах сети и, например, достижимость установить уже не удастся. В любом случае, граф достижимости (покрытия) имеет размер, в общем случае экспоненциально зависящий от количества позиций в сети, так что алгоритмы анализа, как правило, имеют экспоненциальную трудоемкость.

Структурный анализ. Исследование топологии сети в поиске структур с известными свойствами. Этот метод тоже не позволяет установить все свойства.

Как видно, в общем случае доказательство практически важных свойств сети Петри имеет большую временную трудоемкость, но, тем не менее, алгоритмически разрешимо. Поэтому сети Петри имеют меньшую выразительную силу, чем машины Тьюринга, это их недостаток (поскольку сеть Петри нельзя смоделировать произвольную программу) и достоинство (для машин Тьюринга перечисленные выше свойства в общем случае алгоритмически неразрешимы). Вводятся различные расширения сетей Петри, такие как ингибиторные сети, цветные сети, сети с приоритетами, временные сети и т. д., большинство из них тьюринг-полны, так что не поддаются формальному анализу. Они, тем не менее, интересны тем, что имеют формальную семантику и могут использоваться для динамической верификации путем симуляции сети. На этом теоретическое введение заканчивается, подробнее про сети Петри можно прочитать в [27], оценки трудоем-

кости анализа сетей приводятся в [26], подробный обзор работ о разрешимости некоторых проблем в контексте сетей Петри до 1994 г. можно найти в [28], хорошее введение в сети Петри с точки зрения их приложений в промышленности приводится в [29], пример работы о применении в технологическом процессе — работа [30].

В робототехнике сети Петри используются разными способами. В работах [31, 32] сеть Петри используется для моделирования роботов-футболистов и их окружения. При этом инструментальные средства, предлагаемые авторами, позволяют сначала оценить качественные свойства системы (например, возможно ли вообще в данных условиях забить мяч в ворота), рассматривая модель как классическую сеть Петри, затем, с помощью симуляции, количественные свойства (такие как вероятность забить гол и оценка времени для решения этой задачи), рассматривая модель как временную сеть. Аналогично, статья [33] описывает симулятор для разных видов сетей Петри (временных, ингибиторных, с приоритетами), используемый для верификации и оптимизации разных технологических процессов (например, добычи руды в шахте с использованием полуавтономных погрузочно-транспортных машин). Для верификации робототехнических систем расширенные сети Петри используются и в работах [34, 35], причем в последнем случае предлагается новое расширение сетей Петри, и для их формального анализа используются техники проверки моделей.

Следующий важный класс применений сетей Петри в робототехнике — в качестве языка программирования. Все такие работы используют расширенные сети, что делает невозможным их формальный анализ, но наличие четкой семантики и наглядность упрощают разработку программ и написание для них интерпретаторов. Примеры таких работ — [36] (визуальный язык со вставками на языке Python для программирования мультиагентных систем), [37] (интерпретация ингибиторных сетей Петри для управления группой роботов, с визуальным редактором сетей и посылкой команд роботам по радиоканалу), [38] (программи-

рование одновременно нескольких роботов с помощью расширенной сети, где задания для каждого робота «нарезаются» из общей программы, с приложением опять-таки в робофутболе).

Сети Петри также используются для планирования и распределения задач между роботами в группе, хороший пример такой работы – [39], здесь действия каждого робота в группе моделируются ингибиторной сетью, определяется, может ли один робот в группе заменить другого для решения той или иной подзадачи, предлагается алгоритм построения оптимального с точки зрения времени выполнения плана работы группы.

Несколько менее типичный пример использования сети Петри – работа [40], где сеть используется для представления знаний робота при обучении по демонстрации. Человек-учитель перемещает предметы в поле зрения робота, при этом стационарные состояния предметов становятся позициями сети, а сами перемещения – переходами. После этого робот анализирует граф достижимости сети, чтобы найти кратчайшую последовательность перемещений, которые привели бы предметы из одного состояния в другое.

Communicating Sequential Processes

Речь в данном разделе пойдет о формальном языке описания параллельных систем CSP (Communicating Sequential Processes), языке *оссам*⁶, построенном на формализме CSP, и о некоторых работах, посвященных их применению в робототехнике. CSP принадлежит к классу математических теорий под общим названием *алгебры процессов* (*process algebras*) [41, 42]. В простейшем виде алгебра процессов для множества атомарных действий A определяется как замыкание A относительно операций альтернативы («+»), последовательной композиции («·») и параллельной композиции («||»).

Алгебра процессов CSP состоит из мно-

жества атомарных событий и элементарных процессов STOP и SKIP. Первый – процесс над пустым алфавитом, т. е. процесс, не взаимодействующий с другими процессами и окружением и означающий аварийную остановку работы (иначе его называют *тупиком*). Второй означает нормальное завершение работы системы. Каждое событие является атомарной сущностью, соответствующей какому-либо явлению моделируемой предметной области. Все события в модели CSP являются мгновенными, при этом не важен интервал времени, разделяющий события, важна лишь упорядоченность событий.

Для подробного ознакомления с формализмом мы отсылаем читателя к оригинальным работам [43–45]. Здесь будут неформально определены основные операции в алгебре CSP.

- *Префикс* («→»). Выражение $Q = a \rightarrow P$ конструирует новый процесс Q , который ждет события a , а затем ведет себя как P . В качестве примера приведем простейший бесконечный процесс работы часов: $CLOCK = tick \rightarrow CLOCK$. Такая запись может быть развернута бесконечной подстановкой выражения для $CLOCK$ в правую часть в последовательность $tick \rightarrow tick \rightarrow tick \rightarrow \dots$. Запись $Q = (x : B \rightarrow P(x))$ предлагает выбрать любое событие x из множества событий B , и далее процесс ведет себя как $P(x)$. В таком случае B называют *меню* процесса P .

- *Недетерминированный выбор* («□»). Выражение $P = Q \sqcap R$ конструирует процесс, который ведет себя либо как Q , либо как R , при этом выбор имеет внутреннюю природу, которая не определяется, и среда не может влиять на то, в чью пользу будет сделан этот выбор. Другими словами, процесс $(a \rightarrow Q) \sqcap (b \rightarrow R)$ может отказаться принимать a или b , и природа этого отказа носит внутренний характер.

- *Детерминированный выбор* («□»). $P = (a \rightarrow Q) \sqcap (b \rightarrow R)$ конструирует процесс P , который ведет себя как P или Q в зависимости от события, выработанного окружением. В случае, когда окружение произвело событие, принимаемое обоими операндами, выбор между ними будет про-

⁶ Страница *оссам* на сайте WoTUG, URL: <http://www.wotug.org/ossam> (дата обращения: 30.09.2015).



изведен недетерминированно.

- *Параллельная композиция по пересечению* (« $\parallel$ »). Данный оператор позволяет процессам исполняться параллельно, но только в случае, когда оба процесса к этому готовы. Более формально, $(x : B \rightarrow P(x)) \parallel (y : C \rightarrow Q(y)) = (z : (B \cap C) \rightarrow (P(z) \parallel Q(z)))$. Оператор параллельной композиции по пересечению часто применяется для описания синхронизации процессов в формализме CSP.

- *Параллельная композиция чередованием* (*interleaving*, « $\parallel\parallel$ »). В отличие от композиции по пересечению, чередование позволяет исполняться параллельно независимо друг от друга, т. е. каждое событие требует участия одного процесса вместо двух. Формально, если B — меню процесса P , а C — меню процесса Q , то $P \parallel\parallel Q = (x : B \rightarrow (P(x) \parallel\parallel Q)) \sqcap (y : C \rightarrow (P \parallel\parallel Q(y)))$, то есть события P и Q «чередуются» во времени, откуда и произошло название операции.

- *Последовательная композиция* (« $;$ »). Позволяет вести себя процессу (P ; Q) как процессу P до тех пор, пока тот не закончит свою работу, а затем как Q . В частности,

$$\begin{aligned} \text{SKIP}; P &= P, \\ \text{STOP}; P &= \text{STOP}. \end{aligned}$$

- *Соккрытие* (*concealment*, « $\backslash$ »). Процесс $P \backslash b$ ведет себя как P , за исключением того, что все вхождения b удаляются из всех траекторий исполнения P . В частности, $(b \rightarrow P) \backslash b = P \backslash b$.

- *Образ и прообраз*. Пусть f — тотальное отношение из множества событий во множество событий, причем прообраз каждого события конечен. Тогда $f(P)$ определяет процесс, который выполняет $f(a)$ тогда и только тогда, когда P выполняет a , $f^{-1}(P)$ — процесс, который выполняет a тогда и только тогда, когда P выполняет $f(a)$.

Существует множество инструментов для анализа CSP, например, FDR2⁷ и PAT⁸. FDR2, применение которого мы рассмотрим

ниже, является коммерческим инструментом проверки моделей для CSP (*refinement checker*), позволяющим формально доказать выполнимость свойств процесса, описанного на входном языке FDR2, практически полностью совпадающем с версией языка CSP самого Ч.Э. Хоара (автора CSP) из [45]. Также CSP повлиял на такие языки программирования, как оссам-π [46] и Go⁹.

Перейдем к рассмотрению применения описанного формализма к задачам робототехники. В 1998 г. была опубликована статья [47], в которой обсуждается применение формальных методов в целях безопасности на примере реабилитационной робототехники. Работа ведется в контексте бременского автономного кресла — интеллектуальной системы для передвижения людей преклонного возраста и людей с ограниченными возможностями. Предлагается использование подхода с моделированием дерева опасностей [48, 49] посредством CSP. Дерево опасностей представляет собой иерархическое представление всех рисков, которым подвергается система, начиная от самого абстрактного описания в корневых узлах до самого конкретного в листьях. Узлы-дети конкретизируют опасности родителя, объединяясь либо логическим И (родительская опасность случается тогда и только тогда, когда случаются все опасности, соответствующие узлам-детям), либо логическим ИЛИ (родительская опасность случается, когда случается хотя бы одна дочерняя опасность). Полное дерево опасностей для бременского автономного кресла имеет 170 узлов и охватывает опасности от угроз столкновения (активного и пассивного, т. е. произошедших по вине самого кресла и по причине внешних обстоятельств) до угрозы потери связи между программными и аппаратными компонентами кресла. Дерево опасностей (или его поддеревья) далее описывается на входном языке FDR2, происходит моделирование окружения и системы со случайными выработками событий опасностей. Утверж-

⁷ Домашняя страница FDR2, URL: <http://www.fsel.com> (дата обращения: 30.09.2015).

⁸ Домашняя страница PAT, URL: <http://pat.comp.nus.edu.sg> (дата обращения: 30.09.2015).

⁹ Домашняя страница Go, URL: <https://golang.org> (дата обращения: 30.09.2015).

дается, что такой подход более выразителен, чем просто постановка требований в терминах LTL (см. выше в данной статье), т. к. в общем случае дерево опасностей не выражается в LTL. Наконец, производится верификация сетевых протоколов взаимодействия компонент кресла с блокировками и двойной буферизацией, а также обсуждается другое возможное применение для подхода с деревом опасностей.

Работа [50] интересна в контексте образовательной робототехники, а также в контексте визуального моделирования [51]. В ней обсуждается разработка визуального языка для программирования роботов Surveyor SRV-1 в целях обучения студентов Университета Кента параллельному программированию, для чего ранее использовался язык оссам-п [46], основанный на формализмах π -исчисления [46] и CSP.

В работе [52] обсуждается применение языка оссам-п для построения роевой [53] системы роботов сбора отходов. Для реализации системы выбрана архитектура Брукса [54], в которой система представляется как набор уровней ответственности, при этом работоспособность верхних уровней напрямую зависит от нижних, которые могут, например, контролировать работу датчиков или обеспечивать передвижение мобильного робота, избегая при этом столкновений с препятствиями. Характерной чертой такой архитектуры является то, что отказ работы верхних уровней не влечет отказа всей системы, робот все еще в состоянии делать более «примитивные» действия. Использование оссам-п позволило представить каждый уровень ответственности как отдельный процесс, что оказалось удобным при программировании системы сбора отходов группой роботов.

Работа [55] изучает подход к планированию передвижения роботов с использованием алгебр процессов. Общая схема подхода примерно такая же, как у описанного выше в данной статье, однако вместо спецификаций LTL используются алгебры процессов.

Гибридные системы

Робототехническая система, действующая в реальном мире, всегда является *гибридной системой* в том смысле, что совмещает дискретную логику поведения, отражающую внутреннее состояние системы, с непрерывной динамикой (управление скоростями моторов, углом соединения деталей, длина тормозного пути и т. д.). При этом класс гибридных систем не ограничивается одними лишь роботами. Их использование довольно естественно в автомобиле-, авиа- и судостроении, а также в различных отраслях науки (включая такие, как экология хищников-жертв).

На тему теории гибридных систем выходят целые сборники (например, [56, 57]), создано множество инструментов анализа, включая HyTech [58], PHAVer [59], KeYmaera [60]. Мы рекомендуем работу [61] для ознакомления с общими принципами работы верификаторов гибридных систем и эмпирическое исследование о верификации и валидации в гибридных киберфизических системах [62] как хороший обзор научных работ и индустриальной практики в этой области. Интересно, что согласно результатам обзора, почти все участники опроса используют методы проверки моделей, но только для очень ограниченных частей разрабатываемых систем, тогда как основная доля работ по обеспечению качества приходится все-таки на ручное тестирование.

Рассмотрение робота как гибридной системы помогает уточнить и улучшить результаты применения других формальных методов. Например, в работе [63] вводится формализация робота как гибридной системы со сложной динамикой, к которой применяется аппарат темпоральных логик для планирования траектории, близкий к [15]. Планирование выполняется на двух уровнях: высоком (строящем примерный маршрут, максимально «хорошо» удовлетворяющий LTL-спецификации) и низком, где учитывается непрерывная динамика робота. При этом планировщик низкого уровня может влиять на решения планировщика высокого уровня, например, если проложенный маршрут физически невоз-



можен для данного робота. Предложенный авторами подход позволяет не перестраивать LTL-спецификацию при обнаружении препятствий, тем избегая вычислительно сложных операций.

В работе [64] как гибридная система рассматривается промышленная роботизированная сборочная линия. Поведение системы из манипуляторов и контейнеров описывается в виде набора дифференциальных уравнений, решается задача планирования работы системы. Подход позволяет синтезировать непрерывное управление, проводящее систему через набор дискретных состояний (таких как «взять деталь», «передвинуть контейнер» и т. д.) с учетом показаний сенсоров, при этом могут быть удобно выражены неожиданные события, например, появление препятствия в зоне действия робота. После устранения препятствия система может продолжить исполнение плана без необходимости перепланирования. Для достижения этого применяется исключение переменной времени из уравнений и ввод искусственной переменной, с которой ведется работа в уравнениях (reference variable), в которой кодируется и непрерывное, и дискретное состояние системы (пространственное положение манипулятора и задача, которую он в данный момент выполняет). Для удобного выражения последовательных и параллельных процессов применяется max-plus algebra: над множеством $\mathbf{R} \cup \infty$ вводятся операции $\oplus$ и $\otimes$:

$$\begin{aligned}a \otimes b &= a + b \\a \oplus b &= \max(a, b).\end{aligned}$$

Уравнения, описывающие поведение системы из нескольких манипуляторов и конвейера, выражаются в этой алгебре, после чего их относительно несложно анализировать.

Также для анализа гибридных систем используется дифференциальная динамическая логика ($d\mathcal{L}$). В работе [65] представляется инструмент Sphinx, позволяющий описывать модель гибридной системы на визуальном языке (профиле для диаграмм активностей и диаграмм классов языка UML), после чего модель может быть пере-

ведена в текстовую форму и проанализирована инструментом KeYmaera. Средство реализовано на платформе Eclipse Modeling Project¹⁰, поддерживает коллаборативную работу над моделью и доказательствами корректности (используются возможности Eclipse, в частности, EMF Compare¹¹), средства контроля версий. Возможности по обеспечению коллаборативной работы позволяют распределять задачи по формализации и верификации системы, эффективно используя знания членов команды, а использование визуальных языков позволяет сделать модель более наглядной и понятной. Кроме того, авторы упоминают, что возможности по реализации предметно-ориентированных языков платформы Eclipse Modeling Project позволяют описывать систему на разных языках, привычных для разных групп экспертов, и автоматически транслировать их в язык $d\mathcal{L}$.

Марковские модели

Еще одной активно развивающейся областью применения математического аппарата к задачам робототехники является теория марковских процессов принятия решений. О марковских процессах принятия решений написано довольно много литературы, в том числе на русском языке, например, книга [67]. Применение теории марковских процессов принятия решений к задаче планирования, что ближе к материалу данной статьи, довольно подробно рассмотрено в [68].

Марковским процессом принятия решений (Markov Decision Process — MDP) назовем шестерку $M = (S, A, D_0, P, R, \gamma)$, где S — конечное множество состояний; A — конечное множество действий; $D_0 : S \rightarrow \mathbb{R}$ — начальное распределение вероятности по состояниям; $P : S \times A \times S \rightarrow [0, 1]$ — функция вероятности перехода. По данным $s, s' \in S$

¹⁰ Обзор платформы на русском языке можно найти в работе [66].

¹¹ Домашняя страница средства визуального сравнения моделей EMF Compare, URL: <https://www.eclipse.org/emf/compare/> (дата обращения: 04.02.2016).

и $a \in A$ величина $P(s, a, s')$ означает вероятность перехода из состояния s в состояние s' при совершении системой действия a ; $R: S \times A \times S \rightarrow \mathbb{R}$ — функция награды. Значение $P(s, a, s')$ есть величина награды, полученной системой немедленно после перехода из состояния s в состояние s' при выполнении действия a ($s, s' \in S, a \in A$); $\gamma \in [0, 1]$ — фактор скидки, означающий важность ближайших наград относительно будущих наград.

Политикой процесса M с конечной памятью назовем отображение $f: S^* \rightarrow D(A)$, где S^* — множество конечных кортежей, все элементы которого принадлежат S , $D(X)$ — множество функций распределения вероятности над множеством X . Политика f с конечной памятью возвращает по «истории» передвижения по состояниям марковского процесса распределение вероятностей очередного действия системы. *Детерминированная политика* — политика марковского процесса, отображающая конечный путь в одно конкретное действие. В контексте робототехники обычно говорят о детерминированных политиках. Часто используется понятие *политики без памяти*, или *стационарной политики*, когда в действие отображается только текущее состояние: $f: S \rightarrow A$.

В классическом виде задачей анализа MDP является поиск политики f , максимизирующей потенциальную награду над бесконечным горизонтом, иначе говоря, ставится экстремальная задача:

$$\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t, s_{t+1}) \rightarrow \max,$$

где действие a_t выбирается по состоянию s_t в соответствие с политикой f . Значения γ , близкие к нулю, заставляют систему пренебрегать более удаленными во времени наградами, близкие к единице — учитывать их.

MDP применяются для моделирования поведения дискретных систем, поведение которых носит стохастический характер, поэтому часто используются в робототехнике. С их помощью моделируется поведение роботов и их частей (вероятности переходов в таком случае означают вероятности

успешного выполнения задачи, например, захвата предмета клешней), элементов окружения и даже поведения человека [69–71] (например, когнитивного состояния оператора робота или человека как движущегося препятствия).

В контексте формальных методов интересны расширения марковских процессов множеством атомарных предикатов AP и функцией пометок $L: S \rightarrow 2^{AP}$. Такая структура носит название *марковского процесса принятия решений с метками* (*Labeled Markov Decision Process — LMDP*). Для систем, моделируемых при помощи LMDP, часто ставится задача *проверки вероятностных моделей* (*probabilistic model checking*) [72]. Методы, применяемые для проверки вероятностных моделей, в целом аналогичны методам классической проверки моделей. Нередко ограничения ставятся на языке логики PCTL¹² или PCTL*, хотя практически во всех упомянутых ниже работах для постановки задачи роботу используется темпоральная логика линейного времени.

Рассмотрим некоторое применение методов проверки вероятностных моделей к задачам робототехники. В работе [73] описан метод синтеза политики, гарантирующей выполнение требований в форме произвольной формулы LTL с вероятностью равной единице, при этом такая политика будет оптимальной с точки зрения награды (или стоимости, что эквивалентно). Методы синтеза такой политики без учета LTL-ограничений могут быть найдены, например, в [74]. Дополнительные ограничения в форме LTL заставляют рассматривать синхронное произведение MDP системы и автомата Рабина, соответствующего LTL-ограничению, что может сильно увеличивать размер всей системы. Результаты применяются к решению задачи передвижения мобильного робота. Анализ системы в ситуации произвольных формул LTL в худшем случае имеет двойную экспоненциальную сложность, к тому же результирующая политика в общем случае не обязана иметь конечную память (последнее

¹² Probabilistic Computation Tree Logic.



все же не играет особой роли на практике). Существуют и более практичные подходы, использующие фрагменты LTL, что нередко снижает трудоемкость синтеза и анализа вероятностных систем.

Работа [75] описывает простые методы для синтеза стационарной политики передвижения мобильного робота в среде, описываемой марковским процессом принятия решений. При этом ограничения ставятся в форме логики со-safe LTL, описывающей только конечные траектории системы. Модель не поддерживает возможности задать реакцию робота на значения с датчиков. Статья [23], уже упомянутая выше, содержит описание подхода к синтезу политики с конечной памятью для ограничений в форме специфичного фрагмента LTL за полиномиальное время без использования теоретико-автоматного подхода.

В [76] приводится интересное приложение теории проверки вероятностных моделей к задаче *гибридного управления роботом (shared autonomy robot)*. Для робота, у которого есть как подсистема операторского контроля, так и подсистема автономного поведения, строится такая модель поведения системы, которая находится в состоянии парето-оптима между максимизацией вероятности выполнения задания, поставленного LTL-спецификацией, и стоимостью работы оператора. Для синтеза такой стратегии поведение робота и когнитивное состояние оператора представляется марковским процессом.

Другая часто применяемая в контексте робототехники математическая структура — *частично наблюдаемые марковские процессы принятия решений (Partially Observable Markov Decision Process — POMDP)*. От обычных MDP POMDP отличаются тем, что состояние, в котором находится система, неизвестно, и система может только поддерживать набор вероятностей нахождения в определенном состоянии, который она получает из наблюдений о поведении окружающей среды в ответ на выполнение какого-либо действия. Формально, частично наблюдаемый марковский процесс принятия решений — кортеж $(S, A, D_0, P, R, Z, O, \gamma)$, где S, A, D_0, P, R и γ соответствуют

аналогичным элементам из определения MDP, Z — конечное множество наблюдений, $O : S \rightarrow Z$ — отображение, сопоставляющее состоянию наблюдение. Очевидно, что в реальном мире часто приходится сталкиваться с ситуациями, описываемыми POMDP, что подтверждается большим числом работ, посвященных применению POMDP в робототехнике.

Формальный анализ POMDP разделяется на *количественный* и *качественный*. Задача качественного анализа POMDP — нахождение политики, удовлетворяющей цели с вероятностью, равной единице. Количественный анализ задается вопросом об удовлетворении цели с вероятностью $\lambda \in (0, 1)$. В [77] доказывается неразрешимость задачи количественного анализа. В [78] подробно описываются текущие результаты разрешимости задач анализа POMDP, в частности, приводится конструктивное доказательство разрешимости EXPTIME-полноты качественного анализа POMDP.

В [79] исследуется применение качественного анализа POMDP совместно с LTL-требованиями для классических задач планирования поведения робота. Работа [80] описывает применение темпоральной логики линейного времени и POMDP к управлению квадрокоптером, наблюдающим за ситуацией на земле по видеонаблюдению.

Другие используемые формализмы

Process Algebra for Robot Schemas. Существуют формализмы, которые не попали ни в одну из категорий выше. Один из таких формализмов — вариант алгебры процессов — PARS (Process Algebra for Robot Schemas), описанный в работах [81, 82]. Этот подход рассматривает модель программы робота, аппаратного обеспечения робота и среды как набор процессов, каждый из которых имеет набор входных и выходных переменных, преобразует входные переменные в выходные, используя данные, которые он может читать из входных портов, при этом процесс может писать в выходные порты. Каждый процесс может завершиться в двух состояниях — stop и abort, или не завершиться вовсе. Процессы комбинируются с помощью операторов «;»

(последовательное выполнение), «|» (параллельное выполнение до тех пор, пока все процессы не завершатся) и «#» (параллельное выполнение до тех пор, пока хотя один из процессов не завершится). Последовательное выполнение передает управление следующему процессу, только если предыдущий завершился в состоянии stop, что дает возможность моделировать условный оператор:

$$T = (Eq \langle a, b \rangle; P \mid Neg \langle a, b \rangle; Q),$$

где Eq — процесс, который заканчивается в состоянии stop, если его входные переменные a и b равны, и в состоянии abort в противном случае; Neg — аналогично моделирует неравенство. Циклическое исполнение моделируется через хвостовую рекурсию $T \langle a \rangle \langle b \rangle = P \langle a \rangle \langle b \rangle; T \langle b \rangle$.

Таким образом, выразительной мощности алгебры процессов достаточно, чтобы выразить любую программу. Временной аспект вводится в модель посредством процесса $Delay \langle t \rangle$, который останавливается в состоянии stop по истечении времени t , коммуникации между параллельно исполняемыми частями системы — через процессы $In \langle p \rangle \langle x \rangle$ и $Out \langle p, x \rangle$.

Ограничение на систему вводится как выражение на PARS, дополненное логическим условием на переменные, используемые процессами в выражении. Для того чтобы моделировать неопределенность, имеющуюся в реальном мире, переменные рассматриваются как случайные величины с некоторой функцией распределения. Вводится процесс $Ran \langle \Phi \rangle \langle v \rangle$, возвращающий случайную величину v с распределением Φ .

Цель введения такого формализма — подавление комбинаторного взрыва, которому подвержены многие другие методы анализа (сети Петри, подходы, основанные на темпоральных логиках). Достигается это так: рассматриваются только системы, представляющиеся в виде совокупности параллельных циклических процессов (выражаемых с помощью хвостовой рекурсии). Для них считается функция потока, показывающая, как входные параметры преобразуются в выходные за одну итерацию. Это не накладывает серьезных ограничений на модели-

руемые системы, поскольку представление в требуемом виде во многих случаях может быть построено автоматически (причем, полиномиальным относительно количества процессов алгоритмом). Далее переменные, используемые процессами, «пропускаются» через динамическую байесовскую сеть, чтобы определить вероятность удовлетворения ограничений. Таким образом, можно априори определить вероятность успешного выполнения миссии. Авторы поставили более сотни экспериментов, в результате которых установлено, что реальные значения вероятности выполнения миссии статистически соответствуют предсказанным.

Integrated Behavior-Based Control. Еще один интересный формализм, используемый для программирования роботов, — Integrated Behavior-Based Control (i2BC), формализм и архитектура, описанные в [83]. Архитектура базируется на известной в среде робототехников работе [54] и представляет программу в виде сети взаимодействующих *поведений*. Поведение преобразует входные данные в выходные, кроме того, имеет входы *активации* и *ингибиции*. Входы принимают сигнал в виде числа от нуля до единицы, при этом смысл их таков: если ингибиция равна единице, то поведение не должно работать, если нулю, то должно работать полностью. Соответственно, если активация равна единице, то наоборот, поведение должно работать, если нулю, то не должно (обычно их комбинируют как $a^*(1-i)$, где a — это активация, i — ингибиция). Помимо выхода для данных есть еще выходы активности и целевого рейтинга. Это тоже числа от нуля до единицы; активность показывает, насколько поведение активно и готово влиять на состояние системы (с учетом активации и ингибиции), целевой рейтинг — насколько поведение «удовлетворено» тем, что происходит. Наличие такой структуры позволяет разделить поток данных и поток управления и реализовать сеть из влияющих друг на друга поведений без нужды в централизованной координации. Благодаря активностям и ингибициям поведения могут перехватывать управление (например, подсистема уклонения от столкновений может подавлять



управляющее воздействие от поведения более высокого уровня, обрабатывающего действия оператора пульта дистанционного управления). Возможна и более сложная логика комбинирования поведений, для этого применяется *fusion behavior*, специальный вид поведения, имеющий несколько входов данных, активаций и ингибиций, и выдающий результат согласно некоторой внутренней функции комбинирования этих сигналов.

Как архитектура *i2BC* интересна тем, что позволяет разрабатывать сложные системы, сохраняя их модульность, и интегрировать существующие алгоритмы, «оборачивая» их в поведение. Как формализм она интересна тем, что сводима к набору конечных автоматов, для которых оказываются применимы методы проверки моделей, как описано в работе [84]. Каждое поведение представляется в виде набора из пяти автоматов, каждый из которых отвечает за обработку своего входа или выхода. Автоматы связаны друг с другом синхронизационными каналами, по которым они могут обмениваться сообщениями. Автоматное представление системы может быть получено автоматически по модели в виде сети состояний. Авторы [84] проводили эксперименты по оценке влияния отказов датчиков на свойства системы управления мобильным роботом, в результате чего выяснилось, что некоторые свойства доказываются относительно эффективно (порядка секунд на обычном настольном компьютере), но некоторые вообще не доказываются за «разумное» время (за два дня на процессоре Intel XEON 2.7 ГГц с 1024 Гб оперативной памяти).

Формальная логика. Помимо верификации уже существующих систем, формальные методы применяются при нахождении корректных «по построению» и оптимальных решений. Еще один пример такого применения (помимо описанных выше в данной статье) — решение задачи о нахождении оптимального плана реконфигурации модульных роботов путем сведения ее к задаче SAT методами матлогики [85]. Оптимальный (в смысле количества шагов соединения-рассоединения) план для групп

размером примерно в 400 роботов находился существующими решателями SAT за время порядка десятков минут.

Динамическая верификация

Рассмотренные ранее методы предназначены прежде всего для верификации программы до запуска на реальном роботе или создания корректных «по построению» программ, но в реальных условиях этого оказывается недостаточно. Даже если удалось полностью верифицировать модель, необходимо еще убедиться, что модель соответствует реальному миру, в котором будет функционировать робот. Доказать в общем случае невозможно в силу непредсказуемости реального мира, но можно проверить во время выполнения и активировать аварийный механизм поведения, если данные о реальном мире перестали соответствовать предсказанным моделью.

Пример такой работы — средство *ModelPlex*, описанное в [86]. Верифицированная модель робота, программы управления и окружающей среды здесь уже даны, цель работы — сгенерировать код верификатора, который бы запускался на роботе каждый раз, когда вырабатывается новое управляющее воздействие, проверял, что параметры соответствуют предсказанным моделью и запускал бы аварийный режим управления, если не соответствуют.

Система гибридная, в том смысле, что робот непрерывно взаимодействует с окружающей реальностью, но управление и проверки происходят в дискретные моменты времени. Авторы используют дифференциальную динамическую логику для описания моделей и записи условия корректности. Управление системой рассматривается как циклическое исполнение программы управления и верификатора в некоторые моменты времени. Верификатор проверяет сначала переход из предыдущего состояния системы в текущее (описывается ли моделью изменение показаний датчиков за время, прошедшее с последнего запуска верификатора), затем корректность выработанного управляющего воздействия, затем предсказание на момент следующего запуска верификатора: будет ли выполнен

инвариант корректности модели с учетом выработанного управляющего воздействия и возможных отклонений реальности от модели. Разумеется, отклонения от модели должны быть ограничены. Например, если под роботом провалится пол, никакой верификатор не сможет гарантировать его безопасность. В случае, если вероятное отклонение модели от реального мира получается оценить, можно формально доказать корректность и своевременность переключения на аварийный сценарий. При этом сгенерированные верификаторы оказываются, по утверждению авторов, достаточно вычислительно просты, чтобы работать на роботе в реальном времени.

Менее формальный метод обеспечения корректности предложен для системы ROS13 в [87]. Система ROS представляет программу в виде модулей (узлов), связанных каналами, по которым узлы обмениваются сообщениями. В работе предлагается вставлять между двумя взаимодействующими узлами монитор, который бы верифицировал сообщения, передающиеся по каналу, и в случае, если сообщения нарушают заданные ограничения, инициировать аварийное поведение. Монитор описывается на специальном языке, после чего генерируется код для ROS. В работе приводится пример с контролем угла наклона пейнтбольной пушки на роботе, чтобы подавить сигнал о выстреле, если робот может попасть в себя.

В настоящей статье приведен обзор использования формальных методов в робототехнике. Обзор проводился по материалам конференций ICRA и IROS последних лет, а также по данным Scopus и Google Scholar. Наиболее популярный и исследуемый подход на данный момент — постановка задач посредством LTL-спецификаций. Также активно ведутся исследования применения сетей Петри и марковских моделей. Есть и ряд альтернативных направлений, призванных избежать вычислительной сложности более популярных подходов, либо сложности специфицирования систем в таких формализмах. Также в контексте робототехники актуальна задача динамической верификации, когда корректность поведения робота контролируется в процессе его работы в реальном времени.

Следует отметить, что часто формальные методы применяются не для верификации, а для синтеза программы поведения робота по заданной модели среды и желаемым результатам. Это позволяет задавать поведение робота декларативно, и результирующая программа будет корректна по построению, однако такое построение часто требует больших вычислительных ресурсов.

Работа выполнена в рамках «Межвузовской проектной лаборатории робототехники» — совместного проекта компаний JetBrains и Кибер-Тех Лабс.

СПИСОК ЛИТЕРАТУРЫ

1. Kornfeld R.P., Prakash R., Devereaux A.S., et al. Verification and validation of the Mars science laboratory/curiosity rover entry, descent, and landing system // *J. of Spacecraft and Rockets*. 2014. Vol. 51. No. 4. Pp. 1251–1269.
2. Antonioti M., Mishra B. Discrete event models + temporal logic = supervisory controller: Automatic synthesis of locomotion controllers // *Robotics and Automation. Proc. IEEE Internat. Conf.* 1995. Vol. 2. Pp. 1441–1446.
3. Alpern B., Schneider F.B. Defining liveness // *Information processing letters*. 1985. Vol. 21. No. 4. Pp. 181–185.
4. Clarke E.M., Emerson E.A., Sistla A.P. Automatic verification of finite-state concurrent systems using temporal logic specifications // *ACM Transactions on Programming Languages and Systems*. 1986. Vol. 8. No. 2. Pp. 244–263.
5. Карпов Ю.Г. Model Checking. Верификация параллельных и распределенных программных систем. СПб.: БХВ-Петербург, 2010.
6. Clarke E.M., Grumberg O., Peled D. Model checking. MIT press, 1999.
7. Emerson E.A. Temporal and modal logic // *Handbook of Theoretical Computer Science*. Vol. B: Formal Models and Semantics. 1990. Vol. 995. No. 1072. P. 5.
8. Piterman N., Pnueli A., Sa'ar Ya. Synthesis of reactive (1) designs // *Verification, Model Checking, and Abstract Interpretation*. Springer, 2006. Pp. 364–380.
9. Pnueli A., Rosner R. On the synthesis of a reactive module // *Proc. of the 16th ACM SIGPLAN-SIGACT Symp. on Principles of*



Programming Languages. 1989. Pp. 179–190.

10. **Kripke S.A.** Semantical analysis of modal logic i normal modal propositional calculi // *Mathematical Logic Quarterly*. 1963. Vol. 9. No. 5-6. Pp. 67–96.

11. **Fainekos G.E., Kress-Gazit H., Pappas G.J.** Temporal logic motion planning for mobile robots // *Robotics and Automation. Proc. IEEE Internat. Conf.* 2005. Pp. 2020–2025.

12. **Cimatti A., Clarke E., Giunchiglia E., et al.** NuSMV 2: An opensource tool for symbolic model checking // *Computer Aided Verification*. Springer, 2002. Pp. 359–364.

13. **Holzmann G.J.** The SPIN model checker: Primer and reference manual. Addison-Wesley Reading, 2003.

14. **Belta C., Habets L.C.G.J.M.** Constructing decidable hybrid systems with velocity bounds // *Decision and Control*. 43rd IEEE Conf. on. 2004. Vol. 1. Pp. 467–472.

15. **Kress-Gazit H., Fainekos G.E., Pappas G.J.** Where's Waldo? Sensor-based temporal logic motion planning // *Robotics and Automation. IEEE Internat. Conf. on.* 2007. Pp. 3116–3121.

16. **Kress-Gazit H., Fainekos G.E., Pappas G.J.** Temporal logic-based reactive mission and motion planning // *Robotics. IEEE Transactions on.* 2009. Vol. 25. No. 6. Pp. 1370–1381.

17. **Conner D.C., Rizzi A., Choset H., et al.** Composition of local potential functions for global robot control and navigation // *Intelligent Robots and Systems. IEEE/RSJ Internat. Conf. on.* 2003. Vol. 4. Pp. 3546–3551.

18. **Livingston S.C., Murray R.M.** Just-in-time synthesis for reactive motion planning with temporal logic // *Robotics and Automation. IEEE Internat. Conf. on.* 2013. Pp. 5048–5053.

19. **He K., Lahijanian M., Kavraki L.E., et al.** Towards manipulation planning with temporal logic specifications // *Robotics and Automation. IEEE Internat. Conf. on.* 2015. Pp. 346–352.

20. **Srivastava S., Fang E., Riano L., et al.** Combined task and motion planning through an extensible planner independent interface layer // *Robotics and Automation. IEEE Internat. Conf. on.* 2014. Pp. 639–646.

21. **Raman V., Kress-Gazit H.** Synthesis for multi-robot controllers with interleaved motion // *Robotics and Automation. IEEE Internat. Conf. on.* 2014. Pp. 4316–4321.

22. **Vasile Cristian Ioan, Belta Calin.** Reactive sampling-based temporal logic path planning // *Robotics and Automation. IEEE Internat. Conf. on.* 2014. Pp. 4310–4315.

23. **Wolff E.M., Topcu U., Murray R.M.** Efficient reactive controller synthesis for a fragment of linear

temporal logic // *Robotics and Automation. IEEE Internat. Conf. on.* 2013. Pp. 5033–5040.

24. **Бугайченко Д.Ю.** Разработка и реализация методов формально-логической спецификации самонастраивающихся мультиагентных систем с временными ограничениями // Сайт математико-механического факультета СПбГУ. СПб.: СПбГУ, 2007. Т. 2010.

25. **Plünnecke H., Reisig W.** Bibliography of Petri nets 1990. Advances on Petri Nets 1991 // *Lecture Notes in Computer Science*. 1991. Vol. 524. Pp. 312–572.

26. **Yen Hsu Chun.** Introduction to Petri net theory // *Studies in Computational Intelligence*. 2006. Vol. 25. Pp. 343–373.

27. **Murata Tadao.** Petri nets: Properties, analysis and applications // *Proc. of the IEEE*. 1989. Vol. 77. No. 4. Pp. 541–580.

28. **Esparza J., Nielsen M.** Decidability Issues for Petri Nets-a Survey // *Bulletin of the European Association for Theoretical Computer Science*. 1994. Vol. 52. Pp. 245–262.

29. **Zurawski R., Zhou Meng Chu.** Petri nets and industrial applications: A tutorial // *IEEE Transactions on Industrial Electronics*. 1994. Vol. 41. No. 6. Pp. 567–583.

30. **Мартынов В.Г., Масыгин В.Б.** Применение сетей Петри при моделировании управления технологическими процессами сборочного производства // *Омский научный вестник*. 2014. № 1 (127). С. 134–137.

31. **Costelha H., Lima P.** Robotic Tasks Modeling and Analysis Based on Petri Nets. 2003.

32. **Costelha H., Lima P.** Petri net robotic task plan representation: Modelling, analysis and execution // *Autonomous Agents*. 2010. Pp. 65–91.

33. **Конюх В.Л.** Опыт применения сетей Петри для имитации поведения систем // *Имитационное моделирование. Теория и практика. ИММОД-2009*. С. 27–37.

34. **Aguiar A.J.C., Villani E.** Petri Nets and Graphic Simulation for the Validation of Collaborative Robotic Cells in Aircraft Industry // *ABCM Symp. Ser. in Mechatronics*. 2010. Vol. 4. Pp. 364–373.

35. **Fu Yujian, Drabo Mebougna.** Formal Modeling and Verification of Dynamic Reconfiguration of Autonomous Robotics Systems // *Proc. of the Internat. Conf. on Embedded Systems and Applications (ESA)*. 2014.

36. **Simon Jochen, Moldt Daniel.** PyTri, a Visual Agent Programming Language // *Algorithmen und Werkzeuge für Petrinetze*. 2010. Vol. 643. Pp. 106–111.

37. **Kashima Hideharu, Masuda Ryosuke.**

Cooperative Control of Mobile Robots Based on Petri Net // The 10th Internat. Conf. on Advanced Robotics. 2001. Pp. 399–404.

38. **Palamara F., Ziparo V.A. Locchi L., et al.** A Robotic Soccer Passing Task Using Petri Net Plans (Demo Paper) // Proc. of the 7th Internat. Joint Conf. on Autonomous Agents and Multiagent Systems. 2008. Pp. 1711–1712.

39. **Kotb Y.T., Beauchemin S.S., Barron J.L.** Petri Net-Based Cooperation In Multi-Agent Systems // Computer and Robot Vision. 4th Canadian Conf. on. 2007. Pp. 123–130.

40. **Chang Guoting Jane, Kulirc Dana.** Robot Task Learning from Demonstration Using Petri Nets // The 22nd IEEE Internat. Symp. on Robot and Human Interactive Communication. Gyeongju, Korea. 2013. Pp. 31–36.

41. **Baeten J.C.M.** A brief history of process algebra // Theoretical Computer Science. 2005. Vol. 335. No. 2. Pp. 131–146.

42. **Fokkink Wan.** Introduction to process algebra. Springer Science & Business Media, 2013.

43. **Hoare C.A.R.** Communicating sequential processes // Communications of the ACM. 1978. Vol. 21. No. 8. Pp. 666–677.

44. **Brookes S.D., Hoare C.A.R., Roscoe A.W.** A theory of communicating sequential processes // J. of the ACM. 1984. Vol. 31. No. 3. Pp. 560–599.

45. **Hoare C.A.R.** Communicating Sequential Processes. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1985.

46. **Welch P.H., Barnes F.R.M.** Communicating mobile processes // Communicating Sequential Processes. The 1st 25 Years. Springer, 2005. Pp. 175–210.

47. **Lankenau A., Meyer O.** Formal methods in robotics: Fault tree based verification // Proc. of Quality Week. Citeseer, 1999.

48. **Vesely W.E., Goldberg F.F., Roberts N.H., et al.** Fault tree handbook: Tech. Rep. DTIC Document, 1981.

49. **Hansen K.M.** Linking safety analysis to safety requirements: exemplified by railway interlocking systems. Institut for Informationsteknologi. DTU, 1996.

50. **Simpson J., Jacobsen C.L.** Visual Process-Oriented Programming for Robotics // CPA. 2008. Pp. 365–380.

51. **Кознов Д.В.** Основы визуального моделирования. М.: БИНОМ, Лаборатория знаний, 2008.

52. **Posso J.C., Sampson A.T., Simpson J., et al.** Process-Oriented Subsumption Architectures in Swarm Robotic Systems // CPA. 2011. Pp. 303–316.

53. **Sahin Erol.** Swarm robotics: From sources of inspiration to domains of application // Swarm robotics. Springer, 2005. Pp. 10–20.

54. **Brooks R.** A robust layered control system for a mobile robot // IEEE Journal on Robotics and Automation. 1986. Vol. 2. No. 1. Pp. 14–23.

55. **Varricchio V., Chaudhari P., Frazzoli E.** Sampling-based algorithms for optimal motion planning using process algebra specifications // Robotics and Automation. IEEE Internat. Conf. on. 2014. Pp. 5326–5332.

56. **Vaandrager F.W., van Schuppen J.H.** Hybrid Systems: Computation and Control // Proc. of the 2nd Internat. Workshop. HSCC'99. Berg en Dal, The Netherlands. Springer Science & Business Media, 1999. No. 1569.

57. **Lynch N., Krogh B.** Hybrid Systems: Computation and Control // Proc. of the 3rd Internat. Workshop. HSCC 2000. Pittsburgh, PA, USA. Springer Science & Business Media, 2007.

58. **Henzinger T.A., Ho Pei-Hsin, Wong-Toi H.** HyTech: A model checker for hybrid systems // Computer aided verification. Springer, 1997. Pp. 460–463.

59. **Frehse Goran.** PHAVer: Algorithmic verification of hybrid systems past HyTech // Hybrid Systems: Computation and Control. Springer, 2005. Pp. 258–273.

60. **Platzer A., Quesel J.-D.** KeYmaera: A hybrid theorem prover for hybrid systems (system description) // Automated Reasoning. Springer, 2008. Pp. 171–178.

61. **Alur R., Courcoubetis C., Halbwachs N., et al.** The algorithmic analysis of hybrid systems // Theoretical computer science. 1995. Vol. 138. No. 1. Pp. 3–34.

62. **Xi Zheng, Christine Julien, Miryung Kim, et al.** Perceptions on the State of the Art in Verification and Validation in Cyber-Physical Systems // Systems Journal, IEEE. Los-Alamitos, CA. Pp. 1–14.

63. **Matthew R.M., Lahijanian M., Kavraki L.E., et al.** Iterative temporal motion planning for hybrid systems in partially unknown environments // Proc. of the 16th Internat. Conf. on Hybrid Systems: Computation and Control. NY, USA, 2013. Pp. 353–362.

64. **Song M., Tarn T.-J., Xi N.** Integration of task scheduling, action planning, and control in robotic manufacturing systems // Proc. of the IEEE. Los-Alamitos, CA, 2000. Vol. 88. No. 7. Pp. 1097–1107.

65. **Mitsch S., Passmore G.O., Platzer A.** Collaborative Verification-Driven Engineering of Hybrid Systems // Mathematics in Computer Science. Basel, 2014. Vol. 8. Pp. 71–97.

66. **Сорокин А.В., Кознов Д.В.** Обзор Eclipse



Modeling Project // Системное программирование. СПб., 2010. Т. 5. № 1. С. 6–32.

67. Майн Х. Марковские процессы принятия решений. М., 1977.

68. Kolobov A. Planning with Markov decision processes: An AI perspective // Synthesis Lectures on Artificial Intelligence and Machine Learning. 2012. Vol. 6. No. 1. Pp. 1–210.

69. Pentland A., Liu A. Modeling and prediction of human behavior // Neural computation. 1999. Vol. 11. No. 1. Pp. 229–242.

70. Rothkopf C.A., Ballard D.H. Modular inverse reinforcement learning for visuomotor behavior // Biological cybernetics. 2013. Vol. 107. No. 4. Pp. 477–490.

71. McGhan C.L.R., Nasir A., Atkins E.M. Human intent prediction using markov decision processes // J. of Aerospace Information Systems. 2015. Pp. 1–5.

72. Baier C., Katoen J.-P., et al. Principles of model checking. Cambridge: MIT press, 2008. Pp. 2620–2649.

73. Svorenova M., Cerna I., Belta C. Optimal control of MDPs with temporal logic constraints // Decision and Control. IEEE 52nd Annual Conf. on. 2013. Pp. 3938–3943.

74. Bertsekas D.P., et al. Dynamic programming and optimal control. Athena Scientific Belmont, MA, 1995. Vol. 1.

75. Lacerda B., Parker D., Hawes N. Optimal and dynamic planning for Markov decision processes with co-safe LTL specifications // Intelligent Robots and Systems. IEEE/RSJ Internat. Conf on. 2014. Pp. 1511–1516.

76. Fu Jie, Topcu Ufuk. Pareto efficiency in synthesizing shared autonomy policies with temporal logic constraints // arXiv preprint arXiv:1412.6029. 2014.

77. Baier C., Grosser M., Bertrand N. Probabilistic omega-automata // J. of the ACM. 2012. Vol. 59. No. 1. P. 1.

78. Chatterjee K., Chmelik M., Tracol M. What is Decidable about Partially Observable Markov Decision Processes with omega-Regular Objectives // arXiv preprint arXiv:1309.2802. 2013.

79. Chatterjee K., Chmelik M., Gupta R., et al. Qualitative analysis of pomdps with temporal logic specifications for robotics applications // arXiv preprint arXiv:1409.3360. 2014.

80. Svorčnová M., Chmelik M., Leahy K., et al. Temporal logic motion planning using POMDPs with parity objectives: case study paper // Proc. of the 18th Internat. Conf. on Hybrid Systems: Computation and Control. ACM. 2015. Pp. 233–238.

81. Lyons D., Arkin R., Jiang S., et al. Performance Verification for Behavior-based Robot Missions // IEEE Transactions on Robotics. 2015. Vol. 31. No. 3. Pp. 619–636.

82. Damian M.L., Arkin R.C., Nirmal P., et al. A Software Tool for the Design of Critical Robot Missions with Performance Guarantees // Procedia Computer Science. 2013. Vol. 16. C. 888–897.

83. Proetzsch M., Luksch T., Berns K. The Behaviour-Based Control Architecture iB2C for Complex Robotic Systems. 2007.

84. Kiebusch L., Armbrust C., Berns K. Formal Verification of Behaviour Networks Including Hardware Failures // Proc. of the 13th Internat. Conf. on Intelligent Autonomous Systems. 2014.

85. Gorbenko A.A., Popov V.Yu. Programming for modular reconfigurable robots // Programming and Computer Software. 2012. Vol. 38. No. 1. Pp. 13–23.

86. Mitsch S., Platzer A. ModelPlex: Verified Runtime Validation of Verified Cyber-Physical System Models // Runtime Verification. 2014. Pp. 199–214.

87. Huang J., Erdogan C., Zhang Yi, et al. ROSRV: Runtime Verification for Robots // Runtime Verification. 2014. Pp. 247–254.

REFERENCES

1. Kornfeld R.P., Prakash R., Devereaux A.S., et al. Verification and validation of the Mars science laboratory/curiosity rover entry, descent, and landing system. *Journal of Spacecraft and Rockets*, 2014, Vol. 51, No. 4, Pp. 1251–1269.

2. Antoniotti M., Mishra B. Discrete event models + temporal logic=supervisory controller: Automatic synthesis of locomotion controllers. *Robotics and Automation, Proceedings IEEE International Conference*, 1995, Vol. 2, Pp. 1441–1446.

3. Alpern B., Schneider F.B. Defining liveness. *Information processing letters*, 1985, Vol. 21, No. 4,

Pp. 181–185.

4. Clarke E.M., Emerson E.A., Sistla A.P. Automatic verification of finite-state concurrent systems using temporal logic specifications. *ACM Transactions on Programming Languages and Systems*, 1986, Vol. 8, No. 2, Pp. 244–263.

5. Karpov Yu.G. *Model Checking. Verifikatsiya parallelnykh i raspredelennykh programmnykh sistem [Model Checking. Verification of parallel and distributed software systems]*, St. Petersburg: BKhV-Petersburg Publ., 2010. (rus)

6. Clarke E.M., Grumberg O., Peled D. *Model*

checking. MIT press, 1999.

7. **Emerson E.A.** Temporal and modal logic. *Handbook of Theoretical Computer Science, Vol. B: Formal Models and Semantics*, 1990, Vol. 995, No. 1072, P. 5.

8. **Piterman N., Pnueli A., Sa'ar Ya.** Synthesis of reactive (1) designs. *Verification, Model Checking, and Abstract Interpretation*. Springer, 2006, Pp. 364–380.

9. **Pnueli A., Rosner R.** On the synthesis of a reactive module. *Proceedings of the 16th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, 1989, Pp. 179–190.

10. **Kripke S.A.** Semantical analysis of modal logic i normal modal propositional calculi. *Mathematical Logic Quarterly*, 1963, Vol. 9, No. 5-6, Pp. 67–96.

11. **Fainekos G.E., Kress-Gazit H., Pappas G.J.** Temporal logic motion planning for mobile robots. *Robotics and Automation, Proceedings of the IEEE International Conference*, 2005, Pp. 2020–2025.

12. **Cimatti A., Clarke E., Giunchiglia E., et al.** NuSMV 2: An opensource tool for symbolic model checking. *Computer Aided Verification*. Springer, 2002, Pp. 359–364.

13. **Holzmann G.J.** *The SPIN model checker: Primer and reference manual*. Addison-Wesley Reading, 2003.

14. **Belta C., Habetts L.C.G.J.M.** Constructing decidable hybrid systems with velocity bounds. *Decision and Control. Proceedings of the 43rd IEEE Conference on*, 2004, Vol. 1, Pp. 467–472.

15. **Kress-Gazit H., Fainekos G.E., Pappas G.J.** Where'sWaldo? Sensor-based temporal logic motion planning. *Robotics and Automation, IEEE International Conference on*, 2007, Pp. 3116–3121.

16. **Kress-Gazit H., Fainekos G.E., Pappas G.J.** Temporal logic-based reactive mission and motion planning. *Robotics, IEEE Transactions on*, 2009, Vol. 25, No. 6, Pp. 1370–1381.

17. **Conner D.C., Rizzi A., Choset H., et al.** Composition of local potential functions for global robot control and navigation. *Intelligent Robots and Systems, Proceedings of the 2003 IEEE/RSJ International Conference on*. 2003, Vol. 4, Pp. 3546–3551.

18. **Livingston S.C., Murray R.M.** Just-in-time synthesis for reactive motion planning with temporal logic. *Robotics and Automation, IEEE International Conference on*, 2013, Pp. 5048–5053.

19. **He K., Lahijanian M., Kavraki L.E., et al.** Towards manipulation planning with temporal logic specifications. *Robotics and Automation, IEEE International Conference on*, 2015, Pp. 346–352.

20. **Srivastava S., Fang E., Riano L., et al.** Combined task and motion planning through an

extensible planner independent interface layer. *Robotics and Automation, IEEE International Conference on*, 2014, Pp. 639–646.

21. **Raman V., Kress-Gazit H.** Synthesis for multi-robot controllers with interleaved motion. *Robotics and Automation, IEEE International Conference on*, 2014, Pp. 4316–4321.

22. **Vasile Cristian Ioan, Belta C.** Reactive sampling-based temporal logic path planning. *Robotics and Automation, IEEE International Conference on*, 2014, Pp. 4310–4315.

23. **Wolff E.M., Topcu U., Murray R.M.** Efficient reactive controller synthesis for a fragment of linear temporal logic. *Robotics and Automation, IEEE International Conference on*, 2013, Pp. 5033–5040.

24. **Bugaychenko D.Yu.** Razrabotka i realizatsiya metodov formalno-logicheskoy spetsifikatsii samonastroyayushchikhsya multiagentnykh sistem s vremennymi ogranicheniyami [Development and implementation of methods of formal logic specification self-adjusting multi-agent systems with time constraints]. *Sayt matematiko-mekhanicheskogo fakulteta SPbGU [Website of the Faculty of Mathematics and Mechanics St. Petersburg State University]*. SPbGU, 2007, Vol. 2010. (rus)

25. **Plünnecke H., Reisig W.** Bibliography of Petri nets 1990. Advances on I'rtri Net.s 1991, *Lecture Notes in Cornputer Science*, 1991, Vol. 524, Pp. 312–572.

26. **Yen Hsu Chun.** Introduction to Petri net theory. *Studies in Computational Intelligence*, 2006, Vol. 25, Pp. 343–373.

27. **Murata Tadao.** Petri nets: Properties, analysis and applications. *Proceedings of the IEEE*, 1989, Vol. 77, No. 4, Pp. 541–580.

28. **Esparza J., Nielsen M.** Decidability Issues for Petri Nets-a Survey. *Bulletin of the European Association for Theoretical Computer Science*, 1994, Vol. 52, Pp. 245–262.

29. **Zurawski R., Zhou Meng Chu.** Petri nets and industrial applications: A tutorial. *IEEE Transactions on Industrial Electronics*, 1994, Vol. 41, No. 6, Pp. 567–583.

30. **Martynov V.G., Masyagin V.B.** Primeneniye setey Petri pri modelirovanii upravleniya tekhnologicheskimi protsessami sborochnogo proizvodstva [Application of Petri network for modeling of process control of assembly production]. *Omskiy nauchnyy vestnik [Omsk Scientific Bulletin]*, 2014, No. 1 (127), Pp. 134–137. (rus)

31. **Costelha H., Lima P.** *Robotic Tasks Modeling and Analysis Based on Petri Nets*, 2003.

32. **Costelha H., Lima P.** Petri net robotic task plan representation: Modelling, analysis and



execution. *Autonomous Agents*, 2010, Pp. 65–91.

33. **Konyukh V.L.** Opyt primeneniya setey Petri dlya imitatsii povedeniya system [Experience of using Petri nets to simulate the behavior of systems]. *Imitatsionnoye modelirovaniye. Teoriya i praktika [Simulation. Theory and practice. IMMOD-2009]*, 2009, Pp. 27–37. (rus)

34. **Aguiar A.J.C., Villani E.** Petri Nets and Graphic Simulation for the Validation of Collaborative Robotic Cells in Aircraft Industry. *ABCM Symposium Series in Mechatronics*, 2010, Vol. 4, Pp. 364–373.

35. **Fu Yujian, Drabo Mebougna.** Formal Modeling and Verification of Dynamic Reconfiguration of Autonomous Robotics Systems. *Proceedings of the International Conference on Embedded Systems and Applications (ESA)*, 2014.

36. **Simon Jochen, Moldt Daniel.** PyTri, a Visual Agent Programming Language. *Algorithmen und Werkzeuge für Petrinetze*, 2010, Vol. 643, Pp. 106–111.

37. **Kashima Hideharu, Masuda Ryosuke.** Cooperative Control of Mobile Robots Based on Petri Net. *The 10th International Conference on Advanced Robotics*, 2001, Pp. 399–404.

38. **Palamara F., Ziparo V.A. Locchi L., et al.** A Robotic Soccer Passing Task Using Petri Net Plans (Demo Paper). *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems*, 2008, Pp. 1711–1712.

39. **Kotb Y.T., Beauchemin S.S., Barron J.L.** Petri Net-Based Cooperation In Multi-Agent Systems. *Computer and Robot Vision, Proceedings of the 4th Canadian Conference on*, 2007, Pp. 123–130.

40. **Chang Guoting Jane, Kulirc Dana.** Robot Task Learning from Demonstration Using Petri Nets. *The 22nd IEEE International Symposium on Robot and Human Interactive Communication*, Gyeongju, Korea, 2013, Pp. 31–36.

41. **Baeten J.C.M.** A brief history of process algebra. *Theoretical Computer Science*, 2005, Vol. 335, No. 2, Pp. 131–146.

42. **Fokkink Wan.** *Introduction to process algebra*. Springer Science & Business Media, 2013.

43. **Hoare C.A.R.** Communicating sequential processes. *Communications of the ACM*, 1978, Vol. 21, No. 8, Pp. 666–677.

44. **Brookes S.D., Hoare C.A.R., Roscoe A.W.** A theory of communicating sequential processes. *Journal of the ACM*, 1984, Vol. 31, No. 3, Pp. 560–599.

45. **Hoare C.A.R.** *Communicating Sequential Processes*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1985.

46. **Welch P.H., Barnes F.R.M.** Communicating

mobile processes. *Communicating Sequential Processes. The 1st 25 Years*. Springer, 2005, Pp. 175–210.

47. **Lankenau A., Meyer O.** Formal methods in robotics: Fault tree based verification, *Proc. of Quality Week*, Citeseer, 1999.

48. **Vesely W.E., Goldberg F.F., Roberts N.H., et al.** *Fault tree handbook: Tech. Rep.*, DTIC Document, 1981.

49. **Hansen K.M.** *Linking safety analysis to safety requirements: exemplified by railway interlocking systems*. Institut for Informationsteknologi, DTU, 1996.

50. **Simpson J., Jacobsen C.L.** Visual Process-Oriented Programming for Robotics. *CPA*, 2008, Pp. 365–380.

51. **Koznov D.V.** *Osnovy vizualnogo modelirovaniya [Fundamentals of Visual Modeling]*. Moscow: BINOM, Laboratoriya znaniy Publ., 2008. (rus)

52. **Posso J.C., Sampson A.T., Simpson J., et al.** Process-Oriented Subsumption Architectures in Swarm Robotic Systems. *CPA*, 2011, Pp. 303–316.

53. **Sahin Erol.** Swarm robotics: From sources of inspiration to domains of application. *Swarm robotics*, Springer, 2005, Pp. 10–20.

54. **Brooks R.** A robust layered control system for a mobile robot. *IEEE Journal on Robotics and Automation*, 1986, Vol. 2, No. 1, Pp. 14–23.

55. **Varricchio V., Chaudhari P., Frazzoli E.** Sampling-based algorithms for optimal motion planning using process algebra specifications. *Robotics and Automation, IEEE International Conference on*, 2014, Pp. 5326–5332.

56. **Vaandrager F.W., van Schuppen J.H.** Hybrid Systems: Computation and Control. *Proceedings of the 2nd International Workshop, HSCC'99*, Berg en Dal, The Netherlands. Springer Science & Business Media, 1999, No. 1569.

57. **Lynch N., Krogh B.** Hybrid Systems: Computation and Control. *Proceedings of the 3rd Internat. Workshop, HSCC 2000*, Pittsburgh, PA, USA. Springer Science & Business Media, 2007.

58. **Henzinger T.A., Ho Pei-Hsin, Wong-Toi H.** HyTech: A model checker for hybrid systems. *Computer aided verification*. Springer, 1997, Pp. 460–463.

59. **Frehse Goran.** PHAVer: Algorithmic verification of hybrid systems past HyTech. *Hybrid Systems: Computation and Control*. Springer, 2005, Pp. 258–273.

60. **Platzer A., Quesel J.-D.** KeYmaera: A hybrid theorem prover for hybrid systems (system description). *Automated Reasoning*. Springer, 2008, Pp. 171–178.

61. Alur R., Courcoubetis C., Halbwachs N., et al. The algorithmic analysis of hybrid systems. *Theoretical computer science*. 1995, Vol. 138, No. 1, Pp. 3–34.
62. Xi Zheng, Christine Julien, Miryung Kim, et al. Perceptions on the State of the Art in Verification and Validation in Cyber-Physical Systems. *Systems Journal, IEEE*. Los-Alamitos, CA, Pp. 1–14.
63. Matthew R.M., Lahijanian M., Kavraki L.E., et al. Iterative temporal motion planning for hybrid systems in partially unknown environments. *Proceedings of the 16th International Conference on Hybrid Systems: Computation and Control*. New York, USA, 2013, Pp. 353–362.
64. Song M., Tarn T.-J., Xi N. Integration of task scheduling, action planning, and control in robotic manufacturing systems. *Proceedings of the IEEE*. Los-Alamitos, CA, 2000, Vol. 88, No. 7, Pp. 1097–1107.
65. Mitsch S., Passmore G.O., Platzer A. Collaborative Verification-Driven Engineering of Hybrid Systems. *Mathematics in Computer Science*. Basel, 2014, Vol. 8, Pp. 71–97.
66. Sorokin A.V., Koznov D.V. Obzor Eclipse Modeling Project [Eclipse Modeling Project Overview]. *Sistemnoye programmirovaniye [System Programming]*. St. Petersburg, 2010, Vol. 5, No. 1, Pp. 6–32. (rus)
67. Mayn Kh. *Markovskiye protsessy prinyatiya resheniy [Markov decision processes]*. Moscow, 1977. (rus)
68. Kolobov A. Planning with Markov decision processes: An AI perspective. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 2012, Vol. 6, No. 1, Pp. 1–210.
69. Pentland A., Liu A. Modeling and prediction of human behavior. *Neural computation*, 1999, Vol. 11, No. 1, Pp. 229–242.
70. Rothkopf C.A., Ballard D.H. Modular inverse reinforcement learning for visuomotor behavior. *Biological cybernetics*, 2013, Vol. 107, No. 4, Pp. 477–490.
71. McGhan C.L.R., Nasir A., Atkins E.M. Human intent prediction using markov decision processes. *Journal of Aerospace Information Systems*, 2015, Pp. 1–5.
72. Baier C., Katoen J.-P., et al. *Principles of model checking*. Cambridge: MIT press, 2008, Pp. 2620–2649.
73. Svorenova M., Cerna I., Belta C. Optimal control of MDPs with temporal logic constraints. *Decision and Control (CDC), IEEE 52nd Annual Conference on*, 2013, Pp. 3938–3943.
74. Bertsekas D.P., et al. *Dynamic programming and optimal control*. Athena Scientific Belmont, MA, 1995, Vol. 1.
75. Lacerda B., Parker D., Hawes N. Optimal and dynamic planning for Markov decision processes with co-safe LTL specifications. *Intelligent Robots and Systems, IEEE/RSJ International Conference on*, 2014, Pp. 1511–1516.
76. Fu Jie, Topcu Ufuk. *Pareto efficiency in synthesizing shared autonomy policies with temporal logic constraints*, arXiv preprint arXiv:1412.6029, 2014.
77. Baier C., Grosser M., Bertrand N. Probabilistic omega-automata. *Journal of the ACM*, 2012, Vol. 59, No. 1, P. 1.
78. Chatterjee K., Chmelik M., Tracol M. *What is Decidable about Partially Observable Markov Decision Processes with omega-Regular Objectives*, arXiv preprint arXiv:1309.2802, 2013.
79. Chatterjee K., Chmelik M., Gupta R., et al. *Qualitative analysis of pomdps with temporal logic specifications for robotics applications*, arXiv preprint arXiv:1409.3360, 2014.
80. Svorěnová M., Chmelik M., Leahy K., et al. Temporal logic motion planning using POMDPs with parity objectives: case study paper. *Proceedings of the 18th International Conference on Hybrid Systems: Computation and Control*, ACM, 2015, Pp. 233–238.
81. Lyons D., Arkin R., Jiang S., et al. Performance Verification for Behavior-based Robot Missions. *IEEE Transactions on Robotics*, 2015. Vol. 31, No. 3, Pp. 619–636.
82. Lyons D.M., Arkin R.C., Nirmal P., et al. A Software Tool for the Design of Critical Robot Missions with Performance Guarantees. *Procedia Computer Science*, 2013, Vol. 16, Pp. 888–897.
83. Proetzsch M., Luksch T., Berns K. *The Behaviour-Based Control Architecture iB2C for Complex Robotic Systems*, 2007.
84. Kiebusch L., Armbrust C., Berns K. Formal Verification of Behaviour Networks Including Hardware Failures. *Proceedings of the 13th International Conference on Intelligent Autonomous Systems*, 2014.
85. Gorbenko A.A., Popov V.Yu. Programming for modular reconfigurable robots. *Programming and Computer Software*, 2012, Vol. 38, No. 1, Pp. 13–23.
86. Mitsch S., Platzer A. ModelPlex: Verified Runtime Validation of Verified Cyber-Physical System Models. *Runtime Verification*, 2014, Pp. 199–214.
87. J. Huang, C. Erdogan, Yi Zhang, et al. ROSRV: Runtime Verification for Robots. *Runtime Verification*, 2014, Pp. 247–254.

МОРДВИНОВ Дмитрий Александрович — аспирант кафедры системного программирования
математико-механического факультета Санкт-Петербургского государственного университета.

199034, Россия, Санкт-Петербург, Университетская наб., д. 7-9.

E-mail: mordvinov.dmitry@gmail.com

MORDVINOV Dmitrii A. *St. Petersburg State University.*

199034, Universitetskaya emb., 7-9, St. Petersburg, Russia.

E-mail: mordvinov.dmitry@gmail.com

ЛИТВИНОВ Юрий Викторович — старший преподаватель кафедры системного программирования
математико-механического факультета Санкт-Петербургского государственного университета.

199034, Россия, Санкт-Петербург, Университетская наб., д. 7-9.

E-mail: yurii.litvinov@gmail.com

LITVINOV Yurii V. *St. Petersburg State University.*

199034, Universitetskaya emb., 7-9, St. Petersburg, Russia.

E-mail: yurii.litvinov@gmail.com

DOI: 10.5862/JCSTCS.236.9

УДК 004.05

М.К. Ермаков

ПРОВЕДЕНИЕ ДИНАМИЧЕСКОГО АНАЛИЗА ИСПОЛНЯЕМОГО КОДА ФОРМАТА ARM ELF НА ОСНОВЕ СТАТИЧЕСКОГО БИНАРНОГО ИНСТРУМЕНТИРОВАНИЯ

М.К. Ermakov

DYNAMIC ANALYSIS OF ARM ELF EXECUTABLE CODE USING STATIC BINARY INSTRUMENTATION

Статья посвящена возможностям применения статического инструментирования исполняемых файлов и файлов динамических библиотек в формате ELF для проведения динамического анализа программ. Проведен обзор существующих решений в данной и смежных областях. Предложен подход к обработке бинарных файлов ARM ELF. Описаны основные особенности предлагаемого подхода: разметка целевых файлов и генерация внедряемого кода на основе пользовательских спецификаций, использование промежуточного языка описания инструкций ассемблера ARM, компоновка итоговых инструментированных файлов исполняемого кода. Приведены результаты применения реализации данного подхода для создания модулей трассировки помеченных данных, используемых в рамках инструмента автоматического поиска дефектов Avalanche.

ДИНАМИЧЕСКИЙ АНАЛИЗ; БИНАРНОЕ ИНСТРУМЕНТИРОВАНИЕ; АРХИТЕКТУРА ARM.

Dynamic program analysis methods are widely used in a broad range of activities related to software development; practical implementations of dynamic analysis rely on various code transformation and monitoring techniques. In this paper we focus on one of these techniques, static binary code instrumentation. We provide an overview of the existing tools implementing this technique and show that there are no tools directly applicable to our platform of choice, i.e., ARM/Linux and ELF binary format. We present an approach to perform static binary instrumentation for the platform in question and describe in detail the following points: user-specified instrumentation code and insertion point mapping; intermediate instruction representation used in instrumentation engine; code insertion process; offset correction process. Finally we describe a set of practical experiments of applying static binary instrumentation to Avalanche, a dynamic program analysis tool performing automatic input generation and bug discovery.

DYNAMIC ANALYSIS; BINARY INSTRUMENTATION; ARM ARCHITECTURE.

В настоящее время существует большое количество систем и инструментов, производящих автоматическую и полуавтоматическую обработку программного обеспечения (ПО) с целью анализа, профилирования программ, вскрытия алгоритмов или, наоборот, внедрения блоков препятствования взлому.

Приведенные примеры взаимодействия с ПО относятся в первую очередь к аспектам времени выполнения целевой программы. На практике реализация подобной функциональности подразумевает

произведение дополнительных действий во время работы целевой программы: либо на уровне внешнего системного инструмента, имеющего возможность отслеживать и влиять на поведение программы, либо на уровне модифицированного исполняемого кода программы или исполняемого кода, внедренного в программу посредством *инструментирования*.

Различают три схемы проведения инструментирования: модификацию исходного кода программы, модификацию исполняемого кода программы во время



выполнения (динамическое инструментирование) и модификацию файлов исполняемого кода перед выполнением (статическое инструментирование).

Модификация исходного кода программы требует наличия исходного кода и позволяет описывать требуемую функциональность высокоуровневыми конструкциями языков программирования. Системы инструментирования исходного кода позволяют делегировать решение практических подзадач изменения программ системам компиляции и сборки, что обеспечивает отсутствие затрат на поддержку различных платформ. В то же время особенности процесса компиляции и оптимизации кода могут негативно влиять на точность анализа, если требуется извлечение низкоуровневой информации о ходе выполнения программы.

Инструментирование исполняемого кода во время выполнения программы предоставляет высокую точность модификации, заниженные затраты на непосредственный разбор кода и возможность обработки кода, который загружается или создается динамически. В то же время разбор кода и инструментирование проводятся на каждом запуске программы. Для методов анализа, требующих проведение множества запусков, данная особенность негативно влияет на производительность по сравнению с другими режимами инструментирования.

Статическое инструментирование исполняемого кода имеет большую область применения, по сравнению с инструментированием исходного кода, и позволяет легко осуществлять доступ к низкоуровневой информации о выполнении программы. В то же время задача разбора исполняемого кода до запуска программы является достаточно сложной, особенно при работе с обфусцированным или защищенным кодом.

Среди рассмотренного выше применения данных систем можно выделить в высокой степени востребованную задачу поиска дефектов и уязвимостей в программах при наличии только исполняемого кода. Актуальность этой задачи распространяется на мобильные платформы (например, Tizen

и Android), программное обеспечение для которых развивается крайне активно. Большинство мобильных устройств работает на базе процессорной архитектуры ARM. Дополнительно, мобильные устройства обычно имеют ограниченные объемы ресурсов. Как было отмечено выше, динамическое инструментирование повышает объем накладных расходов по сравнению со статическими методами.

Тем самым разумно предположить, что для решения задачи анализа программ по исполняемому коду можно эффективно использовать методы статического инструментирования исполняемого кода. К сожалению, современные свободно доступные системы статического инструментирования не предоставляют поддержки архитектуры ARM. Создание системы инструментирования, сравнимой по функциональности с существующими аналогами, является актуальной задачей.

В рамках данной статьи будет рассматриваться подход к проведению статического инструментирования исполняемого кода в формате ARM ELF и особенности программной системы, реализующей данный подход.

Обзор существующих систем инструментирования

Для большинства существующих систем инструментирования центральным принципом является предоставление возможностей по проведению настраиваемой обработки целевых программ: блоки исполняемого кода, в которых производится модификация или внедрение дополнительного кода, а также функциональная логика необходимых изменений задаются разработчиком модуля инструментирования. Основные факторы, ограничивающие область применимости систем инструментирования, следующие:

- поддерживаемые машинные архитектуры и специфические наборы инструкций для данных архитектур;
- поддерживаемые форматы организации файлов исполняемого кода;
- предоставляемая разметка точек и блоков инструментирования;

- тип организации внедряемого кода;
- ограничения, накладываемые на целевые файлы исполняемого кода.

Ранние системы инструментирования создавались для специфических архитектур и технологий и в основном не применимы для использования на данный момент из-за отсутствия поддержки со стороны авторов или устаревания целевых платформ. Тем не менее данные проекты представляют значительный интерес за счет развития области и формулирования основных подходов.

Система АТОМ [1] была разработана для архитектуры DEC в связке с системой управления и организации исполняемого кода ОМ [2]. Данная система одной из первых предоставляла пользователю возможности описания логики инструментирования, в то время как непосредственно процесс инструментирования выполнялся внутренним модулем АТОМ. Дополнительно пользователю необходимо было осуществить разметку файлов целевой программы для обозначения точек вставки дополнительного кода.

Авторы системы EEL [3] сфокусировали внимание на возможности применения структурных элементов исполняемого кода целевой программы (инструкций, блоков инструкций, функций) в качестве точек инструментирования. Это позволило значительно упростить процесс разметки целевого кода и спецификации логики модуля инструментирования. Дополнительно подход, используемый в системе EEL, обеспечивал гибкую структуру для поддержки различных архитектур: кодирование и декодирование целевого кода и генерацию внедряемого кода осуществлял специфический модуль, реализующий некоторый общий интерфейс.

Среди систем, разработанных под целевые архитектуры, широко используемые в данный момент, можно выделить BIRD [4] (Windows/x86) и BitRaker Anvil [5] (Linux/ARM).

Подход к инструментированию в системе BIRD заключался во вставке инструкций вызова внешних функций в исполняемый код целевой программы и сборке внедряемого кода в виде динамической библиоте-

ки, предоставляющей реализацию данных внешних функций.

Система BitRaker Anvil основана на использовании кросс-платформенного анализа: в исполняемый код целевой программы для платформы ARM вставлялись инструкции вызова внешней функции. При работе целевой программы в среде эмуляции, функционирующей в рамках устройства x86, вызовы внедряемого кода осуществлялись в процессе среды эмуляции, что значительно повышало скорость проведения анализа.

Современные системы инструментирования активно используются для разработки отдельных модулей или полноценных инструментов анализа. Существует несколько направлений, разработки в которых интегрируются в системы инструментирования для повышения их точности, эффективности и гибкости:

- развитие интерфейса инструментирования, доступного пользователю для описания логики необходимого модуля инструментирования;
- поддержка большого количества целевых платформ и разработка специфических оптимизаций для повышения эффективности работы инструментов на данных платформах;
- повышение точности алгоритмов декодирования фрагментов данных, управляющих структур и кода целевых программ и разработка алгоритмов извлечения дополнительных данных о связях и особенностях объектов целевых программ для использования инструментом анализа;
- оптимизация генерируемого внедряемого кода.

Среди наиболее значимых систем статического инструментирования можно выделить MAQAO [6, 7], PEBIL [8] и Dyninst [9], предоставляющие возможности работы с файлами исполняемого кода для платформ Linux/x86 (PEBIL, Dyninst), Linux/x86-64 (MAQAO, PEBIL, Dyninst) и Linux/ppc, Linux/ppc64 (Dyninst).

Система MAQAO изначально была разработана для разбора исполняемого кода и внедрения модулей регистрации событий для профилирования и оценки эффектив-



ности работы компонентов программы. В дальнейшем был разработан язык спецификации точек инструментирования MIL [7] и модуль генерации внедряемого кода для поддержки возможности разработки дополнительных инструментов анализа. Важная особенность системы MAQAO – высокая точность и корректность работы с программами, активно использующими параллельные вычисления. Именно для анализа подобных программ было разработано ядро системы.

Система PEBIL производит разбор исполняемых файлов целевой программы, модификацию секций кода и управляющих структур с целью подготовки к инструментированию и добавляет внедряемый код в виде дополнительной секции. Блоки инструкций в точках инструментирования заменяются на инструкции перехода в добавленные секции. Система PEBIL поддерживает генерацию внедряемого кода в виде внешней динамической библиотеки или коротких блоков инструкций, не требующих подключения дополнительных библиотек. В системе используется двухпроходный декодер инструкций и модуль оптимизации, осуществляющий минимизацию количества инструкций, не относящихся к логике инструмента анализа, но необходимых для корректной передачи управления на внедряемый код.

Система Dyninst предоставляет широкий интерфейс описания внедряемого кода и спецификаций точек инструментирования. Внутренние модули обеспечивают проведение необходимых модификаций, причем возможно проведение как статического инструментирования на основе полномасштабного разбора файлов целевой программы, так и динамического инструментирования на основе стандартного подхода перехвата и модификации блоков инструкций.

Инструментирование файлов ARM ELF

Как было сказано ранее, ни одна из представленных выше современных систем инструментирования не предоставляет поддержку обработки исполняемого кода в инструкциях архитектуры ARM. Система

BitRaker Anvil не предоставляет поддержку современных версий набора инструкций архитектуры. Данное исследование направлено на разработку подхода инструментирования и создание инструментального средства, предоставляющего возможности работы с исполняемым кодом в инструкциях ARM. В качестве первичных задач для применения системы инструментирования рассматриваются задачи, связанные с автоматическим поиском дефектов и уязвимостей в программах.

Среди основных принципов разрабатываемого подхода, определяемых целевыми задачами динамического анализа, можно выделить следующие:

- фокус на инструментирование структурных единиц исполняемого кода (инструкций) для сбора подробной трассы, включающей типы и аргументы совершаемых операций;

- поддержка создания внедряемого кода в виде исходного кода на языке C/C++ с возможностью использования прямых ассемблерных вставок;

- поддержка возможности задания областей исполняемого кода целевой программы на основе фильтрации по функциям и модулям.

Разбор файлов и генерация кода инструментирования. Первым этапом процесса инструментирования является обработка пользовательских спецификаций и проход декодера файлов исполняемого кода целевой программы. Язык задания спецификаций позволяет описывать целевые типы точек инструментирования, набор фильтров для каждой типа и код, выполнение которого необходимо в точках инструментирования. В числе поддерживаемых типов точек инструментирования следующие:

- структурные точки, задающие все инструкции определённой функциональной группы (арифметические и логические инструкции, инструкции доступа к памяти, вызова процедур и т. д.);

- модульные точки, задающие позицию в некотором логическом блоке (входные и выходные инструкции функции, входные инструкции базовых блоков, блоки условного выполнения).

Задаваемые фильтры обеспечивают возможность осуществлять инструментирование для точек определенного типа внутри отдельных функций или групп функций. Язык описания фильтров поддерживает групповые символы для возможности использования префиксов (например, для работы с функциями классов C++).

Задание внедряемого кода осуществляется путем написания коротких блоков исходного кода на языке C/C++, которые автоматически компилируются в исполняемый код, который впоследствии внедряется в файлы целевой программы. Существует возможность использования ассемблерных вставок, глобальных переменных (с помощью задания заголовка, общего для всех блоков) и внешних зависимостей (для которых необходимо задать имя подключаемой библиотеки).

Блоки внедряемого кода привязаны к инструкциям, находящимся в точках инструментирования. При написании внедряемого кода существует возможность использования специального интерфейса, предоставляющего операции над инструкциями двух типов:

- операции доступа к статическим элементам инструкций, таким как номера регистров операндов, значения константных операндов, флаги, специфичные для отдельных групп инструкций;
- операции доступа к динамическим элементам инструкций, таким как значения регистров и содержимого ячеек памяти, значения регистра флагов для работы с блоками условного выполнения и т. д.

Реализация операций первого типа построена таким образом, чтобы приводить к генерации явных константных значений в исходном внедряемом коде. Работа блока оптимизации при компиляции внедряемого кода позволяет понизить накладные расходы по осуществлению доступа к статическим элементам. Для реализации операций второго типа производится генерация блока инструкций, выполнение которого приведет к вычислению необходимых значений при выполнении внедряемого кода.

Подготовка файлов целевой программы к инструментированию. Вторым этапом

инструментирования является изменение структуры файлов целевой программы в рамках формата ARM ELF для получения корректных образов. В список операций, проводимых на данном этапе, входят следующие:

- копирование исходной структуры файлов и добавление дополнительных секций внедряемого кода и данных;
- перемещение и расширение ряда секций, содержащих управляющую информацию и исполняемый код, для добавления внешних зависимостей, указанных во внедряемом коде;
- добавление исполняемого кода и заполнение управляющих структур в расширенных областях секций согласно стандарту ARM ELF и протоколам генерации кода, используемых компиляторами и сборщиками, для корректной работы загрузчика при выполнении инструментированных исполняемых файлов и библиотек.

Инструментирование подготовленных образов. Непосредственное инструментирование заключается в замене одной инструкции или блока инструкций в указанных точках на инструкцию перехода в секцию внедряемого кода, добавленную на предыдущем этапе. Последующие действия связаны с обеспечением корректности модифицированного кода путем вставки дополнительных инструкций сохранения состояния программы и восстановления заменённых инструкций.

Сгенерированный внедряемый код состоит из набора обособленных сегментов, каждый из которых соответствует точке инструментирования. При этом сегменты имеют блочную структуру, разделённую на четыре области: начальный «пустой» блок, состоящий из инструкций, не имеющих никаких побочных эффектов, непосредственно блок кода, реализующий необходимую для анализа функциональность, дополнительный «пустой» блок и блок данных. Начальный и дополнительный «пустые» блоки имеют размер, определяемый особенностями инструкций в точке инструментирования, и используются для вставки служебных инструкций.

В процессе инструментирования в пер-



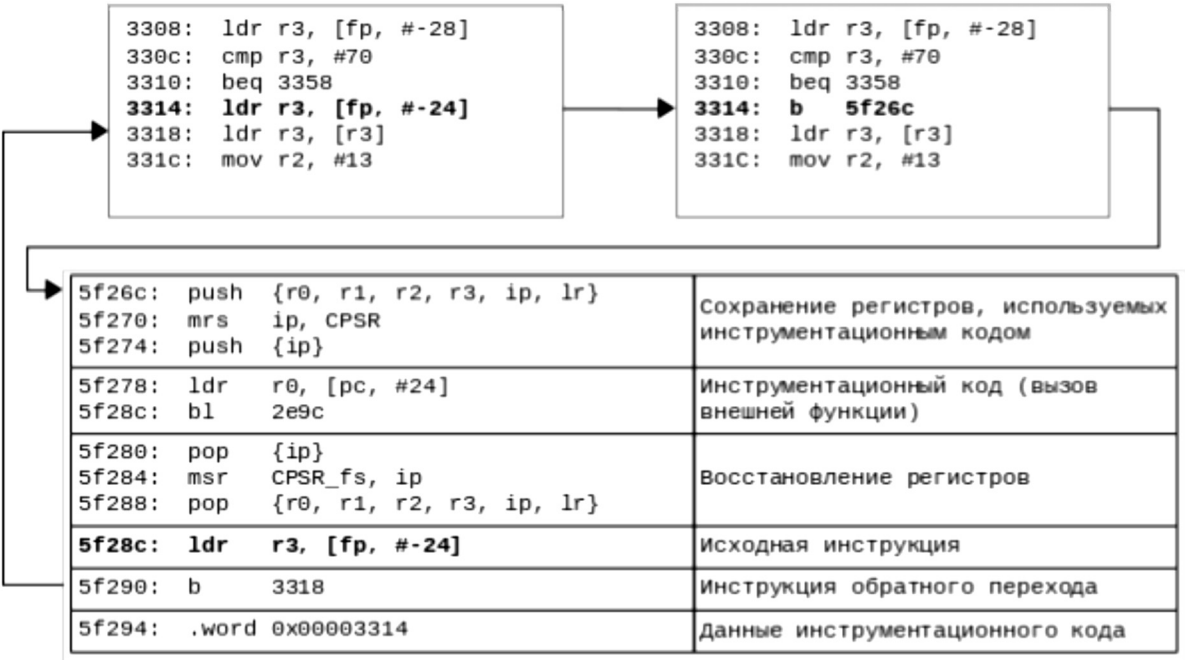
вый «пустой» блок осуществляется вставка инструкций сохранения регистров, изменяемых целевым блоком, во второй «пустой» блок осуществляется вставка инструкций восстановления регистров и вставка инструкций кода, которые были заменены в точке инструментирования на инструкцию перехода. Последней во второй «пустой» блок вставляется инструкция возвратного перехода в основную секцию кода на позицию после точки инструментирования. При наличии двух смежных точек инструментирования с целью снижения накладных расходов вместо возвратного перехода может осуществляться переход на следующий блок внедряемого кода. Пример структуры кода представлен на рисунке, показывающем три блока инструкций: исходный блок инструкций, содержащий точку инструментирования, блок инструкций после вставки перехода на внедряемый код и соответствующий блок внедряемого кода.

Во время переноса исходных инструкций во второй «пустой» блок может возникнуть необходимость модифицировать параметры инструкций или заменить инструкцию на блок, производящий идентичный эффект. В первую очередь подобные

сложности вызывают инструкции, которые напрямую или косвенно используют значение счетчика команд: инструкции перехода по относительному смещению, инструкции доступа к памяти по адресу, вычисляющемуся по смещению от текущего, и т. д.

Коррекция смещений во внедряемом коде.

Последним этапом инструментирования является восстановление смещений, используемых инструкциями доступа к глобальными переменным и внешним зависимостям во внедряемом коде. Стандартный процесс сборки исполняемого кода из нескольких объектных файлов включает в себя выставление корректных смещений в сегментах исполняемого кода по общей секции работы с глобальными данными и зависимостями. В процессе инструментирования сгенерированный внедряемый код подключается к существующему исполняемому файлу или библиотеке сторонними от системы сборки средствами (использование стандартных сборщиков напрямую невозможно из-за отсутствия дополнительных таблиц в исходных файлах целевых файлов) и, соответственно, требует проведения подобного шага.



Пример инструментирования одной точки

Особенности практической реализации подхода

Система инструментирования интегрирована с набором инструментов binutils [10]. Базовые модули binutils включают в себя библиотеку для разбора и управления файлами в формате ELF, причем текущая версия комплекса уже включает в себя средства удаления, добавления и перемещения секций, представления управляющих структур в необходимом формате. Наконец, в составе комплекса binutils существует библиотека liborcodes, предоставляющая гибкий интерфейс для реализации модулей декодирования набора инструкций определённой архитектуры.

Инструменты readelf и objdump в составе комплекса binutils используются для извлечения структуры целевых файлов ARM ELF, разметки точек инструментирования и автоматической генерации исходного кода инструментирования. Инструмент objdump реализует линейное дизассемблирование; данный подход в отличие от метода рекурсивного разбора накладывает дополнительные ограничения на входные файлы (например, наличие таблицы символов) для обеспечения точности результата. Трансформация внедряемого кода в объектные файлы осуществляется с помощью компиляторной системы gcc, конкретная версия которой может быть предоставлена пользователем. Перераспределение исходных секций и добавление новых секций осуществляется с помощью инструмента objcopy. Для коррекции смещений и сегментов, осуществляющих работу с внешними зависимостями, на основе внутренней библиотеки работы с форматом ELF binutils разработан инструмент addsymbol. Наконец, модуль расширения для библиотеки liborcodes, используемый для разметки точек инструментирования, интегрирован с новым инструментом gewriter, проводящим непосредственное инструментирование.

Внутреннее представление инструкций. При работе инструмента objdump происходит разбор инструкций файлов исполняемого кода целевой программы и перевод

инструкций во внутреннее представление, описывающее их основные элементы и параметры. Среди наиболее значимых элементов можно выделить следующие:

- тип аргументов инструкции (регистры, константные значения);

- значения аргументов инструкции (для констант);

- код операции;

- дополнительные параметры, специфичные для групп операций (режимы индексирования, триггеры обновления регистра флагов).

Основная функциональная мощность по обработке данных параметров лежит на внедряемом коде, однако прямой доступ к значениям параметров значительно упрощает структуру внедряемого кода и снижает накладные расходы при проведении анализа за счет единовременного извлечения статических данных на этапе инструментирования.

Практическое применение системы инструментирования. В рамках практических экспериментов по использованию системы статического инструментирования разработаны модули, позволяющие проводить динамический анализ, нацеленный на автоматический поиск дефектов. Данные модули разрабатывались для существующей системы анализа Avalanche [11], стандартно использующей динамическое инструментирование на основе комплекса Valgrind [12] для сбора необходимых данных.

Система Avalanche осуществляет обход возможных путей выполнения целевой программы на основе отслеживания потока помеченных данных, составления систем булевых формул для описания путей выполнения и автоматической генерации новых входных данных с помощью решателя булевых формул. Система осуществляет итеративный анализ, причем на каждой итерации целевая программа выполняется под контролем двух модулей, извлекающих дополнительные данные, – tracegrind (модуль инструментирования с целью получения полной трассы инструкций, оперирующих помеченными данными) и covgrind (модуль инструментирования с целью вычисления метрики покрытия на основе количества

выполненных базовых блоков). Для работы с удалёнными и внешними устройствами, обладающими ограниченными вычислительными мощностями, в системе Avalanche предусмотрен специальный режим, при использовании которого на целевом устройстве осуществляется только запуск программы, а полная обработка трасс и работа решателя булевых формул выполняется на хост-устройстве.

Реализованные на основе статического инструментирования версии модулей covgrind и tracegrind использовались в рамках указанного режима работы и позволили значительно ускорить анализ набора целевых программ. Различия в обработке инструкций системами статического и динамического инструментирования привели к малому изменению порядка обхода ветвей, однако повышение скорости обхода ветвей было зафиксировано на всех проектах. Обнаруженные дефекты соответствуют дефектам, найденным системой Avalanche ранее на рассмотренных проектах для архитектуры x86_64. В таблице приведен краткий обзор результатов работы системы с использованием двух типов инструментирования.

Время анализа для всех запусков было ограничено двумя часами (стандартная практика для инструмента Avalanche), в качестве начальных входных данных использовались файлы, соответствующие предыдущим результатам применения Avalanche.

В список проанализированных программ входят:

утилита swfdump (swftools-0.9.0) для извлечения информации о содержимом файлов формата SWF;

утилита mpeg2dec (libmpeg2-0.5.1) для декодирования файлов формата MPEG-2;

утилита cjpeg (libjpeg-7) для кодирования файлов в формат JPG;

утилита qtdump (libquicktime-1.1.3) для декодирования файлов нескольких форматов (например, AVI);

утилита mpeg3dump (libmpeg3-1.8) для декодирования файлов в формате MPEG-3.

Для четырех целевых программ из пяти при применении статического инструментирования было достигнуто пятикратное увеличение количества итераций анализа. Дефекты, известные по предыдущим результатам работы Avalanche, были получены ранее, и для двух проектов было найдено больше различных дефектов. Для программы mpeg3dump было зафиксировано менее значительное ускорение, что связано с большим количеством ветвей выполнения, приводящих к зацикливанию программы. Зацикливание приводит к задержке по ожиданию срабатывания таймера и к снижению полезного времени анализа.

Рассмотренный подход позволяет создавать точные и эффективные инструменты анализа, осуществляющие трассировку дан-

Результаты работы инструмента Avalanche

Программа		Итерации анализа	Уникальные дефекты	Время обнаружения первого дефекта, с
swfdump	Дин.	105	1	31
	Стат.	575	4	4
mpeg2dec	Дин.	55	1	25
	Стат.	302	1	3
cjpeg	Дин.	236	1	600
	Стат.	1331	1	20
qtdump	Дин.	189	1	1269
	Стат.	1027	1	33
mpeg3dump	Дин.	54	1	112
	Стат.	125	2	40

ных и инструкций. Наибольший выигрыш по производительности по сравнению с существующими динамическими аналогами достигается при выполнении многочисленных запусков модифицированных целевых программ (с целью проверки различных путей выполнения и сценариев работы). На данный момент основная задача реализации подхода заключается в работе с платформой ARM, однако интеграция с системой binutils обеспечивает возможность поддержки других систем (в рамках формата ELF) без необходимости модификации базовых модулей системы инструментирования.

В настоящее время рассматриваются следующие направления будущей деятельности:

Расширение множества поддерживаемых типов точек инструментирования и интерфейса, предоставляемого пользователю для описания внедряемого кода.

Исследования возможностей оптимизации инструментирования за счет уменьшения количества инструкций, обеспечивающих корректность перехвата управления, и реорганизации переходов между исполняемым кодом целевой программы и блоками внедряемого кода.

Повышение точности работы декодера исполняемого кода при работе с файлами ELF, не содержащими дополнительные секции управляющей информации.

Изменение структуры внутреннего представления инструкций для повышения гибкости интерфейса, предоставляемого для написания внедряемого кода, и адаптации к поддержке различных наборов инструкций. Одним из возможных решений в данном направлении является полномасштабная трансформация блоков инструкций в операции промежуточного языка (например, REIL [13]).

СПИСОК ЛИТЕРАТУРЫ

1. **Srivastava A., Eustace A.** ATOM: A System for Building Customized Program Analysis Tools // WRL Research Report 94/2. Western Research Laboratory, 1994.
2. **Srivastava A., Wall D.** A Practical System for Intermodule Code Optimization at Link-Time // J. of Programming Language. 1993. Vol. 1(1). Pp. 1–18.
3. **Larus J., Schnarr E.** EEL: Machine-Independent Executable Editing // Proc. of the ACM SIGPLAN Conf. on Programming Language Design and Implementation. 1995. Pp. 291–300.
4. **Nanda S., Li W., Lam L., Chiueh T.** BIRD: Binary Interpretation using Runtime Disassembly // Proc. of the Internat. Symp. on Code Generation and Optimization. 2006.
5. **Calder B., Austin T., Yang D., Sherwood T., Sair S., Newquist D., Cusac T.** BitRaker Anvil: Binary Instrumentation for Rapid Creation of Simulation and Workload Analysis Tools // Global Signal Processing (GSPx) Conf. 2004.
6. **Djoudi L., Barthou D., Carribault P., Lemuet C., Acquaviva J., Jalby W.** MAQAO: Modular Assembler Quality Analyzer and Optimizer for Itanium 2 // Workshop on Explicitly Parallel Instruction Computing Techniques. 2005.
7. **Charif-Rubial A., Barthou D., Valensi C., Shende S., Malony A. Jalby W.** MIL: A Language to Build Program Analysis Tools through Static Binary Instrumentation // Proc. of the 20th Annual Internat. Conf. on High Performance Computing. 2013.
8. **Laurenzano M., Tikir M., Carrington L., Snaveley A.** PEBIL: Efficient Static Binary Instrumentation for Linux. // Proc. of the Internat. Symp. on Performance Analysis of Systems & Software. 2010. Pp. 175–183.
9. **Miller B., Bernat A.** Anywhere, Any Time Binary Instrumentation // ACM SIGPLAN-SIGSOFT Workshop on Program Analysis for Software Tools and Engineering. 2011. Pp. 9–16.
10. Страница проекта GNU binutils [Электронный ресурс] // URL: <http://www.gnu.org/software/binutils/> (Дата обращения: 10.08.2015).
11. **Исаев И.К., Сидоров Д.В.** Применение динамического анализа для генерации входных данных, демонстрирующих критические ошибки и уязвимости в программах // Программирование. 2010. № 4. С. 51–67.
12. **Nethercote N., Seward J.** Valgrind: A Framework for Heavyweight Dynamic Binary Instrumentation // In Proc. of ACM SIGPLAN Conf. on Programming Language Design and Implementation. San Diego, California, USA, 2007. Pp. 89–100.
13. **Dullien T., Porst S.** REIL: A platform-independent intermediate representation of disassembled code for static code analysis // In CanSecWest. 2009.



REFERENCES

1. **Srivastava A., Eustace A.** ATOM: A System for Building Customized Program Analysis Tools. *WRL Research Report 94/2*, Western Research Laboratory, 1994.
2. **Srivastava A., Wall D.** A Practical System for Intermodule Code Optimization at Link-Time. *Journal of Programming Language*, 1993, Vol. 1(1), Pp. 1–18.
3. **Larus J., Snharr E.** EEL: Machine-Independent Executable Editing. *Proceedings of the ACM SIGPLAN 1995 Conference on Programming Language Design and Implementation*, 1995, Pp. 291–300.
4. **Nanda S., Li W., Lam L., Chiueh T.** BIRD: Binary Interpretation using Runtime Disassembly. *Proceedings of the International Symposium on Code Generation and Optimization*, 2006.
5. **Calder B., Austin T., Yang D., Sherwood T., Sair S., Newquist D., Cusac T.** BitRaker Anvil: Binary Instrumentation for Rapid Creation of Simulation and Workload Analysis Tools. *Global Signal Processing (GSPx) Conference*, 2004.
6. **Djoudi L., Barthou D., Carribault P., Lemuet C., Acquaviva J., Jalby W.** MAQAO: Modular Assembler Quality Analyzer and Optimizer for Itanium 2. *Workshop on Explicitly Parallel Instruction Computing Techniques*, 2005.
7. **Charif-Rubial A., Barthou D., Valensi C., Shende S., Malony A. Jalby W.** MIL: A Language to Build Program Analysis Tools through Static Binary Instrumentation. *Proceedings of the 20th Annual International Conference on High Performance Computing*, 2013.
8. **Laurenzano M., Tikir M., Carrington L., Snaveley A.** PEBIL: Efficient Static Binary Instrumentation for Linux. *Proceedings of the International Symposium on Performance Analysis of Systems & Software*, 2010, Pp. 175–183.
9. **Miller B., Bernat A.** Anywhere, Any Time Binary Instrumentation. *ACM SIGPLAN-SIGSOFT Workshop on Program Analysis for Software Tools and Engineering*, 2011, Pp. 9–16.
10. **GNU binutils.** Available: <http://www.gnu.org/software/binutils/>.
11. **Isayev I.K., Sidorov D.V.** Primeneniye dinamicheskogo analiza dlya generatsii vkhodnykh dannykh, demonstriruyushchikh kriticheskiye oshibki i uyazvimosti v programmakh [The Use of Dynamic Analysis for Generation of Input Data that Demonstrates Critical Bugs and Vulnerabilities in Programs]. *Programmirovaniye [Programming]*, 2010, No. 4, Pp. 51–67. (rus)
12. **Nethercote N., Seward J.** Valgrind: A Framework for Heavyweight Dynamic Binary Instrumentation. *Proceedings of the ACM SIGPLAN Conference on Programming Language Design and Implementation*, San Diego, California, USA, 2007, Pp. 89–100.
13. **Dullien T., Porst S.** REIL: A platform-independent intermediate representation of disassembled code for static code analysis. *CanSecWest*, 2009.

ЕРМАКОВ Михаил Кириллович — младший научный сотрудник Института системного программирования РАН.

109004, Россия, Москва, ул. Александра Солженицына, д. 25.

E-mail: mermakov@ispras.ru

ERMAKOV Mikhail K. *Institute for System Programming of the Russian Academy of Sciences.*

109004, Alexander Solzhenitsyn Str. 25, Moscow, Russia.

E-mail: mermakov@ispras.ru

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»

№ 1(236) 2016

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

д-р техн. наук, профессор *А.С. Коротков* – главный 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

Компьютерная верстка *А.Н. Смирнов*

Перевод на английский язык *А.С. Колгатина*

Лицензия ЛР № 020593 от 07.08.97

Подписано в печать 31.03.2016. Формат 60×84 1/8. Бум. тип. № 1.
Печать офсетная. Усл. печ. л. 13,95. Уч.-изд. л. 13,95. Тираж 1000. Заказ

Санкт-Петербургский политехнический университет Петра Великого
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. 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 выпускается 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 публикациях представлены в РИНЦ, в Реферативном журнале ВИНИТИ РАН, в международной справочной системе «Ulrich's Periodical Directory».

Периодичность выхода журнала — 4 номера в год.

2. ТРЕБОВАНИЯ К ПРЕДОСТАВЛЯЕМЫМ МАТЕРИАЛАМ

2.1. Оформление материалов

1. Рекомендуемый объем статей для авторов с ученой степенью доктора наук, званием профессора, соискателей ученой степени доктора наук (докторантов) 12–20 страниц формата А-4 с учетом графических вложений. Количество графических вложений (диаграмм, графиков, рисунков, таблиц, фотографий и т. п.) не должно превышать 4.

2. Рекомендуемый объем статей для преподавателей, авторов без ученой степени, соискателей ученой степени кандидата наук — 8–15 страниц формата А-4; аспирантов — 8 страниц формата А-4 с учетом графических вложений. Количество графических вложений (диаграмм, графиков, рисунков, таблиц, фотографий и т. п.) не должно превышать 3.

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 7.05.-2008).

4. Число авторов статьи не должно превышать трех человек.

5. Набор текста осуществляется в редакторе **MS Word**, формул — в редакторе **MathType**. 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 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;
- акт экспертизы о возможности опубликования материалов в открытой печати.

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:

на сайте журнала <http://ntv.spbstu.ru>

по телефону редакции +7(812) 552-62-16 с 10⁰⁰ до 18⁰⁰ Галина Александровна

или по e-mail: infocom@spbstu.ru