

Журнал включен в Перечень ведущих периодических изданий, рекомендованных ВАК для публикации результатов диссертационных исследований.

РЕДАКЦИОННЫЙ СОВЕТ

Главный редактор

Емельянов А. А., докт. экон. н., проф., Национальный исследовательский университет «МЭИ»; Национальное общество имитационного моделирования, Санкт-Петербург

Сопредседатели редакционного совета

Рубин Ю. Б., докт. экон. н., проф., чл.-корр. РАО, ректор Университета «Синергия», зав. кафедрой Теории и практики конкуренции

Мешалкин В. П., докт. техн. н., проф., академик РАН, директор Института логистики ресурсосбережения и технологической инноватики, РХТУ им. Д. И. Менделеева

Члены редакционного совета

Брекис Эд., докт. экон. н., ассоциированный проф., зав. кафедрой Эконометрики и бизнес-информатики, Латвийский Университет, Рига, Латвия

Волкова В. Н., докт. экон. н., проф., кафедра Системного анализа и управления Института информационных технологий и управления, СПбГПУ

Дли М. И., докт. техн. н., проф., зав. кафедрой МИТЭ, зам. директора Филиала НИУ «МЭИ» в Смоленске

Козлов В. Н., докт. техн. н., проф., зав. кафедрой Системного анализа и управления Института информационных технологий и управления, СПбГПУ

Краковский Ю. М., докт. техн. н., профессор, кафедра Информационных систем и защиты информации, Иркутский государственный университет путей сообщения

Пецольдт К., докт. экон. н., проф., проректор по международному сотрудничеству с Восточной Европой, Технологический Университет Ильменау, Германия

Росс Г. В., докт. техн. н., докт. экон. н., проф., академик РАН, кафедра «Системный анализ в экономике», Финансовый университет при Правительстве РФ

Стоянова О. В., докт. техн. н., доцент, кафедра Информационных систем в экономике, СПбГУ

Сухомлин В. А., докт. техн. н., проф., зав. лабораторией Открытых информационных технологий, факультет ВМК, МГУ им. М. В. Ломоносова

Халин В. Г., докт. экон. н., проф., зав. кафедрой Информационных систем в экономике, Экономический факультет СПбГУ

Шориков А. Ф., докт. физ.-мат. н., проф., кафедра Прикладной математики УралЭНИИ, Уральский Федеральный Университет им. Первого Президента России Б. Н. Ельцина

Штельцер Д., докт. техн. н., reg. pol., проф., Глава Департамента информации и управления знаниями, Технологический Университет Ильменау, Тюрингия, Германия

Юсупов Р. М., докт. техн. н., проф., чл.-корр. РАН, директор Санкт-Петербургского института информатики и автоматизации РАН, президент Национального общества имитационного моделирования «НОИМ»

Заместители главного редактора

Власова Е. А., научная редакция Университета «Синергия»

Прокимов Н. Н., канд. техн. н., доцент, кафедра Информационных систем, Университет «Синергия»

Журнал выходит с 2006 г. Периодичность издания — 6 раз в год.

Журнал индексируется в российских и зарубежных базах научной периодики eLIBRARY (РИНЦ), Russian Science Citation Index (RSCI) на платформе Web of Science, ВИНТИ, Ulrich's Periodicals Directory

Учредитель и издатель: **Негосударственное образовательное частное учреждение высшего образования «Московский финансово-промышленный университет «Синергия»**

Адрес редакции и издателя:

129090, Москва, ул. Мещанская, д. 9/14, стр.1 (юр.)
125190, Россия, Москва, Планетная ул., д. 36, оф. 301, 302.

Тел.: +7 (495) 987-43-74 (доб. 33-04); e-mail: appliedinformaticsjournal@gmail.com; www.appliedinformatics.ru

© Университет «Синергия»

EDITORIAL BOARD

Editor-in-Chief

A. Emelyanov, Dr of Economics, Professor, National Research University MPEI; Executive board member of NC «National Society for Simulation Modelling», St. Petersburg

Co-Chairs of the Editorial Board

Yu. Rubin, Dr of Economics, Professor, Corresponding Member of the Russian Education Academy, Head of the Theory and Practice of Competition Chair, Rector of the Moscow University for Industry and Finance «Synergy»

V. Meshalkin, Dr of Technique, Professor, Academician of Russian Academy of Sciences (RAS), Director of the Institute of Logistics and Resource Technology Innovation, D. Mendeleev University of Chemical Technology of Russia, Moscow

Members of the Editorial Board

Ed. Brēķis, Dr. Ec., Assoc. professor, Head of The Econometrics and Business Informatics Chair, Faculty of Economics and Management, Rīga, University of Latvia

M. Dli, Dr of Technique, Professor, Head of The MITE Chair, Deputy Director of the National Research University MPEI Branch in Smolensk

V. Hulin, Dr of Economics, Professor, Head of The Economic Information Systems Department, St. Petersburg State University

V. Kozlov, Dr of Technique, Professor, Head of System analysis and management Chair, Institute of Information technologies and management, St. Petersburg State Polytechnical University

Krakovskiy Yu., Dr of Technique, Professor, Information Systems and Information Security Department, Irkutsk State Railway Transport Engineering University

K. Pezoldt, Dr of Economics, Professor, Deputy Rector for International Cooperation with Eastern Europe, Ilmenau University of Technology, Germany

G. Ross, Dr. of Technique, Dr. of Economics, Professor, Academician of Russian Academy of Natural Sciences (RANS), System Analysis in Economics Department, Financial University under the Government of the Russian Federation

A. Shorikov, Dr. of Physics & Mathematics, Professor of The Applied Mathematics Chair, Ural Power Institute of El'cin Ural Federal University (Ekaterinburg)

O. Stoyanova, Dr. of Technique, Asoc. Professor, Economic Information Systems Department, St. Petersburg State University

V. Sukhomlin, Dr of Technique, Professor, Faculty of Computational Mathematics and Cybernetics, Lomonosov Moscow State University

D. Stelzer, Dr., rer. pol., Professor, Head of The Information and Knowledge Management Department of Ilmenau University of Technology (TU Ilmenau), Germany

V. Volkova, Dr of Economics, Professor, System analysis and management Chair, Institute of Information technologies and management, St. Petersburg State Polytechnical University

R. Yusupov, Dr of Technique, Professor, Corresponding Member of Russian Academy of Sciences (RAS), Director of the Saint Petersburg RAS Institute of Informatics and Automation, President of NC «National Society for Simulation Modeling», St. Petersburg

Deputy Chief Editors

E. Vlasova, Scientific Edition Department, Moscow University for Industry and Finance «Synergy»

N. Prokimnov, PhD in Technique, Associate Professor, the Information Systems Chair, Moscow University for Industry and Finance «Synergy»

Published since 2006. Periodicity: six times a year.

The journal is indexed in

Russian Science Citation Index (RSCI) on Web of Science platform, VINITI (Russian Academy of Sciences), Ulrich's Periodicals Directory

Publisher: Moscow University for Industry and Finance «Synergy»

Publisher address: 9/14 s.1, Meshchanskaya str., Moscow, 129090, Russia

Editorial office address: of. 301; 36, Planetnaya st., Moscow, 125319, Russia

Tel: +7 (495) 987-43-74 (ext. 33-04); e-mail: appliedinformaticsjournal@gmail.com; www.appliedinformatics.ru

© Moscow University for Industry and Finance «Synergy»

ИТ-БИЗНЕС

Информационные системы бизнеса

Интервью

Перспективы разработки и внедрения ERP-систем в России: точка зрения руководства ООО «Системные Бизнес-Компоненты» 5

ИТ-МЕНЕДЖМЕНТ

Управление эффективностью

Е. Д. Стрельцова, А. И. Бородин, И. В. Яковенко
Модельный инструментарий стратегического управления промышленным предприятием . . . 15

Управление ресурсами

А. В. Кычкин, А. И. Дерябин, О. Л. Викентьева, Л. В. Шестакова,
Проектирование IoT-платформы для управления энергоресурсами интеллектуальных зданий. 29

ИТ И ОБРАЗОВАНИЕ

Образовательное пространство

А. Н. Терехов, В. Г. Халин, А. В. Юрков
Нужны ли кандидаты и доктора наук по программной инженерии для модернизации и технологического развития российской экономики? 42

SIMULATION

Теория и практика

И. И. Труб
Имитационное моделирование иерархических bitmap-индексов 53

ИНСТРУМЕНТАЛЬНЫЕ СРЕДСТВА

Сетевые технологии

А. Ю. Новиков, П. П. Кейно, Л. Л. Хорошко
Разработка архитектуры интернет-сервиса организации научных мероприятий с автоматизацией документооборота 70

Информационная инфраструктура

Ю. В. Нефедов, А. Ю. Сергеева
Типовые проблемы интеграции ИС и пути их решения 77

Технологии разработки программного обеспечения

А. И. Грачев, О. А. Страхов
Сравнительная характеристика способов построения облачных ERP-систем. 84

ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ

Защита информации

С. С. Бармина, Ф. М. Таджибаева, М. В. Тумбинская
Корреляционный анализ и прогнозирование SYN-флуд атак 93

ЛАБОРАТОРИЯ

Модели и методики

Е. А. Федорова, О. Ю. Рогов, Ф. Ю. Федоров
Модели прогнозирования банкротства предприятий: алгоритм ансамбля классификаторов 103

М. И. Дли, О. В. Булыгина, П. Ю. Козлов
Формирование структуры интеллектуальной системы анализа и рубрицирования неструктурированной текстовой информации в различных ситуациях 111

ВОПРОСЫ ТЕОРИИ

Информационные системы

Л. А. Рейнгольд, Е. А. Рейнгольд
Структурирование информации для целей автоматизации — концептуальные аспекты 124

Правила оформления материалов 140

IT BUSINESS

Business information systems

Interview

Prospects of development and implementation of ERP systems in Russia: point of view of the company’s management «System Business Components» 5

IT MANAGEMENT

Performance management

E. Streltsova, A. Borodin, I. Yakovenko
Model instrumentation of strategic management of industrial enterprise 15

Resource management

A. Kychkin, A. Deryabin, O. Vikentieva, L. Shestakova,
IoT-platform design for smart buildings energy management 29

IT AND EDUCATION

Education environment

A. Terekhov, V. Khalin, A. Yurkov
Do candidates and doctors of science in software engineering need to modernize and technological development of the Russian economy? 42

SIMULATION

Theory and practice

I. Trub
Simulation of hierarchical bitmap-indices. 53

TOOLS

Network technologies

A. Novikov, P. Keyno, L. Khoroshko
Architecture development of internet-service for science conferences organizing with workflow automatization 70

Information infrastructure

Y. Nefedov, A. Sergeeva
Typical problems of information systems integration and ways of their solution 77

Software development technologies

A. Grachev, O. Strakhov
Comparative characteristic of methods of cloud ERP-systems construction 84

INFORMATION SECURITY

Data protection

S. Barmina, F. Tadzhibaeva, M. Tumbinskaya
Correlation analysis and forecasting of SYN-flood attacks. 93

LABORATORY

Models and methods

E. Fedorova, O. Rogov, F. Fedorov
Models of corporate bankruptcy prediction with the ensemble of classifiers algorithm 103

M. Dli, O. Bulygina, P. Kozlov
Formation of the structure of the intellectual system of analyzing and rubricating unstructured text information in different situations 111

THEORETICAL APPROACH

Information systems

L. Reingold, E. Reingold
Structuring information for automation-conceptual aspects 124

Guidelines for authors 140

Перспективы разработки и внедрения ERP-систем в России: точка зрения руководства ООО «Системные Бизнес-Компоненты»

Вопросам разработки и внедрения ERP-систем посвящено достаточно много практических статей и научных работ, тем не менее стремительное развитие информационных технологий, изменение ситуации на рынке ИТ, вызванное введением санкций и ориентацией на импортозамещение, позволяют по-новому оценить возможности и перспективы автоматизации отечественных предприятий.

Для обсуждения этой темы 26 июня 2018 года в Университете «Синергия» была организована встреча генерального директора ООО «Системные Бизнес-Компоненты» Сергея Викторовича Наливайко и заместителя генерального директора по науке и развитию ООО «Системные Бизнес-Компоненты» Олега Алексеевича Страхова с заведующим кафедрой информационных систем Университета «Синергия» Денисом Владимировичем Денисовым.

От редакции журнала «Прикладная информатика» в беседе и подготовке материала принимала участие журналист-стажер Лорета Александровна Малеева, студентка направления «Журналистика» Московского гуманитарного университета.

Л. А. Малеева: Сергей Викторович, вначале хотелось бы попросить Вас рассказать немного о компании «Системные Бизнес-Компоненты». Как возникла идея создания компании в 1995 году? Как развивалась компания? Как менялась продуктовая линейка с 1995 года и до настоящего времени? Что позволяет компании сохранять конкурентоспособность и активно развиваться в условиях острой конкуренции на рынке ИТ?

С. В. Наливайко: Компания образовалась в 1995 году в Республике Беларусь, в городе Гомеле. К созданию предприятия подтолкнул визит во Францию и знакомство с системами автоматизации местных производственных предприятий. Разительный контраст между тем, как подходили к автоматизации на Западе с тем, как это на тот момент выглядело в Белоруссии (или в России), был довольно существенным. Однако уже

в то время российские и белорусские программисты приобрели довольно хорошую репутацию. Проблема была в качественной постановке задач автоматизации, поскольку часто получалось, что задачу программистам ставят такие же программисты, далекие от бизнеса и его потребностей.

Изначально пошли обычным путем: создали учетную систему, автоматизирующую бухгалтерский и налоговый учет, а также некоторые торговые операции. В начале нового тысячелетия стало понятно, что сохранение подобного подхода мешает развитию системы. Было принято решение о постепенном переходе на новую собственную программную платформу, создаваемую в первую очередь для задач управления процессами производства, ремонта и сервисного обслуживания. Уже с этой платформой в 2004 году мы выиграли конкурс по автоматизации процес-

сов нефтедобывающей компании, входящей в крупный российский холдинг. Тогда же был создан московский офис компании, впоследствии ставший основным.

О. А. Страхов: В 2010 году московский офис компании пополнился специалистами компании МКБ «Параллель», имевшими опыт выполнения работ по заказу Москвы в рамках крупных проектов, а также в интересах других госзаказчиков. Это позволило «Системным Бизнес-Компонентам» приобрести уникальные компетенции, связанные с автоматизацией ситуационного управления и обеспечением автоматизации жизненного цикла сложных изделий. Примерно тогда же было принято принципиальное решение о необходимости обеспечения кроссплатформенности — как впоследствии оказалось, очень правильное и своевременное решение. Наверное, это и помогает нам развиваться — умение среди всех новых и популярных технологий определить те, которые реально будут нужны бизнесу, и сосредоточиться на их применении.

Л. А. Малеева: А как все-таки «Системные Бизнес-Компоненты» вышли на российский рынок?

С. В. Наливайко: Наши разработки для белорусских компаний оказались востребованы и в России, мы приняли участие в тендере одной из нефтедобывающих компаний, выиграли его, и таким образом у нас появились партнеры в России, с которыми мы продолжаем успешное сотрудничество.

О. А. Страхов: Дело в том, что условия тендеров большинства российских компаний предусматривают работу с российским юридическим лицом. Собственно это и стало поводом для оформления юридического лица в Российской Федерации...

С. В. Наливайко: Такая территориальная распределенность позволяет использовать преимущества и белорусского и российского рынков, обмениваться наработками, задействовать кадровый потенциал обеих стран, подбирая под любую задачу высококвалифицированных специалистов, в том числе это выпускники белорусских вузов, распределенные в нашу компанию.

О. А. Страхов: Здесь стоит отметить, что в Белоруссии сохранилась система распределения выпускников, т. е. выпускник должен отработать определенный срок у себя на родине в той сфере, в которой он получил об-



О. А. Страхов, С. В. Наливайко и Л. А. Малеева на рабочей встрече в Университете «Синергия»
Nalivayko S. V., Strahov O. A. and Maleeva L. A. at the meeting in the «Synergy» University

разование, тем самым поддерживая народное хозяйство республики. Поэтому белорусские компании, в том числе «Системные Бизнес-Компоненты», имеют надежный и устойчивый канал поступления молодых специалистов из числа выпускников вузов.

С. В. Наливайко: Также следует отметить, что в белорусских вузах внедрены некоторые элементы западных систем подготовки специалистов, благодаря этому выпускник вуза умеет себя позиционировать и успешно решать задачи в соответствии со своей ролью в команде ИТ-проекта. Это очень важно в условиях активного перехода на технологии гибкой разработки и сокращения сроков выпуска релизов программных решений.

Д. В. Денисов: Сергей Викторович, как Вы могли бы охарактеризовать текущее состояние ИТ-отрасли в целом и рынка ERP-систем в частности?

С. В. Наливайко: Если смотреть в международном масштабе, ИТ-рынок можно считать полностью сформировавшимся. Можно уверенно говорить как о состоявшемся международном разделении труда в ИТ, так и о полно-

стью сформировавшейся сегментации рынка с ярко выраженными лидерами и постепенным сокращением числа игроков в основных сегментах. Эта же тенденция наблюдается и в ERP — более крупные игроки поглощают менее крупных, в результате выбор продуктов на рынке, конечно, сильно сокращается.

Что касается российского рынка программных решений (включая ERP и комплексные системы управления предприятиями), то он изначально формировался с позиций «все западное лучше». Изначально был допущен перекося в развитии, в результате которого в ряде чувствительных отраслей мы имеем полную зависимость от зарубежного программного обеспечения. Отечественным решениям была уготована участь «нишевых»: кто нашел свое «место под солнцем», тот и выжил.

В результате мы имеем практически полное доминирование двух-трех западных продуктов. К сожалению, и на сегодняшний момент большинство заказчиков с трудом признают риски, связанные с полной зависимостью от зарубежного ПО. Начавшийся



Д. В. Денисов, О. А. Страхов и С. В. Наливайко на рабочей встрече в Университете «Синергия»
Nalivayko S. V., Strahov O. A. and Denisov D. V. at the meeting in the «Synergy» University

процесс импортозамещения наконец-то способен изменить сложившиеся взгляды, но это не быстрый процесс.

Позитивной же тенденцией является растущее понимание руководителей и владельцев бизнеса, что одним из ключевых факторов на сегодня является оперативность принятия решений и доведения их до исполнителей. А для этого у них в любой точке мира должна быть необходимая информация — оперативная, наглядная и достоверная. То есть продвигаемая в том числе нами концепция ситуационного управления постепенно становится подручным средством, необходимым инструментом.

Л. А. Малеева: Сергей Викторович, Вы упомянули о международном разделении труда в ИТ — уточните, пожалуйста, в чем оно заключается?

С. В. Наливайко: Международное сотрудничество в сфере ИТ, которое заключается в том, что постановкой задач занимаются западные специалисты, а отечественные программисты занимаются их реализацией. Например, в Белоруссии находится достаточно много ИТ-компаний, которые выполняют заказы западных партнеров. Взаимодействие осуществляется через рабочие группы, участники которых могут находиться совершенно в разных странах. Нам это дает возможность приобщиться к зарубежному опыту, а также определить западные тенденции развития ИТ исходя из того, какие задачи ставятся. Они точно так же приобщаются к нашему опыту разработки приложений.

В сфере аппаратного обеспечения это, конечно, производство комплектующих в Китае, Малайзии, на Тайване и в Сингапуре, хотя разработки осуществляются в США и странах Западной Европы. Кстати, у нас постепенно начинают «оживать» разработки в области электроники, но «мировой фабрикой» является Китай.

О. А. Страхов: Добавлю, что заданные системные архитекторы, постановщики задач поступают достаточно интересно: общая за-

дача разделяется на части, и каждая часть задачи ставится отдельной группе разработчиков, таким образом, чтобы ни одна из групп не понимала общей задачи.

Также хочу отметить, что события 2014 года, а именно введение антироссийских санкций, с одной стороны, усугубили зависимость от западных ИТ-компаний, а с другой, обусловили запрос на реализацию собственных разработок как системного, так и прикладного программного обеспечения.

Д. В. Денисов: Сергей Викторович, расскажите, пожалуйста, о флагманской разработке ООО «Системные Бизнес-Компоненты» — «SBC-Предприятие»!

С. В. Наливайко: «SBC-Предприятие» — это комплексное решение, прежде всего, программная платформа, на основе которой создаются прикладные системы. Платформа обеспечивает единый полнофункциональный документооборот всех видов документов (включая, например, конструкторскую документацию), ведение нормативно-справочной информации (в том числе с привязкой НСИ взаимодействующих систем, справочников синонимов и т. п.), средств импорта-экспорта данных и контроля их качества, средств графической визуализации данных, а также средств разработки и генерации отчетов различной сложности.

Способ организации базы данных нашей платформы позволяет обеспечить интеграцию практически с любыми отечественными и зарубежными программными продуктами. Все компоненты платформы (соответствующие термину «модуль» ERP-решений у других производителей), упрощенно говоря, реализуются специальными настройками платформы. Также следует отметить четыре ключевых требования, задававшиеся при разработке системы: низкая требовательность к вычислительным и телекоммуникационным ресурсам без ущерба быстродействию; высокая надежность и отказоустойчивость; низкая стоимость эксплуатации; удобство использования персоналом с низкой квалификацией.

Д. В. Денисов: Давайте конкретизируем, что именно вкладывается в понятие «программная платформа», поскольку это относительно новый термин и разработчики программного обеспечения трактуют его по-разному...

О. А. Страхов: Дело в том, что сейчас достаточно сложно определить, где начинается и где заканчивается программная платформа, это действительно достаточно новый термин... С нашей точки зрения, программная платформа это общая основа для всех решений в некоторой предметной области, позволяющая осуществлять разработку дополнительных компонентов. Применительно к «SBC-Предприятие» программная платформа это:

1) единые средства администрирования системы — пополняемые при подключении новых компонентов;

2) инструменты ведения нормативно-справочной информации;

3) единый документооборот;

4) средства разработки и формирования отчетов;

5) средства интеграции, в том числе поддержка сервисно-ориентированной архитектуры, средства извлечения и обработки данных, поддержка обмена данными в разных форматах и с разными приложениями.

По нашему мнению, это обязательно должно быть элементами современной программной платформы. Все остальное — настраивается и интегрируется.

Конечно, есть элементы средств разработки, различные дополнительные компоненты (именно компоненты, а не модули), но в большинстве случаев прикладные решения являются компонентами, настроенными на платформе, что позволяет существенно снизить требования к эксплуатирующему персоналу на стороне заказчика, т. е. нет необходимости содержать штат высококвалифицированных и дорогостоящих специалистов, умеющих работать только в определенной среде. Например, мы не стали разра-

батывать собственный язык для построения отчетов, для этого может использоваться либо ObjectPascal либо JavaScript.

Д. В. Денисов: А может ли заказчик разработать собственный функционал?

О. А. Страхов: Обязательно! Более того, заказчик может разрабатывать дополнительный функционал на чем ему это удобно и подключить его как библиотеки. Например, таким образом было создано специфическое прикладное решение по управлению персоналом под нужды заказчика, которое не вписывалось идеологически в платформу, и оно просто было подключено как набор библиотек. С точки зрения пользователя это абсолютно не заметно. Вообще мы продвигаем концепцию, что пользователь должен работать не с программным продуктом, а с рабочим местом.

С. В. Наливайко: У нас есть примеры, когда на достаточно крупном предприятии сопровождением и доработкой системы занимаются всего два человека.

О. А. Страхов: Важно отметить, что установка обновлений происходит без остановки работы системы. Обновления устанавливаются по регламенту, а применяются либо директивно (пользователи уведомляются о необходимости «выйти-войти»), либо просто при следующем входе в систему. За счет специальной организации базы данных введение новых реквизитов и новых представлений данных никак не сказывается на работе ранее созданных приложений.

Д. В. Денисов: Я бы хотел вернуться немного назад и поговорить подробнее о концепции ситуационного управления. Что вы вкладываете в это понятие, и как именно ситуационное управление реализовано в «SBC-Предприятие»?

С. В. Наливайко: Давайте начнем с понятия «цифровая экономика». Что это означает? Это означает, что пользователь информации должен иметь адекватные инструменты ее сбора, конфигурирования и представления, а главное, что эти процессы должны происхо-

дить очень оперативно, буквально в режиме онлайн. Олег Алексеевич, расскажите, пожалуйста, дальше.

О. А. Страхов: Мы стали использовать термин «Ситуационное управление» в задачах, связанных с поддержкой принятия решений, чуть менее 10 лет назад. Дополнительным стимулом к развитию подхода стали задачи по автоматизации ситуационных центров заказчиков. Дело в том, что недостаточно обеспечить передачу и отображение информации в удобном для заказчика виде, необходима предварительная обработка информации в наглядный вид, пригодный для принятия оперативных решений. Так родилась концепция ситуационного управления, по сути ориентированная на реализацию функций системы поддержки принятия решений (СППР), которая включает четыре составляющих:

1. Сбор, предварительный анализ и агрегирование данных, в том числе слабоструктурированных.

2. Удобный интерфейс для представления информации лицу, принимающему решение.

3. Средства обратной связи, т. е. быстрое реагирование на ситуацию и оперативное доведение принятых решений до исполнителей, в том числе на местах.

4. Возможности моделирования развития ситуаций на основе накопленных знаний, в том числе принятых решений и последствий их реализации, только важно понимать, что соответствующая библиотека нарабатывается годами, поэтому реализация этой возможности требует определенных усилий со стороны персонала по вводу информации.

С. В. Наливайко: Мало того, эта библиотека должна быть полной, потому что отсутствие полной информации может привести к принятию неправильных решений, т. к. в модели знаний будут установлены некорректные связи между исходными данными, описывающими ситуацию, принятыми решениями и их последствиями.

О. А. Страхов: Например, в задачах обеспечения ремонта и сервисного обслуживания

(PCO) зачастую важно не только учитывать наработку на отказ оборудования, но и условия эксплуатации, которые существенно влияют как раз на этот показатель. Если анализировать только наработку на отказ без учета условий эксплуатации, можно прийти к неверным выводам. Да, мы предлагаем инструмент, а не готовые решения, так называемые «лучшие практики». Откровенно говоря, ни одна серьезная компания своими лучшими практиками делиться не будет.

С. В. Наливайко: У нас был опыт, когда мы со своими партнерами исследовали лучшие практики на предмет их реализации. Оказалось, что необходимой информации нет, а анализ и осмысление того, что удалось найти, занял достаточно много времени. Поэтому вы никогда не получите те сведения, которые дают выгоды бизнесу. В лучшем случае можно получить информацию 10–15-летней давности.

О. А. Страхов: Или без важных деталей.

Д. В. Денисов: Согласно информации, размещенной на сайте ООО «Системные Бизнес-Компоненты», в качестве средства разработки «SBC-Предприятие» используется мультиплатформенная среда Lazarus и язык FreePascal. Чем обусловлен такой выбор? Какая используется СУБД?

С. В. Наливайко: Та часть системы, которая устанавливается на рабочих местах клиентов или на терминальных серверах, изначально разрабатывалась в среде Delphi. Причиной этому стала низкая требовательность к аппаратным ресурсам решений, разработанных в этой среде, а также широкая доступность на тот момент квалифицированных разработчиков, работающих в среде Delphi. Когда встал вопрос обеспечения переносимости на операционные системы, использующие ядро Linux, были рассмотрены различные варианты. В результате наилучшие результаты показала связка Lazarus+FreePascal (FPC).

О. А. Страхов: Что касается выбора СУБД, то изначально ориентировались на отечественные дистрибутивы операционных

систем, разрешенные к применению в системах, обрабатывающих конфиденциальную информацию — AltLinux, AstraLinux и Циркон. То есть на практике выбор был между MySQL и PostgreSQL. PostgreSQL подошел практически идеально благодаря хорошей организации работы с хранимыми процедурами и возможностям оптимизации. Кстати, связка Linux + PostgreSQL при работе с довольно большим объемом данных на тех же серверных ресурсах показала заметно лучшую производительность, чем Windows + Oracle.

Следует отметить, что вследствие самой архитектуры системы СУБД оказывает значительно большее влияние на производительность и функциональные возможности системы, нежели клиентская часть, которая выступает всего лишь в качестве средства интерфейса с пользователем — по аналогии с веб-браузером, обеспечивающим доступ к сайтам в Интернете.

При развитии системы, в том числе для мобильных клиентов, используются другие, более современные инструменты разработки. Возможности платформы SBC-Предприятие это позволяют.

Что касается среды разработки Lazarus, то мы были рады ее использовать в проекте миграции на Linux/FreeBSD/Android, признательны разработчикам за их труд по созданию этого продукта. Однако утверждать, что платформа SBC-Предприятие создана в среде Lazarus, было бы крайне неверно и не корректно.

Д. В. Денисов: «SBC-Предприятие» функционирует на Windows, а так же Linux, Android и FreeBSD. А возможно ли развертывание решения в компании, где используются ПК под управлением разных операционных систем и установлены разные СУБД? Какие предъявляются требования к серверу для развертывания «SBC-Предприятие»? Каковы возможности решений для Android? Система может быть установлена на смартфон, и пользователь получит удаленный доступ к корпоративной ИС? Реализованы ли

возможности доступа через «тонкого клиента»? Планирует ли компания «Системные Бизнес-Компоненты» развивать облачные решения на базе «SBC-Предприятие»?

С. В. Наливайко: Да, система успешно работает в смешанной среде, и у нас есть такой опыт. Изначально задача ставилась таким образом, чтобы создать ERP-платформу с широкими интеграционными возможностями. Для актуальной версии нашей системы нет существенной разницы, под какой из поддерживаемых операционных систем работает пользователь — на экране он будет видеть ту же «картинку». Несколько иначе дела обстоят с офисными пакетами, которые используются для представления отчетов — средства формирования отчетов для Microsoft Office и для Libre Office/Open Office отличаются, поэтому работать сразу с двумя «офисами» на одном компьютере одному пользователю вряд ли получится. Впрочем, и необходимости в подобном мы не видим. В части требований к серверным ресурсам — они полностью определяются теми задачами, которые должны быть решены системой. В качестве примера можно привести такой факт: офисный компьютер с двухъядерным процессором и 4 Гбайтами оперативной памяти вполне справляется с задачами сервера «SBC-Предприятие» для небольшой компании (15–20 пользователей) и при максимальном количестве компонентов. Правда, таким компаниям мы рекомендуем использовать терминал-серверный режим, что сильно упрощает администрирование рабочих мест, но существенно повышает требования к серверу.

Что касается Android, то «классическое» настольное приложение, равно как и терминальный режим, изначально предусмотрены для использования на планшетах. Такая потребность возникла при решении производственных задач выездными бригадами в сложных условиях работ — когда ноутбук развернуть не так-то просто. Примером могут служить задачи ремонта и сервисного обслуживания оборудования на местах эксплуатации. Поскольку

многие модули ввода системы адаптированы для ввода информации одной рукой, использование планшета оказывается удобным.

Что касается облачных версий и мобильных клиентов, то в настоящее время мы ведем с одним из наших партнеров работы по созданию облачной платформы с использованием отечественного серверного оборудования. В состав этой платформы войдет и «облачный» вариант клиента к платформе СБК-Предприятие. Планируем появление такой версии до конца 2018 года, к этому же времени собираемся широко представить мобильные клиенты для смартфонов на iOS и Android. Сейчас пока такие приложения делаются под конкретных заказчиков.

Д. В. Денисов: Как, по вашему мнению, сказались на российском рынке ERP-систем санкции, вводимые западными странами? Как введение санкций сказалось на процессе взаимодействия с западными ИТ-компаниями? Какие в связи с этим вы ожидаете изменения в отечественной ИТ-отрасли в целом и, в частности, в сегменте ERP-систем. Какие в связи с этим появляются возможности для российских компаний?

С. В. Наливайко: Как уже было сказано ранее, российский рынок изначально был ориентирован на зависимость от западного ПО. Непосредственно введение санкций привело к двум тенденциям.

Во-первых, западные производители начали искать пути обхода — и в большинстве своем нашли такие пути, оказывая услуги и осуществляя продажи через посредников как в России, так и за рубежом.

Во-вторых, лица, принимающие решения в отрасли, наконец-то начали осознавать опасность сложившейся ситуации на рынке и озаботились поиском отечественных решений-аналогов. Но здесь стоит заметить, что зарубежные компании тратят весьма и весьма значительные средства на исследования и разработку, что отечественные компании, вынужденно довольствовавшиеся «нишами», просто не могли себе позволить.

Для создания качественного, отвечающего современным требованиям продукта необходимо пройти полный проектный цикл. Отсюда идеи создать полноценный продукт — операционную систему или ERP-систему — с нуля за пару лет, даже при бесперебойном полномасштабном финансировании, изначально обречены на провал. Определенную поддержку, конечно, способен оказать валютный курс, установившийся после известных событий. Однако этот фактор также не стоит переоценивать: с развитием технологий удаленной разработки уровень зарплат квалифицированных разработчиков привязан к иностранным валютам. Иначе говоря, за полцены ту же работу серьезный специалист делать не будет.

С учетом сказанного основные изменения на рынке для отечественных разработчиков могут быть связаны, прежде всего, с активизацией импортозамещения в госсекторе, а также на системообразующих предприятиях. Тормозящим фактором останется менталитет руководства предприятий и их ИТ-служб, привыкших ориентироваться исключительно на иностранное. Однако реальной альтернативы созданию и использованию отечественных продуктов — программных, аппаратных, микроэлектроники — для применения в жизненно важных сферах просто не существует. Понимание этого, по-видимому, есть как у руководства страны, так и у самих разработчиков. Остается донести неизбежность этого процесса до руководства соответствующих предприятий и холдингов, их объединяющих.

Л. А. Малеева: А какие инновации ожидают ИТ-отрасль в целом и рынок ERP-систем в частности? Какие видятся наиболее перспективные направления развития ERP-систем?

С. В. Наливайко: Предсказывать будущее сложно, слишком много влияющих факторов. Однако можно отметить несколько тенденций, которые уже себя проявляют.

Например, существенно возросшие объемы информации, нужной для принятия решений, требуют использования интеллектуальных методов и средств извлечения данных

и их оперативной систематизации. И эти методы должны быть интегрируемыми в прикладные решения — в том числе, в ERP-системы, в системы ситуационного управления и тем более в системы управления жизненным циклом.

Интересны технологии смарт-контрактов, хоть и находящиеся пока еще на начальном этапе развития. Следует ожидать появления новых способов и средств визуализации информации, поскольку оперативно оценивать множество показателей, представленных в разном виде, становится все сложнее. Несомненно, будут развиваться «облачные» решения.

Особо стоит отметить так называемые «частные» облака, обеспечивающие наилучшую безопасность данных. Этот подход, при всей его затратности на сегодняшнем этапе, имеет хорошие перспективы по мере выхода на рынок быстро разворачиваемых ЦОД и высокоскоростных сетей связи. Что же касается конкретно ERP-систем, то уже сейчас они могут и должны рассматриваться как «часть большего», то есть как платформа для объединения модулей и отдельных программных систем в единую интегрированную систему, в том числе — облачную.

Л. А. Малеева: Хотелось бы услышать также, что, по вашему мнению, необходимо знать выпускнику, планирующему карьеру в ИТ-отрасли, а именно в сфере разработки и внедрения ERP-систем? Знание каких операционных систем, СУБД, средств и технологий будут востребованы работодателями в ближайшие 2–3 года? Потребуется ли какие-то специфические навыки и умения?

С. В. Наливайко: С учетом сказанного будущий разработчик должен, прежде всего, обладать широким кругозором, уметь работать с источниками информации и, что немаловажно, с технической документацией. Важно понимать, что не документированный должным образом продукт, будь то ERP-платформа или, скажем, соковыжималка, есть предмет разового использования. Сломалось — выбросить...

Что касается знания программных продуктов, то, прежде всего, надо знать и уметь использовать по назначению средства коллективной разработки, особенно основанные на гибких методологиях. Но при этом владеть и «классическими» моделями разработки, четко осознавая границы их применимости.

В части операционных систем будущему администратору или разработчику придется уметь работать со всеми распространенными системами, но прежде всего — с системами на ядре Linux (включая Android).

Что касается СУБД, то, наряду с популярными реляционными, стоит присматриваться к не реляционным (объектным) СУБД. Будущим администраторам систем стоит хорошо ориентироваться в средствах мониторинга и управления сложными системами, понимать основные принципы действия таких систем. Но самое главное — уметь учиться, извлекать новые знания и их систематизировать. ИТ-отрасль весьма динамична, и знания десятилетней давности могут оказаться безнадежно устаревшими.

Д. В. Денисов: И наконец, заключительный вопрос нашей беседы: Сергей Викторович, Олег Алексеевич, возможно, у вас есть какие-то идеи, соображения, что-то, что вы хотели бы особо донести до аудитории журнала «Прикладная информатика»...

С. В. Наливайко: Главное, чтобы нам хотелось донести — это понимание того, что самолеты надо строить самим.

О. А. Страхов: Я присоединяюсь к Сергею Викторовичу. Мы уже говорили о зависимости от западных разработок, и от системных архитекторов, и от зарубежных постановщиков задач, при этом мы не развиваем собственные технологии, собственные школы подготовки высококвалифицированных ИТ-специалистов. И мне бы хотелось призвать руководителей компаний, ИТ-директоров более внимательно анализировать отечественные разработки не только прикладного, но и системного программного обеспечения.

С. В. Наливайко: Я добавлю, что есть «свет в конце туннеля», поскольку сейчас на государственном уровне перед государственными структурами и предприятиями, в том числе оборонными, а это критически важно для безопасности государства, так вот сейчас на государственном уровне ставится задача перехода на отечественное программное обеспечение. Мы надеемся, что это по-

служит стимулом к внедрению отечественных разработок, их развитию и совершенствованию, которое требует значительного финансирования, а оно возможно только при соответствующем объеме заказов.

Материал подготовили

*Денис Денисов, ddenisov@synergy.ru,
и Лорета Малеева, lorimaleeva@gmail.com*

Interview

Prospects of development and implementation of ERP systems in Russia: point of view of the company's management «System Business Components»

There are a lot of practical articles and scientific works devoted to development and implementation of ERP systems, however, the rapid development of information technology, the change in the situation in the IT market caused by the introduction of sanctions and focus on import substitution allow to evaluate the possibilities and prospects of automation of domestic enterprises. To discuss this topic on June 26, 2018 a meeting with the General Director of the «System Business Components» company Nalivayko S. V. and the Deputy General Director for Science and Development of the «System Business Components» company Strahov O. A. with the head of Information Systems Department of the «Synergy» University Denisov D. V. was organized. Maleeva L. A., studying journalism in the Moscow Humanitarian University and now our intern, represented the «Applied Informatics Journal» editorial board in the conversation and the preparation of the material. The result of this meeting — interview published below.

About authors:

D. Denisov, *PhD in Economics, Head of Information Systems Department, «Synergy» University, Moscow, ddenisov@synergy.ru*

L. Maleeva, *Student of the Journalism Department, Moscow Humanitarian University, lorimaleeva@gmail.com*

For citation:

Interview. Prospects of development and implementation of ERP systems in Russia: point of view of the company's management «System Business Components». Prikladnaya informatika — Journal of Applied Informatics, vol. 13, no. 4 (76), pp. 5–14 (in Russian).

Е. Д. Стрельцова, докт. экон. наук, доцент, Южно-Российский политехнический университет имени М. И. Платова, г. Новочеркасск, eL_strel@mail.ru

А. И. Бородин, докт. экон. наук, доцент, РЭУ им. Г. В. Плеханова, г. Москва, aib-2004@yandex.ru

И. В. Яковенко, канд. экон. наук, доцент, Южно-Российский политехнический университет имени М. И. Платова, г. Новочеркасск, iranyak@mail.ru

Модельный инструментарий стратегического управления промышленным предприятием

Целью статьи является разработка инструментария, поддерживающего принятие решений при выборе стратегических ориентиров развития предприятий с использованием экономико-математических методов. Разработана модель процесса стратегического управления в виде логической структуры, содержащей объединенную решающую процедуру решения задач, начиная от исследования внешней среды и разработки миссии и заканчивая созданием экономико-математического инструментария и его использованием для оценки принимаемых решений. Построен экономико-математический инструментарий для поддержки принятия решений относительно стратегических ориентиров развития промышленного предприятия в условиях стохастической неопределенности.

Ключевые слова: стратегическое управление, системный анализ, стохастическая неопределенность, рентабельность, имитационная модель.

Введение

Современная экономическая ситуация характеризуется происходящими бифуркационными изменениями, приводящими к неустойчивости функционирования промышленных предприятий в условиях флуктуаций бизнес-среды [1]. Эти флуктуации вызваны развитием глобализации, активизацией инновационной составляющей, повышением роли нематериальных активов и др. Перспективным направлением, обеспечивающим эффективное развитие промышленных предприятий, становится применение методов стратегического управления, позволяющего решать проблемы его долговременного развития. Формирование в организации миссии, долговременных планов развития не является доказательством наличия на промышленном предприятии механизмов стра-

тегического управления. Построение долгосрочных планов не позволяет вести учет влияния изменений внешней среды. Важнейшим результатом разработки процессов стратегического управления представляется создание и использование встроенного в структуру предприятия экономико-математического инструментария, адекватно реагирующего на изменения, происходящее в бизнес-сфере. Высокий динамизм внешней среды, а также происходящие в ней процессы неравновесности и неопределенности обуславливают необходимость в рамках стратегического управления осуществлять количественную оценку различных сценариев развития промышленного предприятия на длительную перспективу, как залога его устойчивого развития в рыночной среде.

Происходящие в мировой экономике кризисные процессы в сочетании с антироссий-

скими санкциями и курсом на импортозамещение, а также жесткая конкурентная борьба с зарубежными предприятиями актуализируют проблему создания эффективной системы управления, чувствительной к флуктуациям внешней среды и способной оказывать влияние на эффективность процессов производства и качество выпускаемой продукции. Использование традиционных методов планирования, основанных на системе учета и анализа, не дает требуемого временем эффекта в управлении и диктует необходимость их усовершенствования, в связи с чем значительно повышается интерес промышленных предприятий к применению принципов и методов стратегического управления для повышения конкурентоспособности выпускаемой продукции и обеспечения стабильного экономического роста. Стратегическое управление промышленными предприятиями необходимо производить на основе применения современных экономико-математических методов и моделей с целью количественной оценки и научной обоснованности принимаемых решений.

Как современное научное направление теория стратегического управления создана на рубеже 1980–1990 гг. Как управленческий термин стратегическое управление стало известно в середине XX в., когда в организациях обострилась проблема влияния на процессы их функционирования факторов внешней среды, что породило большое количество публикаций отечественных и зарубежных исследователей, в которых приведены различные трактовки понятия «Стратегия». Так, А. Томпсон и А. Стрикленд [2] представляют стратегию планом управления, выполнение которого укрепляет позиции организации, обеспечивает удовлетворение потребностей и достижение поставленных ею целей. М. Мескон, М. Альберт и Ф. Хедоури [3] считают, что стратегия должна представлять собой комплексный план, обеспечивающий выполнение предприятием своей миссии. К подобному выводу пришли Л. С. Шеховцева,

А. В. Тебекин [4, 5] и др. Авторы настоящей статьи поддерживают позицию, что ключевой вопрос стратегического управления — обеспечение развития организации в условиях воздействий нестабильной внешней среды. В связи с этим основным направлением создания применяемого в стратегическом управлении инструментария является обеспечение адекватного реагирования на вариации параметров внешней среды на основе экономико-математических моделей. Разработке экономико-математических моделей, имитирующих погруженных в случайную среду объектов, посвящен широкий спектр работ отечественных ученых. Вопросы имитационного моделирования для изучения экономических систем и процессов отражены в исследованиях О. В. Бульгиной, А. А. Емельянова, Н. З. Емельяновой [6]. Общие приемы имитационного моделирования изложены в статье Н. Н. Прокимнова [7]. Подходы к созданию акторных моделей корпоративных информационных систем предложены Е. А. Власовой [8]. Результаты применения методов имитационного моделирования для описания поведения инвестиционных процессов рассмотрены А. А. Емельяновым, Е. А. Власовой, Емельяновой Н. З., Н. Н. Прокимновым [9].

В настоящей статье решается проблема усовершенствования экономико-математического инструментария, применяемого в стратегическом управлении промышленным предприятием, являющегося драйвером его экономического развития.

Стратегическое управление промышленным предприятием на основе системного подхода

Проведя анализ функционирования промышленного предприятия, авторы определили, что отличительной чертой протекающих на нем процессов является, во-первых, постоянно изменяющиеся экономические условия, вызванные как технологическими аспектами, так и различными инновациями,

необходимостью применения наукоемких технологий и др. Во-вторых, ключевой проблемой управления промышленным предприятием представляется необходимость максимизации эффективности производства, получаемой прибыли и рентабельности, удержания старых рынков и завоевания новых и как следствие удовлетворения потребности потребителей. В связи с этим авторами статьи решается вопрос исследования стратегического управления промышленными предприятиями с точки зрения развивающихся систем. Процесс исследования проведен на основе методологии системного анализа с использованием концепции диалектического развития. Системные исследования проведены на основе подхода, предложенного Е. Д. Стрельцовой [10]. В соответствии с этим подходом в качестве внутреннего источника развития рассматривается диалектическое единство целевой категории деятельности и категории средств достижения цели. В роли целевой категории выступает выбираемый сценарий, стратегический ориентир деятельности, к которой приводит последовательность действий как категорий средств достижения цели. К этой последовательности действий относится и стратегическое управление, конечным результатом которого является разработка стратегии промышленного предприятия. В такой постановке методологических исследований процедуре системного анализа подвергается логически связанная четверка объектов: <процесс разработки стратегии, относящийся к категории средств достижения цели> ↔ <стратегия, как объект разработки, относящийся к целевой категории> ↔ <стратегия в виде последовательности действий, относящихся к категории средств достижения цели> ↔ <стратегический ориентир (целевая категория)>.

Понятие логической связи означает, что все подцели размещены в единой логической цепочке деятельности в процессе поэтапного достижения главной цели в системе стратегического управления как развиваю-

щейся системе, где подцели трансформируются в средства, а затем средства — в подцели. В понятие «качество принимаемых решений» вкладывается смысл совокупности свойств, благодаря которым объект управления переводится в состояние, обеспечивающее достижение поставленной цели. Что касается эффективности, то в это понятие вкладывается смысл свойства действий, которые способны привести к нужным результатам. В качестве показателя эффективности стратегического управления в статье рассматривается рентабельность. Применение метода системного анализа к предложенному объекту обусловило рассмотрение сложной цели управления, представляющей собой диалектическое единство подцелей: <повышение эффективности процесса разработки стратегии управления> ↔ <повышение качества управленческих решений> ↔ <повышение эффективности принимаемых решений по определению стратегического направления развития> ↔ <повышение качества решений>.

Применение методологии системного анализа к предложенному объекту исследования имплицировало концептуальное представление стратегического управления взаимодействием систем *SISTEM1* и *SISTEM2* (рис. 1).

Система *SISTEM1* идентифицирует стратегическое управление, в котором объектом принятия решений *MEN1* является выбор стратегии управления. Система *SISTEM2* осуществляет целевое использование принятой стратегии в процессе выбора стратегического ориентира функционирования промышленного предприятия. Процесс стратегического управления представляется автором в виде последовательности действий по установлению траектории роста производительности и эффективности промышленного предприятия по отношению к уровню конкурентов: <исследование состояния бизнес-среды> ↔ <назначение миссии и целей функционирования предприятия> ↔ <формирование множества

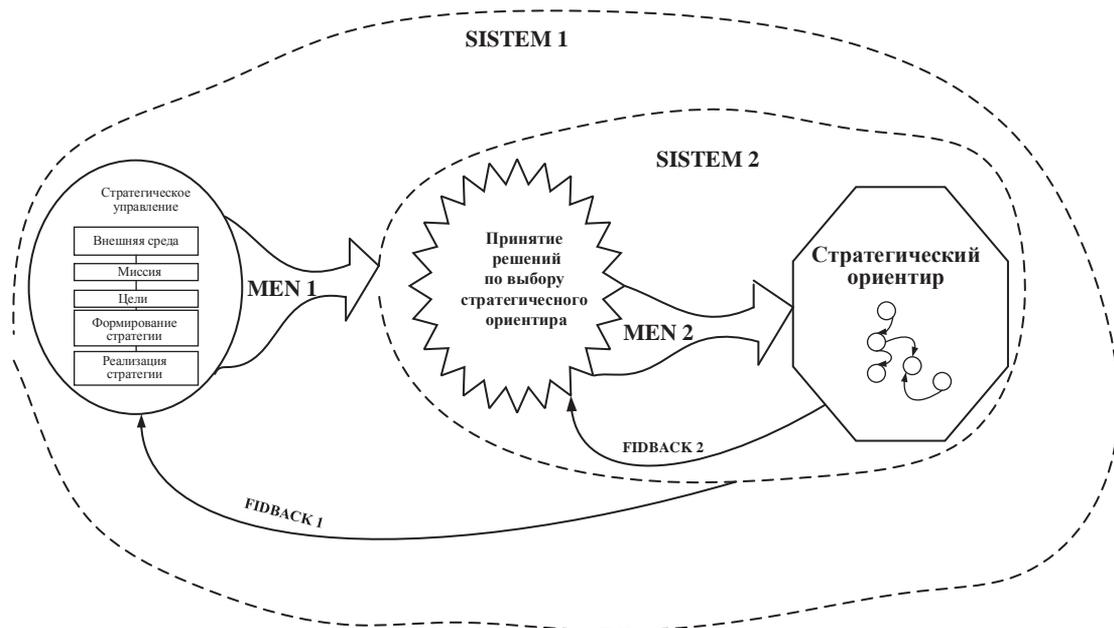


Рис. 1. Концептуальная модель стратегического управления как развивающейся системы
Fig. 1. Conceptual model of strategic management as a developing system

стратегических ориентиров><=><<выбор стратегического ориентира>.

Реализация каждого из обозначенных действий предполагает выполнение множества задач, решаемых в ходе итеративного процесса с обратными связями *FIDBACK1*, *FIDBACK2*, обеспечивающими корректировку решений задач на предшествующих этапах. Следуя методологии системного подхода, задача «Стратегическое управление» *SU* разбивается на подзадачи $SU = \{SU_{ij}\}$, индексы *i* и *j* которых означают, соответственно, стадию стратегического управления и номер задачи, где *i* = 1 идентифицирует стадию разработки стратегии, а *i* = 2 — стадию ее реализации.

Авторами статьи предложена модель формального представления стратегического управления в виде логической структуры последовательности принятия решений. В логической схеме локальные задачи SU_{ij} описаны кортежем $\langle NAME_{ij}, RP_{ij}, M_{ij} \rangle$ с компонентами, означающими, соответственно: наименование решаемой задачи; применяемую

для решения процедуру или инструментарий; управленческое решение. Логическая структура дает возможность задать порядок на множестве задач SU_{ij} , рассматривая их как целостный, логически связанный, управляемый комплекс, решаемый с помощью целостной процедуры $RP = \{RP_{ij}\}$, применение которой позволяет получить стратегический ориентир предприятия [11]. Целостная решающая процедура $RP = \{RP_{ij}\}$ включает построенный авторами модельный инструментарий, дающий оценку выбираемому стратегическому ориентиру и изложенный в следующем разделе настоящей статьи. На рисунке 2 приведена логическая структура стратегического управления промышленным предприятием.

В предложенной логической структуре задачи SU_{11} , SU_{14} , SU_{21} , SU_{24} решаются посредством применения решающих процедур как инструментария стратегического управления. Стадия разработки стратегии (*i* = 1) использует решающие процедуры $RP = \langle RP_{11}, RP_{12}, RP_{13}, RP_{14} \rangle$, в состав которых включены как формальные экономико-

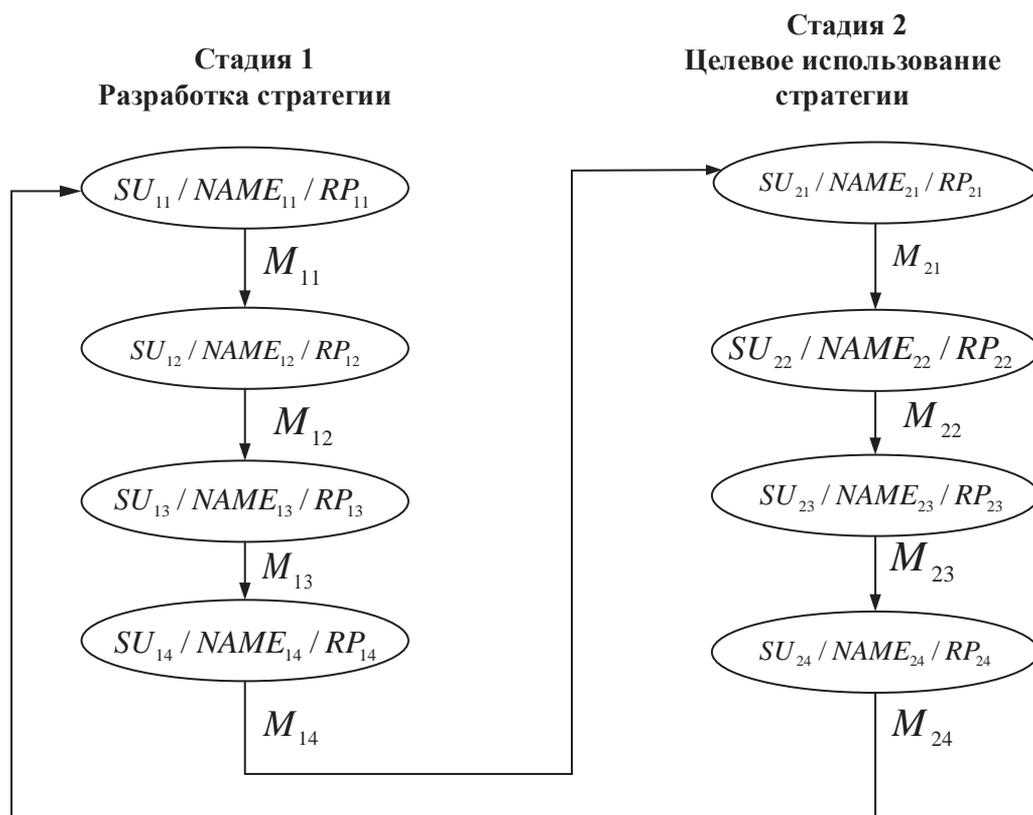


Рис. 2. Логическая структура стратегического управления промышленным предприятием
 Fig. 2. Logical structure of strategic management of industrial enterprise

математические методы (RP_{11} — математико-статистические методы, RP_{14} — методы компьютерного моделирования), так и эвристические, не являющиеся формальными методы решения RP_{12} , RP_{13} .

На базе применения формальных методов созданы следующие модельные инструменты:

- M_{11} — математическая модель построения законов распределений для финансовых потоков, отражающих величины прибыли и затрат капитала (далее в статье эта модель обозначена как $SIMULATE_1$);
- M_{14} — модель, позволяющая прогнозировать рентабельность стратегического ориентира (далее в статье модель обозначена как $SIMULATE_2$).

На следующей стадии, касающейся реализации стратегии ($i=2$), решения M_{11} , M_{14}

представляют собой средства, применяемые в процессе выбора управленческих стратегических решений. Таким образом, на стадии целевого использования решения M_{11} , M_{14} выполняют функции решающих процедур RP_{21} , RP_{24} , тем самым демонстрируя трансформацию целей в средства, как в развивающейся системе. Задачи $SU = \{SU_{ij}\}$, на множестве которых задано отношение порядка посредством логической структуры, описаны в табл. 1.

Развитие экономико-математического аппарата и компьютерной техники позволит в дальнейшем создавать инструментарий для решения поставленных в логической структуре стратегического управления слабоструктурированных задач и использовать вместо неформальных решающих процедур формализованные процедуры, модели, алгоритмы.

Таблица 1. Задачи и решающие процедуры в логической структуре стратегического управления**Table 1.** Tasks and decision procedures in the logical structure of strategic management

Стадии	Обозначение задач	Задачи	Решающие процедуры	Управляющие решения
$i = 1$ разработка стратегии	SU_{11}	$NAME_{11}$ — подходы и инструментарий для исследования воздействий среды	RP_{11} — статистические методы обработки информации	M_{11} — экономико-математические модели описания законов распределения влияний окружающей среды ($SIMULATE_1$)
	SU_{12}	$NAME_{12}$ — разработка миссии промышленного предприятия	RP_{12} — неформальный метод	M_{12} — миссия промышленного предприятия
	SU_{13}	$NAME_{13}$ — формирование множества стратегических решений	RP_{13} — неформальный метод	M_{13} — множество стратегических решений
	SU_{14}	$NAME_{14}$ — разработка подходов и инструментов оценки результатов управленческих решений, применяемых на этапе выбора стратегических ориентиров	RP_{14} — методы компьютерного моделирования	M_{14} — компьютерная модель составления прогноза рентабельности ($SIMULATE_2$)
$i = 2$ целевое использование стратегии	SU_{21}	$NAME_{21}$ — исследование окружающей среды: формализация случайных величин «прибыль» и «затраты капитала»	RP_{21} — экономико-математическая модель формального описания случайных величин «прибыль» и «затраты капитала» ($SIMULATE_1$)	M_{21} — формальное описание случайных величин «прибыль» и «затраты капитала» ($SIMULATE_1$) в виде закона распределения
	SU_{22}	$NAME_{22}$ — тестирование соответствий стратегических решений принятой на предприятии миссии	RP_{22} — неформализованный метод	M_{22} — вывод о соответствии стратегических решений принятой на предприятии миссии
	SU_{23}	$NAME_{23}$ — составление множества (портфеля) стратегических ориентиров	RP_{23} — неформализованный метод	M_{23} — включенные в портфель стратегические альтернативы
	SU_{24}	$NAME_{24}$ — составление прогноза рентабельности, характеризующей стратегический ориентир	RP_{24} — компьютерная модель составления прогноза рентабельности ($SIMULATE_2$)	M_{24} — количественное обоснование стратегического ориентира с помощью показателя рентабельности

Модельный инструментарий стратегического управления

Перед изложением материала по созданию инструментов стратегического управления в статье, руководствуясь принципом тожде-

ства формальной логики, определено понятие «инструмент» стратегического управления. Значение слова «инструмент», заимствованное из латинского *instrumentum*, означает средство труда или встроенный в машину механизм, выполняющий некоторую работу.

Авторами статьи в понятие «инструмент стратегического управления» вкладывается смысл семейства концептуальных позиций, теоретико-методологических подходов, методов, моделей и характеристик, используемых в ходе принятия и исполнения управляющих решений, позволяющих при их реализации осуществлять выбор стратегического ориентира, обеспечивающего эффективное эволюционирование предприятий в долгосрочной перспективе в условиях неопределенности влияния внешней среды [1].

Ключевой проблемой стратегического управления промышленным предприятием в условиях неопределенности воздействия внешней среды является обеспечение его эффективности и конкурентных преимуществ.

Во множестве показателей эффективности функционирования промышленных предприятий одну из главных ролей играют показатели, отражающие рентабельность. Управление этими показателями, отражающими соотношение между величинами прибыли и вложенного капитала, имеет стратегическую направленность и происходит в условиях неопределенности, так как прибыль и затраты изменяются во времени случайным образом. Формальное описание неопределенности и создание для поддержки принятия решений экономико-математических моделей продиктовано необходимостью количественного обоснования принимаемых стратегических решений.

Рассматривая множество относительных показателей «рентабельность», можно заключить, что это относительные характеристики $\{Rentabl_i\}_{i \in I}$, $I = 1, 2, 3, \dots, N$ экономической эффективности того, как используются различного рода ресурсы:

$$Rentabl_i = \frac{Profit_i}{K_i}, \quad i = \overline{1, N},$$

где $Rentabl_i$ — показатель рентабельности вида i ; $Profit_i$ — прибыль, полученная при доходности вида i ;

K_i — капитал, формирующий величину прибыли.

Если обозначить набор стратегических ориентиров в портфеле как $\{ST_i\}_{i \in I}$, а их характеристики рентабельности как $Profitability_{ij}$, где i — стратегический ориентир, а j — его показатель рентабельности, то задачу управления принятием решений на основе показателей рентабельности можно представить как выбор такой стратегии $Strateg_\alpha \in \{Strateg_i\}_{i \in I}$, при которой рентабельность $Profitability_{\alpha j}$, $\alpha = \overline{1, k}$, $j = \overline{1, N}$ является максимальной (рис. 3).

На рисунке 3 альтернативы $\{Strateg_i\}_{i \in I}$ играют роль управляющих переменных, $Rentabl_i$ — выходов, а величины прибыли $Profit_i$ и капитала K_p , используемого при создании прибыли, — роль возмущений. Элементы K_i представляют собой активы, ресурсы, потоки и др. Формальная постановка задачи стратегического управления имеет вид:

$$\forall Strateg_1, Strateg_2, \dots, Strateg_N \exists \\ \exists Strateg^* / Rentabl_i(Strateg^*) \Rightarrow \max.$$

Решение задачи управления выбором стратегических ориентиров требует создания экономико-математической модели прогнозирования прибыли $Profit_i$ и затрат капитала K_i для реализации стратегического ориентира промышленного предприятия $Strateg \in \{Strateg_i\}_{i \in I}$. Ввиду случайного характера изменения величин $Profit_i$ и K_i , $i = \overline{1, N}$ их формальное описание выполнено на базе применения методов математико-статистического исследования. Авторами предложен инструментарий в виде семейства экономико-математических моделей $SIMULATE = \langle SIMULATE_1, SIMULATE_2, SIMULATE_3 \rangle$ оценки результатов принимаемых решений по выбору стратегических ориентиров $Stratrg \in \{Strateg_i\}_{i \in I}$ функционирования предприятия, где $SIMULATE_1$ экстраполирует статистические данные, описывающие величины $Profit_i$ и K_p , $i = \overline{1, N}$, $SIMULATE_2$ формально описывает потоки текущих затрат и прибыли для реализации стратегического ориентира, $SIMULATE_3$ предназначена для

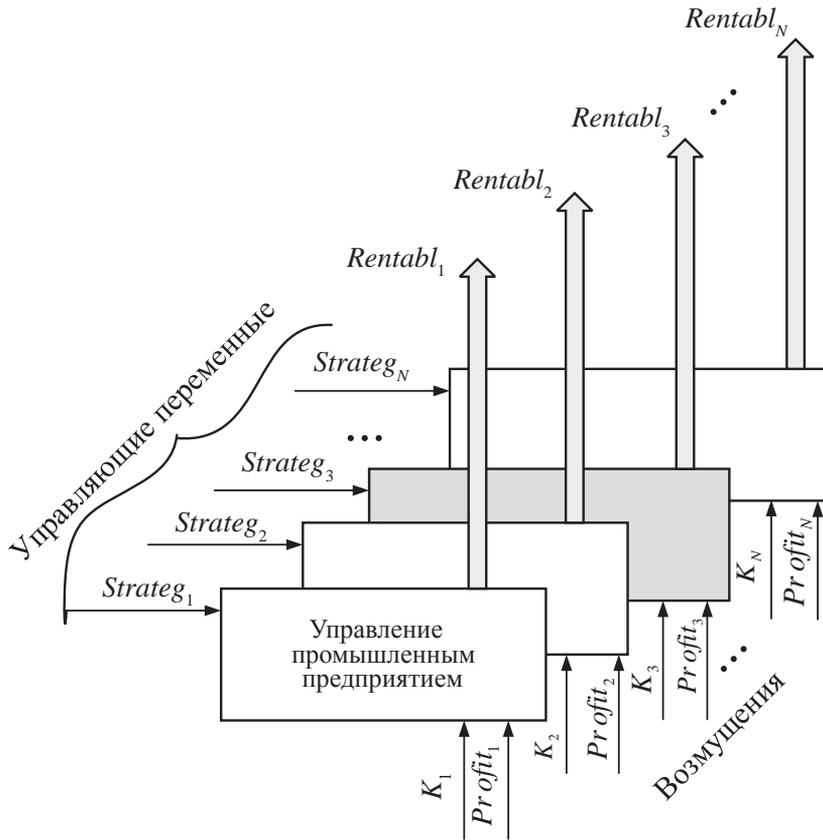


Рис. 3. Концептуальная модель постановки задачи стратегического управления
 Fig. 3. A conceptual model of the problem statement strategic management

прогнозирования рентабельности и выполнена в классе имитационных моделей.

Экстраполяция тенденций развития прибыли и затрат капитала

Математическая модель $SIMULATE_1$ осуществляет экстраполяцию собранных статистических данных, характеризующих динамику величин $Profit_i$ и K_i , $i=1, N$, на основе использования математического аппарата искусственных нейронных сетей. При этом с помощью инструмента Neural Networks Toolbox создана модель нейронной сети, изображенной на рис. 4. При этом использовалась двухслойная нейронная сеть с одним входом. Первый слой нейронной сети содержит 20 нейронов, а второй — один нейрон.

Преобразование сигналов в первом слое осуществляется искусственными нейронами с помощью активационной функции «гиперболический тангенс»:

$$F(Profit) = \frac{e^{Profit} - e^{-Profit}}{e^{Profit} + e^{-Profit}}; F(K) = \frac{e^K - e^{-K}}{e^K + e^{-K}}.$$

Обучение нейронной сети производилось на тестовых выборках, представленных в табл. 2 и 3. Результаты обучения нейронной сети приведены на рис. 5.

Результаты прогнозирования величин $Profit$ и K приведены на рис. 6 и 7.

Реальные статистические данные величин прибыли $Profit_i$ и капитала K_i , $i=1, N$ после объединения с результатами прогнозирования использовались для построения моделью

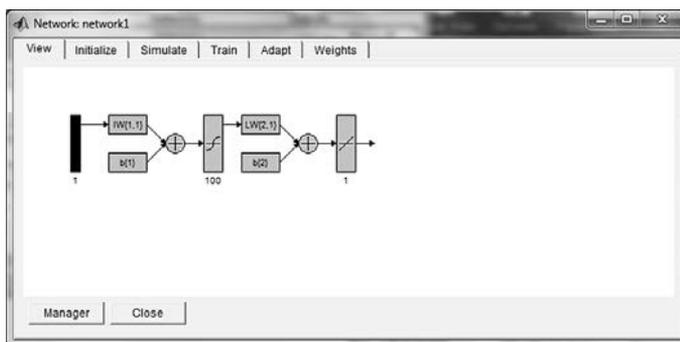


Рис. 4. Искусственная нейронная сеть для экстраполяции тенденций развития величин $Profit_i$ и K_i , $i = \overline{1, N}$

Fig. 4. Artificial neural network for extrapolation of trends in the development of quantities $Profit_i$ and K_i , $i = \overline{1, N}$

Таблица 2. Статистические данные случайной величины $Profit_i$
Table 2. Statistics of a random variable $Profit_i$

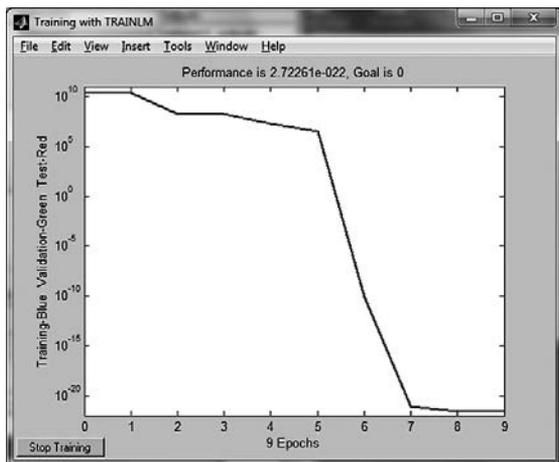
№ п/п	$Profit_i$, тыс. руб.	№ п/п	$Profit_i$, тыс. руб.	№ п/п	$Profit_i$, тыс. руб.
1	56 485	8	97 352	15	198 578
2	65 982	9	129 749	16	215 436
3	65 874	10	145 786	17	216 939
4	76 523	11	146 837	18	223 163
5	78 451	12	175 842	19	241 592
6	89 641	13	184 693	20	251 648
7	95 675	14	188 936	21	254 623

Таблица 3. Статистические данные случайной величины K_i
Table 3. Statistics of a random variable K_i

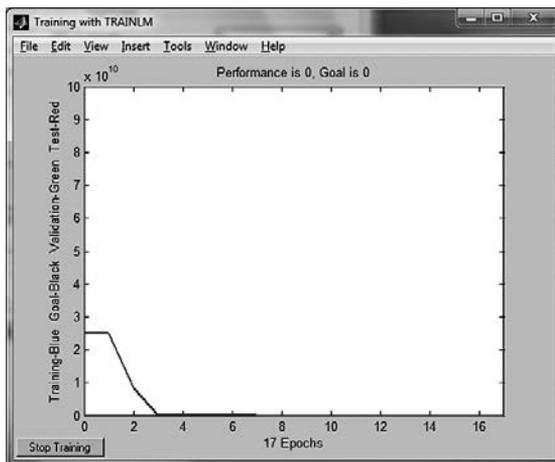
№ п/п	K_i , тыс. руб.	№ п/п	K_i , тыс. руб.	№ п/п	K_i , тыс.руб.
1	125 915	8	197 397	15	396 986
2	128 678	9	199 588	16	428 563
3	146 413	10	265 385	17	498 395
4	156 590	11	298 479	18	501 836
5	179 853	12	345 731	19	582 436
6	183 755	13	345 815	20	586 373
7	189 346	14	359 629	21	689 154

$SIMULATE_2$ интервальных рядов распределения. Построенные законы распределения являются исходными данными для работы имитационной модели $SIMULATE_3$, функционирующей на базе применения метода статистических испытаний для генерации случайных

чисел по заданным рядам распределения, описывающим потоки прибыли $Profit_i$ и капитала K_i , $i = \overline{1, N}$. Модель $SIMULATE_3$ представляется развитием исследований, проведенных А. И. Бородиным, Е. Д. Стрельцовой, А. В. Ковалёвой [12]. Разыгранные случайные



a



b

Рис. 5. Результат обучений нейронной сети, *a* — по тестовой выборке величины *Profit*; *b* — по тестовой выборке величины *K*
Fig. 5. The result of neural network training, (*a* test sample size *Profit*; *b* the test sample values *K*)

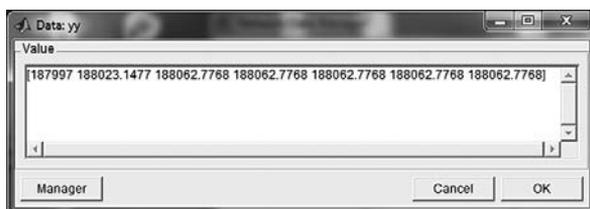


Рис. 6. Результаты прогнозирования величины *Profit*
Fig. 6. The results of forecasting the magnitude *Profit*

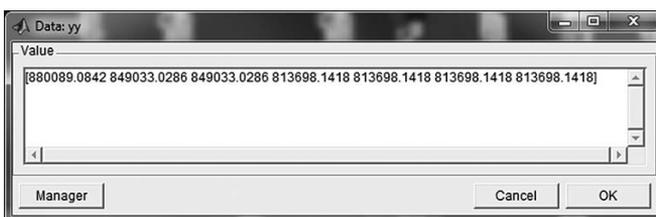


Рис. 7. Результаты прогнозирования величины *K*
Fig. 7. The results of forecasting the magnitude *K*

числа, рассматриваемые как возможные значения случайных величин $Profit_i$ и K_i , участвуют в выражении для определения рентабельности

$$Rentabl_i = \frac{\sum_{t=1}^T Profit_i(t)}{\sum_{t=1}^T K_i(t)},$$

где T – временной интервал.

Семейство моделей

$$SIMULATE = \langle SIMULATE_1, SIMULATE_2, SIMULATE_3 \rangle$$

реализовано в виде программного продукта PROFIT, главное окно которого представлено на рис. 8.

Верхняя часть главного окна программного продукта PROFIT содержит меню пользова-

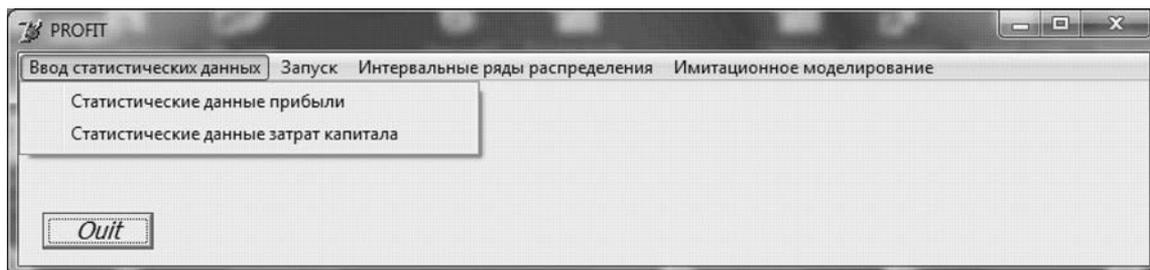


Рис. 8. Главное окно программы PROFIT
Fig. 8. The main window of the PROFIT program

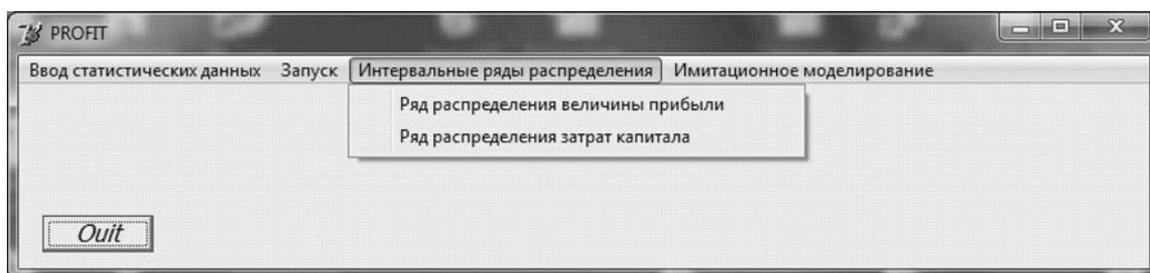


Рис. 9. Вид главного окна программы PROFIT при выполнении функции построения интервальных рядов распределения случайных величин *Profit* и *K*

Fig. 9. The view of the main window of the PROFIT program when performing the function of constructing interval series of distribution of random variables *Profit* and *K*

теля, состоящее из разделов «Ввод статистических данных», «Запуск», «Интервальные ряды распределения», «Имитационное моделирование». Первый раздел предназначен для ввода статистических данных, после активизации которого появляется каскадное меню, включающее пункты для ввода статистических данных, характеризующих прибыль ($Profit(t)$) и затраты капитала ($K(t)$). После активизации команды «Запуск» осуществляется построение интервальных рядов распределения для введенных статистических данных, которые появляются на экране компьютера при активизации раздела «Интервальные ряды распределения» (рис. 9).

По команде «Имитационное моделирование» на основе применения метода статистических испытаний воспроизводится динамика потоков величин $Profit_i$ и K_i , $i = 1, N$ в соответствии с построенными законами рас-

пределения вероятностей и осуществляется прогнозирование величины рентабельности $Rentabl_i$. Результатом работы программы являются сгенерированные прогнозные значения прибыли $Profit_i$ и затрат K_i , $i = 1, N$, а также величины рентабельности (рис. 10).

Разработанный на основе экономико-математических моделей программный продукт PROFIT позволяет составлять прогнозы относительно рентабельности стратегического ориентира развития промышленного предприятия и оценить различные альтернативы принятия решений.

Заключение

Современная экономическая среда, в которую погружены промышленные предприятия России, характеризуется широким спектром проблем, среди которых развитие гло-

Результат по временному периоду:

№	Величина Прибыли	Величина Затрат	Коэффициент риска
1	65993,4586	149227,1816	0,4422
2	72332,4310	149227,1816	0,4847
3	91349,3481	149227,1816	0,6121
4	85010,3757	177201,7994	0,4797
5	91349,3481	177201,7994	0,5155
6	91349,3481	158552,0542	0,5761
7	65993,4586	167876,9268	0,3931
8	65993,4586	177201,7994	0,3724
9	91349,3481	158552,0542	0,5761
10	65993,4586	167876,9268	0,3931
11	85010,3757	167876,9268	0,5064
12	65993,4586	149227,1816	0,4422
13	91349,3481	149227,1816	0,6121
14	65993,4586	177201,7994	0,3724
15	85010,3757	149227,1816	0,5697

Результат итоговый:

Σ Прибыли	Σ Затрат	Средний Коэф. Риска	Рентабельность
789835,1979	16051027,7441	0,4948	0,4921

Рис. 10. Результат определения рентабельности

Fig. 10. The result of the determination of profitability

бализации и повсеместной информатизации, активизация инновационной деятельности, кризисные явления, экономические санкции стран запада и Америки и др. Следствие происходящих изменений — необходимость кардинальной перестройки системы управления промышленными предприятиями с ориентацией на стратегический менеджмент. Процесс принятия стратегических решений подразумевает создание и использование модельного инструментария, основанного на экономико-математических моделях, адекватно реагирующих на неопределенность воздействий окружающей среды и позволяющих на формализованной основе принимать решения относительно стратегических ориентиров развития предприятия. Авторами предложены экономико-математические модели, позволяющие с количественной точки зрения оценивать различные альтернативы стратегического развития организаций в условиях стохастической неопределенности. В каче-

стве характеристик эффективности функционирования предприятия выбраны показатели рентабельности, носящие стратегический характер. Построенные экономико-математические модели алгоритмизированы и встроены в программный продукт PROFIT, поддерживающий принятие стратегических решений в режиме диалога менеджера с компьютером. В статье продемонстрированы результаты экспериментов, проведенных в процессе принятия стратегических решений на базе применения созданного программного продукта.

В результате исследований авторами получены следующие научные результаты.

1. Проведено исследование стратегического управления промышленным предприятием на основе методологии системного анализа, объектом которого является логическая связь целевых категорий и категорий средств достижения цели в процессе выбора стратегических ориентиров.

2. Процесс стратегического управления промышленным предприятием представлен развивающейся системой с обратными связями, посредством введения которых принятие решений рассматривается как итеративный процесс с корректировкой результатов.

3. Предложена логическая структура решения задач стратегического управления с помощью объединенной решающей процедуры, сочетающей формальные и неформальные методы на стадиях разработки и целевого использования стратегии.

4. Построены экономико-математические модели оценки принимаемых в процессе стратегического управления решений на основе прогнозирования рентабельности посредством имитации потоков прибыли и затрат.

5. Осуществлена программная реализация построенных экономико-математических моделей в программном продукте PROFIT.

Список литературы

1. Булыгина О. В., Емельянов А. А., Емельянова Н. З., Кукушкин А. А. Системный анализ в управлении / Под ред. А. А. Емельянова. М.: ФОРУМ — ИНФРА-М, 2017. — 450 с.
2. Томпсон А., Стрикленд А. Стратегический менеджмент: искусство разработки и реализации стратегии. М.: Банки и биржи, 1998. — 576 с.
3. Мескон М., Альберт М., Хедоури Ф. Основы менеджмента. М.: Дело, 1997. — 704 с.
4. Шеховцева Л. С. Стратегический менеджмент. Калининград: Изд-во РГУ им. И. Канта, 2006. — 153 с.
5. Тебекин А. В. Стратегический менеджмент: учебник для бакалавров. М.: Юрайт, 2012. — 320 с.
6. Булыгина О. В., Емельянов А. А., Емельянова Н. З. Имитационное моделирование в экономике и управлении / Под ред. А. А. Емельянова. М.: НЕОЛИТ — ИНФРА-М, 2018. — 592 с.
7. Прокимнов Н. Н. Об одном приеме имитационного моделирования // Прикладная информатика. 2010. №3. С. 78–83.
8. Власова Е. А. Акторные модели корпоративных информационных систем // Прикладная информатика. 2007. № 3.
9. Емельянов А. А., Власова Е. А., Емельянова Н. З., Прокимнов Н. Н. и др. Имитационное моделирование инвестиционных процессов // Прикладная информатика. 2012. № 2 (38). С. 93–99.
10. Стрельцова Е. Д. Системное проектирование инструментальных средств поддержки принятия финансовых решений // Известия высших учебных заведений. Северо-Кавказский регион. Серия: технические науки. 2003. №5. С. 127–128.
11. Стрельцова Е. Д., Богомякова И. В., Стрельцов В. С. Совершенствование инструментария стратегического управления межбюджетным регулированием // Вестник Удмуртского университета. Серия экономика и право. 2014. №2–3. С. 112–116.
12. Бородин А. И., Стрельцова Е. Д., Ковалёва А. В. Экономико-математическая модель оценки стратегического риска // Вестник Московского авиационного института. 2012. Т. 19. №5. С. 222–232.

References

1. Bulygina O. V., Emelyanov A. A., Emelyanova N. Z., Kukushkin A. A. *Sistemnyi analiz v upravlenii. Pod red. A. A. Emelyanova. 2-e izdanie* [System analysis in management. Edited by A. A. Emelyanov. 2nd edition]. Moscow, FORUM–INFRA-M Publ., 2017, 450 p.
2. Thompson A., Strickland A. D. *Strategicheskij menedzhment: iskusstvo razrabotki i realizacii strategii* [Strategic management: the art of developing and implementing a strategy]. Moscow, Banki i birzhi, 1998, 576 p.
3. Meskon M., Albert M., Hedouri F. *Osnovy menedzhmenta* [Basics of management]. Moscow, Delo, 1997, 704 p.
4. Shekhovtseva L. S. *Strategicheskij menedzhment* [Strategic management]. Kaliningrad, Izd-vo RGU im. I. Kanta, 2006, 153 p.
5. Tepikin A. V. *Strategicheskij menedzhment: uchebnik dlya bakalavrov* [Strategic management: textbook for bachelors]. Moscow, Yurajt, 2012, 320 p.
6. Bulygina O. V., Emelyanov A. A., Emelyanova N. Z. *Imitatsionnoe modelirovanie v ekonomike i upravlenii. Pod red. A. A. Emelyanova* [Simulation modeling in economics and management / Edited by A. A. Emelyanov]. NEOLITH — INFRA-M, 2017, 592 p.
7. Prokimnov N. N. *Ob odnom prieme imitacionnogo modelirovaniya* [About one technique of simulation modeling]. *Prikladnaya informatika — Journal of Applied Informatics*, 2010, no. 3, pp. 78–83.
8. Vlasova E. A. Actor models of corporate information systems // *Prikladnaya informatika — Journal of Applied Informatics*, 2007, no. 3.
9. Emelyanov A. A., Vlasova E. A., Emelyanova N. Z., Prokimnov N. N. *Imitatsionnoe modelirovanie investitsionnykh processov* [Simulation modeling of investment processes]. *Prikladnaya informatika — Journal of Applied Informatics*, 2012, no. 2 (38), pp. 93–99.

10. Streltsova E. D. *Sistemnoe proektirovanie instrumental'nyh sredstv podderzhki prinyatiya finansovyh reshenij* [System design of financial decision support tools]. *Izvestiya vysshih uchebnyh zavedenij. Severo-Kavkazskij region. Seriya: tekhnicheskie nauki* — News of higher educational institutions. North Caucasus region. Series: technical Sciences, 2003, no. 5, pp. 127–128.
11. Streltsova E. D., Bogomyagkova I. V., Streltsov V. S. *Sovershenstvovanie instrumentariya strategicheskogo upravleniya mezhybudzhetyym regulirovaniem* [Improvement of strategic management tools for inter-budget regulation]. *Vestnik udmurtskogo universiteta. Seriya ehkonomika i pravo* — Bulletin of the Udmurt University. Economy and law series, 2014, no. 2–3, pp. 112–116.
12. Borodin A. I., Streltsova E. D., Kovaleva A. V. *Ehkonomiko-matematicheskaya model' ochenki strategicheskogo riska* [Economic and mathematical model of strategic risk assessment]. *Vestnik Moskovskogo aviacionnogo institute* — Bulletin of the Moscow aviation Institute, 2012, vol. 21, no. 5, pp. 222–232.

E. Streltsova, South Russian State Technical University, Novocherkassk, Russia, el_strel@mail.ru

A. Borodin, Department of Financial Management of Plekhanov, Moscow, Russia, aib-2004@yandex.ru

I. Yakovenko, South Russian State Technical University, Novocherkassk, Russia, iranyak@mail.ru

Model instrumentation of strategic management of industrial enterprise

The article is devoted to solving the problem of improving the strategic management of industrial enterprises in conditions of uncertainty created by fluctuations in the external and internal environment conditioned by the current economic and political situation in the country, as well as market conditions. The purpose of the article is to develop tools that support decision making when choosing strategic guidelines for the development of enterprises using economic and mathematical methods. A study is made of the system of strategic management of an industrial enterprise with the property of development based on the application of the methodology of system analysis. The model of the process of strategic management is developed in the form of a logical structure containing a joint solving procedure for solving problems, starting with the study of the external environment and the development of the mission, and ending with the creation of an economic and mathematical toolkit and its use to evaluate decisions. Economic and mathematical tools have been built to support decision-making regarding the strategic guidelines for the development of an industrial enterprise in conditions of stochastic uncertainty. As a characteristic of the profitability of strategies chosen profitability. The toolkit is created in a class of simulation models and allows to reproduce the dynamics of a company's profit and capital expenditures by the method of statistical tests, as well as to forecast on this basis the profitability of various options for strategic orientations of the enterprise's functioning. The constructed economic-mathematical models are algorithmized and implemented in a software product that allows the strategic manager to scientifically and quantitatively substantiate the decisions made in the dialogue with the computer. The article demonstrates the work of the software product in the course of simulation experiments and their processing in assessing the selected strategic benchmark in terms of profitability. The built model toolkit is intended for use in industrial enterprises to support strategic decisions.

Keywords: strategic management, system analysis, stochastic uncertainty, profitability, imitation model.

About authors: E. Streltsova, *Dr of Economics, Associate Professor*; A. Borodin, *Dr of Economics, Associate Professor*; I. Yakovenko, *PhD in Economics, Professor*

For citation: Streltsova E., Borodin A., Yakovenko I. Model instrumentation of strategic management of industrial enterprise. *Prikladnaya informatika* — Journal of Applied Informatics, 2018, vol. 13, no. 4 (76), pp. 15–28 (in Russian).

А. В. Кычкин, канд. техн. наук, Национальный исследовательский университет «Высшая школа экономики», филиал в г. Перми, avkuchkin@hse.ru

А. И. Дерябин, канд. техн. наук, доцент, Национальный исследовательский университет «Высшая школа экономики», филиал в г. Перми, aderyabin@hse.ru

О. Л. Викентьева, канд. техн. наук, доцент, Национальный исследовательский университет «Высшая школа экономики», филиал в г. Перми, ovikenteva@hse.ru

Л. В. Шестакова, канд. техн. наук, доцент, Национальный исследовательский университет «Высшая школа экономики», филиал в г. Перми, lshestakova@hse.ru

Проектирование IoT-платформы для управления энергоресурсами интеллектуальных зданий

Авторами рассматривается задача проектирования киберфизической системы, применяемой в качестве сервиса для управления интеллектуальными зданиями с использованием технологий Интернета вещей — Internet of Things (IoT). Такие программные платформы входят в состав комплексных систем класса BEMS — Building Energy Management System и являются инструментальным средством для реализации энергоресурсосберегающих мероприятий в зданиях. Сервера и контроллеры IoT образуют инфраструктуру системы управления, в которой особую роль играют механизмы адаптации и интеллектуального анализа данных, поступающих непрерывно с большого числа узлов. В условиях отсутствия стандартов по реализации BEMS систем на базе IoT, а также возрастания требований, предъявляемых к составу и функциям аналитического обеспечения, возникает необходимость в унификации программных архитектур и обеспечении их эффективной реализации при решении практических задач.

Ключевые слова: киберфизические системы, Интернет вещей, интеллектуальное здание, интеллектуальный анализ данных, облачные технологии, беспроводная передача данных, туманные вычисления, проактивное управление.

Введение

Среди программных систем мониторинга и управления энергоресурсами EMS (Energy Management System) наиболее популярны многофункциональные информационно-аналитические платформы таких производителей, как Oracle, Microsoft, IBM, Siemens, Schneider Electric, CentraLine, АО «Группа ЭНТЕР», ГК «АйТи», НТЦ «АРГО», ЗАО «НПФ Прорыв» и других. Применение этих решений значительно повышает эффективность работы предприя-

тий за счет анализа энергетических данных и оперативного принятия решений в рамках системного энергетического менеджмента в аспекте различных видов ресурсов.

Несмотря на стремительный рост рынка EMS-систем, большинство доступных на рынке программных инструментальных средств для реализации веб-сервисов для управления энергоресурсами зданий остаются слишком сложными и дорогими, чтобы их правильно использовать. Их закрытая архитектура, невозможность адаптации аналитических компонент к конкретным сооруже-

ниям или условиям эксплуатации, высокие требования к квалификации энергоменеджеров не позволяют создавать эффективные объекты городской среды в условиях применения технологий Интернета вещей — Internet of Things (IoT). Наблюдается противоречие между уровнем развития технологий системного энергоменеджмента и формами реализации в широкой практике энергоресурсосберегающих мероприятий в зданиях. В связи с этим перспективным направлением исследований является построение открытых архитектур программных платформ для систем управления энергоресурсами зданий класса BEMS — Building Energy Management System, включающих средства обработки многомерных энергоданных с позиции энергоменеджмента.

Вопросы, связанные с использованием технологий IoT в управлении различными объектами, рассматриваются в работах [3–8] и др. IoT может помочь преодолеть проблему изолированных систем зданий [14] и поддержать создание единой среды. В этом новом контексте системы автоматизации зданий потенциально потребуют фундаментальных изменений в том, как они разрабатываются и устанавливаются.

В работах [6–7] обсуждаются различные технологии и проблемы, связанные с зеленым IoT, который дополнительно снижает потребление энергии. В работе [7] рассматривается облако датчиков, которое предполагает их использование в качестве интерфейса между физическим и цифровым миром. Облако датчиков (Sensor Cloud) [8] позволяет использовать их в ИТ-инфраструктуре путем виртуализации физического датчика на облачных вычислениях, т.е. предоставлять виртуальные датчики как услугу (SNaaS). Пользователям не нужно беспокоиться о местоположении физических датчиков, вместо этого они могут контролировать виртуальные датчики. Интеграция датчиков с облаком позволяет создать открытую, расширяемую, масштабируемую среду. В кон-

тексте SNaaS представляют интерес исследования [7], связанные с построением энергоэффективной архитектуры системы; стратегией использования энергоэффективных услуг; ситуационной и контекстной осведомленностью пользователей и приложений, например, обучение и прогнозирование; энергоэффективным управлением WSN (Wireless Sensor Networks); энергоэффективным управлением облаками.

В ряде работ [1, 2, 9–12] предлагается использовать облачные и туманные технологии при реализации программных систем для управления объектами. В работе [1] рассматривается возможность использования модели туманных вычислений как универсальной технологической платформы для проектов с применением Интернета вещей (IoT), мобильной связи 5G, искусственного интеллекта. В большей степени туманные вычисления подходят для работы с системами межмашинного взаимодействия и устройствами для сбора данных, генерируемых серийно расставленными сенсорами с частотой до долей секунд. Подобные устройства встречаются в системах безопасности и комплексах управления промышленными объектами. Эталонная архитектура туманных вычислений Openfog была опубликована в 2015 г. Особенностью туманных вычислений является локальная обработка собранных данных непосредственно на месте их генерации, что является определяющим фактором для систем реального времени. Данные размещаются на клиентских узлах, где предстоит их последующая обработка. Для сбора данных зачастую используется беспроводная связь.

В работе [2] предлагается технология конвергентных распределенных вычислений для сбора и обработки больших сенсорных данных, основанная на интеграции моделей GRID, облачных и туманных вычислений. Модель туманных вычислений используется для обработки и агрегирования сенсорных данных на уровне узлов сенсор-

ной сети и промышленных контроллеров. Для анализа агрегатов данных в центральном вычислительном кластере используются модели GRID и «облачных» вычислений.

В работе [8] рассматривается среда для разработки IoT-приложений, позволяющая использовать возможности облачных вычислений. IoT-приложение должно обеспечить:

- поддержку чтения потоков данных либо от сенсорных датчиков напрямую, либо из базы данных;
- реализацию функций обработки (анализа) данных в облачной инфраструктуре;
- визуализацию событий, обнаруженных в результате анализа данных.

В работе [9] предлагается архитектура, которая интегрирует IoT и аналитику больших данных. Данные IoT, собранные с различных датчиков и объектов, отличаются от обычных больших данных тем, что включают неоднородность, шум, разнообразие и быстрый рост. Статистические исследования [12] показывают, что к 2030 г. количество датчиков будет увеличено на 1 трлн. Это увеличение повлияет на рост больших данных. Авторы [9] выделяют сенсорный слой, содержащий сенсорные устройства и объекты, подключенные через беспроводную сеть. Беспроводная сетевая связь может быть реализована через RFID, Wi-Fi, ZigBee, Bluetooth и пр. Шлюз IoT позволяет осуществлять связь через Интернет и различные сети. Большой объем данных, полученных от датчиков, хранится в облаке. Данные доступны через приложения аналитики больших данных. Эти приложения содержат управление API и панель инструментов для обработки данных.

В работе [13] рассматривается программное обеспечение с открытым кодом — платформа VOLTTRON, предоставляющая услуги для управления различными активами в энергосистеме. Платформа разработана в Тихоокеанской северо-западной национальной лаборатории (США) и функционирует на основе программных агентов, которые могут принимать решения на основе информа-

ции, предоставленной оператором, другими агентами или онлайн-источниками в режиме реального времени. VOLTTRON-агенты делятся на три класса: агенты платформы служат поставщиками услуг на платформе; облачные агенты служат мостами между платформой и другими удаленными платформами; контрольные агенты управляют фактически аппаратными устройствами.

Типовое архитектурное решение системы управления энергоресурсами здания

В качестве типового решения для реализации системы управления энергоресурсами здания выступает архитектура смешанного типа, включающая в себя архитектурные стили: вызов с возвратом (клиент-сервер), брокер сообщений и шина (рис. 1). Доступ к системе реализуется как с помощью настольного приложения, так и из веб-браузера и мобильного приложения [15]. К системе может быть подключено большое количество устройств, для этого в составе сервера реального времени может быть использован распределенный программный брокер сообщений Apache Kafka.

Сервера мобильных клиентов, настольного приложения, веб-клиента взаимодействуют с соответствующими им клиентами с помощью REST API. Сервер визуализации метрик Grafana доступен пользователю по HTTP-запросам.

Уровень клиента отвечает за взаимодействие системы с пользователями, графическое представление данных о потреблении ресурсов в реальном времени и статистической информации. На этом уровне осуществляется сбор данных с датчиков и формирование команд управления инженерными подсистемами зданий, включая подсистемы теплоснабжения, электроснабжения, освещения, водоснабжения, вентиляции, кондиционирования и др., с помощью исполнительных механизмов [16].

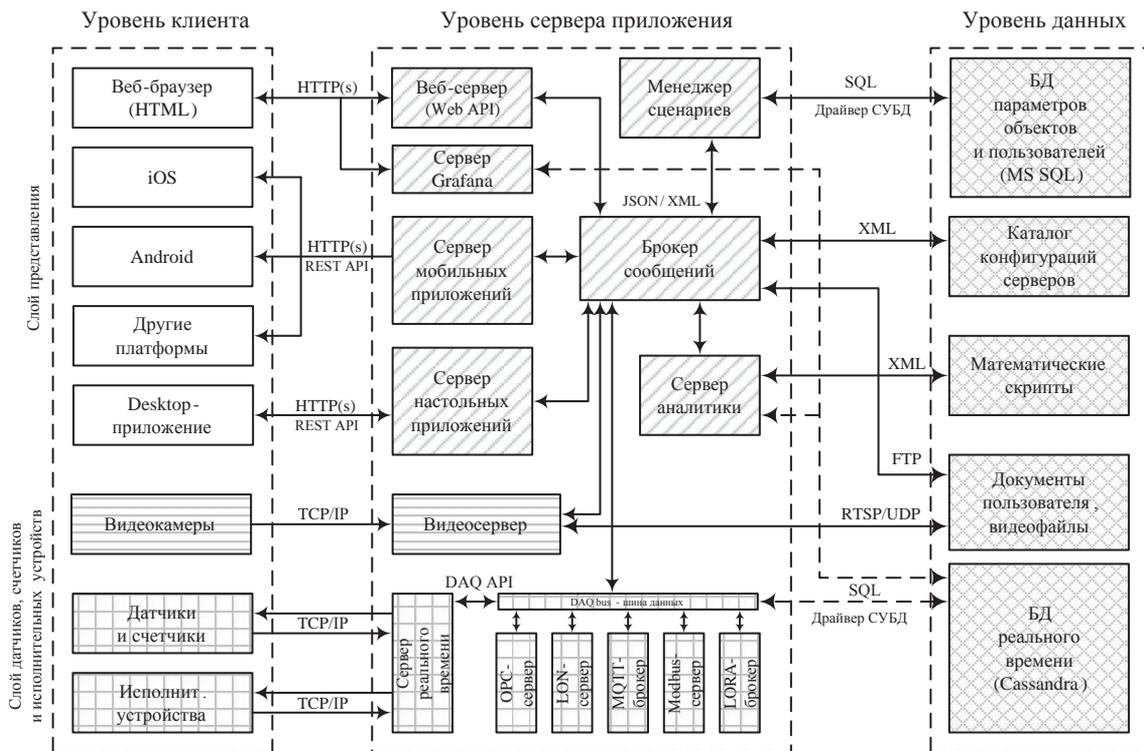


Рис. 1. Обобщенная архитектура системы управления энергоресурсами здания смешанного типа
Fig. 1. The mixed type generalized building energy management system architecture

Уровень сервера приложений необходим для реализации функций управления, сбора и обработки информации как для последующего предоставления ее клиенту или исполнительным механизмам, так и для записи в базах данных. На данном уровне выделено шесть основных серверов: сервер мобильного приложения, сервер настольного приложения, сервер веб-приложения, сервер, обеспечивающий передачу данных в сервис Grafana, аналитический сервер, а также сервер, обрабатывающий данные с видеокamer, и сервер реального времени для связи с устройствами. Чтобы синхронизировать данные между серверами, используется брокер сообщений — отдельный модуль, занимающийся приемом и передачей сообщений между компонентами и модулями системы. Для обеспечения высокой скорости работы сервера реального времени с устройствами интеллектуального зда-

ния по различным протоколам целесообразно применение шины данных, доступ к которой реализуется посредством DAQAPI. Сервер аналитики выполняет оперативную обработку информации, анализ данных об энергопотреблении здания, распознавание аварийных ситуаций, динамическое построение отчетов в различных разрезах, прогнозирование событий с помощью OLAP-технологий, [17].

Уровень данных обеспечивает подключение к базам данных, а также прием и передачу информации от уровня сервера приложений. Уровень содержит СУБД MS SQL Server, предназначенную для хранения параметров системы, конфигураций серверов, информации о пользователе, а также видеоданных; кластеризуемую базу данных Cassandra или InfluxDB, предназначенные для хранения данных с датчиков в формате временных рядов.

Организация и принцип работы менеджера сценариев

Менеджер сценариев сервера прикладной программной платформы осуществляет создание, редактирование, удаление, выбор и активацию сценариев управления объектами зданий, передает выполняемый в данный момент сценарий на сервер аналитики для моделирования его исполнения.

Зададим ограничения на подачу энерго-ресурсов j -го типа (электроэнергия, тепловая энергия, вода, пар, сжатый воздух, холод и др.) в здании для каждого момента времени на интервале T :

$$\forall t \in T : E_i^j \in \left[\underline{E}_i^j, \overline{E}_i^j \right], i = \overline{1, m}, j = \overline{1, n}$$

и общее ограничение на объем используемых энергоресурсов:

$$\sum_{i=1}^m \int_{t \in T_i} E_i^j(t) dt \leq E_0^j, \quad (1)$$

где \underline{E} , \overline{E} — нижняя и верхняя границы подаваемой энергии или ресурса j -го типа в помещении i ; E_0 — максимальный объем ресурса на временном интервале T , $T_i \subseteq T$ — временной интервал снабжения энергоресурсом i -го помещения.

Если $P = \{P_1, P_2, \dots, P_y; y = \overline{1, v}\}$ — сценарии, реализуемые алгоритмами работы программных компонентов, то оптимальный результат управления энергопотреблением здания будет достигнут при минимальных энергозатратах (1) по критерию:

$$K_j = \sum_{i=1}^m \int_{t \in T_i} E_i^j(t; P_y) dt \rightarrow \min(E_i^j), \quad (2)$$

где P — программно реализуемый сценарий управления инженерным оборудованием здания для i -го помещения. Выбор оптимального сценария может быть достигнут с помощью идентификации нелинейных зависимостей целевых значений климатических и других

показателей и объемов затрачиваемой энергии для заданных параметров, что реализуется в сервере аналитики с помощью механизмов интеллектуального анализа данных [18].

С программной точки зрения менеджер сценариев описывается классом «ScenarioManager». Атрибутом данного класса является коллекция экземпляров класса «Scenario», который описывает конкретный сценарий работы системы.

Ключевой метод — «setScenario()» по установленному набору сценариев управления объектами здания выбирает наиболее рациональный с точки зрения энергопотребления и активирует его исполнение на контроллере.

Это реализуется следующим образом. На первом этапе менеджер сценариев получает информацию о том, какой сценарий используется в данный момент на контроллере в исследуемом помещении здания.

Для заданных условий работы оборудования и помещения менеджер производит поиск в базе сценариев альтернативных вариантов, подходящих по классу управляемых переменных.

С помощью алгоритмов, реализуемых сервером аналитики, менеджер сценариев получает прогнозные значения по потреблению энергоресурсов в зависимости от анализируемого сценария и внешних условий.

Менеджер производит распознавание потенциальных аварийных ситуаций — делает оценку того, насколько прогнозируемая ситуация является аварийной в зависимости от сценария и последующих показаний датчиков.

Далее он решает задачу нахождения оптимального сценария работы оборудования на основе статистических и прогнозных данных по критерию K_j (2).

Формирует команду на активацию выбранного сценария для последующей отправки на контроллер управления с помощью следующих методов:

- «TaskSignalCreator ()» — формирование сигнала управления.

- «SignalSender ()» — отправка сигналов управления на контроллер.

Многоуровневая архитектура сервера аналитики уровня сервера приложений

Для реализации задач

$$Ts = \{Ts_1, Ts_2 \dots Ts_a; a = \overline{1, n}\}$$

обработки и анализа временных рядов и числовых параметров об энергопотреблении здания на базе сервера аналитики предложены следующие архитектурные решения:

1. Компонентная архитектура, предусматривающая раздельное представление сервера аналитики с помощью нескольких компонентов $C = \{C_1, C_2 \dots C_b; b = \overline{1, m}\}$, реализующих отдельные задачи.

2. Многоуровневая архитектура, где каждый вышестоящий уровень может обратиться только к нижестоящему.

3. Брокер сообщений на основе системы программных менеджеров

$$M = \{M_1, M_2 \dots M_u; u = \overline{1, w}\},$$

реализующих управление вызовами функций и координирующих работу подсистем сервера аналитики $S = \{S_1, S_2 \dots S_k; k = \overline{1, l}\}$.

В этом случае на верхнем первом уровне будет размещен менеджер анализа, координирующий работу подсистем S .

На втором уровне — компоненты, реализующие статистические расчеты и методы интеллектуального анализа данных по группам параметров.

На третьем уровне — менеджеры анализа временных рядов, содержащих информацию о потреблении энергии и ресурсов здания. Также этот уровень содержит компоненты подготовки данных для последующей обработки.

На четвертом уровне — обработчики временных рядов.

Обобщенную архитектуру A сервера аналитики программной платформы можно запи-

сать как операцию отображения целей и задач, а также состава менеджеров и подсистем в структурированное описание процесса обработки и анализа данных об энергопотреблении здания:

$$A : (Ts, M, S) \rightarrow P^C.$$

На рисунке 2 приведены основные компоненты и их взаимодействие друг с другом для программного блока, реализующего решение аналитических задач управления энергоресурсами зданий на уровне сервера приложений.

Представленная архитектура содержит следующие менеджеры:

- M_1 (класс «*Analysis manager*») — менеджер анализа, отвечающий за обработку системных и пользовательских вызовов функций обработки данных и служащий для координации работы подсистем S_1 – S_7 . Входными данными являются запросы на выполнение какой-либо операции, а выходными — результаты их выполнения.

- M_2 (класс «*Time series manager*») — внутренний менеджер, управляющий подготовкой одного временного ряда для последующего анализа, включая выборку из БД и сохранение, предусматривающую поиск и исправление некорректных значений временных меток, устранение несистематических выбросов или провалов в измерениях, масштабирование и нормализацию.

- M_3 (класс «*Time series collection manager*») — внутренний менеджер, реализующий подготовку для последующего анализа нескольких временных рядов, включая масштабирование и калибровку значений по первому ряду, выравнивание временных меток, если они сделаны с разным шагом, фильтрацию и сглаживание.

- M_4 (класс «*Data manager for histogram presentation*») — внутренний менеджер, отвечающий за подготовку к представлению временного ряда в формате гистограмм. Формирует новый ряд данных, иллюстрирующий распределение случайной величины.

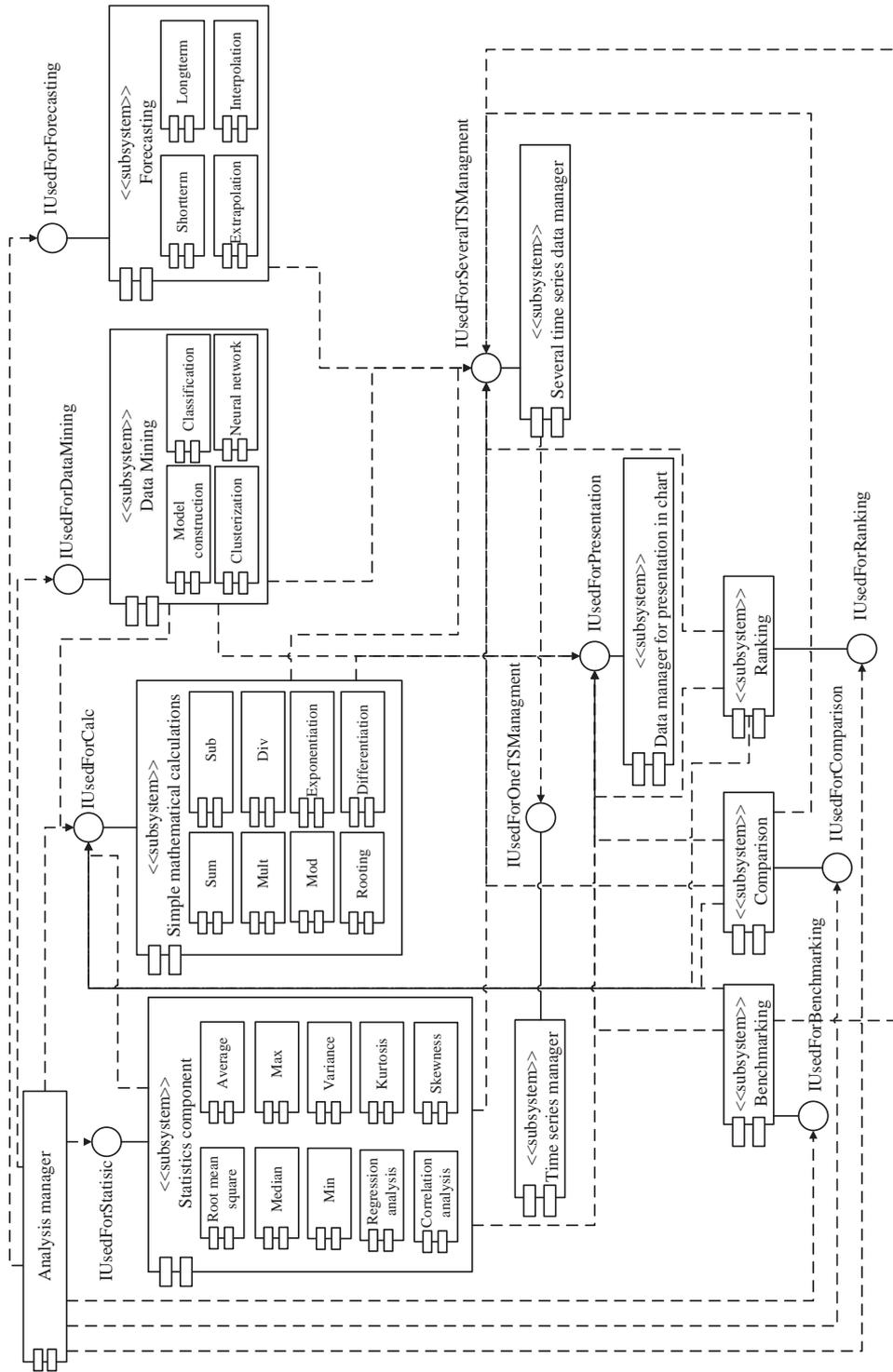


Рис. 2. Компонентная архитектура сервера аналитики программной платформы системы управления энергоресурсами здания
 Fig. 2. Component architecture of the BEMS software platform analytics server

Архитектура содержит следующие подсистемы:

- S_1 (класс «*Statistics*») — подсистема статистических расчетов, включая расчет среднего и среднеквадратичного значения энергопотребления, медианы, регрессионный анализ с построением нелинейных и линейных целевых функций энергопотребления, корреляционный анализ, используемый для оценки влияния различных факторов на энергопотребление здания.

- S_2 (класс «*BasicMath*») — подсистема реализации простейших математических операций над временными рядами данных.

- S_3 (класс «*Forecasting*») — подсистема прогнозирования, предназначенная для оценки тенденций и выявления закономерностей статистических показателей энергопотребления здания. Реализует функции краткосрочного прогнозирования (до 1 суток), долгосрочное прогнозирование (до 1 года), экстраполяции и интерполяции.

- S_4 (класс «*DataMining*») — подсистема интеллектуального анализа данных, предназначенная для выявления скрытых закономерностей и зависимостей в данных, а также идентификации модели исследуемых систем по результатам долгосрочных наблюдений на основе нейросетевых методов машинного обучения, кластеризации и классификации.

- S_5 (класс «*Comparison*») — подсистема количественной оценки отклонения нескольких временных рядов друг от друга в абсолютном и относительном эквиваленте по каждому измерению в масштабе эталонного (образцового) временного ряда.

- S_6 (класс «*Benchmarking*») — подсистема бенчмаркинга, реализующая упорядочивание временных рядов по значениям рассчитываемых интегральных показателей, например, площади, ограничиваемой профилем энергопотребления.

- S_7 (класс «*Ranking*») — подсистема ранжирования, осуществляющая поиск соотношения временного ряда к тому или иному рангу.

Реализация архитектуры программной системы управления энергоресурсами здания с помощью паттернов проектирования

Среди шаблонов проектирования выделяют порождающие, структурные и поведенческие. Несмотря на большое разнообразие паттернов, их применимость для решения поставленной задачи серьезно отличается.

Так, например, порождающий паттерн «Прототип» (*Prototype*) применим для создания экземпляров классов различных IoT-устройств, в том числе датчиков и счетчиков, обладающих схожей структурой и принципами работы в составе системы.

Порождающий паттерн «Строитель» (*Builder*) позволяет создавать сценарии в системе и регулировать доступ к функциям управления.

В качестве интерфейса высокого уровня для организации работы системы с сервером аналитики целесообразно применение структурного паттерна — «Фасад» (*Facade*). Фасад позволяет отделить компоненты сервера аналитики от системы и работать с ними независимо. В этом случае все внешние вызовы к сложно организованным аналитическим средствам сводятся к обращению к менеджеру анализов, распределяющему сообщения по подсистемам.

Поведенческий паттерн «Стратегия» (*Strategy*) позволяет выбирать один из нескольких созданных заранее сценариев управления по команде пользователя либо исходя из внешних условий, включая показания датчиков, время суток/года/день недели и т. д. С помощью данного шаблона создается гибкая архитектура классов сценариев, отвечающих за формирование нужных алгоритмов. Поскольку число компонентов, реализующих сценарии, в системе может быть велико, и каждый компонент может иметь несколько алгоритмов реали-

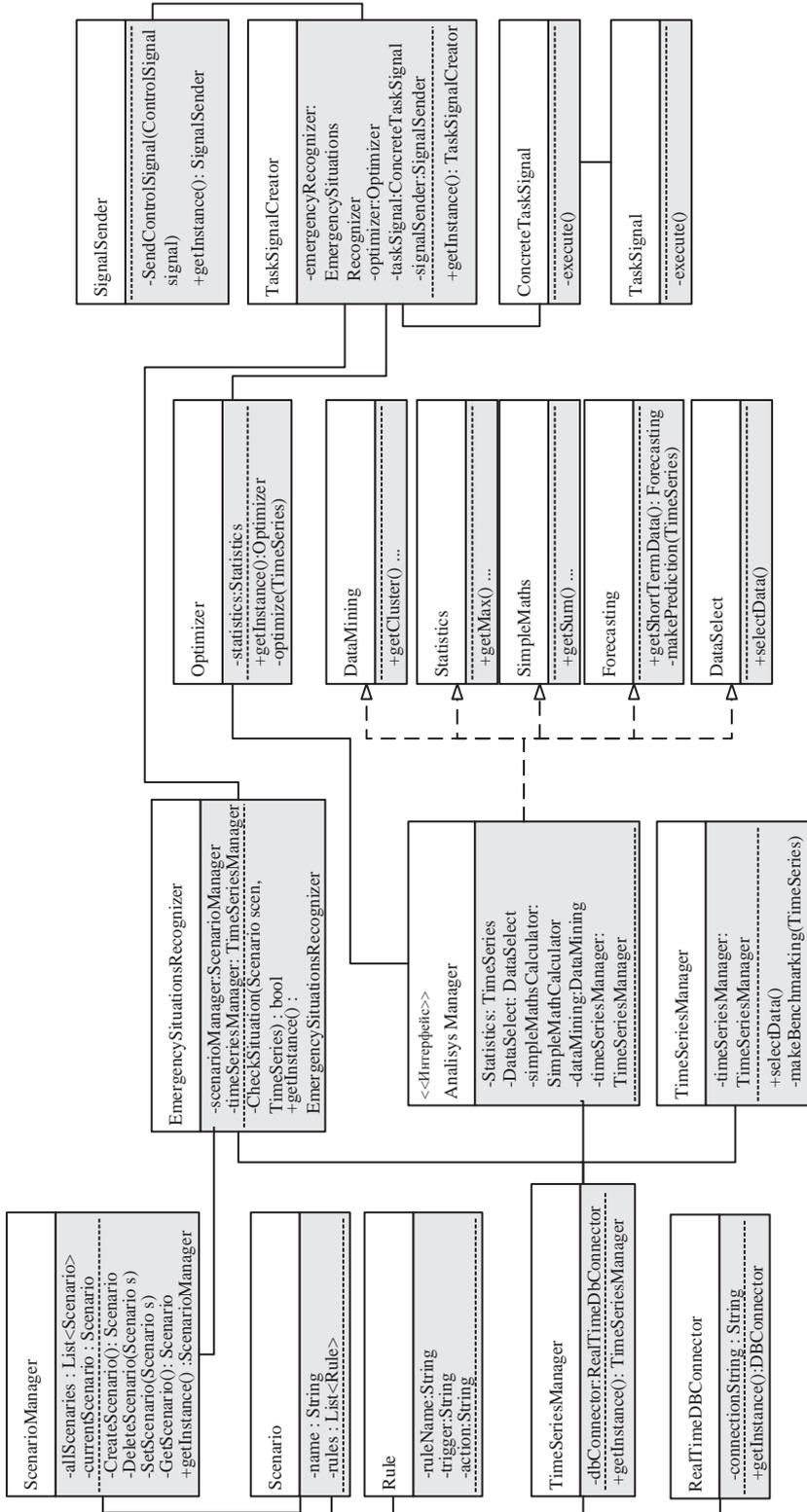


Рис. 3. Диаграмма классов сервера программы сервера аналитики с применением поведенческих паттернов

Fig. 3. Class diagram of an analytics server program using behavioral patterns

зации, стратегия обеспечивает эффективные переключения между ними во время работы системы, в том числе с учетом возможных аварий и тревог.

Поведенческий паттерн «Команда» (*Command*) позволяет эффективно организовать выполнение последовательностей запросов к серверу аналитики, таких как расчет статистических показателей за определенный период времени, обработку и анализ данных, прогнозирование и т. д. Этот паттерн позволяет отменить затянувшийся процесс обработки данных и вернуть систему в исходное состояние.

Применение поведенческого паттерна «Посредник» (*Mediator*) снижает связность большого количества компонентов, которые работают совместно, что наблюдается в менеджере анализа класс — «*Analysis manager*». Данный класс содержит ссылки на все классы, содержащие алгоритмы обработки данных и отвечает за их последовательность выполнения (рис. 3).

Поведенческий паттерн «Цепочка обязанностей» (*Chain of responsibility*) подразумевает взаимодействие программных компонентов с целью обработки потока событий, запросов или сообщений. В данном случае

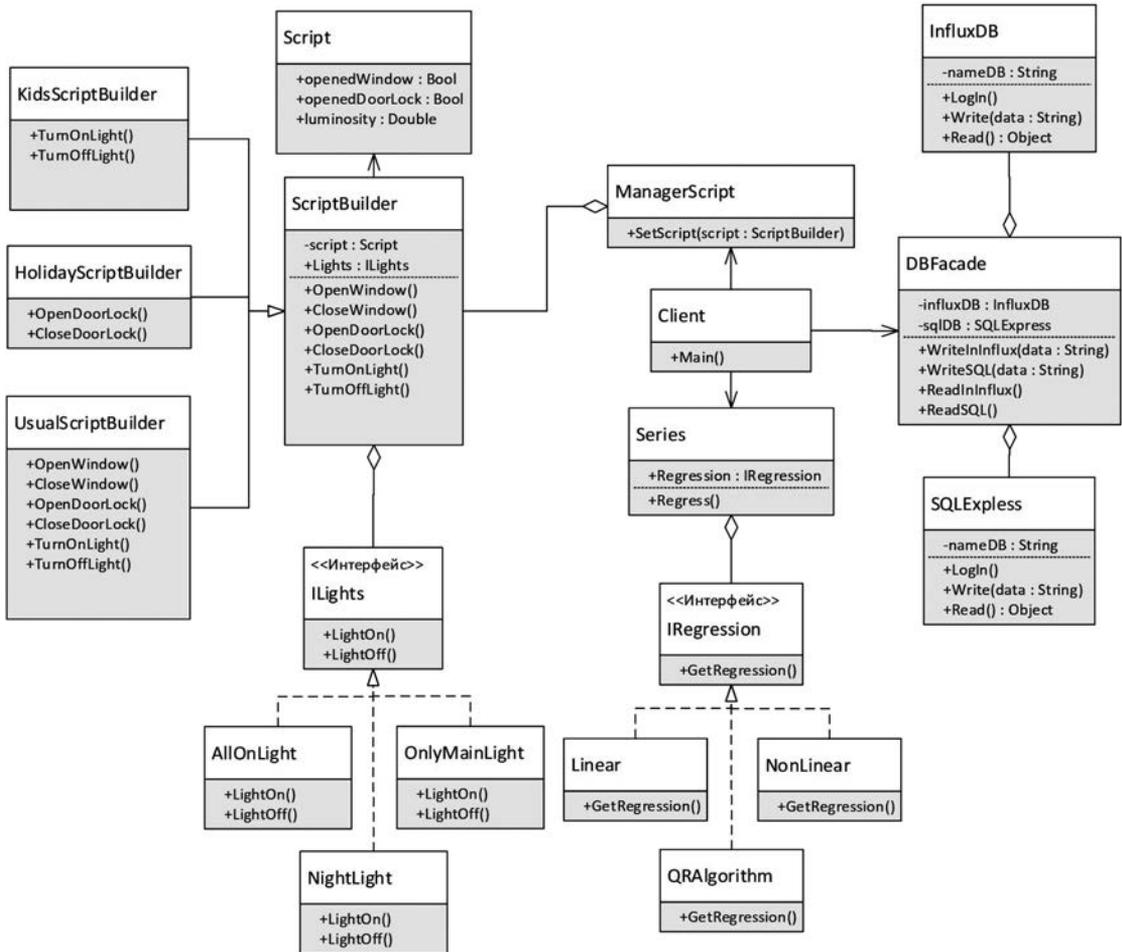


Рис. 4. Диаграмма классов составной части менеджера сценариев «ScenarioManager» с применением порождающего, структурного и поведенческого паттернов

Fig. 4. The ScenarioManager class diagram using the creational, structural and behavioral patterns

паттерн реализует последовательную обработку измерительной информации, передавая поток данных из одного компонента в другой, например, от менеджера M_2 подготовки временных рядов к подсистеме S_3 для формирования прогнозных значений. Применение «Цепочки обязанностей» позволяет менеджерам определять, куда дальше отправлять полученные в ходе обработки данные. Он наиболее эффективен в случаях, когда один запрос должен быть выполнен несколькими компонентами, порядок которых может определяться динамически.

На диаграмме классов (рис. 4) представлен состав одного из программных блоков менеджера сценариев (*ScenarioManager*), реализованного с помощью таких паттернов, как «Фасад», «Стратегия» и «Строитель».

В данном случае с помощью конкретных строителей *Kids Script Builder*, *Holiday Script Builder* и *Usual Script Builder* создаются объекты *ScriptBuilder* с определенным набором действий, определяющих сценарий управления зданием в зависимости от факторов человеческого присутствия. В роли распорядителя выступает класс менеджера *ManagerScript*, который вызывает только доступные методы конкретных строителей.

Заключение

Унификация организации и принципов работы базовых блоков проектируемой системы, включая менеджер сценариев, обеспечивает оперативную программную реализацию с использованием различных информационных технологий, в том числе облачных или туманных вычислений.

Представленные архитектурные шаблоны наглядно иллюстрируют возможности масштабирования и развития сервисов для управления интеллектуальными зданиями с использованием IoT.

Проектируемая программная платформа и комплекс реализуемых на ее основе сер-

висов являются в совокупности эффективным средством управления сложным энергетическим объектом или инженерной системой зданий с контуром интеллектуальной аналитики, построенной на основе открытой SaaS-архитектуры, способствующей развитию и тиражированию знаний об энергоэффективности зданий, включая использование возобновляемых источников энергии.

Список литературы

1. Новиков И. Консорциум OpenFog опубликовал эталонную архитектуру туманных вычислений. URL: <https://www.itweek.ru/infrastructure/article/detail.php?ID=192666>.
2. Финогеев А. А., Финогеев А. Г., Нефедова И. С. Технология конвергентной обработки данных в защищенной сети системы мониторинга // Фундаментальные исследования. 2015. № 11. С. 923–927.
3. Zanella A., Bui N., Vangelista L., Zorzi M. Internet of Things for Smart Cities // IEEE Internet of Things Journal. 2014. Vol. 1. No. 1. P. 22–32.
4. Thanh Nam Pham, Ming-Fong Tsai, Duc Binh Nguyen, Chyi-Ren Dow, Der-Jiunn Deng. A Cloud-Based Smart-Parking System Based on Internet-of-Things Technologies // IEEE Access. 2015. Vol. 3. P. 1581–1591.
5. Arshad Rushan, Saman Zahoor, Munam Ali Shah, Hongnian Yu. Green IoT: An Investigation on Energy Saving Practices for 2020 and Beyond // IEEE Access. URL: https://www.researchgate.net/publication/318797858_Green_IoT_An_Investigation_on_Energy_Saving_Practices_for_2020_and_Beyond?ev=publicSearchHeader&_sg=dWIS1SnLTai9MRQEVbgBVlfx72XEbjgG2z1yHDC_ibUM354FF-CIOPg2dj_m43cLCXu86hqLzJWYkoc.
6. Chunsheng Zhu, Victor C. M. Leung, Lei Shu, Edith Ngai. Green Internet of Things for Smart World // IEEE Access. 2015. Vol. 3. P. 2151–2162.
7. Atif Alamri, Wasai shadab ansari, Mohammad Mehedi Hassan, M. Anwar Hossain. A Survey on Sensor-Cloud: Architecture, Applications, and Approaches // International Journal of Distributed Sensor Networks. 2013. URL: <http://dx.doi.org/10.1155/2013/917923>.
8. Кычкин А. В., Артемов С. А., Белоногов А. В. Распределенная система энергомониторинга реального времени на основе технологии IoT // Датчики и системы. 2017. № 8–9 (217). С. 49–55.
9. Gubbi J., Buyya R., Marusic S., Palaniswami M. Internet of Things (IoT): A vision, architectural elements, and future directions // Future Generation Computer Systems. 2013. Vol. 29. P. 1645–1660.

10. Mohsen Marjani, Fariza Nasaruddin, Abdullah Gani Mohsen. Big IoT Data Analytics: Architecture, Opportunities, and Open Research Challenges // *IEEE Access*. Vol. 5. URL: <https://ieeexplore.ieee.org/document/7888916/>.
11. Chen M., Mao, S., Zhang, Y., Leung V. C. Big Data: Related Technologies, Challenges and Future Prospects. Springer Briefs in Computer Science, 2014. — 89 p.
12. Кычкин А. В. Программно-аппаратное обеспечение сетевого энергоучетного комплекса // Датчики и системы. 2016. №7 (205). С. 24–32.
13. Katipamula S., Haack J., Hernandez G., Akyol B., Hagerman J. VOLTTRON: An Open-Source Software Platform of the Future // *IEEE Electrification Magazine*. 2016. №4 (4). P. 15–22.
14. Stluka P., Parthasarathy G., Gabel S., Samad T. Architectures and Algorithms for Building Automation — An Industry View // *Intelligent Building Control Systems*. 2017. P. 11–43.
15. Викентьева О. Л., Дерябин А. И., Шестакова Л. В., Кычкин А. В. Синтез информационной системы управления подсистемами технического обеспечения интеллектуальных зданий // *Вестник МГСУ*. 2017. Т. 12. №10 (109). С. 1191–1201.
16. Кычкин А. В. Синтез системы удаленного энергетического мониторинга производства // *Металлург*. 2015. №9. С. 20–27.
17. Костыгов А. М., Кычкин А. В., Борковец К. А. Прогнозирование электропотребления здания на основе селективного выбора нейронной сети // *Информационно-измерительные и управляющие системы*. 2017. Т. 15. №9. С. 75–82.
18. Кычкин А. В., Микрюков Г. П. Метод обработки результатов мониторинга группы энергопотребителей // *Энергобезопасность и энергосбережение*. 2016. №6. С. 9–14.
4. Thanh Nam Pham, Ming-Fong Tsai, Duc Binh Nguyen, Chyi-Ren Dow, Der-Jiunn Deng. A Cloud-Based Smart-Parking System Based on Internet-of-Things Technologies. *IEEE Access*, 2015, vol. 3, pp. 1581–1591.
5. Arshad Rushan, Saman Zahoor, Munam Ali Shah, Hongnian Yu. Green IoT: An Investigation on Energy Saving Practices for 2020 and Beyond. *IEEE Access*, Available at: https://www.researchgate.net/publication/318797858_Green_IoT_An_Investigation_on_Energy_Saving_Practices_for_2020_and_Beyond?ev=publicSearchHeader&_sg=dWIS1SnLTai9MRQEVJvbgBVlfx72XEbjG2z1yHDC_ibUM354FF-CIOPg2dj_m43cLCxu86hqLzJWYkoc (accessed 06.06.2018).
6. Chunsheng Zhu, Victor C. M. Leung, Lei Shu, Edith Ngai. Green Internet of Things for Smart World. *IEEE Access*, 2015, vol. 3, pp. 2151–2162.
7. Atif Alamri, Wasai shadab ansari, Mohammad Mehedi Hassan, M. Anwar Hossain. A Survey on Sensor-Cloud: Architecture, Applications, and Approaches. *International Journal of Distributed Sensor Networks*, 2013. Available at: <http://dx.doi.org/10.1155/2013/917923> (accessed 06.06.2018).
8. Kichkin A. V., Artemov S. A., Belonogov A. V. *Raspredeleonnaya sistema energomonitoringa real'nogo vremeni na osnove tekhnologii IoT* [Distributed real-time energy monitoring system based on IoT technology]. *Datchiki i sistemy — Sensors and systems*, 2017, no. 8–9 (217), pp. 49–55.
9. Gubbi J., Buyya R., Marusic S., Palaniswami M. Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, 2013, vol. 29, pp. 1645–1660.
10. Mohsen Marjani, Fariza Nasaruddin, Abdullah Gani Mohsen. Big IoT Data Analytics: Architecture, Opportunities, and Open Research Challenges. *IEEE Access*. Vol. 5. Available at: <https://ieeexplore.ieee.org/document/7888916/> (accessed 06.06.2018).
11. Chen M., Mao, S., Zhang, Y., Leung V. C. Big Data: Related Technologies, Challenges and Future Prospects. Springer Briefs in Computer Science, 2014, 89 p.
12. Kichkin A. V. *Programmno-apparatnoye obespecheniye setevogo energouchetnogo kompleksa* [Software and hardware of the network energy-accounting complex]. *Datchiki i sistemy — Sensors and systems*, 2016, no. 7 (205), pp. 24–32.
13. Katipamula S., Haack J., Hernandez G., Akyol B., Hagerman J. VOLTTRON: An Open-Source Software Platform of the Future. *IEEE Electrification Magazine*, 2016, no. 4 (4), pp. 15–22.
14. Stluka P., Parthasarathy G., Gabel S., Samad T. Architectures and Algorithms for Building Automation — An Industry View. *Intelligent Building Control Systems*, 2017, pp. 11–43.

References

1. Novikov I. *Konsortsium OpenFog opublikoval etalonnyu arkhitekturu tumannykh vychisleniy* [The Open-Fog Consortium published a reference architecture for foggy computing]. Available at: <https://www.itweek.ru/infrastructure/article/detail.php?ID=192666> (accessed 06.06.2018).
2. Finogeev A. A., Finogeev A. G., Nefedova I. S. *Tekhnologiya konvergentnoy obrabotki dannykh v zashchishchennoy seti sistemy monitoring* [The technology of convergent data processing in the protected network of the monitoring system]. *Fundamental'nyye issledovaniya — Fundamental Research*, 2015, no. 11, pp. 923–927.
3. Zanella A., Bui N., Vangelista L., Zorzi M. Internet of Things for Smart Cities. *IEEE Internet of Things Journal*, 2014, vol. 1, no. 1, pp. 22–32.

15. Vikentyeva O. L., Deryabin, A. I., Shestakova L. V., Kichkin A. V. *Sintez informatsionnoy sistemy upravleniya podsystemami tekhnicheskogo obespecheniya intellektual'nykh zdaniy* [Synthesis of the information management system for the subsystems of the technical support of intellectual buildings]. *Vestnik MGSU — MSSU Bulletin*, 2017, vol. 12, no. 10 (109), pp. 1191–1201.
16. Kichkin A. V. *Sintez sistemy udalennogo energeticheskogo monitoringa proizvodstva* [Synthesis of the system of remote energy production monitoring]. *Metallurg — Metallurgist*, 2015, no. 9, pp. 20–27.
17. Kostygov A. M., Kichkin A. V., Borkovets K. A. *Prognozirovaniye elektropotrebleniya zdaniya na osnove selektivnogo vybora neyronnoy seti* [Forecasting power consumption of a building on the basis of a selective choice of a neural network]. *Informatsionno-izmeritel'nyye i upravlyayushchiye sistemy — Information-measuring and control systems*, 2017, vol. 15, no. 9, pp. 75–82.
18. Kichkin A. V., Mikryukov G. P. *Metod obrabotki rezul'tatov monitoringa gruppy energopotrebitel'ey* [Method of processing the results of monitoring the group of energy consumers]. *Energobezopasnost' i energobezopasnost' i energobezopasnost' i energobezopasnost'* — Energy Safety and Energy Economy, 2016, no. 6, pp. 9–14.

A. Kychkin, National Research University Higher School of Economics Campus in Perm, Perm, Russia, avkychkin@hse.ru

A. Deryabin, National Research University Higher School of Economics Campus in Perm, Perm, Russia, aderyabin@hse.ru

O. Vikentjeva, National Research University Higher School of Economics Campus in Perm, Perm, Russia, ovikentjeva@hse.ru

L. Shestakova, National Research University Higher School of Economics Campus in Perm, Perm, Russia, lshestakova@hse.ru

IoT-platform design for smart buildings energy management

The problem of designing a cyberphysical system used as a service for Smart buildings control using Internet technologies — Internet of Things (IoT) is considered. Such software platforms are part of the complex systems of the BEMS — Building Energy Management Systems and are an instrument for implementing energy savings in buildings. IoT servers and controllers form a management infrastructure in which the mechanisms of adaptation and data mining, which are continuously received from a large number of nodes, play a special role. In the absence of standards for the implementation of BEMS based on IoT, as well as increasing requirements for the composition and functions of analytical support, there is a need to unify software architectures and ensure their effective implementation in solving practical problems. Software component unification, including the scenario manager, provides operational software implementation using various information technologies, including cloud or foggy computing. The presented architectural templates clearly illustrate the possibilities of scaling and developing services for managing intelligent buildings using IoT. The projected open SaaS architecture software platform is an effective means for buildings energy managing with a contour of Data Mining. It can be used for replication of knowledge about the energy efficiency in buildings, including the use of renewable energy sources.

Keywords: cyberphysical systems, Internet of Things, Smart Building, data mining, cloud computing, wireless data transmission, fog computing, proactive control.

About authors:

A. Kychkin, *PhD in Technique*;

A. Deryabin, *PhD in Technique, Associate Professor*;

O. Vikentjeva, *PhD in Technique, Associate Professor*;

L. Shestakova, *PhD in Physics & Mathematics, Associate Professor*

For citation:

Kychkin A., Deryabin A., Vikentjeva O., Shestakova L. IoT-platform design for smart buildings energy management. *Prikladnaya informatika — Journal of Applied Informatics*, 2018, vol. 13, no. 4 (76), pp. 29–41 (in Russian).

*А. Н. Терехов, докт. физ.-мат. наук, профессор,
Санкт-Петербургский государственный университет, a.terekhov@spbu.ru*

*В. Г. Халин, докт. экон. наук, профессор,
Санкт-Петербургский государственный университет, v.halin@spbu.ru*

*А. В. Юрков, докт. физ.-мат. наук, профессор,
Санкт-Петербургский государственный университет, a.v.yurkov@spbu.ru*

Нужны ли кандидаты и доктора наук по программной инженерии для модернизации и технологического развития российской экономики?¹

Создание информационной и коммуникационной инфраструктуры цифровой экономики будущего требует специалистов, способных подготавливать необходимые для этого проекты, разрабатывать и внедрять программные продукты, информационные ресурсы и технологии. Однако в России до сих пор нет научной специальности под названием *программная инженерия (Software Engineering)*, по которой бы осуществлялась подготовка научных кадров высшей квалификации — аспирантов и докторантов, призванных профессионально решать названные задачи. Более того, программная инженерия отсутствует в списке научных специальностей, по которым в России создаются и работают советы по защите диссертаций на соискание ученых степеней кандидата и доктора наук (диссертационные советы) и присуждаются соответствующие ученые степени. В статье представлен анализ современного состояния российского высшего образования в сфере подготовки бакалавров и магистров по направлению программная инженерия и близким специальностям, а также сформулировано и обосновано предложение о включении новой научной специальности «программная инженерия» в Номенклатуру научных специальностей Российской Федерации. Реализация данного предложения позволит организовать подготовку необходимых специалистов и на третьем уровне высшего профессионального образования, а именно подготовку соискателей научной степени PhD — кандидатов и докторов наук — в области программной инженерии.

Ключевые слова: цифровая экономика, программная инженерия, высшее образование, научные специальности, диссертационные советы, присуждение ученых степеней (кандидата наук и доктора наук), PhD Software Engineering, SWEBOOK.

Введение

Ключевыми ресурсами современного развития общества являются знания, информация и образование. Эти ресурсы призваны создавать современные ис-

следовательские университеты, которые справедливо относят к числу основных учреждений экономики знаний XXI в. [1]. В условиях глобализации [2] информационно-коммуникационные технологии (ИКТ) стали предметом специального внимания университетов, стремящихся соответствовать вызовам цифровой экономики будущего, что отражается в их программах развития («Дорожных

¹ Статья подготовлена при поддержке гранта РФФИ 13.15.202.2016.

картах») — см., например, [3]. Так, признанный лучшим среди университетов — участников проекта «5–100»² План мероприятий Московского физико-технического университета (МФТИ) по реализации его Дорожной карты содержит специальный раздел, посвященный информатизации, а сформулированные ключевые стратегические инициативы в области образования и инновационных исследований предполагают создание и использование отвечающих вызовам времени ИКТ [4].

6 января 2015 г. Правительство Российской Федерации распоряжением № 7-р утвердило перечень из 104 направлений и специальностей подготовки высшего образования, которые соответствуют приоритетным направлениям модернизации и технологического развития российской экономики [5]. В этот перечень вошли 38 направлений подготовки бакалавров и магистров, 35 специальностей подготовки специалистов с высшим образованием и 31 специальность подготовки кадров высшей квалификации (аспирантура и докторантура), которые, в первую очередь, связаны с развитием основных технологических областей России, таких как энергетика; оборона и безопасность; авиационная и ракетно-космическая техника; электроника и компьютеры; программное обеспечение и управление информацией; медицина и биотехнологии; нанотехнологии и новые материалы; железнодорожный и автомобильный транспорт. Российские вузы, реализующие данные направления, получают специальную государственную поддержку, которая выражается в повышенном ресурсном и финансовом обеспечении учебного процесса, ускоренном развитии и модернизации их материально-техни-

ческой базы, повышении стипендий для бакалавров, магистров, специалистов, аспирантов и докторантов, обучающихся по этим специальностям и направлениям. Следует также заметить, что доля приоритетных специальностей и направлений, входящих в утвержденный Правительством России перечень, в общем списке специальностей и направлений подготовки бакалавров, магистров, специалистов, аспирантов и докторантов, по которым сегодня ведется обучение в российской высшей школе, достаточно незначительна и в целом составляет менее 15%. Так, например, из 188 направлений подготовки бакалавров и магистров, реализуемых в российской высшей школе, в приоритетный перечень входят лишь 38 направлений, что составляет около 20%. Ситуация с аспирантурой и докторантурой еще более сложная. Так, например, в Номенклатуру специальностей научных работников [6], утвержденную приказом Минобрнауки России от 10.01.2012 № 5, по которым в России происходят защиты кандидатских и докторских диссертаций, а также присуждение ученых степеней кандидата и доктора наук, входят 424 научные специальности, а в приоритетный перечень направлений подготовки аспирантов и докторантов, утвержденный Правительством России, — лишь 31 позиция, что составляет около 7%. Приведенные выше данные показывают, насколько значимыми и важными для модернизации и технологического развития российской экономики являются те направления подготовки высшего образования, которые попали в утвержденный Правительством Российской Федерации приоритетный перечень.

Анализ упомянутого выше распоряжения Правительства России от 06.01.2015 № 7-р, действующих Федеральных государственных образовательных стандартов российской высшей школы и нормативных документов ВАК России позволяет выявить серьезную системную проблему, сложившуюся в настоящее время в российской высшей школе с подготовкой специалистов в области

² Государственный проект повышения конкурентоспособности ведущих российских университетов среди ведущих мировых научно-исследовательских центров, инициированный в мае 2013 г. в соответствии с положениями Указа от 07.05.2012 № 599 Президента Российской Федерации «О мерах по реализации государственной политики в области образования и науки».

программной инженерии. С одной стороны, направление подготовки бакалавров и магистров «Программная инженерия» отнесено Правительством нашей страны к приоритетным направлениям модернизации и технологического развития российской экономики, и тем самым подчеркивается его важность и значимость для дальнейшего развития страны, а с другой стороны, оказывается, что в российской высшей школе в настоящее время вообще отсутствует и не ведется подготовка кадров высшей квалификации (аспирантов и докторантов) по данному направлению. Более того, «Программная инженерия» отсутствует и в Номенклатуре специальностей научных работников, утвержденной приказом Минобрнауки Российской Федерации от 10.01.2012 № 5, что не позволяет в принципе проводить в России защиты кандидатских и докторских диссертаций, а также присуждение ученых степеней кандидата и доктора наук специалистам по программной инженерии.

Создание информационно-коммуникационной инфраструктуры цифровой экономики будущего, по мнению авторов, определенно требует не только хороших специалистов в области программной инженерии уровня бакалавра и магистра, но и кадры высшей научной квалификации — кандидатов и докторов наук, для которых программная инженерия является предметом профессиональной деятельности: именно они призваны инициировать, разрабатывать и осуществлять управление масштабными ИКТ-проектами будущего нашей страны [7].

Подготовка специалистов по направлению «Программная инженерия» в России: ситуация на федеральном уровне

Анализируя сделанное за последние годы, необходимо отметить, что в российской высшей школе разработаны и в 2011 г. Мини-

стерством образования и науки утверждены Федеральные государственные образовательные стандарты подготовки бакалавров и магистров по направлению «Программная инженерия» (Software Engineering). Это позволило в нашей стране открыть подготовку бакалавров и магистров по указанному направлению. В настоящее время в 91 университете России осуществляется подготовка бакалавров и магистров по этому направлению, в том числе в вузах из первой десятки рейтинга качества приема:

- СПбГУ — Санкт-Петербургский государственный университет, www.spbu.ru;
- НИУ ВШЭ — Национальный исследовательский университет «Высшая школа экономики», www.hse.ru;
- МГТУ им. Н. Э. Баумана — Московский государственный технический университет имени Н. Э. Баумана, www.bmstu.ru;
- НИЯУ МИФИ — Национальный исследовательский ядерный университет «МИФИ», www.mephi.ru;
- ТПУ — Национальный исследовательский Томский политехнический университет, www.tpu.ru;
- ТГУ — Национальный исследовательский Томский государственный университет, www.tsu.ru;
- СПбГПУ — Санкт-Петербургский государственный политехнический университет; www.spbstu.ru;
- УрФУ — Уральский федеральный университет имени первого Президента России Б. Н. Ельцина, www.urfu.ru;
- СПбНИУ ИТМО — Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики, www.ifmo.ru;
- СПбГЭТУ ЛЭТИ — Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В. И. Ульянова (Ленина), www.eltech.ru;
- ЮУрГУ — Южно-Уральский государственный университет (национальный исследовательский университет), www.susu.ru.

Подготовка в области программной инженерии в СПбГУ

Санкт-Петербургский государственный университет имеет лицензию на подготовку бакалавров и магистров по направлению «Программная инженерия», начиная с 2011 г. Она ведется на кафедре системного программирования математико-механического факультета [8] по программам 09.03.04 «Программная инженерия» (для бакалавров) и 09.04.04 «Программная инженерия» (для магистров). В реализации названных образовательных программ участвуют следующие ключевые кафедры факультета: системного программирования; информационно-аналитических систем; информатики; параллельных алгоритмов; исследования операций. Программы включают профильные дисциплины [9]:

- Алгоритмы и структуры данных;
- Практическое программирование;
- Программная инженерия;
- Архитектура и проектирование программного обеспечения;
- Компьютерная графика;
- Управление программными проектами;
- Теория автоматов и формальных языков;
- Базы данных и СУБД;
- Архитектура ЭВМ;
- Анализ алгоритмов.

Список ключевых профессий выпускников:

- Программист;
- Архитектор программного обеспечения;
- Администратор баз данных;
- Специалист по информационным системам;
- Руководитель проектов в области информационных технологий;
- Специалист по тестированию в области информационных технологий;
- Руководитель разработки программного обеспечения;
- Технический писатель (специалист по технической документации в области информационных технологий);

- Системный аналитик;
- Специалист по технической поддержке информационно-коммуникационных систем;
- Системный программист.

В реализации программ принимают участие более 30 профессоров — докторов наук, среди них: А. Н. Терехов, О. Н. Граничин, Б. А. Новиков, Н. К. Косовский, Ю. К. Демьянович, В. М. Нестеров, Д. В. Кознов.

Кроме подготовки непосредственно по направлению «Программная инженерия» для бакалавров и магистров СПбГУ осуществляет обучение аспирантов, магистров и бакалавров по ряду специальностей, близких к профессиональной области программной инженерии [10]. Среди них:

- Образовательные программы подготовки аспирантов:

02.06.01 — компьютерные и информационные науки;

09.06.01 — информатика и вычислительная техника;

27.06.01 — управление в технических системах.

- Образовательные программы подготовки магистров:

01.04.02 — прикладная математика и информатика;

02.04.02 — фундаментальная информатика и информационные технологии;

02.04.03 — математическое обеспечение и администрирование информационных систем;

03.04.01 — прикладная математика и физика;

09.04.03 — прикладная информатика;

27.04.03 — системный анализ и управление;

38.04.05 — бизнес-информатика.

- Образовательные программы подготовки бакалавров:

01.03.02 — прикладная математика и информатика;

01.03.04 — прикладная математика;

02.03.01 — математика и компьютерные науки;

02.03.02 — фундаментальная информатика и информационные технологии;

02.03.03 — математическое обеспечение и администрирование информационных систем;

03.03.01 — прикладная математика и физика;

09.03.03 — прикладная информатика;

27.03.03 — системный анализ и управление;

38.03.05 — бизнес-информатика.

Реализация образовательных программ подготовки магистров и бакалавров по направлениям «Фундаментальная информатика и информационные технологии» и «Системный анализ и управление» осуществляется на факультете прикладной математики — процессах управления СПбГУ кафедрами:

- Компьютерных технологий и систем [11];
- Компьютерного моделирования и многопроцессорных систем [12];
- Моделирования электромеханических и компьютерных систем [13].

Образовательные программы подготовки магистров и бакалавров по направлению «Бизнес-информатика» реализуются на экономическом факультете СПбГУ кафедрой информационных систем в экономике [14].

Профильные курсы образовательной программы по направлению «Бизнес-информатика», коррелирующие с перечисленными в Своде знаний в области программной инженерии SWEBOOK [15]:

Базы данных;

- Объектно-ориентированный анализ и программирование;
- Бизнес-анализ информации;
- Информационные системы;
- Анализ и моделирование бизнес-процессов;
- Управление ИТ-инфраструктурой;
- Информационное право;
- Имитационное моделирование;
- Системы поддержки принятия решений;
- Теория принятия решений.

По опыту выпусков 2011–2017 гг. выпускники работают по следующим профессиям:

- Администратор баз данных;
- Администратор вычислительной сети;
- Бизнес-аналитик;
- Системный аналитик;
- Специалист по внедрению информационных систем;
- Инженер-системный программист;
- Программист информационных систем;
- Консультант по информационным системам;
- Сервис-инженер по информационным системам;
- Программист-проектировщик информационных систем;
- Экономист по планированию;
- Директор по экономике.

Это те самые профессии, которые создают цифровую экономику будущего.

Взаимодействие с ИТ-индустрией

Невозможно научить программного инженера традиционными средствами «с мелом у доски». Одно из главных отличий программной инженерии от информатики — это как раз работа с людьми. Крупные программные комплексы и продукты создаются большими коллективами в течение длительного времени. Программный инженер должен уметь планировать временные рамки проекта, бюджет, обеспечивать своих подчиненных необходимой техникой и другими ресурсами. Кроме того, ежедневно возникают какие-то проблемы, конфликты внутри коллектива, часто возникают проблемы с заказчиками — всему этому можно научить только на практике.

Много лет назад на кафедре системного программирования СПбГУ начала развиваться система студенческих проектов под руководством сотрудников ИТ-компаний, как отечественных, так и зарубежных. Каждым проектом руководят как минимум два сотрудника ИТ-компаний (один может заболеть или уехать в командировку, но учебный процесс прерыв-

вать нельзя). Они предлагают темы, интересные их компаниям, а кафедра следит, чтобы темы были достаточно наукоемкими. Весь процесс организован в соответствии с промышленными традициями: планирование, еженедельная отчетность, версионный контроль, управление конфигурациями, QA (quality assurance), взаимодействие внутри коллектива. На взгляд авторов, это win-win-win ситуация:

- студенты совершенно бесплатно получают дополнительное промышленное образование в области ИТ;
- университет, опять-таки совершенно бесплатно, получает профессиональных руководителей курсовых и дипломных работ и доступ к современным технологиям (если сотрудник ИТ-компании ведет занятие по какой-то теме, то на это занятие может прийти любой студент или даже преподаватель, интересующийся данной темой);
- наконец, предприятие, которое, разумеется, тратит деньги на зарплату своих сотрудников, занимающихся со студентами, получает сотрудников, которые без всякой дополнительной подготовки могут приступить к созидательной работе на предприятии.

Руководство студенческими проектами — отличная школа подготовки программных инженеров, способных не только программировать, но и руководить коллективами.

Проблема создания новой научной специальности

Какими бы обнадеживающими не казались перечисленные выше заделы, необходимо констатировать, что в настоящее время в России не существует научной специальности с точным названием «Программная инженерия», по которой осуществлялась бы подготовка аспирантов и докторантов — см. Номенклатуру специальностей научных работников [5].

При этом в Российской Федерации накоплен значительный опыт по подготовке высококвалифицированных кадров (аспирантов и докторантов) с присуждением ученых сте-

пеней кандидата и доктора наук по научным специальностям, которые близки к профессиональной области программной инженерии». К ним можно, например, отнести следующие научные специальности, по которым в СПбГУ действуют диссертационные советы по присуждению ученых степеней кандидата и доктора наук и ведется подготовка аспирантов, соискателей, докторантов [16]:

- Информатика, вычислительная техника и управление:
 - 05.13.01 — системный анализ, управление и обработка информации;
 - 05.13.11 — математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей;
 - 05.13.17 — теоретические основы информатики;
 - 05.13.18 — математическое моделирование, численные методы и комплексы программ.
- Физико-математические науки:
 - 01.01.07 — вычислительная математика;
 - 01.01.09 — дискретная математика и математическая кибернетика.
- Экономические науки:
 - 08.00.13 — математические и инструментальные методы экономики.

В России наиболее близкой научной специальностью к зарубежной специальности Software Engineering, по которой защищаются диссертации на ученую степень PhD, является специальность 05.13.11 — математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей (технические науки). В паспорте данной научной специальности сказано, что защищаемые по ней диссертации должны содержать результаты, обеспечивающие повышение эффективности процессов обработки данных и знаний в вычислительных машинах, комплексах и компьютерных сетях и сокращение сроков их создания, а присуждение ученых степеней кандидата и доктора наук по техническим наукам происходит за исследования, дающие существенный технический эффект

Таблица 1. Сравнение с компетенциями специалиста в области программной инженерии, требуемыми стандартом SWEBOK

Table 1. Comparison of the specialist competencies in software engineering with the requirements of the SWEBOK standard

Компетенции \ Специальность	05.13.11	08.00.13	08.00.05
Экономические			
Базовые основы экономики		+	+
Экономика жизненного цикла		+	+
Оценка риска и неопределенности		+	+
Методы экономического анализа		+	+
Вычислительные			
Методы решения проблем	+	+	
Абстракции	+	+	
Базовые основы программирования	+	+	
Знание базовых языков программирования	+		
Техники и инструменты отладки	+		
Представление и структура данных	+	+	
Алгоритмы	+	+	
Базовые основы системного анализа	+		+
Организация компьютеров	+		
Основы компиляции	+		
Основы ОС	+		
Основы БД и управления данными	+	+	
Основы сетей	+		
Параллельные и распределенные вычисления	+		
Анализ требований			
Управление человеческими ресурсами			+
Информационная безопасность и поддержка	+	+	+
Математические			
Множества, отношения, функции	+	+	
Основы логики	+	+	
Проверка гипотез	+	+	
Графы и деревья	+		
Дискретная вероятность	+	+	
Конечный автомат	+		
Теория чисел	+	+	
Алгебраические структуры	+		
Инженерные			
Эмпирические методы и экспериментальные техники			
Статистический анализ	+	+	+
Измерения	+	+	
Инженерный дизайн			
Моделирование, симуляция, прототипирование	+	+	+
Анализ причин	+		
Всего компетенций:	35 (100%)	Совпадающих:	Совпадающих:
		27 (77%)	19 (54%)
			Совпадающих:
			9 (26%)

при использовании и внедрении результатов. Сравнительный анализ компетенций специалистов, защищающих кандидатские диссертации по специальности 05.13.11, с таковыми для специалиста в области программной инженерии, требуемыми стандартом SWEBOOK [15], свидетельствует, что они совпадают на 77% (табл. 1).

Особенности подготовки специалистов в области программной инженерии

Анализ отечественной и зарубежной практики подготовки кадров в области программной инженерии показывает, что эта подготовка имеет целый ряд специфических особенностей [17]. Являясь по сути междисциплинарной, она включает десятки областей знаний от фундаментальных (математика, информатика, моделирование) до технических и технологических: проектирование программного обеспечения, процессы разработки, управление программными продуктами и др. Приобретение необходимых компетенций в последних невозможно без соучастия в коллективах разработчиков, выполняющих реальные проекты. Многолетняя практика подготовки таких специалистов кафедрой системного программирования СПбГУ убедительно свидетельствует [18] о том, что успеха удается достичь, если при организации учебного процесса должным образом учитываются потребности индустрии в ИТ-специалистах; методологической базой обучения являются российские и международные стандарты; обеспечена поддержка обучения предприятиями, которые выступают и как соучастники процесса обучения, и как заинтересованные потребители подготавливаемых кадров; наконец, технологической платформой образовательной и научной деятельности представляется современный технопарк, на базе которого в командах профессиональных разработчиков будущие специалисты соучаствуют в подготовке и выполнении актуальных ИКТ-проектов.

Основанный на принципиальном переплетении с производственным обучением в софтверных компаниях способ подготовки начинающих специалистов в области программной инженерии, внедренный А. Н. Тереховым, оказался экономически обоснованным, востребованным и взят на вооружение не только отечественными университетами, но и зарубежными ИТ-компаниями [19].

Особенности подготовки и специфика производственной деятельности программных инженеров и привели к выделению соответствующей специальности как самостоятельной и разработке федеральных образовательных стандартов подготовки бакалавров и магистров по направлению «Программная инженерия». В подготовке специалистов высшей научной квалификации в области программной инженерии фундаментальные и технические области знания существенным образом дополняются нетривиальными компетенциями в экономике производства и управлении исследованиями и разработками, а производственная составляющая должна обеспечивать приобретение опыта руководства программными проектами и научными исследованиями в области информационных и коммуникационных технологий.

Заключение

Представляется целесообразным подготовить от имени СПбГУ и других ведущих российских вузов обоснованное ходатайство в адрес Минобрнауки России о включении новой специальности «Программная инженерия» в Номенклатуру специальностей научных работников. Это позволит в России проводить защиты кандидатских и докторских диссертаций, а также присуждать ученые степени кандидата и доктора наук по специальности «Программная инженерия». Дополнение двух существующих уровней высшего профессионального образования (бакалавриата и магистратуры) отсутствующим в настоящее время третьим — аспирантурой по направлению

«Программная инженерия» станет важным системообразующим решением, создающим условия для подготовки кадров, призванных осуществлять разнообразные ИКТ-проекты цифрового будущего.

Для обоснования сформулированного предложения намечены следующие конкретные исследования:

- Детализация обоснования ходатайства в адрес Минобрнауки России о включении новой специальности «Программная инженерия» в Номенклатуру специальностей научных работников Российской Федерации в разделах: актуальность, проект паспорта новой специальности, ее уникальность и отличие в сравнении с уже существующими и др.

- Выявление рисков управления в российской высшей школе в контексте открытия новой научной специальности с целью получения ответа на вопрос: как создать благоприятную систему управления [20] в ведущих российских университетах для эффективной подготовки специалистов высшей научной квалификации — кандидатов и докторов наук в области программной инженерии.

- В рамках государственной программы «Цифровая экономика Российской Федерации» инициировать создание на базе Санкт-Петербургского государственного университета единой пилотной аспирантской и магистерской школы подготовки высококвалифицированных кадров по направлению «Программная инженерия».

Список литературы

1. Дорога к академическому совершенству: Становление исследовательских университетов / под ред. Ф. Дж. Альтбаха, Д. Салми; пер. с англ. М.: Издательство «Весь Мир», 2012. — 381 с.
2. *Kosov Y., Plotnikov V.* The Concept of a «Gateway to the Global World» in Relation to Saint Petersburg — Moscow «Axis of Development» // The Transition of Cosmopolis: the Roles of Government and Markets. The Bilateral Symposium between Shanghai Administration Institute and the North-West Institute of Management. Shanghai, Shanghai Administration Institute, 2014. Vol. 5. С. 38–47.
3. Дорожная карта НИИ ВШЭ 5/100. URL: http://www.hse.ru/data/2014/06/20/1310645347/7_Дорожная_карта_ВШЭ_5-100.pptx (Дата обращения 10.11.2017).
4. Дорожная карта МФТИ (3 этап — 2017 год). URL: <https://mipt.ru/mki/%D0%BD%D0%B5%D1%83%D0%B4%D0%B0%D0%BB%D1%8F%D1%82%D1%8C/%D0%94%D0%9A3%20%D0%9C%D0%A4%D0%A2%D0%98%2027.05.17.pdf> (Дата обращения 10.11.2017).
5. Распоряжение Правительства Российской Федерации от 6 января 2015 г. № 7-р «О Перечне специальностей и направлений подготовки высшего образования, соответствующих приоритетным направлениям модернизации и технологического развития российской экономики». URL: <http://www.garant.ru/products/ipo/prime/doc/70742752/> (Дата обращения 10.11.2017).
6. ВАК/Нормативно-справочная информация/Справочные материалы. URL: http://vak.ed.gov.ru/help_desk (дата обращения 01.08.2017).
7. *Шилова О. Н., Юрков А. В.* Об информационных технологиях в образовании в контексте обеспечения глобальной конкурентоспособности университета // Университет, Бизнес и Власть: итоги взаимодействия за 10 лет. Материалы X Международного Форума «От науки к бизнесу». «Мономакс». 2016. С. 140–142.
8. Сайт кафедры системного программирования СПбГУ. URL: <http://se.math.spbu.ru/SE> (Дата обращения 10.11.2017).
9. Компетентностно-ориентированный учебный план основной образовательной программы высшего образования бакалавриата «Программная инженерия» по направлению 09.03.04 «Программная инженерия». URL: http://math.spbu.ru/ru/mmeh/PLANS/1/15_5080_090304b_PrIng_16_12_15.pdf (Дата обращения 10.11.2017).
10. Приложение № 1.1 к лицензии СПбГУ на право оказывать образовательные услуги по реализации образовательных программ. URL: http://spbu.ru/sites/default/files/prilozhenie_no_1.1_k_licenzii.pdf (Дата обращения 10.11.2017).
11. Сайт кафедры компьютерных технологий и систем СПбГУ. URL: <http://www.apmath.spbu.ru/ru/structure/depts/kts/> (Дата обращения 10.11.2017).
12. Сайт кафедры компьютерного моделирования и многопроцессорных систем СПбГУ. URL: <http://www.apmath.spbu.ru/ru/structure/depts/kmms/> (Дата обращения 10.11.2017).
13. Сайт кафедры моделирования электромеханических и компьютерных систем СПбГУ. URL: <http://www.apmath.spbu.ru/ru/structure/depts/mems/> (Дата обращения 10.11.2017).

14. Сайт кафедры информационных систем в экономике СПбГУ. URL: <http://www.spbu-bi.ru/ru/> (Дата обращения 10.11.2017).
15. Bourque P., Fairley R. E. (eds.) Guide to the Software Engineering Body of Knowledge, Version 3.0, IEEE Computer Society, 2014. URL: www.swebok.org (Дата обращения 10.11.2017).
16. Советы по защите диссертаций на соискание ученой степени кандидата наук, на соискание ученой степени доктора наук на базе СПбГУ. URL: <https://disser.spbu.ru/> (Дата обращения 10.11.2017).
17. Терехов А. Н. Что такое программная инженерия? // Программная инженерия. 2010. № 1. С. 40–45.
18. Терехов А. Н. Вспоминая о статье «Как готовить системных программистов» // Компьютерные инструменты в образовании. 2007. № 4. С. 3–12.
19. Терехов А. Н., Терехова К. А. Экономика подготовки кадров на предприятии // Software Engineering Conference «Path Competitive Advantage». Москва, 21–25 октября, 2008. Материалы 4 международной конференции. М.: Издательство «Текама», 2008. С. 31–44. URL: http://2008.secr.ru/en/etc/secr2008_andrey_n_terekhov_economics_of_staff_training_at_enterprise.pdf (Дата обращения 10.11.2017).
20. Халин В. Г. Модернизация национальной системы высшего образования в контексте выбора управленческих решений. Научное издание. СПб.: Изд-во С.-Петербург. ун-та, 2008. — 264 с.
- 0%9A3%20%D0%9C%D0%A4%D0%A2%D0%98%2027.05.17.pdf (accessed 10.11.2017).
5. Order of the Government of the Russian Federation of January 6, 2015, no. 7-п «O Perechne spetsial'nostei i napravlenii podgotovki vysshego obrazovaniya, sootvetstvuyushchikh prioritetnym napravleniyam modernizatsii i tekhnologicheskogo razvitiya rossiiskoi ekonomiki» [About the List of specialties and directions of training of the higher education corresponding to priority directions of modernization and technological development of the Russian economy]. Available at: <http://www.garant.ru/products/ipo/prime/doc/70742752/> (in Russian, accessed 10.11.2017).
6. Higher Attestation Commission of the Russian Ministry of Education and Science (VAK) / Reference. Available at: http://vak.ed.gov.ru/help_desk (in Russian).
7. Shilova O. N., Yurkov A. V. *Ob informatsionnykh tekhnologiyakh v obrazovanii v kontekste obespecheniya global'noi konkurentosposobnosti universiteta* [On ensuring the global competitiveness of the university]. X International Forum «From science to business» «Academia — Industry — Government: The Results of a Decade Cooperation». May 11–13, 2016, Saint-Petersburg, Russia. Proceedings. Monomax Publ., 2016, pp. 140–142. Available at: http://elibrary.ru/download/elibrary_26157161_19167764.pdf (accessed 10.11.2017) (in Russian).
8. Department of Software Engineering of the Saint Petersburg State University, <http://se.math.spbu.ru/SE>.
9. *Kompetentnostno-orientirovannyi uchebnyi plan osnovnoi obrazovatel'noi programmy vysshego obrazovaniya bakalavriata «Programmnyaya inzheneriya» po napravleniyu 09.03.04 «Programmnyaya inzheneriya»* [Software Engineering: Competent-oriented curriculum of the basic educational program of higher education of the baccalaureate in the direction 09.03.04 — Software Engineering]. Available at: http://math.spbu.ru/ru/mmeh/PLANS/1/15_5080_090304bPrIng_16_12_15.pdf (accessed 10.11.2017) (in Russian).
10. *Prilozhenie № 1.1 k litsenzii SPbGU na pravo okazyvat' obrazovatel'nye uslugi po realizatsii obrazovatel'nykh programm* [Appendix No 1.1 to the License of Saint Petersburg State University for the right to provide educational services for the implementation of educational programs]. Available at: http://spbu.ru/sites/default/files/prilozhenie_no_1.1_k_licenzii.pdf (accessed 10.11.2017) (in Russian).
11. Department of Computer Applications and Systems of the Saint Petersburg State University. Available at: <http://www.apmath.spbu.ru/ru/structure/depts/kts/>
12. Department of Computer Modelling and Multiprocessor Systems of the Saint Petersburg State University. Available at: <http://www.apmath.spbu.ru/ru/structure/depts/kmms/>

References

1. Altbach P. G., Jamil Salmi (eds.). The road to academic excellence: The making of world-class research universities. World Bank Publications, 2011. Available at: <http://documents.worldbank.org/curated/en/584591468324279676/pdf/646680PUB0Russ0371975B00PUBLIC00PDF.pdf> (in Russian, accessed 10.11.2017).
2. Kosov Y., Plotnikov, V. The Concept of a «Gateway to the Global World» in Relation to Saint Petersburg — Moscow «Axis of Development». *The Transition of Cosmopolis: the Roles of Government and Markets. The Bilateral Symposium between Shanghai Administration Institute and the North-West Institute of Management*. Shanghai, Shanghai Administration Institute, 2014, vol. 5, pp. 38–47.
3. Roadmap for Implementing the Global Competitiveness Programme of the National Research University «Higher School of Economics» in 2013–2020 (SUMMARY). Available at: <https://strategy.hse.ru/en/summary> (accessed 10.11.2017).
4. Roadmap of the MIPT (2017). Available at: <https://mipt.ru/mki/%D0%BD%D0%B5%20%D1%83%D0%B4%D0%B0%D0%BB%D1%8F%D1%82%D1%8C%D0%94%D0%9A3%20%D0%9C%D0%A4%D0%A2%D0%98%2027.05.17.pdf> (accessed 10.11.2017).

13. Department of Electromechanical and Computer Systems Modelling of the Saint Petersburg State University. Available at: <http://www.apmath.spbu.ru/ru/structure/depts/mems/>.
14. Department of Information Systems in Economics of the Saint Petersburg State University. Available at: <http://www.spbu-bi.ru/ru/>.
15. Bourque P., Fairley R. E. (eds.). *Guide to the Software Engineering Body of Knowledge, Version 3.0*, IEEE Computer Society, 2014. Available at: www.swebok.org.
16. *Sovety po zashchite dissertatsii na soiskanie uchenoi stepeni kandidata nauk, na soiskanie uchenoi stepeni doktora nauk na baze SPbGU* [Dissertation Councils for the defences of degrees of Candidate of sciences and Doctor of Sciences at the Saint Petersburg State University]. Available at: <https://disser.spbu.ru/> (accessed 01.08.2017) (in Russian).
17. Terekhov A. N. *Chto takoe programmnyaya inzheneriya?* [What is the program engineering?]. *Programmnyaya Ingeneria* — J. Software Engineering, 2010, no. 1, pp. 40–45 (in Russian).
18. Terekhov A. N. *Vspominaya o stat'e «Kak gotovit' sistemykh programmistov»* [Recalling the article 'How to prepare system programmers']. *Komp'yuternye instrumenty v obrazovanii* — Computer Tools in Education, 2007, no. 4, pp. 3–12 (in Russian).
19. Terekhov A., Terekhova K. The economics of hiring and staff retention for an IT company in Russia. URL: http://www.math.spbu.ru/user/ant/all_articles/086_Terekhov_Terekhova_HR_economics.pdf (accessed 17.11.2017).
20. Khalin V. G. *Modernization of the national system of higher education in the context of the choice of management decisions* [Modernization of National System of Higher Education in the Context of Making Administrative Decisions.]. Saint-Petersburg University Press, 2008, 264 p.

A. Terekhov, Saint Petersburg State University, Saint Petersburg, Russia, a.terekhov@spbu.ru

V. Khalin, Saint Petersburg State University, Saint Petersburg, Russia, v.khalin@spbu.ru

A. Yurkov, Saint Petersburg State University, Saint Petersburg, Russia, a.v.yurkov@spbu.ru

Do candidates and doctors of science in software engineering need to modernize and technological development of the Russian economy?¹

Creation of an information and communication infrastructure of the digital economy requires specialists capable of initiating, developing, and implementing projects the necessary software products, information resources and technologies. In Russia there is still no scientific specialty exactly named Software Engineering for the training of post-graduate students and doctoral candidates. Moreover, such a specialty is missing in the list of scientific specialties on which the degrees of the candidate and the doctor of sciences are defended. The article provides an analysis of the status of Russian higher education in the field of bachelor's and master's training in Software Engineering and related specialties and formulates a proposal to include Software Engineering in the Nomenclature of Specialties of Scientists of the Russian Federation. To solve this problem, it is necessary to organize training for specialists in this area at the third level of higher professional education, namely the training of PhDs — candidates and doctors of sciences — in the field of Software Engineering. The research is partially supported by the grant RFBR 13.15.202.2016.

Keywords: digital economy, Software Engineering, higher education, scientific specialties, PhD training.

About authors:

A. Terekhov, Dr. of Physics & Mathematics, Professor;

V. Khalin, Dr. of Economics, Professor;

A. Yurkov, Dr. of Physics & Mathematics, Professor

For citation:

Terekhov A., Khalin V., Yurkov A. Do candidates and doctors of science in software engineering need to modernize and technological development of the Russian economy? *Prikladnaya Informatika* — Journal of Applied Informatics, 2018, vol. 13, no. 4 (76), pp. 42–52 (in Russian).

¹ This work was supported by grants RFBR 13.15.202.2016.

И. И. Труб, канд. техн. наук, доцент, вед. инженер-программист,
ООО «Исследовательский центр Samsung», Москва, itrub@yandex.ru

Имитационное моделирование иерархических bitmap-индексов

В статье рассмотрено построение имитационной модели для иерархических bitmap-индексов на языке С. Индексы строятся по свойству, являющемуся значением времени занесения записи в базу данных. Модель позволяет проектировщику выбрать наиболее эффективную иерархию индексов по критерию минимизации логических операций при выполнении поисковых запросов. Отдельное внимание уделено верификации модели путем сравнения с частными случаями известных аналитических решений.

Ключевые слова: иерархические bitmap-индексы, дизъюнкция, исключающее ИЛИ, случайный поток событий, плотность распределения вероятностей, имитационное моделирование, доверительный интервал.

Введение

Bitmap-индексы являются широко распространенным встроенным инструментом СУБД для повышения производительности обработки поисковых запросов. Они используются в таких известных СУБД, как Oracle и Cache [6, 12, 20]. Напомним, что использование bitmap-индексов для некоторого столбца/свойства заключается в создании отдельных битовых строк для каждого возможного значения свойства, где каждому биту соответствует строка/объект (далее для определенности будем пользоваться термином *запись*) таблицы базы данных с этим значением. Если бит равен 1, то это означает, что запись, соответствующая позиции бита, содержит индексированное значение для данного свойства. Описание реализации и использования битовых индексов в Cache можно найти в [7], в Oracle — в [15]. Типичным примером является генерация отчетов по значению свойства, являющегося временем занесения записи в базу данных. Например, в CRM-системе (Customer Relationship Management) имеется некоторая сущность, скажем, «Входящие

звонки», экземпляры которой хранятся в отдельной структуре (для Cache таковой является глобал, но можно представлять ее и таблицей — общая постановка задачи никак не будет привязана к особенностям конкретной СУБД). Одним из атрибутов сущности является время поступления звонка с точностью до секунды, причем упорядоченности записей таблицы по этому атрибуту разработчик ожидать не вправе, т. к. информация о звонке заносится в БД не в момент поступления звонка, а позже. Интенсивность поступления звонков в масштабах всей компании, эксплуатирующей приложение, достаточно высока, т. е. объем таблицы велик. Типовым сервисом приложения, востребованным заказчиком, является генерация отчетов по входящим звонкам, причем основным фильтром при задании параметров отчета служит временной диапазон, например, «Вывести информацию обо всех звонках, поступивших с такого-то времени по такое». Для этого примера использование bitmap-индексов означает, что для каждого уникального значения времени, для которого зафиксирован хотя бы один звонок, создается битовая строка, возможно (в случае большого

количества записей), состоящая из нескольких подстрок (chunks). При запросе на отчет берутся все имеющиеся индексы для значений, входящих в заданный пользователем диапазон, и путем их дизъюнкции вычисляется результирующая битовая строка, единицы которой соответствуют ID записей, удовлетворяющих запросу и выводимых в итоговый отчет.

Однако практика показывает, что даже применение bitmap-индексов для промышленных объемов данных не дает приемлемой для потребителя производительности — отчеты генерируются слишком медленно. Ключевым параметром является в данном случае количество операций дизъюнкции между битовыми строками, т. е. количество индексов, попадающих под условия фильтра запроса. При большой величине временного диапазона в фильтре и высокой интенсивности занесения в БД новых записей оно оказывается очень велико, что и приводит к ощутимой задержке. В этой связи актуальны расширения и улучшения технологии, одной из которых является концепция иерархических bitmap-индексов. Оценить эффективность применения этой концепции можно различными способами, например, с помощью имитационного моделирования. Построению и анализу результатов такой имитационной модели и посвящена настоящая статья. Она построена следующим образом. В первой части описаны иерархические bitmap-индексы применительно к временному свойству, дан краткий обзор соответствующей литературы, введены используемые обозначения и поставлена задача построения имитационной модели. Во второй части представлена сама модель и алгоритмы, лежащие в ее основе. В третьей части описаны особенности анализа выходных данных модели. В четвертой части приведены примеры реализации модели для различных вариантов входных данных. В заключении указаны направления дальнейшего расширения и развития модели.

Иерархические bitmap-индексы

Предположим, что занесение новых записей в конкретную таблицу базы данных образует поток событий в непрерывном времени, описываемый функцией распределения случайной величины X_1 интервала между двумя последовательными событиями $F_1(t)$, а длина временного интервала, задаваемого в запросе на выборку записей, является случайной величиной X_2 с распределением $F_2(t)$. Шкала времени разбита на равные интервалы единичной длины. Начало интервала для X_2 всегда совмещено с началом одного из единичных интервалов. Назовем полуоткрытый единичный интервал $[i; i + 1)$ *помеченным*, если он содержит хотя бы одно событие из потока X_1 . Дискретную случайную величину Y определим как количество помеченных интервалов, полностью покрываемых на временной шкале случайным интервалом X_2 . Тогда Y как раз и является количеством индексов (дизъюнкций), используемых для вычисления результата запроса. Чем меньше эта величина, тем меньше времени занимает обработка запроса. Аналитическая модель вычисления распределения для Y была построена в [8] и [9]. При этом в [8] в предположении экспоненциальности $F_1(t)$ было получено точное решение, а в [9] для произвольной $F_1(t)$ — численное (вид $F_2(t)$ не оказывает принципиального влияния на сложность аналитической модели).

Идея иерархических bitmap-индексов применительно ко времени заключается в следующем: количество индексов, необходимых для получения результата запроса, можно уменьшить, если наряду с *мелким* индексом, соответствующим основной единице времени (секунде), СУБД будет поддерживать еще некоторое количество *крупных* индексов, соответствующих интервалам времени, кратным основному. Например, если наряду с *посекундными* индексами мы располагаем также и *поминутными*, то для обслуживания запроса длиной в минуту (если его левая гра-

ница совпадает с границей минуты) нам понадобится максимум одна индексная битовая строка, а не 60. Идея проста и понятна, но ее использование требует точных определений и обоснованной методики расчета показателей эффективности. В частности, необходимо с той или иной точностью дать ответ на следующие вопросы:

- как вычислить распределение случайной величины Y для заданной иерархии крупных индексов;
- как выбрать наиболее эффективную иерархию индексов и какие критерии эффективности использовать.

Понятие иерархических bitmap-индексов (hierarchical bitmap indexes) было впервые введено в [21] для индексирования свойств, являющихся множествами значений некоторых простых типов (set-valued attribute). Тот смысл, в котором термин используется в настоящей работе, известен в англоязычной литературе также как *binning* — объединение отдельных значений данных в группы по некоторому правилу и создание укрупненных индексов по этим группам. Обзор результатов по *binning* с подробным объяснением основных концепций на примерах содержится в [23]. Отмечено, что данная технология хорошо подходит для больших массивов результатов различных научных и технических измерений, таких как давление, температура и пр. Близко к рассматриваемым задачам примыкает сравнительно недавняя работа американских специалистов [22]. В ней рассматривается задача выбора между OR- и XOR-операциями, которые названы *inclusive query plan* и *exclusive query plan*, соответственно; сформулирована проблема оптимального выбора плана запроса (*cut selection problem*), предложены некоторые эвристические алгоритмы ее решения, применение которых проиллюстрировано на синтетических наборах данных. Но в этой статье рассмотрена оптимизация запроса на конкретной иерархии индексов, а не оптимальный выбор самой иерархии. В [10], являющейся первой работой

подобного рода, задача оценки эффективности использования иерархических bitmap-индексов решена с помощью формально обоснованной математической модели методами теории вероятностей. В этой работе, а также в [14], описанная выше постановка задачи была продолжена на иерархию индексов и построена модель, в которой распределение и среднее для случайной величины Y доводятся до вычисленных по выведенным формулам значений. Следуя [10], приведем эту постановку, необходимую для дальнейшего понимания работы.

Определим *иерархию индексов* как упорядоченное множество $K = \{k_i\}$, $i = 1, \dots, N$, где каждый элемент множества является натуральным числом больше единицы. Смысл этого понятия заключается в следующем: наряду с мелким индексом поддерживаются N крупных индексов, кратность первого по отношению к мелкому равна k_1 , кратность второго по отношению к первому равна k_2 и т. д. N назовем *размером иерархии*. Например, $K = \{60, 60\}$ соответствует совокупности трех индексов размерами одна секунда, одна минута и один час.

Введем также множество $L = \{l_i\}$, $i = 0, \dots, N$, где $l_0 = 1$, $l_i = \prod_{j=1}^i k_j$, $i > 0$. Из определения очевидно, что i -й элемент этого множества есть размер i -го крупного индекса относительно основной единицы измерения — мелкого индекса. В [10] и [14] рассматривались только такие интервалы времени, левая граница которых совпадает с границей самого крупного индекса, имеющего в иерархии K порядковый номер, равный размеру иерархии N . В примере $\{60, 60\}$ это начало любого часа. Это предположение было названо *элементарным граничным условием (ЭГУ)*, и его выполнение позволяет любой интервал времени длины T представить единственным образом в виде упорядоченного множества $B = \{b_i\}$, $i = 0, 1, \dots, N$, где i -й элемент множества есть количество единиц i -го индекса иерархии, а $T = \sum_{j=0}^N b_j l_j$. Такое представление

позволило построить вероятностную модель иерархии индексов и рассчитать ее.

Итак, любому запросу на выборку данных по времени при условии выполнения ЭГУ соответствует некоторое множество B разложения длины запроса по иерархии индексов. Очевидно, что значение b_N может быть произвольным неотрицательным целым числом, а для всех $0 \leq i \leq N - 1$ должно выполняться соотношение $0 \leq b_i \leq k_{i+1}$, т. к. количество индексов i -го уровня не может быть больше соответствующей ему кратности следующего уровня. Однако такая интерпретация модели является не вполне экономичной. Поясним это примером. Предположим, что требуется построить выборку, где начало интервала совпадает с началом минуты, а длина интервала равна 45 сек. Если вычислять результат запроса исключительно с помощью дизъюнкций (*OR-operation* или *inclusive query plan*), потребуется дизъюнкция 45 индексных битовых строк — от нулевой до 44-й сек. Но их количество можно сократить до 16, если последовательно осуществить операцию «исключающее ИЛИ» (*XOR-operation* или *exclusive query plan*) полного минутного индекса с индексами от 45-й до 59-й сек. Иными словами, вместо «складывания» событий, произошедших от 0 до 44-й секунды, из всех событий, произошедших в течение минуты, следует «вычесть» те события, которые в этот интервал не попали. Формализация указанного соображения приводит к следующей связи между множествами B и K :

$$0 \leq b_i \leq R_i, \quad i = 0, \dots, N - 1,$$

$$R_i = \begin{cases} \frac{k_{i+1}}{2}, & k_{i+1} - \text{четное} \\ \frac{k_{i+1} + 1}{2}, & k_{i+1} - \text{нечетное} \end{cases}.$$

Интуитивно очевиден тот факт, что, с одной стороны, чем больше размер иерархии N , тем меньше будет значение $Y_{\text{ср}}$, т. е. тем меньше в среднем потребуется использовать битовых индексных строк для обслу-

живания запроса. Но, с другой стороны, поддержание лишнего крупного индекса требует определенных накладных расходов от СУБД, т. к. этот индекс нужно обновлять при каждой операции добавления и удаления записей, кроме того, требуется место, чтобы его хранить. Поэтому вполне естественно, что на размер иерархии должны накладываться некоторые ограничения. Это позволило поставить в [10] содержательные проблемы условной оптимизации по выбору наилучшей в некотором смысле иерархии K .

Но для решения любой задачи оптимизации необходим движок — метод вычисления значения минимизируемой функции в заданной точке. В нашей задаче это вычисление значения $Y_{\text{ср}}$ для заданной иерархии K , что представляет собой самостоятельную и весьма непростую задачу. Как уже было сказано, эта задача была решена аналитически в [10] при наличии двух упрощающих предположений: все запросы удовлетворяют ЭГУ и распределение $F_1(t)$ является экспоненциальным. Однако на практике ожидание выполнения этих ограничений не вполне оправдано. Записи могут искаться в интервале, не начинающемся с границы часа, а входной поток записей в базу данных не будет пуассоновским. Ввиду того, что отбрасывание любого из них сразу же разрушает построенную в [10] модель, решение задачи в общем виде возможно только средствами имитационного моделирования, т. к. имитационная модель позволяет реализовать любую внутреннюю логику системы. Таким образом, точка, в которой вычисляется значение минимизируемой функции — это множество K , а в качестве процедуры вычисления выступает прогон имитационной модели.

Имитационная модель

Рассмотренная задача имеет особенности, отличающие ее от традиционных задач на построение имитационной модели. В первую очередь моделируемая система не является

системой массового обслуживания (СМО) в классическом смысле этого термина. Что имеется в виду? В СМО необходимо разыгрывать две последовательности случайных величин — интервалы между поступлениями заявок и длительности обслуживания заявок. И обе эти последовательности можно разыгрывать в динамике, т. е. в момент поступления в систему очередной заявки генерируется по очередному члену каждой последовательности. В нашем же случае логика несколько иная. Для моделирования запроса на выборку данных необходимо, чтобы значения временного атрибута были уже назначены всем записям базы данных. Т.е. сначала в полном объеме должна быть сформирована реализация распределения $F_1(t)$ (будем для краткости называть ее *поле*), и только после этого реализуется распределение $F_2(t)$. Очередное значение искомой случайной величины Y вычисляется для каждого элемента этой реализации, после чего реализуется следующий элемент. Иными словами, имея поле, иерархию индексов и интервал запроса, мы по определенному алгоритму, являющемуся собственно алгоритмическим движком моделирования, вычисляем, сколько индексов понадобится для обслуживания этого запроса. Указанная особенность сводит имитационный эксперимент к вычислению Y_{cp} , как среднего значения последовательности $Y_{cp,1}, Y_{cp,2}, \dots$, где $Y_{cp,i}$ — среднее значение для реализации Y , вычисленной на одном и том же поле. Назовем эту последовательность *глобальной*, в то время как последовательность реализаций Y на одном и том же поле для каждого i — *локальной*.

Оригинальность моделируемой системы делает малопригодным использование каких-либо стандартных универсальных пакетов, таких как AnyLogic [2, 5], т. к. применение к данной задаче либо невозможно, либо требует самостоятельного написания большого количества кода на языке JAVA. Поэтому для создания адекватной модели был использован подход, близкий к тому, на котором осно-

ваны работы [11] и [13], — собственная реализация на универсальном языке программирования C/C++, позволяющая полностью учесть все алгоритмические нюансы задачи. По существу, процесс создания модели полностью, вплоть до языка программирования, соответствует классическому плану, изложенному в [1], а именно:

- 1) выбор объекта, системы процессов, которые необходимо смоделировать;
- 2) формулирование вопросов, на которые должна отвечать модель;
- 3) построение графа модели;
- 4) формулирование требований к внутренней логике;
- 5) кодирование графа модели на языке программирования C;
- 6) кодирование внутренней логики на C;
- 7) компиляция и сборка;
- 8) отладка модели.

Перейдем к рассмотрению архитектуры модели (рис. 1). На рисунке 1 отображены функции, из которых состоит программный код, и связи между ними, где наличие направленной связи обозначает факт вызова из одной функции другой функции. Опишем семантику этих функций.

- *cycleGlo* — цикл, на каждой итерации которого вычисляется очередное значение $Y_{cp,i}$;
- *batchingGlo, estimationGlo* — функции анализа выходных данных, используемые для определения момента прекращения моделирования, т. е. момента, когда доверительный интервал для Y_{cp} с требуемым уровнем достоверности уже получен. Подробнее они рассмотрены в третьей части работы;
- *spectralVarAnalysis* — функция, лежащая в основе анализа выходных данных;
- *leastSquares* — реализует интерполяцию методом наименьших квадратов;
- *getSingleDataGlo* — реализует вычисление одного значения $Y_{cp,i}$;
- *getGround* — вычисляет новое поле, т. е. последовательность целочисленных случайных величин, интервалы между которыми описываются функцией $F_1(t)$. Генерация поля

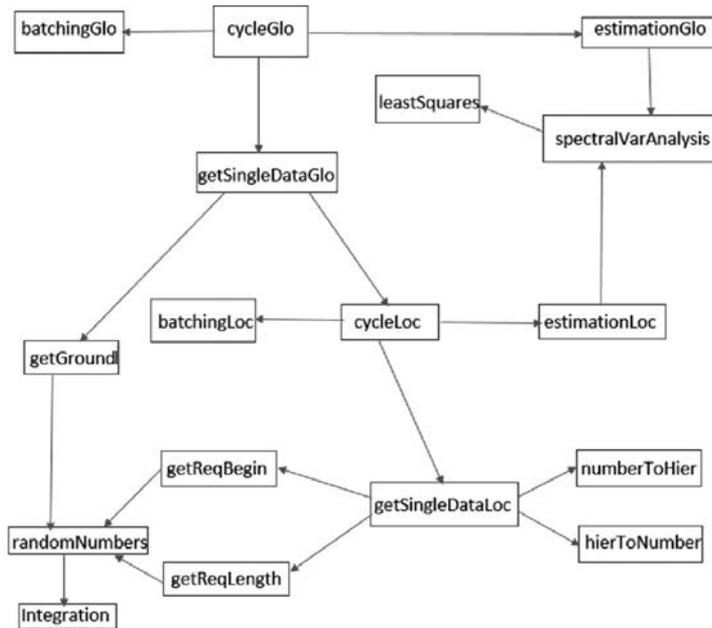


Рис. 1. Программная структура имитационной модели
 Fig. 1. Structure of simulation software

продолжается до достижения значения 86 400 (количество секунд в сутках). В зависимости от условий и целей моделирования может быть выбрано и другое значение;

- *batchingLoc* и *estimationLoc* — анализ выходных данных в локальном цикле. Реализация та же, что и для соответствующих глобальных функций. Выделены в отдельные блоки, чтобы еще раз подчеркнуть двухуровневую структуру модели;

- *getSingleDataLoc* — собственно, движок моделирования. Генерирует очередной запрос на выборку данных и на основе моделируемой иерархии вычисляет количество индексов для его обслуживания;

- *numberToHier*, *hierToNumber* — вспомогательные функции, конвертирующие абсолютное значение времени в секундах в его иерархическое представление $B = \{b_i\}$, $i = 0, 1, \dots, N$ и наоборот;

- *getReqBegin* — генерирует левую границу интервала запроса при отсутствии выполнения ЭГУ. Алгоритм подробнее рассмотрен далее;

- *getReqLength* — генерирует длину запроса в соответствии с распределением $F_2(t)$;

- *randomNumbers* — генератор случайных чисел. Реализован с помощью метода обратной функции. Многочисленные примеры реализации на языке C приведены в [11];

- *Integration* — реализация численного интегрирования, необходимого генератору случайных чисел для вычисления некоторых функций распределения.

Опишем обобщенную математическую модель, используемую для генерации начальной точки интервала запроса при снятии ЭГУ. В примере иерархии секунда-минута-час это означает, что таковой может быть не только момент времени, кратный часу, а любой момент времени. Зададим два множества: $NB = \{NB_i\}$ и $PB = \{PB_i\}$, $i = 1, \dots, Z$, удовлетворяющие условиям:

- NB_i нацело делится на NB_{i-1} , $i > 1$;

- $\sum_{i=1}^Z PB_i = 1$.

Смысл данной модели состоит в следующем: с вероятностью PB_i начало интер-

вала приходится на момент времени, который в абсолютном исчислении (в сек.) кратен NB_i , но не кратен NB_{i+1} . Например, частный случай выполнения ЭГУ будет выражен так: $Z = 1, NB = \{1_N\}, PB = \{1.0\}$. Заметим, что множество NB не имеет никакого отношения к иерархии индексов и может быть задано совершенно независимо от множества K . В листинге 1 представлена реализация этой модели с комментариями.

Разберем далее алгоритм для функции `getSingleDataLoc`. Последовательным вызовом функций `getReqLength` и `numberToHier` получаем два момента времени в их иерархическом представлении $B^{(1)} = \{b_i^{(1)}\}, B^{(2)} = \{b_i^{(2)}\}, i = 0, \dots, N$. Теперь, зная иерархию индексов и располагая ранее

сгенерированным полем, необходимо подсчитать, сколько индексов попадет в этот интервал. Введем понятие *седловой точки* интервала как $h(B^{(1)}, B^{(2)}) = \max(b_i^{(2)} > b_i^{(1)})$, и с ее помощью запишем принципиальный алгоритм этого подсчета.

1. $i := 0, s := 0$.

2. Сопоставлением с полем подсчитать количество индексов i -го уровня в интервале

$$(0, 0, \dots, 0, b_i^{(1)}, b_{i+1}^{(1)}, \dots, b_h^{(1)}, \dots, b_N^{(1)}),$$

$$(0, 0, \dots, 0, 0, b_{i+1}^{(2)} + 1, \dots, b_h^{(2)} + 1, \dots, b_N^{(2)}).$$

Нарастить значение s .

3. $b_{i+1}^{(1)} := b_{i+1}^{(1)} + 1$.

4. $i := i + 1$. Если $i < h$, перейти к шагу 2. Иначе — к шагу 5.

Листинг 1. Функция `getReqBegin ()`

Listing 1. Source code of `getReqBegin ()` function

```
int getRequestBegin() {
    int k,r_num,n1,n2,m;
    float s;
    //генерируем случайное целое число до тысячи, чтобы разыграть какому элементу
    // множества NB будет соответствовать начало интервала
    r_num = rand() % 1000;
    s=0.0;
    for (k=0;k<Z;k++) {
        s=s+Pb[k]*1000;
        if (r_num < s) break;
        break;
    }
    // вычисляем n1 - количество моментов времени, соответствующих разыгранному k
    if (k==Z-1) n1 = 86400/Nb[k];
    else n1=86400/Nb[k] - 86400/Nb[k+1]; //уменьшаем на число кратных следующему
    элементу
    //теперь из n1 значений нужно выбрать какое-то одно случайным образом
    m = (rand() % n1) * Nb[k];
    if (k != (Z-1)) {
        //из m чисел, делящихся на Nb[k], n2 делятся также и на Nb[k+1], поэтому нужно
        отсчитать вперед после m еще n2 чисел, которые делятся на Nb[k], но не делятся
        на Nb[k+1]
        n2 = m/Nb[k+1]+1;
        for(i=0;i<n2;i++) {
            m=m+Nb[k];
            if (m % Nb[k+1] == 0) m=m+Nb[k];
        }
    }
    return(m);
}
```

5. Сопоставлением с полем подсчитать количество индексов h -го уровня в интервале

$$(0, 0, \dots, 0, b_h^{(1)}, b_{h+1}, \dots, b_N),$$

$$(0, 0, \dots, 0, b_h^{(2)}, b_{h+1}, \dots, b_N)$$

(напомним, что согласно определению седловой точки $b_i^{(1)} = b_i^{(2)}$, при $i > h$). Нарастить значение s .

6. $i := h - 1$.

7. Сопоставлением с полем подсчитать количество индексов i -го уровня в интервале $(0, 0, \dots, 0, b_{i+1}^{(2)}, b_{i+2}^{(2)}, \dots, b_N^{(2)})$, $(0, 0, \dots, b_i^{(2)}, b_{i+1}^{(2)}, b_{i+2}^{(2)}, \dots, b_N^{(2)})$. Нарастить значение s .

8. Если $i = 0$, конец, s — результат. Иначе $i := i - 1$ и перейти к шагу 7.

Отметим, что при выполнении шагов 2, 5 и 7 количество индексов подсчитывается с учетом OR- либо XOR-операции, в зависимости от того, каких будет меньше. Так, на шаге 2 моделируется OR-операция, если $b_i > k_{i+1}/2$, и XOR-операция — в противном случае. На шаге 7, соответственно, наоборот: XOR-операция, если $b_i > k_{i+1}/2$, и OR-операция — в противном случае. На шаге же 5, в случае $h < N$ правило таково: OR-операция в случае $2(b_h^{(2)} - b_h^{(1)}) \leq k_{h+1} + 1$ и XOR-операция — в противном случае. Доказательство достаточно очевидно, пред-

ставим читателю возможность провести его самостоятельно.

Проиллюстрируем особенности алгоритма некоторыми примерами. В иерархии (секунда-минута-час) пусть $B^{(1)} = (6, 10, 3)$, $B^{(2)} = (3, 12, 5)$. Здесь $N = 2$, $h = 2$. Согласно алгоритму на шаге 2 находим количество секундных индексов на интервале $(6, 10, 3)$ — $(0, 11, 3)$ XOR-операцией, т. к. $6 < 60/2 = 30$. Затем на шаге 2 находим количество минутных индексов на интервале $(0, 11, 3)$ — $(0, 0, 4)$ XOR-операцией. Затем на шаге 5 количество часовых индексов на интервале $(0, 0, 4)$ — $(0, 0, 5)$ (это будет 0 или 1, т. к. интервал — один час). Далее на шаге 7 находим количество минутных индексов на интервале $(0, 0, 5)$ — $(0, 12, 5)$ OR-операцией, затем на шаге 7 количество секундных индексов на интервале $(0, 12, 5)$ — $(3, 12, 5)$ OR-операцией. Таким образом, стартовав от левой границы интервала и действуя согласно алгоритму, мы финишировали на его правой границе.

Разберем далее пример изначального интервала $(6, 10, 3)$ — $(20, 45, 3)$. Здесь $h = 1$. На шаге 2 находим количество секундных индексов на интервале $(6, 10, 3)$ — $(0, 11, 3)$ XOR-операцией. Затем на шаге 5 находим количество минутных индексов на интервале $(0, 10, 3)$ — $(0, 45, 3)$ XOR-операцией,

Листинг 2. Функция numberToHier ()

Listing 2. Source code of numberToHier () function

```
void numberToHier(int a, int res[hierN+1]) {
//массив res соответствует множеству В, константный массив hierarchy - множеству К
    int i, pr=1;
    for(i=0; i<hierN; i++) pr=pr*hierarchy[i]; //вычисляем размер самого крупного индекса иерархии
    for(i=hierN; i>=0; i--) {
//последовательно раскладываем абсолютное значение времени по иерархии сверху вниз
        res[i]=a/pr; //целая часть от деления
        if (i != 0) {
            a=a % pr; //остаток от деления
            pr=pr/hierarchy[i-1];
        }
    }
}
```

Листинг 3. Функция `hierToNumber ()`**Listing 3.** Source code of `hierToNumber ()` function

```

int hierToNumber(int res[hierN+1]) {
//массив res соответствует множеству В, константный массив hierarchy - множе-
ству К
int pr=1,sum=0,i;
for(i=0;i<=hierN;i++) {
sum=sum+pr*res[i];
if (i==hierN) break;
pr=pr*hierarchy[i];
}
return(sum);
}

```

т. к. $2 \cdot (45 - 10) = 70 > k_2 + 1 = 61$. Затем на шаге 7 находим количество секундных индексов на интервале $(0,45,3) - (20,45,3)$ OR-операцией.

В заключение второй части приведем коды вспомогательных функций и `numberToHier` (листинг 2) и `hierToNumber` (листинг 3), т. к. с одной стороны они невелики по объему, с другой помогут глубже понять детали рассматриваемой задачи.

Анализ выходных данных

Целью анализа выходных данных в рассматриваемой задаче является определение момента, когда моделирование следует прекратить, т. е. доверительный интервал для величины Y_{cp} с требуемым уровнем достоверности уже получен. Особенностью задачи является то, что этот анализ нужно проводить на двух уровнях — локальном (моделирование на одном поле) и глобальном (последовательность результатов моделирования на разных полях). Основы анализа выходных данных (*output analysis*) в имитационном моделировании изложены в [16, chapter 11]. Отметим, что наша задача, в отличие от классических моделей для систем с очередями, где анализируется, например, количество заявок в системе, обладает двумя особенностями, значительно облегчающими анализ выходных данных:

1) нет необходимости определять момент установления стационарного режима, все на-

блюдения анализируемой случайной величины — количества индексов для обслуживания запроса — являются одинаково репрезентативными;

2) все наблюдения случайной величины являются независимыми от предыдущих наблюдений, поэтому не требуется никаких дополнительных вычислений, чтобы отбросить влияние фактора автокорреляции.

Тем не менее при отладке модели требуется остановить выбор на достаточно четкой, хорошо апробированной и не требующей больших усилий при реализации процедуре определения момента завершения наблюдений (*terminating simulations*). В качестве таковой был выбран метод *групповых средних (batching means)*. Метод был впервые подробно описан в [18], развит в работах тех же авторов [17] и [19]. Наиболее доступное его изложение со схемой реализации приведено в обзорной статье [24]. Следуя этой работе, приведем описание метода, реализованного в модели в функциях *Batching (Glo/Loc)*, *Estimation (Glo/Loc)* и *SpectralVarAnalysis*, заодно исправив некоторые неточности, допущенные в оригинале (листинг 4). Итак, введем обозначения (их смысл становится более понятным из того, как они используются внутри метода):

M — количество групп, выбирается вычислителем, по умолчанию $M = 100$;

n_{max} — максимально возможное число наблюдений, не меньше M ;

n_v — количество наблюдений, используемое для оценки дисперсии $\sigma^2[X(n)]$, значение по умолчанию 100;

γ_a — коэффициент инкремента для контроля точности, значение по умолчанию 1,5;

$(1-\alpha)$ — уровень достоверности для итогового результата, значение по умолчанию 0,05;

ε_{\max} — максимальное значение для относительной точности доверительного интервала, значение по умолчанию 0.05.

Листинг 4. Функции выходного анализа. Псевдокод

Listing 4. Pseudocode of output analysis procedures

Procedure OutputAnalysis

const $M = 100$;

procedure *Batching*:

{преобразование отдельных наблюдений в $2M$ групп с последовательным возрастанием размера группы}

begin

{вычисление среднего $\bar{X}_j(m)$ на m наблюдениях и запись в j -й элемент буфера *AnalysedSequence*}

$\bar{X}_j(m) := \text{sum} / m$;

if $j = 2M$ **then**

{укрупнение $2M$ групп размера m каждая в M групп размером $2m$ каждая}

for $s:=1$ **to** M **do**

$\bar{X}_s(2m) := 0.5 (\bar{X}_{2s-1}(m) + \bar{X}_{2s}(m))$;

enddo

$m:=2m$; $j:=M$;

endif

$j:=j + 1$; $\text{sum}:=0$;

{начало формирования следующей группы}

end *Batching*;

procedure *Estimation*:

{вычисляет оценку среднего и проверяет ее точность, до тех пор пока требуемая точность не будет достигнута}

begin

находим оценку дисперсии $\sigma^2[\bar{X}(n_v)]$ для последовательности из последних n_v значений в *AnalysedSequence* и определяем степень свободы k (применением процедуры *SpectralVarAnalysis*);

вычисляем полуширину доверительного интервала для уровня достоверности $(1 - \alpha)$

в текущей контрольной точке w_k $\varepsilon = \frac{t_{k,1-\alpha/2} \sigma^2[\bar{X}(n_v)]}{\bar{X}(jm)}$, где $\bar{X}(jm) = \frac{\sum_{s=1}^j X_s(m)}{j}$ — теку-

щее значение оцененного среднего после jm наблюдений, $t_{k,1-\alpha/2}$ — табличное значение распределения Стьюдента.

{проверяем условие прекращения моделирования}

if $(\varepsilon \leq \varepsilon_{\max})$ **then**

{выводим результаты и заканчиваем моделирование}

StopSimulation:=true;

else

{требуемая точность еще не достигнута, устанавливаем следующую контрольную точку}

$k:=k+1$;

$w_k = \min(\lfloor \gamma_a w_{k-1} \rfloor, n_{\max})$;

end

end *Estimation*

begin {главная процедура, примерно соответствует функции *cycle (Glo/Loc)*}

```

m:=1; {начальный размер группы}
k:=1; {номер первой контрольной точки}
w0 = 0; i:=1;
w1:=2M; {значение по умолчанию для первой контрольной точки}
sum:=0; j:=1;
StopSimulation:=false;
while (not StopSimulation) do
{собираем и обрабатываем новые wk-wk-1 наблюдений}
моделируем очередное значение/наблюдение x [wk-1+i], соответствует функции
getSingleData (Glo/Loc)
sum:=sum + x [wk-1 + i];
if (i mod m = 0) then Batching
endif;
if (i = wk) then Estimation
endif
if (not StopSimulation) then
i:=i + 1;
if (i > nmax) then
write («Решение с требуемой точностью не может быть получено»);
StopSimulation:=true;
endif
endif
enddo
write («За jm наблюдений с уровнем достоверности (1-α) получено решение
X(jm)(1 ± ε)»)
end OutputAnalysis
procedure SpectralVarAnalysis
{Предусловия:
x (1), x (2), ..., x (nv) – последовательность nv наблюдений стационарного процес-
са (nv ≥ 100)
nap. – количество точек усредненной периодограммы, используемых для приведения
в соответствие полиному применением метода наименьших квадратов (nap ≤ nv/4, зна-
чение по умолчанию 25);
δ – степень полинома, соответствующего логарифму усредненной периодограммы
(значение по умолчанию 2);
Co – нормализующая константа, выбранная так, чтобы px(0) было несмещенной оцен-
кой. Для значений по умолчанию nap. и δ Co=0.882 [17], где эти константы обозна-
чены k, d и C1 (k, d).
Шаг 1.
Вычислить 2 nap значений периодограммы для последовательности x (1), x (2), ..., x
(nv)

$$\Pi \left( \frac{j}{n_v} \right) = \left| \sum_{s=1}^{n_v} x_s \exp \left[ - \frac{2\pi i(s-1)j}{n_v} \right] \right|^2 / n_v$$

для j = 1, 2, ..., 2 nap., i – корень из -1, мо-
дуль интерпретируется как модуль комплексного числа.
Шаг 2.
Вычислить nap. значений функции {L (fj)}, j = 1, 2, ..., nap., где fj = (4j - 1)/2
nv и L (fj) = log { [Π ( (2j - 1)/ nv) + Π (2j/ nv)]/2}.
Шаг 3.
Методом наименьших квадратов находим коэффициент a0 полинома g(f) = ∑s=0δ asfs, явля-
ющегося интерполяционным на множестве точек {L (fj) + 0.27}, j = 1, 2, ..., nap.;
{a0 – это несмещенная оценка логарифма px (0)}.

```

Шаг 4.

Вычислить $p_x(0) = C_{\sigma} e^{\sigma}$, $\sigma_{sp}^2[\bar{X}(n_v)] = p_x(0) / n_v$. Определяем k .

{ таблица степеней свободы k для распределения хи-квадрат оценки $\sigma_{sp}^2[\bar{X}(n_v)]$ для выбранных n_{ap} и δ приведена в [17]; для $n_v = 100$, $n_{ap} = 25$ и $\delta = 2$, $k = 7$ }

end SpectralVarAnalysis

Рассмотренный метод анализа выходных данных достаточно прост и удобен в реализации и показал свою практическую работоспособность.

Апробация модели

Прежде чем перейти к описанию имитационных экспериментов и представлению их результатов, опишем метод верификации модели. Как уже упоминалось, в [10] построена аналитическая модель, работающая с двумя ограничениями — выполнением элементарного граничного условия и экспоненциальным $F_1(t)$. Поэтому имитационную модель можно считать достоверной, если при тех же входных данных она дает приемлемые по точности результаты в смысле совпадения с результатами аналитической модели. Многочисленные эксперименты показали, что это действительно так — расхождение в итоговом значении Y_{cp} составляет не более 0.1, относительная погрешность — не более 1%.

При выборе экспериментов акцент делался на показе преимуществ, которые дает имитационная модель по сравнению с аналитической, в частности, как отличаются результаты моделирования при выполнении ЭГУ и невыполнении такового. На рисунке 2 показаны зависимости Y_{cp} от k_1 для $N = 1$, $K = \{k_1\}$, $F_1(t) = 1 - e^{-\lambda t}$, $\lambda = 1/30$, $F_2(t) = 1 - e^{-\mu t}$, $\mu = 0.001$. Это соответствует появлению новой записи в БД примерно каждые 30 сек. и средней длине интервала запроса примерно 17 мин. Для нижней линии ЭГУ выполняется, т.е. $NB = \{3600\}$, $PB = \{1.0\}$, для верхней были выбраны параметры $NB = \{1,60,3600\}$, $PB = \{0.33, 0.33, 0.34\}$, т.е. начало запроса может с равной вероятностью приходиться на границу секунды, минуты или часа. Обратим внимание на принципиальное отличие случая выполнения ЭГУ и невыполнения такового. Если ЭГУ выполняется, то каждому множеству $K = \{k_1\}$ соответствует свое множество $NB = \{k_1\}$, что значительно снижает практическое значение результата, т.к. на практике выбирается ие-

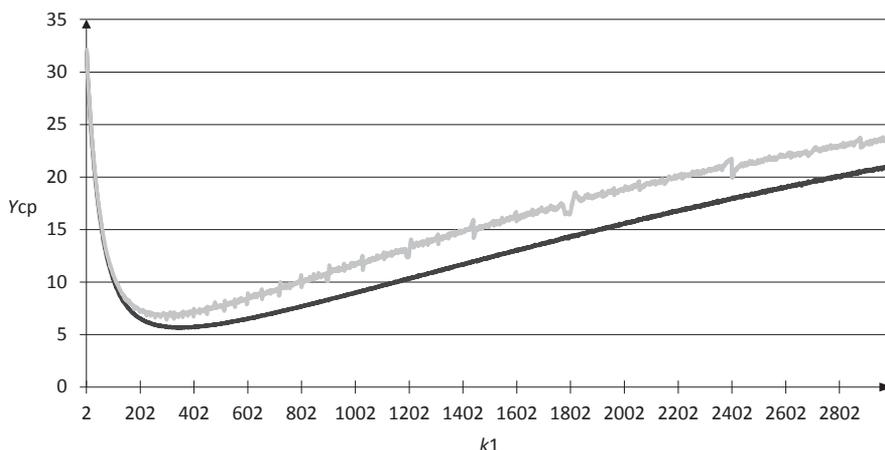


Рис. 2. $N = 1$. Зависимость Y_{cp} от $K = \{k_1\}$
 Fig. 2. $N = 1$. Dependency average of Y on $K = \{k_1\}$

рархия, а множества NB и PB являются входными данными модели; если же требования выполнения ЭГУ нет, множество NB постоянно для каждого множества K . Данное обстоятельство объясняет две особенности полученного результата:

1) кривые практически совпадают при малых k_1 и все более удаляются друг от друга с его ростом. В самом деле чем больше k_1 , тем дальше от левой границы интервала запроса находится левая граница ближайшего интервала для крупного индекса (при выполнении ЭГУ расстояние между этими левыми границами равно нулю), а следовательно, тем больше в начале интервала запроса нужно учесть мелких индексов. Поэтому с ростом k_1 расхождение между кривыми растет;

2) по той же причине наблюдаются флуктуации верхней кривой, т. к. на результат точно влияет не только значение k_1 , но и степень его «кратности» с границами крупных индексов.

Вместе с тем принципиальное поведение обеих зависимостей одно и то же. При выполнении ЭГУ минимум достигается при $k_1 = 354$ и равен 5.6, при невыполнении — примерно в диапазоне 320–340 и равен 6.5–6.6 (более точно сказать затруднительно, т. к. эти данные сильнее флуктуируют). С ростом k_1 Y_{cp} асимптотически стремится к предельному

значению, соответствующему случаю, когда иерархии индексов нет. В [8] в предположении экспоненциальности обоих распределений для него была получена точная формула

$$Y_{cp} = \frac{1 - e^{-\lambda}}{e^{\mu} - 1}$$

При $\lambda = 1/30$, $\mu = 0.001$ эта формула дает значение 32.45. Из этого следует вывод о значительном увеличении производительности обработки запросов, которое дает использование иерархических индексов. На рисунке 3 те же две зависимости изображены для $N = 2$, $K = \{k_1, 3600/k_1\}$, т. е. для каждого k_1 в иерархию добавлен еще один крупный индекс, равный часу.

Эксперименты проводились только для делителей числа 3600. Нетрудно видеть некоторое улучшение результатов при добавлении второго уровня. Так, минимумы, достигаемые при $k_1 = 300$, равны 5.1 и 5.9, соответственно. Гладкость верхней линии в отличие от рис. 2 не должна удивлять, т. к. модель отработала только для делителей 3600, а соединение между ними выполнено простой линейной интерполяцией.

Рисунок 4 аналогичен рисунку 2 с той разницей, что для функций $F_1(t)$ и $F_2(t)$ принято распределение Вейбулла $F(t) = 1 - e^{-(\lambda t)^c}$. λ и c для $F_1(t)$ и $F_2(t)$ выбраны таким обра-

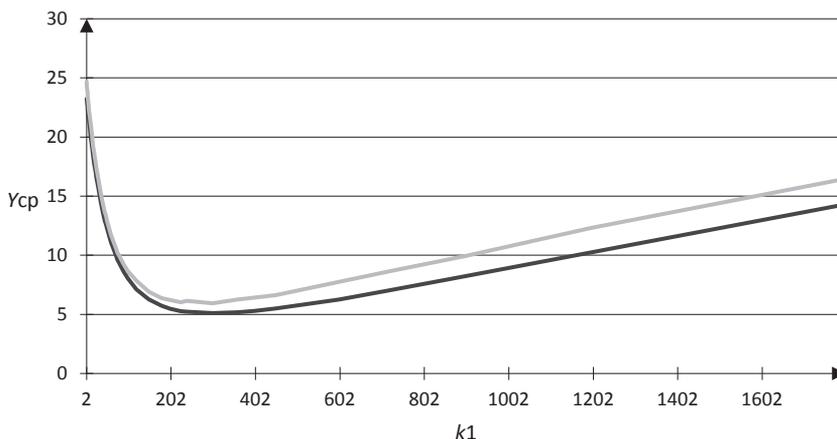


Рис. 3. $N = 2$. Зависимость Y_{cp} от $K = \{k_1, 3600/k_1\}$
 Fig. 3. $N = 2$. Dependency average of Y on $K = \{k_1, 3600/k_1\}$

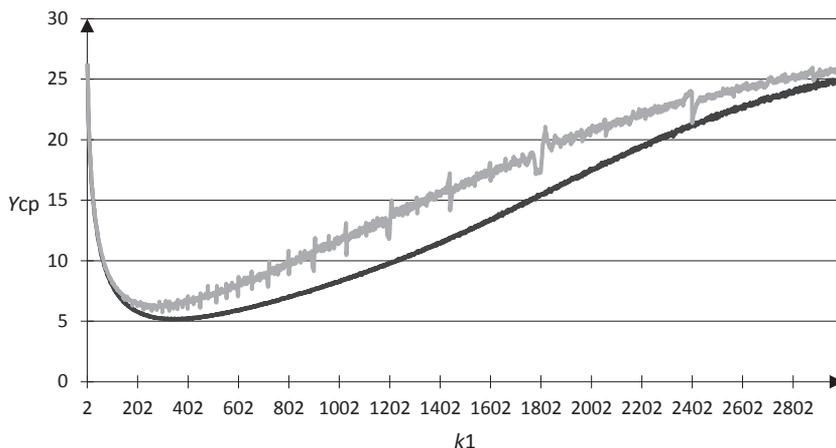


Рис. 4. $N = 1$. Распределение Вейбулла. Зависимость Y_{cp} от $K = \{k_1\}$

Fig. 4. $N = 1$. Weibull distribution. Dependency average of Y on $K = \{k_1\}$

зом, чтобы сохранить математическое ожидание соответствующих экспоненциальных распределений из рис. 2, что позволит увидеть влияние такого фактора как вид функции распределения. Для распределения Вейбулла первый момент, вычисляемый через гамма-функцию, равен

$$m_1 = \frac{\Gamma(1 + \frac{1}{c})}{\lambda}$$

Поэтому значения $c = 2, \lambda = 0.000887$ сохраняют для $F_2(t)$ среднее значение 1000; $c = 0.5, \lambda = 1/15$ сохраняют для $F_1(t)$ среднее значение 30. Как известно [13], при $c > 1$ распределение Вейбулла является стареющим, при $c < 1$ — молодеющим, поэтому для моделирования были выбраны оба варианта.

Минимумы равны приблизительно 5.1 и 5.9, причем для модели без ЭГУ минимум достигается несколько раньше.

Заключение

Дальнейшее развитие построенной имитационной модели можно производить в сторону как систематизации, так и углубления. Создан инструмент исследования, и важно разработать стратегию его наиболее эффективного использования. Речь идет о плани-

ровании имитационных экспериментов для выявления особенностей и закономерностей, присущих моделируемой системе, т. к. ввиду трудностей ее теоретического изучения они вряд ли могут быть выявлены каким-либо иным способом. Факторов, которыми можно варьировать в модели и выявление влияния которых представляет интерес, здесь довольно много. К ним относятся:

- виды распределений $F_1(t)$ и $F_2(t)$ и параметры этих распределений;
- размер иерархии N ;
- множество K , задающее непосредственно иерархию;
- множества NB и PB , регулирующие распределение начальной точки интервала запроса

Не исключено, что правильные вопросы, «заданные» модели, могут привести к весьма интересным и полезным в инженерном отношении ответам. Что же касается углубления модели, то и здесь то, что было описано в работе, можно рассматривать только как первый шаг в данном направлении. Помимо реализованного алгоритма существует еще множество алгоритмических вариаций применения иерархических bitmap-индексов, например:

- 1) динамический выбор между OR- и XOR-операциями на каждом уровне иерархии. Данная дисциплина описана и исследована в [14]. В ней решение принимается

на основании того, где фактически находится больше индексов — слева или справа от b_i ;

2) использование стратегии реализации запроса, описанное в [22] как *cut selection*. Она позволяет предельно минимизировать число операций с индексами при обслуживании запроса, правда в описываемом случае — ненамного;

3) кэширование индексов. До сих пор предполагалось, что индексные битовые строки берутся откуда-то сами по себе, т.е. считываются с диска в память и вносят одинаковый вклад в трудоемкость обслуживания запроса. На самом деле таковую можно снизить, если какие-то из них сохранить в памяти и при повторном использовании брать готовыми из кэша. Вопрос, какие именно и от чего это может зависеть, ждет своего исследования.

Для всех описанных вариаций требуется расширение возможностей имитационной модели путем ее алгоритмической доработки. Кроме того, интерес могут представлять и такие задачи:

1) получение с помощью модели не только Y_{cp} , но и распределения величины Y — количества индексов, участвующих в обслуживании запроса. Примеры графиков таких распределений приведены в [10];

2) нетрудно увидеть, что зависимости Y_{cp} от величины крупного индекса в целом однотипны (крутой спуск, затяжной минимум, где функция близка к константе, и далее медленный подъем с перегибом), поэтому вполне могут быть приближены каким-то параметризуемым семейством функций. Было бы целесообразно построить такую аппроксимацию. Заманчивые перспективы выполнения такого рода исследований открывает работа российского специалиста по имитационному моделированию [3];

3) наконец, фундаментальное решение поставленных в [10] задач оптимизации с помощью имитационного моделирования, содержательность которых проиллюстрирована данной работой, требует выбора соответству-

ющей методики оптимизации. Пример такой методики дан в статье известного российского ученого [4].

Список литературы

1. Артюхин В. В. О некоторых особенностях проектирования и реализации имитационных моделей процессов в сложных технических системах // Прикладная информатика. 2011. № 3 (33). С. 93–99.
2. Боев В. Д. Моделирование в AnyLogic. СПб.: Военная академия связи, 2016. — 412 с.
3. Девятков Т. В. Некоторые вопросы создания систем автоматизации имитационных исследований // Прикладная информатика. 2010. № 5 (29). С. 102–116.
4. Емельянов А. А. Планирование экстремальных экспериментов с имитационными моделями // Прикладная информатика. 2013. № 3 (45). С. 76–90.
5. Карпов Ю. Имитационное моделирование систем. Введение в моделирование с AnyLogic 5. СПб.: БХВ-Петербург, 2005. — 400 с.
6. Кирстен В., Ирингер М., Кюн М. СУБД Cache 5: объектно-ориентированная разработка приложений. 2-е изд. М.: Бином, 2005. — 416 с.
7. Мартынов Д. Bitmap-индексы в Cache на глобалах. URL: <http://habrahabr.ru/company/intersystems/blog/174657/>.
8. Труб И. И. Аналитическое вероятностное моделирование bitmap-индексов // Программные системы и вычислительные методы. 2016. № 4. С. 315–323.
9. Труб И. И. О распределении количества bitmap-индексов для произвольного потока занесения записей в базу данных // Программные системы и вычислительные методы. 2017. № 1. С. 11–21.
10. Труб И. И. Вероятностная модель иерархических индексов баз данных // Программные системы и вычислительные методы. 2017. № 4. С. 15–31.
11. Труб И. И. Объектно-ориентированное моделирование на C++. СПб.: Питер, 2005. — 416 с.
12. Труб И. И. СУБД Cache: работа с объектами. М.: Диалог-МИФИ, 2006. — 480 с.
13. Труб И. И., Труб Н. В. Имитационное моделирование вычислительных систем. Учебный практикум. LAP Lampert Academic Publishing, 2015. — 344 с. URL: simulation.su/uploads/files/default/2015-uchpraktikum-trub-trub.pdf.
14. Труб И. И., Труб Н. В. Модель иерархических индексов баз данных с принятием решений и ее сравнение с минимаксной моделью // Программные системы и вычислительные методы. 2018. № 1. С. 18–36.
15. Шарма В. Bitmap-индекс или B*tree-индекс: какой и когда применять? URL: http://citforum.ru/database/oracle/bb_indexes/.

16. Discrete-Event Simulation. 3rd ed. // Banks J., Carson J. S., Nelson B. L., Nicol D. M. Prentice Hall Englewood Cliffs, NS, 2000. — 594 p.
17. Heidelberg P., Welch P. D. A spectral method for confidence interval generation and run length control in simulations // In Comm. of the ACM. Special issue on simulation modeling and statistical computing. Vol. 24. Issue 4. April 1981. P. 233–245.
18. Heidelberg P., Welch P. D. Adaptive spectral methods for simulative output analysis // IBM J. Research Develop. Vol. 25. No. 6, 1981. P. 860–876.
19. Heidelberg P., Welch P. D. Simulation run length control in the presence of an initial transient // Operation Research. Vol. 31. No. 6. 1983. P. 1109–1144.
20. Kadow P. M. CacheObjectScript and MUMPS // Amazon.co.uk, Ltd., Great Britain, 2012. — 498 p.
21. Morzy M., Morzy T., Nanopoulos A., Manolopoulos Y. Hierarchical Bitmap Index: an Efficient and Scalable Indexing Technique for Set-Valued Attributes // In Proc. «Advances in Databases and Information Systems, Seventh East European Conference ADBIS 2003, Dresden, Germany, September 3–6, 2003». Vol. 2798 of Lecture Notes in Computer Science. Springer, 2003. P. 236–252.
22. Nagarkar P., Candan K. HCS: Hierarchical Cut Selection for Efficiently Processing Queries using Hierarchical Bitmap Indices // In Proc. 17-th International Conference on Extending Database Technology (EDBT), March 24–28, 2014, Athens, Greece. P. 271–282.
23. Otoo E., Wu K. Accelerating Queries on Very Large Datasets // Scientific Data Management. Challenges, Technology and Deployment (edited by Arie Shoshani, Doron Rotem). Chapman & Hall/CRC, 2010. P. 183–234.
24. Pawlikowski K. Steady-State Simulation of Queuing Processes: a Survey of Problems and Solutions // ACM Computer Surveys. Vol. 22. Issue 2. June 1990. P. 123–170.
4. Emelyanov A. *Planirovaniye ekstremal'nykh eksperimentov s imitacionnymi modelyami* [Designing extreme experiments based on simulation models]. *Prikladnaya informatika* — Journal of Applied Informatics, 2013, no. 3 (45), pp. 76–90.
5. Karpov Yu. *Modelirovaniye Sistem. Vvedeniye v Modelirovaniye s AnyLogic* [System Simulation. Introduction to Simulation with AnyLogic]. Saint-Petersburg, BHV-Peterburg Publ., 2005, 400 p.
6. Kirsten W., Ihringer M., Kuhn M., Rohrig B. *Object-oriented Application Development Using the Cache Postrelational Database*. 2th ed. Springer, 2003. (Russ. ed.: Kirsten V., Ihringer M., Kyun M., Rerig B. *Postrelyatsionnaya SUBD Cache 5: Ob'ektno-orientirovannaya razrabotka prilozhenii*. Moscow, BINOM Publ., 2005, 416 p.)
7. Martynov D. *Bitmap-indeksy v Cache na globalakh* [Cache bitmap-indices on globals]. Available at: <http://habrahabr.ru/company/intersystems/blog/174657/> (accessed 10.03.2018).
8. Trub I. I. *Analiticheskoe veroyatnostnoye modelirovaniye bitmap-indeksov* [Analytical probabilistic modeling of bitmap-indices]. *Programmnyye sistemy i vychislitel'nye metody*, 2016, no. 4, pp. 315–323.
9. Trub I. I. *O raspredelenii kolichestva bitmap-indeksov dlya proizvol'nogo potoka zaneseniya zapisey v bazu dannykh* [About bitmap-indices distribution for arbitrary random flow of database updating]. *Programmnyye sistemy i vychislitel'nye metody*, 2017, no. 1, pp. 11–21. Available at: www.nbpublish.com/library_read_article.php?id=21790 (accessed 10.03.2018).
10. Trub I. I. *Veroyatnostnaya model' ierarkhicheskikh indeksov baz dannykh* [Probabilistic model of hierarchical bitmap-indices]. *Programmnyye sistemy i vychislitel'nye metody*, 2017, no. 4, pp. 15–31. Available at: www.nbpublish.com/library_read_article.php?id=24437 (accessed 10.03.2018).
11. Trub I. I. *Ob'ektno-orientirovannoe modelirovaniye na C++* [Object oriented simulation on C++]. Saint-Petersburg, Piter Publ., 2005, 416 p.
12. Trub I. I. *SUBD Cache: rabota s ob'ektami* [Cache Database: Working with objects]. Moscow, Dialog MIFI Publ., 2006, 480 p.
13. Trub I. I., Trub N. V. *Imitatsionnoye modelirovaniye vychislitel'nykh sistem* [Simulation of computer systems]. LAP Lampert Academic Publ., 2015, 344 p.
14. Trub I. I., Trub N. V. *Model' ierarkhicheskikh indeksov baz dannykh s prinyatiem resheniy i ee sravneniye s minimaksnoy model'yu* [Model of hierarchical bitmap-indices with making decisions and its comparison with min-max model]. *Programmnyye sistemy i vychislitel'nye metody*, 2018, no. 1, pp. 18–36. Available at: www.nbpublish.com/library_read_article.php?id=25369 (accessed 23.03.2018).

References

1. Artyuhin V. V. *O nekotorykh osobennostyakh proektirovaniya i realizacii imitacionnykh modeley processov v slozhnykh tehnicheskikh sistemah* [Designing and implementing simulation models of complex engineering systems]. *Prikladnaya informatika* — Journal of Applied Informatics, 2011, no. 3 (33), pp. 93–99.
2. Boev V. D. *Modelirovaniye v AnyLogic* [Simulation in AnyLogic]. Saint-Petersburg, Voennaya Akademiya Svyazi Publ., 2016, 412 p.
3. Devyatkov T. V. *Nekotorye voprosy sozdaniya sistem avtomatizacii imitacionnykh issledovaniy* [Some Problems of Computer-Aided Simulation Study Systems Development]. *Prikladnaya informatika* — Journal of Applied Informatics, 2010, no. 5 (29), pp. 102–116.

15. Sharma V. Bitmap Index vs B*tree Index: Which and When? Available at: http://citforum.ru/database/oracle/bb_indexes/ (accessed 10.03.2018).
16. Discrete-Event Simulation. 3rd ed. Banks J., Carson J. S., Nelson B. L., Nicol D. M. Prentice Hall Englewood Cliffs, NS, 2000, 594 p.
17. Heidelberg P., Welch P. D. A spectral method for confidence interval generation and run length control in simulations. In *Comm. of the ACM. Special issue on simulation modeling and statistical computing*, 1981, vol. 24, issue 4, April, pp. 233–245.
18. Heidelberg P., Welch P. D. Adaptive spectral methods for simulative output analysis. *IBM J. Research Development*, 1981, vol. 25, no. 6, pp. 860–876.
19. Heidelberg P., Welch P. D. Simulation run length control in the presence of an initial transient. *Operation Research*, 1983, vol. 31, no. 6, pp. 1109–1144.
20. Kadow P. M. CacheObjectScript and MUMPS. Amazon.co.uk, Ltd., Great Britain, 2012, 498 p.
21. Morzy M., Morzy T., Nanopoulos A., Manolopoulos Y. Hierarchical Bitmap Index: an Efficient and Scalable Indexing Technique for Set-Valued Attributes. In *Proc. «Advances in Databases and Information Systems, Seventh East European Conference ADBIS 2003, Dresden, Germany, September 3–6, 2003»*, vol. 2798 of Lecture Notes in Computer Science. Springer, 2003, pp. 236–252.
22. Nagarkar P., Candan K. HCS: Hierarchical Cut Selection for Efficiently Processing Queries using Hierarchical Bitmap Indices. In *Proc. 17-th International Conference on Extending Database Technology (EDBT)*, March 24–28, 2014, Athens, Greece. P. 271–282.
23. Otoo E., Wu K. Accelerating Queries on Very Large Datasets. *Scientific Data Management. Challenges, Technology and Deployment* (edited by Arie Shoshani, Doron Rotem). Chapman & Hall/CRC, 2010, pp. 183–234.
24. Pawlikowski K. Steady-State Simulation of Queuing Processes: a Survey of Problems and Solutions. *ACM Computer Surveys*, 1990, vol. 22, issue 2, June. P. 123–170.

I. Trub, Samsung R&D Institute Rus, Moscow, Russia, itrub@yandex.ru

Simulation of hierarchical bitmap-indices

The article describes simulation model of search queries to database with logical operations on hierarchical bitmap indices. Proposed approach considers bitmaps on time property and combines multileveling of bitmaps with their binning on sequence of different time units. The input data are distribution of creation new records flow and distribution of query time interval, the varying condition is bitmap hierarchy, the model output is average number of OR/XOR logical operations to satisfy query. The target is to choose bitmap indices hierarchy, that minimizes this number. Simulation technique for such kind of model is very special, that is why model is manually implemented on C language in order to carefully cover all its features. Much attention is paid to output analysis as stopping rule when required confidence interval for target parameter is obtained. Mathematical background of model includes random number generators and some numerical methods, such as least square method, integration and spectral analysis. Model verification is provided by several ways, in particular, by comparison with existing analytical model results when input distribution is exponential. Several experiments confirm existence of optimal hierarchy and were carried out for different kinds of distributions including heavy-tailed. Graphical illustrations of simulation results are presented.

Keywords: hierarchical bitmap-indices, OR and XOR operation, random event flow, distribution density function, simulation, confidence interval.

About authors:

I. Trub, PhD in Technique

For citation:

Trub I. Simulation of hierarchical bitmap-indices. *Prikladnaya informatika — Journal of Applied Informatics*, 2018, vol. 13, no. 4 (76), pp. 53–69 (in Russian).

А. Ю. Новиков, магистрант кафедры системного моделирования и автоматизированного проектирования» ФГБОУ ВО «Московский авиационный институт (национальный исследовательский университет)», г. Москва, science@conflab.ru

П. П. Кейно, канд. техн. наук, старш. преподаватель кафедры системного моделирования и автоматизированного проектирования ФГБОУ ВО «Московский авиационный институт (национальный исследовательский университет)», г. Москва, science@blockset.ru

Л. Л. Хорошко, канд. техн. наук, доцент, зав. кафедрой системного моделирования и автоматизированного проектирования ФГБОУ ВО «Московский авиационный институт (национальный исследовательский университет)», г. Москва, smig@mati.ru

Разработка архитектуры интернет-сервиса организации научных мероприятий с автоматизацией документооборота

В статье показана актуальность разработки архитектуры облачного интернет-сервиса организации документооборота заявок на научные конференции. Авторами описан набор функционала для автоматизации развертывания пользовательских веб-ресурсов и рассматривается применение стандарта децентрализованной авторизации OpenID Connect. В работе также описана концепция использования гибкой модели полей, предоставляющих широкий функционал конструирования форм подачи заявок.

Ключевые слова: веб-приложение, SaaS, интернет-сервис, документооборот, научные конференции, Conflab, Лаборатория Конференций, ConflD.

Введение

Проведение научных конференций часто сопровождается большим количеством организационных аспектов [1, 2], большую часть которых целесообразно решать, используя цифровой документооборот. Такой способ повышает эффективность взаимодействия участников и руководителей, позволяя наглядно представить процесс рецензирования научных материалов.

Применение для этих целей электронной почты не самый эффективный способ, так как в этом случае увеличивается количество рутинной работы секретарей (руководителей секций). В современных реалиях все чаще

используются платформы для автоматизации документооборота заявок. В данной статье рассматривается функционал сервиса, реализующего полный цикл проведения мероприятия, начиная от приема заявок и заканчивая генерацией готового сборника тезисов.

Для сокращения времени, которое расходуется участником на повторную регистрацию на каждом веб-ресурсе, была поставлена задача унифицировать данные профиля путем разработки отдельного вспомогательного сервиса.

Важной составляющей проекта является сбор сведений о результатах научной деятельности пользователя, а также каталогизация и агрегирование всех конференций для предоставления возможности выбора на ос-

нове его предпочтений и интересов, а также научной тематики мероприятий, в которых он принимал участие.

Описание сервиса ConfLab

Предлагаемая платформа предоставляет широкие возможности для создания и управления веб-ресурсом конференции. Каждый пользователь получает доступ к единому личному кабинету, функционал которого зависит от его роли в проекте. В связи с этим возникла необходимость разделения прав доступа на следующие категории [3]:

- **Администратор проекта.** Изначально данный статус присваивается пользователю, создавшему проект, однако права могут быть делегированы любому зарегистрированному пользователю. Данная группа пользователей наделена наибольшим функционалом для контроля процесса документооборота.

- **Руководитель секции.** Права данной группы назначаются администратором и могут распространять свое действие как на определенные секции, так и на весь проект.

- **Участник.** Имеет доступ только к собственным заявкам.

- **Гость.** Не имеет доступа к разделам личного кабинета.

Развертывание проекта

Процесс создания проекта на базе платформы «Лаборатория конференций» разделен на четыре шага.

1. Ввод основной информации: наименование (полное и сокращенное), краткое описание ресурса.

2. Выбор типа домена (внешний или внутренний) и его резервирование. В процессе проверки введенный домен проверяется на занятость и на соответствие списку недопустимых и зарезервированных значений (таких, как `admin`, `root`, `home` и т. д.).

3. Затем пользователю предлагается настроить параметры сроков: дата начала и за-

вершения приема заявок, дата завершения редактирования заявок автором и секретарем, а также дата глобального завершения действий на проекте.

4. На данном шаге выбирается наличие секций и, если мероприятие разделяется на секции или подсекции, необходимо ввести их названия.

Далее происходит копирование предустановленной стандартной темы оформления в директорию проекта. Благодаря использованию шаблонизатора Smarty и, как следствие, разделению бизнес-логики и представления администратор получает возможность доступа к редактированию шаблонов [4]. Для добавления динамических данных в шаблоны администратору предоставляется API в виде набора структурированных объектов [5]. В качестве примера — генерация списка программы секции, для формирования которой происходит соответствующий запрос к базе данных.

Ведение истории статусов заявок

Процесс рецензирования научных работ подразумевает их принятие и отклонение, для этого были внедрены следующие статусы заявок:

- **Утверждено** — заявка одобрена руководителем.

- **В обработке** — по данной заявке ожидается решение секретаря (руководителя секции).

- **Недооформлено** — данный статус заявка получает, если пользователь при подаче заполнил не все обязательные поля либо секретарь ожидает доработки заявки пользователем.

- **Отклонено** — заявку отклонил руководитель. В данном случае возможность доработки заявок недоступна.

Стоит отметить, что участник имеет возможность обновления своих заявок, а значит, хранение только последнего статуса недостаточно. Для полного понимания действий

обеих сторон внедрено хранение истории статусов, даты изменения и комментария. При обновлении заявки пользователем в качестве комментария добавляются названия полей, подвергшихся изменению. Если изменение статуса выполняется секретарем, имеется возможность добавления текстового комментария.

Применение данного функционала позволяет пользователям получать оповещение о любых изменениях, а секретарям не пренебрегать важными нюансами.

Ключевой особенностью проекта является разбиение проводимой конференции на секции с индивидуальными параметрами, набором полей и правами доступа. Например, в проекте имеется возможность создания секции на английском языке и сокрытия ее от русскоязычных пользователей и наоборот. Кроме того, каждая секция может иметь как общие поля, так и индивидуальные.

Описание гибкой архитектуры метамодели полей

Различные мероприятия оперируют определенным набором данных. Одной из задач интернет-сервиса являлось создание архитектуры элементов — полей для ввода данных пользователями. Каждое поле — объект, обладающий набором свойств:

- **Type (string)** — тип поля, по аналогии с атрибутом `type` в элементах HTML-формы.
- **Label (string)** — название поля, отображаемое в виде подзаголовка.
- **Description (string)** — текстовая подсказка к заполнению.
- **Required (int)** — устанавливает поле обязательным к заполнению. Поддерживается возможность отложенного обязательного заполнения, т. е. пока данные не введены, заявка имеет статус «недооформлено».
- **Restrictions (string)** — определяет правила валидации данных с помощью шаблона регулярных выражений либо по расширению файла [6].

- **Status (bool)** — флаг наличия ошибок в заполнении.

- **Value (string)** — значение поля. Если тип поля является файлом, возможно указание ссылки на него.

- **Options (string)** — дополнительные данные в JSON-представлении, используемые для таких типов, как радиокнопки и раскрывающиеся списки. В данном свойстве содержится соответствие пар: «ключ-значение» для элементов поля.

- **Sort (int)** — численное значение, отвечающее за порядок сортировки выводимых полей.

- **Spreadsheet (bool)** — при положительном состоянии флага указанное поле генерируется в выходном файле электронных таблиц (например, формата Excel).

Поведение алгоритмов обработки напрямую зависит от установленных значений свойств, а значит, набор данных, необходимый для приема заявок на любую конференцию, можно описать с помощью предложенной модели.

Таким образом, внедрив обработку таких объектов, проект «Лаборатория конференций» позволяет обслуживать широкий спектр конференций, предоставляя администраторам интерфейс визуального редактора [7] метамоделирования полей, что позволяет моделировать формы любого уровня сложности в зависимости от данных, требуемых от заявителей на каждой конференции [8].

Генерация выходных документов

Результаты научной работы большинство конференций представляют в виде сборника статей или тезисов, однако верстка сборника зачастую выполняется вручную. Предлагаемый программный комплекс позволяет автоматизировать формирование выходных файлов благодаря использованию библиотек для генерации документов в форматах MS Word и MS Excel [9].

В результате выполнения запроса к базе данных получаем список заявок, каждая из которых имеет свой набор значений полей. Используя эти данные вместе с конфигурацией стилей, формируется документ, готовый к отправке на печать в типографию. Функционал используемой библиотеки позволяет добавлять оглавление, предметный указатель, а также иллюстративно-графические материалы [10]. Сборник отправляется в печать в формате MS Word практически без внесения каких-либо изменений, поскольку не требует дополнительной установки стилей, а текст тезисов редактируется оргкомитетом еще на этапе обработки заявки.

По итогам мероприятия происходит церемония награждения лучших участников. Имеется возможность отметить их заслуги и в личном кабинете. Администратор проекта может выбрать несколько типов грамот: за 1-е, 2-е, 3-е места, а также сертификат победителя, благодарности в организации конференции и др. В личном кабинете участника может быть сгенерировано неограниченное число грамот различного типа, которые он может в последующем скачать в формате PNG, SVG или PDF и распечатать.

Возможности сервиса позволяют по такому же принципу генерировать бейджи, раздаточные материалы, официальные протоколы, таблицы с перечнем участников и т. д.

Интеграция с сервисами поиска некорректных заимствований

Во время экспертной оценки научных материалов определение количества некорректных заимствований позволяет отсеивать работы с низкой степенью научной ценности. Функционал платформы позволяет получать процент оригинальности работы и формировать подробный отчет о найденных заимствованиях благодаря взаимодействию через REST-API с сервисом РуКонтекст [11].

Обрабатывающий скрипт вызывается каждые 15 минут в автоматическом режиме

благодаря использованию классического планировщика задач «cron», запускаемого системой UNIX в фоновом режиме. Происходит поиск заявок, значение оригинальности которых является пустым (равно NULL), а затем отправка HTTP-запросов к сервису РуКонтекст. Далее принимается ответ — строка в формате JSON, после анализа которой обновляются соответствующие значения в базе данных. Среди таких значений степень оригинальности, факт нахождения попыток обхода антиплагиата, идентификатор заявки проверяемого документа и др.

Интеграция с сервисом РуКонтекст выполнена на гибком уровне с использованием унифицированного шаблона проектирования «Адаптер». Это позволило обеспечить легкую миграцию с одного сервиса проверки заимствований на другой. Разработан единый интерфейс доступа в рамках соответствующего класса. Таким образом, достаточно переписать реализацию класса-адаптера, не затрагивая основную часть системы.

Описание сервиса ConfID

Разработанный сервис подразумевает создание множества проектов, оперирующих данными о научной деятельности пользователя. Для создания единой точки входа был создан вспомогательный интернет-сервис — ConfID. На данном ресурсе сохраняется информация профиля, которая копируется при входе в личный кабинет сайта конференции. Полная схема работы сервиса представлена на рис. 1.

Кроме того, по завершении мероприятия в профиль участника добавляется соответствующая отметка об участии. Страница профиля по умолчанию является приватной, однако в настройках имеется возможность активировать публичный режим, в этом случае она будет доступна по адресу следующего вида: <https://confid.ru/username>, где username — логин пользователя [12].

По мере увеличения количества проведенных на базе платформы научных конферен-

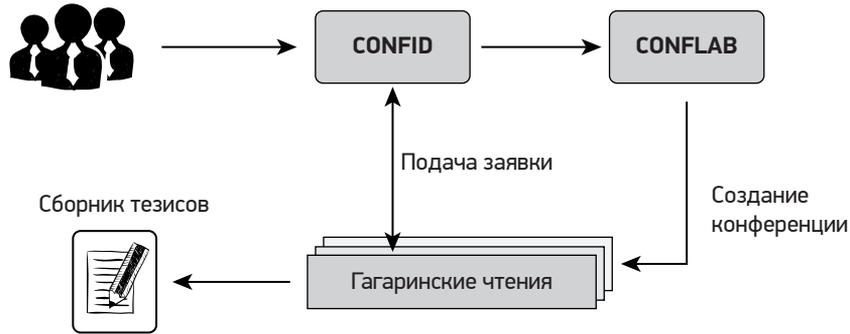


Рис. 1. Схема взаимодействия сервисов «ConfID» и «Лаборатории конференций»

Fig. 1. ConfID and Conference Laboratory services interaction schema

ций формируется научный каталог, данные которого используются для поиска мероприятий, совпадающих с интересами участника. Таким образом решается проблема поиска мероприятий, сокращаются временные затраты на повторную регистрацию, а также появляется возможность делиться своим профилем с участниками научного сообщества.

Подведем итоги вышесказанному: целью данной платформы является формирование электронной научной среды, объединяющей в себе множество разрозненных сообществ, издательств, мероприятий, студентов и научных работников. Подобное объединение позволяет решить множество проблем при поиске профильной научной конференции, журнала, а также коллег-ученых, имеющих схожие научные интересы, например, при необходимости наличия рецензентов на статью или диссертацию. Формирование научной среды может быть завершено только при наличии соответствующего «костяка» — наиболее активной группы студентов, ученых, пользователей системы. Это станет возможным только при внедрении основного проекта — «Лаборатории конференций» — в большое количество научных организаций и университетов.

Заключение

При проведении исследования в области автоматизации документооборота научных конференций авторы пришли к выводу

о том, что оптимальный результат по упрощению решения рутинных задач может быть достигнут только путем разработки специализированного программного комплекса, обладающего достаточной степенью гибкости по конфигурированию форм подачи заявок и валидации поступающих данных.

В связи с этим актуальной задачей является разработка программного комплекса, позволяющего автоматизировать процесс подачи заявок. Рассмотренная архитектура легла в основу программного комплекса «Лаборатория конференций». Программный комплекс апробирован в Московском авиационном институте при проведении научных мероприятий и признан победителем конкурса научных работ на конференции «Гагаринские чтения» в 2018 г.

Сервис идентификации ConfID внесет элемент социального взаимодействия. Система облегчит поиск рецензируемых журналов, профильных конференций и мероприятий, а также ученых со смежным профилем. Молодому ученому система также позволит найти рецензента на статью или оппонента на диссертацию, поскольку многие профили будут открыты по желанию их владельцев. Кроме того, ConfID является неким «научным паспортом» ученого, данные которого могут использовать внешние сервисы на базе спецификации OpenID Connect с согласия владельца профиля.

Сервис «Лаборатория конференций» является конструктором веб-приложений, ориентированных на проведение научных мероприятий. В основе сервиса лежит разбиение мероприятий по секциям с индивидуальными настройками прав доступа каждой секции, а также, что наиболее важно, полей различных типов, как индивидуальных, так и общих для всех секций. Сервис обладает мультиязычностью и позволяет создавать проект на нескольких языках. Каждая заявка, поступающая на обработку, может иметь один из нескольких статусов, что позволяет визуализировать ход ее рассмотрения, делая его прозрачным как для автора, так и для организатора конференции (секретаря секции). Генерация сопутствующих документов позволяет снизить другой аспект рутинной работы: сервис предоставляет возможность генерировать перечень докладчиков, сборник поданных работ, который можно сразу отправить в печать, а также различные грамоты, бейджи и многое другое.

Экосистема, включающая сервис идентификации ConFID и сервис организации научных конференций «Лаборатория конференций», образуют совместно виртуальную среду, являющуюся хорошей почвой для объединения множества разрозненных научных сообществ в едином информационном пространстве.

Список литературы

1. *Da Silva Almendra V., Enăchescu D., Enăchescu C.* Ranking computer science conferences using self-organizing maps with dynamic node splitting // *Scientometrics*. 2015. Т. 102. №. 1. С. 267–283.
2. *Chan L., Loizides F.* (ed.). *Expanding Perspectives on Open Science: Communities, Cultures and Diversity in Concepts and Practices: Proceedings of the 21st International Conference on Electronic Publishing*. IOS Press, 2017.
3. *Кейно П. П.* Унификация типов прав доступа к информационным ресурсам облачных Web-сервисов // *Безопасность информационных технологий*. 2015. Т. 22. №. 1. С. 87–89.
4. *Shklar L., Rosen R.* *Web application architecture*. John Wiley & Sons, 2009. С. 259–262.
5. *Welling L., Thomson L.* *PHP and MySQL Web development*. Sams Publishing, 2003. С. 569–595.
6. *Darie C., Bogdan B., Hendrix A.* *AJAX and PHP: Building Modern Web Applications* // Packt Publishing — 2015. С. 155–160.
7. *Ветров А. А., Хаустов П. А.* Визуальный редактор математических формул для систем электронного документооборота // *Молодежь и современные информационные технологии: сборник трудов XIV Международной научно-практической конференции студентов, аспирантов и молодых ученых, г. Томск, 7–11 ноября 2016 г. Т. 2. Томск, 2016. Изд-во ТПУ, 2016. Т. 2. С. 64–65.*
8. *StröEle V., ZimbrăO G., Souza J. M.* Group and link analysis of multi-relational scientific social networks // *Journal of Systems and Software*. 2013. Т. 86. №. 7. С. 1819–1830.
9. *Диметурев П. П.* Методы генерации word-документов в веб-приложениях // *Молодежный Вестник УГАТУ. Ежемесячный научный журнал*. 2015. №1 (13). С. 56–60.
10. *Сова Е. В., Рыбанов А. А.* Сравнительный анализ библиотек генерации отчетов в веб-ориентированных информационных системах // *Международный журнал прикладных и фундаментальных исследований*. 2012. №. 7.
11. *Masse M.* *REST API Design Rulebook: Designing Consistent RESTful Web Service Interfaces*. O'Reilly Media, Inc., 2011.
12. *OpenID Connect Core 1.0 Incorporating errata set 1*. 2018. URL: http://openid.net/specs/openid-connect-core-1_0.html (Дата обращения 01.02.2018).

References

1. *Da Silva Almendra V., Enăchescu D., Enăchescu C.* Ranking computer science conferences using self-organizing maps with dynamic node splitting. *Scientometrics*, 2015, vol. 102, no. 1, pp. 267–283.
2. *Chan L., Loizides F.* (ed.). *Expanding Perspectives on Open Science: Communities, Cultures and Diversity in Concepts and Practices: Proceedings of the 21st International Conference on Electronic Publishing*. IOS Press, 2017.
3. *Keyno P. P.* Unifying types of access permissions to information resources of cloud Web-services. *IT Security* — 2015, no. 1, pp. 87–89 (in Russian).
4. *Shklar L., Rosen R.* *Web application architecture*. John Wiley & Sons, 2009. pp. 259–262.
5. *Welling L., Thomson L.* *PHP and MySQL Web development*. Sams Publishing, 2003, pp. 569–595.
6. *C. Darie, B. Bogdan, A. Hendrix* *AJAX and PHP: Building Modern Web Applications*. Packt Publishing, 2015, pp. 155–160.

7. Vetrov A. A., Haustov P. A. *Vizual'nyj redaktor matematicheskikh formul dlja sistem jelektronnogo dokumentooborota. Molodezh' i sovremennye informacionnye tehnologii: sbornik trudov XIV Mezhdunarodnoj nauchno-prakticheskoj konferencii studentov, aspirantov i molodyh uchenyh*, Tomsk, 7–11 November 2016. Vol. 2. Tomsk, 2016. Izd-vo TPU, 2016, vol. 2, pp. 64–65.
8. StröEle V., Zimbrão G., Souza J. M. Group and link analysis of multi-relational scientific social networks. *Journal of Systems and Software*, 2013, vol. 86, no. 7, pp. 1819–1830.
9. Dimetirev R. R. *Metody generacii word-dokumentov v veb-prilozhenijah* [Generation methods of Word-documents in Web-applications]. *Molodezhnyj Vestnik UGATU. Ezhemesjachnyj nauchnyj zhurnal*, 2015, no. 1 (13), pp. 56–60.
10. Sova E. V., Rybanov A. A. *Sravnitel'nyj analiz bibliotek generatsii otchetov v veb-orientirovannykh informatsionnykh sistemakh* [Comparasion Analysis of report generation libraries in Web-oriented informational systems]. *Mezhdunarodnyj zhurnal prikladnykh i fundamental'nykh issledovanij*, 2012, no. 7.
11. Masse M. REST API Design Rulebook: Designing Consistent RESTful Web Service Interfaces. O'Reilly Media, Inc., 2011.
12. OpenID Connect Core 1.0 Incorporating errata set 1, 2018. Available at: http://openid.net/specs/openid-connect-core-1_0.html (accessed 01.06.2018).

A. Novikov, Department of System Modeling and Computer Aided Design, Moscow Aviation Institute (National Research University), Russia, science@confiab.ru

P. Keyno, Department of System Modeling and Computer Aided Design, Moscow Aviation Institute (National Research University), Russia, science@blockset.ru

L. Khoroshko, Department of System Modeling and Computer Aided Design, Moscow Aviation Institute (National Research University), Russia, smig@mati.ru

Architecture development of internet-service for science conferences organizing with workflow automatization

The article examines the relevance of development architecture the cloud internet-service for organization application workflow on the scientific conferences. The main goal of the development is the building of digital scientific community to unite scientists, publishers and conference hosts. Article describes a functional set for automatic deployment of a customizable web-resource that can be easily configured to provide individual interface for processing application based on organizer's purposes. Organizer can operate flexible model if fields which provides a wide range of functions for designing applications forms. Such model can provide a fields order, their types and additional options for each application in different sections. Every project can be divided into sections and each section can be operated by different persons. Each application has a status that shows a progress of revisited work. There are a four kind of status: approved, declined, in process, incomplete. Approved works automatically publishing on a user web-resource. All approved works also will be published on a automatically-generated digest that can be printed or uploaded on a web-site as electronic resource. The system can generate several kinds of a output documents like a digest, spreadsheet with statistics, programme of the event and so on. The internet-service that developed and described in the article can be interested by universities and scientific organizations. It optimizes a workflow of different events and decrease of routine tasks by digitalization.

Keywords: web-application, SaaS, internet-service, workflow, scientific conferences, ConfLab, ConfID.

About authors:

A. Novikov, Assistant;

P. Keyno, PhD in Engineering, Senior Lecturer;

L. Khoroshko, PhD in Engineering, Head of the department

For citation:

Novikov A., Keyno P., Khoroshko L. Architecture development of internet-service for science conferences organizing with workflow automatization. *Prikladnaya Informatika* — Journal of Applied Informatics, 2018, vol. 13, no. 4 (76), pp. 70–76 (in Russian).

Ю. В. Нефедов, канд. экон. наук, доцент РЭУ им. Г. В. Плеханова, г. Москва, Nefedov.YV@rea.ru
А. Ю. Сергеева, магистр, консультант SAP CRM, «Мастердата», г. Москва, Sergeeva-AY@edu.rea.ru

Типовые проблемы интеграции ИС и пути их решения

В статье приведена классификация проблем интеграции в соответствии с уровнями взаимодействия систем. Перечислены этапы реализации подхода к обеспечению взаимодействия систем в соответствии с государственным стандартом. Авторами рассматривается архитектурный подход к обеспечению интероперабельности систем. На примере ритейл-компании приводится решение задачи интеграции разнородных информационных систем путем применения подхода микросервисной архитектуры.

Ключевые слова: интеграция информационных систем, уровни интеграции, микросервисная архитектура.

Введение

Деятельность большинства компаний, входящих в топ-50 крупнейших компаний российского IT-рынка, связана с системной интеграцией. Рост выручки этих компаний за 2015–2016 гг. варьировался от –18,3% до 84,6% [7]. Такая изменчивость объясняется частым появлением новых мировых трендов в технологиях, которые решают проблемы бизнеса.

Специфичность российского рынка усложняет процесс трансформации, поэтому заказчики IT-услуг по внедрению и интеграции сталкиваются с проблемами еще на этапе планирования проекта. Эти проблемы в каждом из случаев имеют особенности, но существуют паттерны, используемые компаниями-интеграторами для решения или предупреждения их появления. Некоторые подходы к решению типовых проблем рассматриваются в статье «Проблемы интеграции корпоративных информационных систем» А. А. Кусовым [5].

Проблемы интеграции связаны не только с программным обеспечением, но и с IT-инфраструктурой компании в целом. Интеграция систем должна осуществляться на всех уровнях взаимодействия: техническом, семан-

тическом и организационном для поддержки бизнес-процессов, обеспечивая гибкость и масштабируемость. Также большое значение имеет и взаимосогласованность систем и приложений. Об этом говорит О. А. Морозова в учебном пособии по интеграции корпоративных информационных систем [6].

Таким образом рост количества соответствующих публикаций подтверждает актуальность заявленной темы. В целях роста результативности выполняемых проектов по интеграции информационных систем авторами поставлена следующая задача исследования: сопоставить выявленные на практике трудности с интеграцией ИС с наиболее перспективными подходами к их решению и тем самым наметить предпосылки к созданию полноценной методики интеграции ИС. Практическая значимость данной статьи связана с растущей сложностью проектов по интеграции информационных систем и необходимостью повышения эффективности работы проектных команд.

Классификация проблем интеграции

Согласно ГОСТу Р 55062–2012, уровень взаимодействия систем определяет интероперабельность, т. е. способность систем и компонентов к обмену информацией и ее исполь-

зованию. Каждый из уровней взаимодействия описывает ключевые аспекты интеграции и содержит рекомендации по использованию стандартов. «Основным способом решения проблемы интероперабельности или “прозрачности” гетерогенной среды выступает последовательное применение принципов открытых систем и методологии функциональной стандартизации» [1]. Ниже приведены часто возникающие проблемы в соответствии с уровнями интеграции.

Технический уровень:

- Отсутствие единого формата интеграции (при организации взаимодействия двух систем обмен данными усложняется в связи с необходимостью приведения их к требуемому каждой из систем формату).

- Необходимость промежуточного программного слоя для объединения систем (Middleware).

- Автономность систем. Рассматривается случай, когда разработка и эксплуатация системы производятся независимо друг от друга.

Семантический уровень:

- Сложность семантической обработки информации с различных ресурсов.

- Пересечение смысловых фрагментов сообщений в разных форматах обмена данными (избыточность информации и ее дублирование).

- Конфликты именования.

- Различия формата данных.

Организационный уровень:

- Неактуальная документация или ее отсутствие.

- Отсутствие квалификации или низкий уровень подготовки интеграционных архитекторов.

- Сложность согласования процессов, связанная с различиями в структурах и реализации функций систем.

- Сложность поддержки интеграционных интерфейсов (связана с необходимостью поддержки специалистами обеих систем одновременно, а также с необходимостью обладать знаниями о функционале обеих систем).

- Отсутствие доступа и понятного пользовательского интерфейса для получения данных, содержащихся в различных источниках.

- Долгие сроки внесения изменений.

«Задачей интеграции является консолидирование всех уровней управления предприятием в единую цепочку. Данные уровни могут использоваться в совокупности в зависимости от поставленных целей, задач и условий предприятий» [10].

Этапы реализации подхода к интеграции систем

Существует методология Accelerated SAP (ASAP), используемая для внедрения и интеграции систем. Ее применение позволяет повысить качество внедрения, сократить время интеграции и затраты на оптимизацию других ресурсов.

Основываясь на успешном опыте, ASAP содержит шаблоны интеграции, сценарии внедрений и инструменты для реализации проектов. «Эта методология базируется на накопленном опыте тысяч внедрений SAP, проведенных за последние несколько лет, и включает в себя технические руководства по проведению всех стадий ускоренного внедрения» [4]. Методология ускоренного внедрения является устоявшейся практикой и используется в настоящее время не только для систем SAP.

Перечислим основные этапы методологии ASAP:

1. Подготовка проекта.
2. Концептуальное проектирование.
3. Реализация.
4. Заключительная подготовка.
5. Запуск и обслуживание.

Для каждого из этапов интеграции систем характерны свои проблемы, большую часть которых можно избежать. Ниже приведены примеры проблем и действия, выполняемые для предотвращения появления типовых проблем.

В процессе создания интероперабельной системы часто возникает проблема несогла-

сованности в сроках прохождения этапов. В случае задержки на одном из этапов сроки по всем последующим стадиям сдвигаются. Эта проблема решается с помощью методологий внедрения программного обеспечения. Они позволяют организовать процесс внедрения или интеграции в виде проекта, где есть ответственный за сроки и управление проектом в целом. Использование методологий предполагает понимание командой, на каком этапе сейчас находится проект, какие задачи стоят перед исполнителями на текущем этапе и какие проблемы мешают их решению.

Еще одной типовой проблемой является неграмотная оценка ресурсов. С этой проблемой сталкиваются как заказчики IT-услуг, так и исполнители. В случае, если оценка сроков выполнения работ была недостаточной, исполнитель будет вынужден заканчивать работу в неоплаченные часы, что скажется на общей прибыли от проекта. Для заказчика, помимо переплаты за выполнение работ с превышенной оценкой, проблемой может стать отсутствие необходимых человеческих ресурсов для контроля выполненных задач. Несмотря на то, что работу выполняет компания-интегратор, заказчику требуются сотрудники, которые смогут убедиться, что работы выполняются в соответствии с согласованной документацией и в установленные сроки.

Для решения этой проблемы выбирается подходящий тип договора — фиксированная оплата за объем работ или оплата отработанных часов в соответствии с оценкой. В случае выбора первого варианта описанная проблема не возникнет. А при выборе договора с оплатой часов требуются компетентные сотрудники, которые проведут анализ задач, оценят каждый из этапов решения и учтут время, требуемое на документацию, разработку, тестирование и показ выполненных работ.

Не менее важным, чем корректные планы и оценка ресурсов, является выбор технологий для обеспечения согласованности систем.

Архитектурный подход к интеграции систем

В стандарте ГОСТ Р 55062–2012 «Системы промышленной автоматизации и их интеграция. Интероперабельность. Основные положения», описывающем единый подход к обеспечению взаимодействия систем широкого класса, приведена эталонная модель интероперабельности (рис. 1), т. е. взаимодействия систем, и основные этапы обеспечения интероперабельности. С помощью этой модели и этапов создаются масштабные интероперабельные системы для различных областей применения, учитывающие их особенности.

Также стандартом руководствуются интероперабельные предприятия, обеспечивающие взаимодействие внутри компании (внутренняя интероперабельность). Этот стандарт определяет подход к реализации взаимодействия систем. Для непосредственной реализации требуется выбрать подходящее архитектурное решение.

Решением большинства типовых проблем технического уровня интеграции может являться использование микросервисного архитектурного стиля. Он заключается в создании сервисной шины (фасада), позволяющей интегрировать распределенные приложения. Эти приложения развертываются и разрабатываются в виде служб, запускающихся изолированно друг от друга. «Поддержка сервис-фасада — решает проблему доступа к конечным системам, не имеющим вид сервиса, посредством сервис-ориентированного интерфейса» [2].

Технологически метод микросервисов реализует управление структурой систем. «Микросервисы реализуют возможность управления структурированным информационным ресурсом посредством ограниченного набора операций: чтения и обновления отдельных частей» [8]. Это позволяет реализовать конкретные бизнес-требования пользователей, выделить схожие фрагменты логики

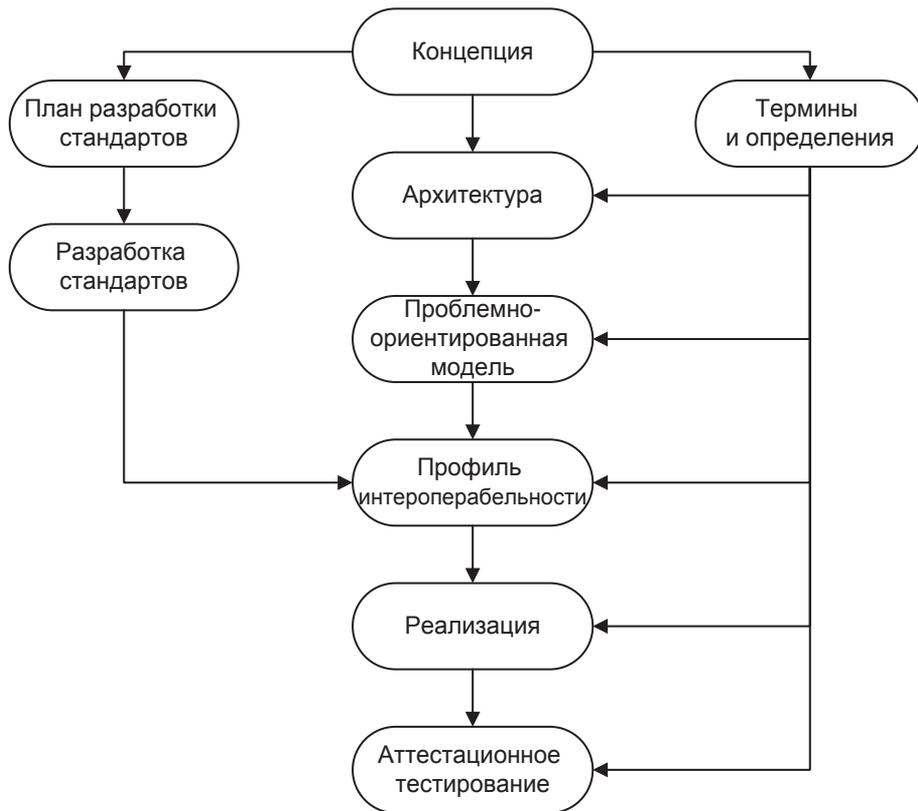


Рис. 1. Эталонная модель интероперабельности

Fig. 1. Reference model of interoperability

и унифицировать их, сохранив при этом возможность масштабирования.

Перечислим основные преимущества микросервисного архитектурного стиля:

- Децентрализованное управление данными.
- Возможность добавить сервис для экспериментов и тестирований.
- Возможность объединения и разъединения микросервисов.
- Независимое развертывание микросервисов.
- Скорость внесения изменений.
- Ориентация на потребности бизнеса вместо ограничений используемых технологий.
- Возможность поэтапной автоматизации процессов.

Недостатками этого архитектурного стиля можно считать:

- Сложность реализации мониторинга и прогнозирования системных проблем.
- Различия в стандартах и алгоритмах разработки сервисов.
- Необходимость учета вероятности отказа сервисов при проектировании приложений.

При выборе подхода следует учитывать то, что «интеграционные продукты обычно реализуют не только узкоспециальные представления сервисной шины как инфраструктуры для взаимодействия и визуализации сервисов в контексте сервис-ориентированной архитектуры, но и более широко — как инфраструктуру интеграции сервисов в самом общем смысле» [9].

Для оценки эффективности применения выбранного подхода используются методики оценки АИС в зависимости от стадии жизненного цикла. «Для каждой из этих стадий характерны отличия в целях оценки эффективности, составе выбираемого набора показателей эффективности (частных и обобщенных) и используемых методических подходах» [3].

Пример интеграции разнородных систем в ритейл-компанию

В компании «К», крупной сети магазинов ритейл-отрасли, используется большое количество информационных систем, таких как ERP, CRM, HR логистические, финансовые и т. д. У компании имеется несколько каналов продаж: сайт, колл-центр, розничные магазины и мобильное приложение, а также планируется добавление новых. Все эти каналы используют одни и те же данные для осуществления продаж: данные клиентов, данные заказов, информация о товарах и т. д. Таким образом, одним из важнейших условий является централизованный доступ к информационному ресурсу.

После согласования концепции и интеграционной схемы взаимодействия ключевых систем было принято решение о создании сервисной платформы. Она выступает в роли промежуточного программного обеспечения для хранения, преобразования и передачи данных.

Для предупреждения проблем на организационном уровне был создан репозиторий на базе «Wiki», в котором централизованно хранится проектная документация по всем существующим и планируемым интерфейсам и доработкам, выполняющимся в сервисной платформе. Удачным решением также является указание ответственных лиц на каждом этапе при создании технической документации. Это позволило ускорить процессы согласования технических решений, т. к. исполнители видели, кто из сотрудни-

ков компании-заказчика заинтересованное лицо.

На этапе реализации концепции архитекторы столкнулись с проблемой поддержки текущих потоков данных. В связи с тем, что одновременно ведутся разработки по нескольким проектам, было высказано требование, касающееся необходимости непрерывного взаимодействия систем до этапа продуктивного запуска сервисной платформы. На время реализации и тестирования производилась отправка данных по старым каналам и в сервисную платформу одновременно. Это позволило проверить готовность сервисной платформы к эксплуатации, не останавливая основные работы.

Для компании «К» выбор микросервисного архитектурного стиля оказался удачным, т. к. стоимость реализации такого подхода и техническая поддержка сервисной платформы требуют меньше ресурсов, чем создание централизованной системы с единой базой данных. Реализация сервисной платформы была проведена в срок, дополнительные соглашения на доработку системы заключены не были.

Подход, рассмотренный в примере, будучи основанным на положениях стандарта, может быть применим и к другим организациям, требующим взаимодействия большого количества информационных систем и их бесперебойной работы.

Заключение

Интеграция на каждом уровне, начиная с организационного, требует некоторого набора предварительных условий, значительно снижающих риски возникновения проблем. В процессе проектной деятельности по интеграции систем требуется обращать внимание на устоявшиеся практики внедрения. Опыт предыдущей деятельности позволяет минимизировать риски, связанные с организацией работы исполнителей, избежать неправильной оценки объема ра-

бот, выбрать соответствующий тип сотрудничества.

Важной частью первичной интеграции является выбор архитектурного подхода. Требуется обращать внимание на актуальные решения, такие как микросервисная архитектура. Ее преимуществом является возможность добавлять новые сервисы и изменять существующие в максимально короткие сроки и без перебоев в работе связанных систем. При этом следует обратить внимание на недостатки подхода, чтобы учесть их при построении оптимального взаимодействия систем и их обслуживания.

Представим основные результаты исследования:

1. Подтверждена актуальность действующего национального стандарта ГОСТ 55062–2012 как для систематизации потенциальных проблемных ситуаций, так и для выстраивания плана предстоящей интеграции ИС.

2. Экспериментально подтверждена актуальность микросервисного архитектурного стиля для решения большинства типовых проблем интеграции ИС технического уровня.

Список литературы

1. ГОСТ Р 55062–2012. Системы промышленной автоматизации и их интеграция. Интероперабельность. Основные положения. М.: Стандартинформ, 2012.
2. Даниленко И. Н., Кузин Д. А., Игнатова К. С. Архитектурные шаблоны интеграции разнородных информационных систем на основе сервисной шины // Вестник кибернетики. 2015. №4. С. 100–104.
3. Зацаринный А. А., Ионенков Ю. С. Некоторые аспекты оценки эффективности автоматизированных информационных систем на различных стадиях их жизненного цикла // Системы и средства информатики. 2016. Т. 26. №3. С. 122–135.
4. Кале В. Внедрение SAP R/3: Руководство для менеджеров и инженеров. М.: Компания АйТи, 2006. — 511 с.
5. Кусов А. А. Проблемы интеграции корпоративных информационных систем // Управление экономическими системами: электронный научный журнал. 2011. №4 (28). URL: <http://uecs.mcnip.ru>.

6. Морозова О. А. Интеграция корпоративных информационных систем: учебное пособие. М.: Финансовый университет, 2014. — 140 с.
7. Рейтинг российских IT-компаний — 2016 / ЗАО «Росбизнесконсалтинг». 1995–2018. URL: <http://www.rbcplus.ru/news/59013cb47a8aa90a969da442> (Дата обращения: 17.04.2018).
8. Смирнов М. Микросервисная архитектура в корпоративном IT-ландшафте // Открытые системы. СУБД. 2017. №4. С. 38–42.
9. Уайли Х., Ламброс П. Шаблоны взаимодействия приложений в корпоративных системах: Интеграционные решения с использованием продуктов IBM Enterprise Service Bus. 2010. URL: <http://www.ibm.com/developerworks/ru/library/ws-enterpriseconnectivitypatterns/>.
10. Хисамутдинов П. М., Хисамутдинов М. П. Научно-производственные проблемы интеграции разнотипных информационных систем // East European Scientific Journal. 2016. №11.

References

1. GOST R 55062–2012 *Sistemy promyshlennoy avtomatizatsii i ih integratsiya. Interoperabel'nost'. Osnovnye polozheniya* [State Standard 55062–2012. Systems of industrial automation and integration of them. Interoperability. The basic position]. Moscow, Standartinform Publ., 2012, 20 p.
2. Danilenko I. N., Kuzin D. A., Ignatova K. S. *Arhitekturnye shablony integratsii raznorodnykh informatsionnykh sistem na osnove servisnoy shiny* [Service-oriented architectural patterns for heterogeneous information systems integration]. *Vestnik kibernetiki*, 2015, no. 4, pp. 100–104.
3. Zatsarinny A. A., Iononkov Y. S. On aspects of automated information system efficiency evaluation at different stages of lifecycle. *Sistemy i sredstva informatiki* — Systems and Means of Informatics, 2016, vol. 26, no. 3, pp. 122–135 (in Russian).
4. Kale V. *Vnedrenie SAP R/3: Rukovodstvo dlya menedzherov i inzhenerov*. [SAP R/3 implementation: Manual for managers and engineers]. Moscow, Kompaniya AyTi, 2006, 511 p.
5. Kusov A. A. *Upravlenie ehkonomicheskimi sistemami* [Problems of corporate information systems integration]. 2011, no. 4 (28). Available at: <http://uecs.ru/instrumentalnii-metody-ekonomiki/item/411-2011-04-25-10-08-37> (accessed 22.07.2018).
6. Morozova O. A. *Integratsiya korporativnykh informatsionnykh sistem* [Integration of corporate information systems]. Moscow, Finance University Publ., 2014, 140 p.

7. *Reyting rossiyskih IT-kompaniy — 2016 ot ZAO «Rosbizneskonsalting»*. [Rating of Russian IT-companies — 2016]. Available at: <http://www.rbcpl.us.ru/news/59013cb47a8aa90a969da442> (accessed 27.07.2018).
8. Smirnov M. *Mikroservisnaya arhitektura v korporativnom IT-landshafte* [Microservice architecture in corporate IT-landscape] *Otkrytye sistemy.SUBD — Open Systems. DBMS*, 2017, no. 4, pp. 38–42.
9. Whily H., Lambros P. *Shablony vzaimodeystviya prilozheniy v korporativnyh sistemah: Integracionnye resheniya s ispol'zovaniem produktov IBM Enterprise Service Bus*. 2010. [Application interaction patterns in enterprise systems: solution based on IBM Enterprise Service Bus. 2010]. Available at: <http://www.ibm.com/developerworks/ru/library/ws-enterpriseconnectivitypatterns/>.
10. Hisamutdinov R. M., Hisamutdinov M. R. *Nauchno-proizvodstvennyye problemy integracii raznoplatformnykh informacionnykh sistem* [The scientific production integration issues for information systems based on different platforms]. *East European Scientific Journal*. 2016, no. 11.

Y. Nefedov, Plekhanov Economics University, Moscow, Russia, Nefedov.YV@rea.ru

A. Sergeeva, Masterdata, Moscow, Russia, Sergeeva-AY@edu.rea.ru

Typical problems of information systems integration and ways of their solution

The task of the study: to compare the difficulties identified in practice with the integration of information systems (IS) with the most promising approaches to their solution; thus, to identify the prerequisites for the creation of a full-fledged method of IS integration. The practical significance of this article is associated with the growing complexity of projects for the integration of information systems and the need to improve the efficiency of project teams. Before the trial integration project in a large retail company there was attempt of classification of issues for three levels: organizational, semantic and technical (according to GOST 55062–2012). The search for promising approaches to solve the most typical problems arising at the technical level was carried out.

The most significant conclusions are:

1. The relevance of the current national standard GOST 55062–2012 both for systematization of potential problem situations and for building a plan for the future integration of information systems is confirmed.

2. The relevance of micro-service architectural style for solving most typical problems of integration of is of technical level is experimentally confirmed.

Keywords: integration of information systems, integration levels, micro service architecture.

About authors:

Y. Nefedov, *PhD in Economics, Associate Professor*;

A. Sergeeva, *SAP CRM consultant*

For citation:

Nefedov Y., Sergeeva A. Typical problems of information systems integration and ways of their solution.

Prikladnaya informatika — Journal of Applied Informatics, 2018, vol. 13, no. 4 (76), pp. 77–83 (in Russian).

*А. И. Грачев, магистр прикладной информатики,
Университет «Синергия», г. Москва, g7alexg7@mail.ru*
*О. А. Страхов, канд. техн. наук, член-корреспондент МАН ИПТ,
Университет «Синергия», г. Москва, ostrakhov@synergy.ru*

Сравнительная характеристика способов построения облачных ERP-систем

Статья посвящена сравнению характеристик различных способов построения облачных ERP-систем. В работе описаны особенности каждой модели, их преимущества и недостатки как самостоятельно, так и применительно к конкретным условиям внедрения. Приведены примеры выбора оптимальных облачных решений для различных ситуаций. Статья предназначена для облегчения выбора способа развертывания ERP-систем руководством предприятия, а также компаниями-провайдерами и интеграторами, занимающимися предоставлением рассматриваемого программного обеспечения.

Ключевые слова: облачные технологии, ERP, гибридное облако, SaaS, модель внедрения.

Введение

С развитием современных информационных технологий выбор способа, применяемого для установки программного обеспечения в облаке, становится все более актуальным. Чаще всего данный вопрос возникает, когда дело касается автоматизации деятельности предприятий, относящихся по величине к диапазону от малого до среднего, что многие авторы отмечают в своих книгах [1–3]. Облачные вычисления определяются несколькими моделями установки, каждая из которых обладает специфическими преимуществами и недостатками. По причине наличия существенных различий между данными способами всем заинтересованным лицам необходимы четкие представления о характеристиках каждого метода.

В основе настоящей статьи лежит анализ характеристик способов установки облачных ERP-систем, выявление их особенностей, эффективных при определенных сценариях, и разбор оптимальности выбора на конкретных примерах. Целью дан-

ной работы является разработка алгоритма, позволяющего наиболее эффективно выбирать метод установки облачных ERP-систем. Механизм, описанный в статье, позволяет руководству компании принять взвешенное решение, поставщику же — повысить качество и эффективность оказываемых услуг. В качестве предмета исследования выступает процесс выбора способа установки облачных ERP-систем.

Содержание исследования

В настоящее время облачные вычисления стали доминантой индустрии информационных технологий — почти каждый человек, обладающий базовыми техническими навыками, способен получить доступ к обширному набору масштабируемых ресурсов с применением подписки. Облако устраняет большое количество затрат, связанных с покупкой оборудования, его настройкой и поддержанием в рабочем состоянии. Однако ведущей функцией, вызвавшей значительное распространение данной технологии, явля-

ется эластичность — способность услуги предоставлять и высвобождать вычислительные мощности в зависимости от текущего требования. Для многих клиентов вопрос доверия облачным службам до сих пор является актуальным. Передача конфиденциальных данных или инфраструктуры приложений провайдеру подобных услуг должна сопровождаться уверенностью, что поставщик обеспечит необходимый уровень безопасности и организует надежную защиту от посторонних. Настоящий вопрос был частично решен путем комплексного использования технологий авторизации и аутентификации, управления доступом, а также шифрования [4]. Другим подходом, позволяющим сочетать достоинства традиционных облачных решений с надежностью и безопасностью корпоративных систем, является создание так называемых частных облаков (private cloud). Частные облака предполагают выделение ресурсов ИТ-инфраструктуры предприятия (собственной и арендуемой) для размещения облачных сервисов. При этом на всю инфраструктуру распространяются общие требования по информационной безопасности и защите информации, действующие на предприятии.

С ростом компании пропорционально увеличивается количество и сложность рабочей нагрузки. Такие изменения требуют расширения функциональности информационных систем (ИС) предприятия, увеличения количества контролируемых показателей. Это в первую очередь отражается на ERP-системах. С ростом функциональности растет и сложность поддержки ИС, усложняется их обновление и сопровождение. Возникновение проблем в процессах и управлении означает необходимость перехода к облачной ERP-системе. Компаниям, работающим на ритейлеров или связанным внешне и внутренне с производством, требуются технологические инструкции для обеспечения функционирования промышленных процессов. Переход в облака позволяет разработать новые стратегические планы для повышения эффективности про-

изводства. В некоторых случаях компании начинают терять доходы по причине поздних обновлений клиентской информации, сведений о продажах и пробелов в коммуникации. Следовательно, возникает нужда в управлении рисками, управлении процессами, автоматизации отчетов и анализе данных для принятия решений, что также является веской причиной для перехода к облачной ERP-системе. Однако существенным сдерживающим фактором, препятствующим переносу ИТ-сервисов в облако, является хорошо проработанная за годы эксплуатации существующих клиент-серверных решений нормативно-справочная информация, описания производственных, технологических и бизнес-процессов, отлаженные схемы электронного документооборота внутри систем, а также надежные и проверенные интерфейсы и технологии межсистемного взаимодействия. Зачастую преимущества облачных технологий кажутся не сопоставимыми с возможными потерями. Следовательно, задача поиска оптимального способа внедрения облачных ERP-решений приобретает особую актуальность.

Для решения указанной задачи необходимо:

- 1) изучить возможность переноса существующих ERP-систем в облачную среду;
- 2) проанализировать существующие способы установки облачных ERP-систем;
- 3) сравнить характеристики различных способов установки облачных ERP-систем;
- 4) определить оптимальность применения того или иного способа внедрения облачной ERP-системы для различных специфик предприятий.

Теоретическая основа построения облачных ERP-систем

ERP-системы в настоящее время широко используются по всему миру компаниями для управления бизнес-активами и практической реализации стратегии ERP (Enterprise Resource Planning, планирование ресурсов

предприятия). Данный вид программного обеспечения интегрируется с рабочими процессами и деятельностью, такой как покупки, продажи, управление складом, производство и т. д. Функции включают в себя помощь в планировании, отслеживании и использовании всех бизнес-ресурсов для достижения поставленных целей. Таким образом, компания получает возможность управлять всеми операциями бизнес-подразделений для контроля рабочего потока [5–7].

Облачное ПО — программное обеспечение, размещенное в облаке и предоставляемое на основе подписки. Оно отличается повышенным уровнем доступности и мобильности по сравнению с традиционными способами. Облако — это сеть серверов, которые хранят данные и обладают собственной функцией. На одних серверах запущены приложения (происходит «предоставление услуги»), в то время как на других просто размещена информация, другими словами, облако — это среда, доступная удаленно. Основным преимуществом облака является возможность получения доступа к нему через сеть Интернет. Таким образом, можно воспользоваться размещенным приложением или хранимыми данными с любого компьютера, имеющего соединение с глобальной Сетью. Услуга включает в себя предоставление через Интернет инфраструктуры, хранилищ данных, платформ и различное программное обеспечение, в том числе и ERP-системы.

Выбор ERP-системы больше не заключается в определении подходящего поставщика, как, например, Microsoft, SAP и т. д. Первоначально стандартной схемой была покупка лицензии на программное обеспечение с последующим получением установочного диска. Со временем методы установки сильно поменялись, и поставщики могут поддерживать не все варианты, а иногда только некоторые или даже один.

Все три способа имеют одно общее свойство — доступ к ERP-системе получается удаленно и происходит через сеть Интернет.

Методы установки отличаются друг от друга по многим признакам, например, времени внедрения, затратам, применяемым технологиям. Выделяют следующие основные модели поставки облачного программного обеспечения: «Облачный хостинг», SaaS и «Гибридное облако».

Прежде, чем рассмотреть облачные способы установки, необходимо вспомнить о классическом методе. «На месте» — традиционная, самая распространенная и до сих пор самая используемая модель. При покупке лицензии поставщик предоставляет программное обеспечение посредством скачивания или на CD-/DVD-диске. Далее происходит установка на собственном сервере и при необходимости на отдельных рабочих станциях. При использовании модели «На месте» информация хранится в организации, что играет существенную роль в вопросе обеспечения конфиденциальности данных. При этом нет нужды ежемесячно оплачивать использование программного обеспечения. Однако следует учитывать затраты на поддержание работоспособного состояния серверов и улучшение оборудования, чего нет в случае с SaaS-моделью, где происходит масштабирование вычислительных ресурсов в зависимости от потребностей клиента.

Виртуальный хостинг означает, что клиент покупает программное обеспечение у вендора, а затем устанавливает его в дата-центре или хостинг-центре, где расположены физические или виртуальные сервера, на которые подписано предприятие-клиент [2–4]. Последующее внедрение решения происходит очень схоже с привычной схемой «На месте» или в офисах предприятия. Большинство бизнес-приложений имеют в своей основе архитектуру клиент-сервер. Доступ к программному обеспечению происходит через VPN и удаленный рабочий стол или терминальные службы, для чего на рабочей станции устанавливается специальное приложение. Таким образом, для запуска программы требуется сетевая инфраструктура, например,

использование тонкого клиента для получения доступа к терминальному серверу.

На сегодняшний день большинство приложений серийного программного обеспечения доступны в качестве «программного обеспечения как услуги» (Software as a Service, SaaS). SaaS — определенное приложение, которое управляется на облачном сервисе. Несмотря на то, что данные приложения поставляются на услугу публичного облака, на многих предприятиях существует тенденция к использованию аналогичного подхода при работе с приложениями, размещенными на их закрытых облаках. Таким образом, при SaaS ERP-система принадлежит и управляется вендором, в то время как клиент оформляет подписку и осуществляет оплату на регулярной основе. Пользователи получают выгоду из простоты использования и быстрого предоставления приложений, поставленных на открытое облако. Информационные технологии могут использовать частное облако для размещения и поставки внутренних приложений, чтобы удовлетворить потребности его внутренних бизнес-пользователей. Как правило, SaaS обычно поставляется в мультиарендной среде, предоставляя выравнивание нагрузки и сервис самообслуживания. Следовательно, многочисленные клиенты совместно с другими пользователями и компаниями делят физическую вычислительную среду. Реализация каждого пользователя выполнена отдельно от остальных клиентов. Существенным преимуществом распространения приложений, поставляемых через облако, является отсутствие ответственности потребителя за обновление программного обеспечения и обслуживание приложения. Однако в отличие от традиционного подхода «на месте», у пользователя нет бесконечной лицензии на приложение и вместо этого оплата происходит ежемесячно или ежегодно (или в некоторых случаях — за каждого пользователя). Таким образом, провайдер целиком и полностью несет ответственность за все решение, в том числе поддержание и улучшение

осуществляет техподдержку, отвечает за конфиденциальность данных клиента. Со стороны последнего требуется наличие подключения к сети Интернет, в то время как остальное программное и аппаратное обеспечение запущены поставщиком. Игрет роль мобильность — клиент имеет возможность получать доступ к приложению с мобильных устройств. Из сказанного выше следует, что перенос ERP-систем в облачную среду представляется не просто вполне оптимальным, но и достаточно эффективным и грамотным ходом, позволяющим свести к минимуму затраты на оборудование и расширить доступные функции рядом дополнительных возможностей.

Гибридное облако — это среда, совмещающая традиционные информационные технологии с комбинацией общедоступных, частных или управляемых облачных сервисов. В сущности, гибридное облако становится виртуальной вычислительной средой, объединяющей службы в открытом облаке со службами полученных от комбинации сред, чтобы обеспечить подходящий уровень обслуживания и гибкость для удовлетворения поступающих потребительских запросов.

Поскольку данная модель объединяет в себе принципы, используемые в двух вышеназванных методах, следовательно, преимущества и недостатки гибридного варианта одинаковы. Также к первым добавляются:

- возможность распределить нагрузку между облачным хостингом и SaaS, чтобы использовать для каждой задачи свою технологию, повышая производительность работы;
- при наступлении чрезвычайного положения возможность быстро и эффективно мигрировать от одного способа установки к другому.

Недостатки включают в себя увеличение затрат — по сути, предприятие одновременно использует оба способа, большее время установки, а также возрастание угроз, связанных с безопасностью данных.

В сфере разработки программного обеспечения под многозвенной или многослойной архитектурой понимается клиент-серверная архитектура, в которой функции представления, обработки приложений и управления данными физически разделены. Наиболее широкое применение получила трехзвенная архитектура. Кроме обычных преимуществ модульного программного обеспечения с четко определенными интерфейсами, трехзвенная архитектура предназначена для того, чтобы позволить любому из трех звеньев быть доработанным или замененным отдельно в ответ на изменения в требованиях или технологии.

Выбор оптимального способа построения облачной ERP

У всех трех подходов есть свои уникальные преимущества. Прежде чем решить, какой подход оптимален для конкретного бизнеса, необходимо учесть текущую инфраструктуру предприятия, схему организации техподдержки, численность и кадровый состав ИТ-служб, возможности и надежность корпоративной сети, местоположение клиентских компьютеров и имеющееся программное обеспечение, используемое для других бизнес-потребностей. Также важную роль играет бюджет, поскольку не каждая модель установки одинаково экономически выгодна в различных ситуациях. Сравнение облачных способов построения ERP-систем представлено в табл. 1.

При виртуальном хостинге, если рассчитывать по порядку потока оплат, организации необходимо для начала оплатить значительную сумму за предоставленные лицензии, затем оплатить работы по внедрению (по нормочасам). В ряде случаев также потребуются авансовая оплата хостеру за первоначальное предоставление ресурсов. Не следует забывать и о ежемесячной оплате самих услуг.

Несомненным преимуществом данного подхода является тот факт, что предприя-

тие-клиент владеет программным обеспечением и обладает лицензией, не ограниченной по времени использования. Таким образом, оплата происходит только один раз, любые колебания курсов валют не смогут в дальнейшем повлиять на стоимость владения лицензиями.

Однако у данного метода есть и недостатки, которые могут оказаться существенными и при определенных условиях сыграть решающую роль. Например, ежемесячные затраты на хостинг могут превысить стоимость организации работы «на месте» в долгосрочной перспективе. В случае сбоя в работе Интернета доступ к ERP-системе утрачивается. Разумеется, снизить влияние таких рисков возможно, используя резервированные каналы связи различных провайдеров, но это также ведет к удорожанию проекта в целом. Клиенту важно обеспечить высокую пропускную способность, особенно в случае, если приложение работает с отсканированными изображениями и фотографиями, так как их загрузка и отображение способны существенно снизить пропускную способность сети.

SaaS-решения обладают очень привлекательной начальной ценой при одновременном отсутствии существенных начальных затрат, а приложениям для нескольких клиентов требуется для поддержания сравнительно низкая оплата, так как используются общие информационные ресурсы. Регулярные платежи уже включают в себя затраты на внедрение и улучшение программного обеспечения. Данная модель имеет самую высокую оперативность развертывания, что может сыграть роль в ситуациях, когда важна скорость внедрения решений. Кроме того, ответственность за поддержание работоспособности всего решения полностью лежит на провайдере.

Зачастую регулярные платежи за использование облачных решений имеют тенденцию к росту [4, 8]. Причинами этого являются, с одной стороны, расширение возмож-

Таблица 1. Сравнение облачных способов построения ERP-систем

Table 1. Comparison of cloud methods ERP-system construction

Параметр \ Модель	Виртуальный хостинг	SaaS	Гибридное облако
Владение ПО	Клиент	Провайдер	Клиент
Затраты на развертывание	Средние	Низкие	Высокие
Затраты на эксплуатацию	Низкие	Средние	Высокие
Потенциал модернизации	Средний	Низкий	Высокий
Возможности масштабирования	Низкие	Высокие	Средние
Требуемое обеспечение для рабочих мест	Рабочая станция или тонкий клиент	Соединение с Интернетом и веб-браузер	Тонкий клиент, или веб-браузер, или рабочая станция
Требуемый ИТ персонал и навыки	Максимальное количество. Высокий уровень	Минимальное количество. Начальный уровень	Среднее количество. Средний уровень
Защищенность критической информации	Средняя	Низкая	Высокая
Гибкость	Средняя	Низкая	Высокая

ностей «облачных» сервисов, с другой, — не всегда очевидные модели предоставления услуг, при которых для полноценного функционирования может потребоваться значительный объем «дополнительных» услуг, не всегда очевидных при начальной оценке стоимости. Увеличение стоимости подписки (регулярных платежей) способно принести SaaS-вендору значительную выгоду, поскольку активный клиент, привыкший к конкретной системе и вложивший в нее значительное количество своих средств и времени, будет вынужден продолжать пользоваться ею. С точки зрения потребителя услуг такие риски также необходимо учитывать.

Несмотря на указанные ранее преимущества, SaaS не является идеальным решением для любой организации. Данный подход интересен в первую очередь малым предприятиям и предприятиям с типовой структурой и функциями, в то время как облачный хостинг оптимален для крупных корпораций,

обладающих комплексной и одновременно нетиповой инфраструктурой. Следует отметить, что большинство компаний на рынке не попадает ни в одну из этих категорий. По этой причине адекватная оценка при выборе оптимального решения может оказаться затруднительной.

Зачастую при внедрении новых ERP-систем в организации основное внимание уделяется информационным технологиям, в то время как вопросы переподготовки персонала, его мотивирования к работе с новыми технологиями отходят на второй план [3, 5, 6]. Такой подход чреват значительным превышением сроков внедрения и фактических затрат над плановыми показателями. В подобном случае достичь необходимого результата поможет перенесение действующей ERP-системы на виртуальный хостинг в корпоративное облако (например, с использованием технологии тонкий клиент). В этом случае решаются задачи единства управле-

ния ИТ-ресурсами, снижается трудоемкость поддержки пользователей на рабочих местах, при этом для сотрудников сохраняется привычный пользовательский интерфейс и все технологии взаимодействия. Данный подход в большинстве случаев способен сократить затраты на внедрение и обеспечить бесперебойную работу сотрудников с системой (переключать рабочие места в облако можно поэтапно). Однако в остальном такое решение обладает всеми достоинствами и недостатками модели «на месте».

Гибридное облако позволяет пользоваться возможностями виртуального хостинга для выполнения наиболее важной и критичной рабочей нагрузки, а менее критические функции передавать в SaaS. Данная модель очень подходит для ведения работы с динамичной или высокопеременной нагрузкой, как, например, системы, нагрузка на которые может резко возрасти в определенные периоды времени (сезонность), что позволит с использованием технологии cloud bursting оперативно перемещаться с виртуального хостинга к SaaS и привлекать дополнительные ресурсы. Одно из существенных преимуществ данной модели заключается в низких затратах на ИТ-оборудование. Данный подход позволяет с легкостью перенести локальную ERP-систему в облако, достаточно обеспечить тонкий клиент. Организация получает преимущества одновременного использования технологии клиент-сервер и облачных вычислений, мобильные сотрудники, работающие за пределами предприятия, смогут пользоваться возможностями системы с мобильных приложений, что положительно повлияет на оперативность их работы. В то же время за счет сохранения «традиционных» технологий межсистемного взаимодействия, взаимодействия с периферийными устройствами и специализированными аппаратно-программными системами (измерительными, диагностическими, АСУ ТП и др.) подобные возможности могут оказаться критически важными в ряде ERP-систем — например,

входящих в системы поддержки жизненного цикла сложных изделий [9].

В качестве примера ниже рассмотрены три случая внедрения ERP-систем в трех разных организациях, каждая из которых обладает специфическими характеристиками и потребностями.

Первый случай. Организация малого бизнеса, занимающаяся торговлей и не имеющая собственного производства и специфического оборудования. В связи с расширением деятельности и увеличением ассортимента товаров возникла срочная необходимость автоматизировать процессы продаж, управление, управление рисками, учет затрат и доходов по продуктам и логистические операции. Самым оптимальным вариантом является подписка на облачную ERP-систему по SaaS-модели. Определяющими будут такие параметры, как время развертывания и первоначальный платеж (включая отсутствие затрат на серверное оборудование). Очевидно, что остальные способы внедрения будут значительно более затратными, со значительно большим сроком окупаемости.

Второй случай. Предприятие среднего бизнеса, обладающее собственным производством и поставляющее свою продукцию другим компаниям. Некоторые аспекты являются динамичными, требуется взаимодействие с ИТ-инфраструктурой организации. В организации внедрена ERP-система по модели «на месте», однако собственных вычислительных мощностей стало не хватать в связи с расширением производства, а строительство собственного ЦОД сопряжено со значительными капитальными вложениями. Компания уже владеет необходимыми лицензиями на программное обеспечение, обеспечивающими рост на ближайшую перспективу, и сотрудники полностью обучены работе с существующей системой. Принято решение по переносу ERP-системы на виртуальный хостинг, что в данной ситуации является оптимальным методом создания облачного решения.

Третий случай. Крупная компания, имеющая собственное производство и сеть распространения продукции. У компании сеть филиалов и представительств, в том числе за рубежом. В процессе деятельности ей приходится решать множество разноплановых задач, связанных с различными видами деятельности. Руководством принято решение о долгосрочной инвестиции в гибридное облако, позволяющее объединить на основе одной базы данных единую интегрированную систему предприятия со способностью привлекать дополнительные ресурсы провайдера. Немаловажную роль сыграло наличие в организации ERP-системы, которая внедрена изначально в архитектуре клиент-сервер. Перенос этой системы в архитектуру «тонкий клиент-терминал-сервер-сервер» в значительной мере облегчает процесс внедрения гибридного облака и экономит значительные финансовые и человеческие ресурсы предприятия. При этом сохранены все уникальные технологические процессы, с одновременным обеспечением новыми облачными сервисами руководителей проектов, коммерческих представителей и других сотрудников, чья деятельность требует мобильности.

Заключение

Проведенный анализ показывает, что выбор модели и способа внедрения облачных ERP-решений представляет собой сложную оптимизационную задачу. Для ее успешного решения необходимо провести тщательный анализ потребностей организации, специфики ее деятельности, ИТ-окружения и уровня подготовки персонала. Многие компании находят SaaS подходящим решением исходя из экономических соображений в краткосрочной перспективе и простоте. В то же время другим компаниям необходимы гибридные решения для сохранения отработанных производственных процессов при одновременном обеспечении гибкости процес-

сов, не связанных с производством. Несмотря на сохраняющуюся распространенность, модель «на месте» все больше утрачивает актуальность, поскольку функциональные возможности и надежность облачных технологий постоянно растут, совершенствуются и набирают популярность. Сдерживающим фактором внедрения облачных ERP является наличие у компаний внедренных решений — однако, как показано выше, технологии гибридных облаков позволяют даже этим компаниям использовать облачные решения с сохранением сделанных ранее инвестиций в ERP-продукты.

Список литературы

1. Singh C. D., Kaur H., Khan A. A. An industry view on turning ERP into a competitive advantage. BookRix, 2018. — 48 с.
2. Alistarh D., Delis A., Pallis G. Algorithmic Aspects of Cloud Computing: Third International Workshop. Springer, 2018. — 171 с.
3. Harwood S. ERP: The Implementation Cycle. Routledge, 2017. — 200 с.
4. Bhowmik S. Cloud Computing. Cambridge University Press, 2017. — 426 с.
5. Kapp K. M., Latham W. F., Ford-Latham H. Integrated Learning for ERP Success. CRC Press, 2016. — 368 с.
6. Bradford M. Modern ERP: Select, Implement, and Use Today's Advanced Business Systems. Lulu.com, 2015. — 284 с.
7. Степанов А. А. Автоматизация учета транзакций средствами ERP-систем. ERPACADEMY.RU, 2015. — 52 с.
8. Mahmood Z. Continued Rise of the Cloud: Advances and Trends in Cloud Computing. Springer, 2014. — 410 с.
9. Страхов А. Ф., Страхов О. А. Особенности применения информационных технологий на стадиях жизненного цикла сложных технических систем // Прикладная информатика. 2017. Т. 12. № 3 (69). С. 80–87.

References

1. Singh C. D., Kaur H., Khan A. A. An industry view on turning ERP into a competitive advantage — BookRix, 2018, 48 p.
2. Alistarh D., Delis A., Pallis G. Algorithmic Aspects of Cloud Computing: Third International Workshop — Springer, 2018, 171 p.

3. Harwood S. ERP: The Implementation Cycle — Routledge, 2017, 200 p.
4. Bhowmik S. Cloud Computing — Cambridge University Press, 2017, 426 p.
5. Kapp K. M., Latham W. F., Ford-Latham H. Integrated Learning for ERP Success — CRC Press, 2016, 368 p.
6. Bradford M. Modern ERP: Select, Implement, and Use Today's Advanced Business Systems — Lulu.com, 2015, 284 p.
7. Stepanov, A. A. Automation of transaction accounting by means of ERP-systems — ERPACADEMY.RU, 2015, 52 p.
8. Mahmood Z. Continued Rise of the Cloud: Advances and Trends in Cloud Computing — Springer, 2014, 410 p.
9. Strakhov A., Strakhov O. The Use of Information Technologies at The Stages of The Life Cycle of Complex Technical Systems. *Prikladnaya informatika* — Journal of Applied Informatics, 2017, vol. 12, no. 3 (69), pp. 80–87 (in Russian).

A. Grachev, Moscow University for Industry and Finance «Synergy», Moscow, Russia, g7alexg7@mail.ru

O. Strakhov, Moscow University for Industry and Finance «Synergy», Moscow, Russia, ostrakhov@synergy.ru

Comparative characteristic of methods of cloud ERP-systems construction

The present article is devoted to the analysis of characteristics of methods of installation of cloud ERP-systems. The goal of this work is to develop the algorithm that allows to select most effective method of installation of cloud ERP-systems. This can help the head management of the company in decision making. It also helps provider to improve services. An object of research is process of selection of a method of installation of cloud ERP-systems. With growth of the company the operational load increases, such changes require extension of functionality of enterprise information systems that first of all is reflected in ERP-systems. The problems in processes and control mean a need of transition to cloud ERP. This article provides study of the possibility of transfer of the existing ERP-systems to the cloud environment, comparement of characteristics of different installation methods and contains conclusions regarding the efficiency of installation methods for different specifics of the enterprises. The analysis shows that it is necessary to study needs of the organization and specifics of its business activity first. SaaS suits many companies in terms of simplicity and short term. Hybrid solutions allow some companies to save old production scheme while granting the flexibility of the processes that are not connected to production. The on-premises model loses its relevance as the functional capabilities and reliability of cloud computing constantly grow. The implemented solutions stop companies from using cloud ERP. However, hybrid solutions allow to save investments made into ERP while providing efficient cloud solution.

Keywords: cloud computing, ERP, hybrid cloud, on-premises model, SaaS, implementation model.

About authors:

A. Grachev, *master of applied computer science;*

O. Strakhov, *PhD in Technique*

For citation:

Grachev A., Strakhov O. Comparative characteristic of methods of cloud ERP-systems construction.

Prikladnaya informatika — Journal of Applied Informatics, 2018, vol. 13, no. 4 (76), pp. 84–92.

С. С. Бармина, студентка, Казанский национальный исследовательский технический университет им. А. Н. Туполева – КАИ, molibdenbora@yandex.ru

Ф. М. Таджибаева, студентка, Казанский национальный исследовательский технический университет им. А. Н. Туполева – КАИ, frida.t.1465@gmail.com

М. В. Тумбинская, канд. техн. наук, Казанский национальный исследовательский технический университет им. А. Н. Туполева – КАИ, tumbinskaya@inbox.ru

Корреляционный анализ и прогнозирование SYN-флуд атак

Веб-ресурсы являются неотъемлемой частью жизни современного человека. Так же, как и другие элементы IT-сферы, они подвержены хакерским атакам. Среди лидеров атак можно выделить внедрение операторов SQL и межсайтовое выполнение сценариев. Несмотря на это, DDoS-атаки продолжают входить в топ-10 сетевых атак и приводить к серьезным сбоям работы веб-ресурсов. В статье рассмотрены DDoS-атаки, их классификация и методы защиты. Особое внимание уделено наиболее распространенному типу DDoS-атак — SYN-флуд атак, анализу их временного ряда и прогнозированию.

Ключевые слова: DDoS-атаки, SYN-флуд, прогнозирование, веб-ресурсы, защита информации.

Введение

DoS (Denial of Service) — хакерская атака на вычислительную систему с целью довести ее до отказа. DoS-атака представляет собой генерацию «мусорного» трафика с одного устройства (IP-адреса) на ресурс-жертву. Схема DoS — основа современных кибератак на отказ в обслуживании, при реализации которой не остается юридически значимых улик. DDoS-атаки являются подтипом DoS-атак и применяются там, где последние могут оказаться неэффективны. Атаки при этом осуществляются с нескольких компьютеров сети, которые объединяются, и каждый производит атаку на систему жертвы. Часто DDoS-атаки реализуют при помощи зараженных специальными программами компьютеров (ботнетов), которые часто называют «компьютерами-зомби».

Авторами работы [1] описывается инцидент, который принято считать первой в мире DDoS-атакой, — атака на IRC-сервер универ-

ситета Миннесоты в 1999 г. Тогда сервер был выведен из строя на несколько дней. DDoS-атаки стали быстро распространяться, и уже к 2000 г. с их помощью была парализована работа компаний *eBay*, *Amazon*, *CNN*, *Yahoo*. Эти инциденты продемонстрировали эффективность атак типа «отказ в обслуживании», вследствие чего скорость DDoS-атак стала стремительно повышаться, а к 2010 г. они выросли до политического уровня. В этот год сайты мировых платежных систем *PayPal*, *Visa* и *Mastercard* были атакованы группой *Anonymous*.

Цель злоумышленников, использующих DDoS-атаки, согласно работе [2] чаще всего заключается в следующем: выразить свой протест политике страны или отдельной компании, тем самым заявив о себе. Однако часто такие атаки прикрывают реальные намерения злоумышленника. Это своего рода отвлекающий маневр: в то время как специалисты ресурса-жертвы пытаются отразить атаку и восстановить нормальную работу сервиса, на самом деле злоумышленники похищают

денежные средства, конфиденциальную информацию или заражают систему для дальнейшего контроля и шпионажа [3–8].

Классификация DDoS-атак

DDoS-атаки можно классифицировать по протоколам и по механизму воздействия.

1. Классификация по протоколам.

В России их чаще всего подразделяют на три большие группы: UDP (User Datagram Protocol), TCP (Transmission Control Protocol), прочие. Такая классификация считается упрощенной, базируется на основных протоколах, которые используют для передачи данных в сети. Уязвимости данных протоколов позволяют хакерам производить атаки. UDP и TCP используются чаще, поэтому атаки с их использованием выделяют в отдельные группы. Категория «прочие» включает в себя атаки на такие протоколы, как ICMP, GRE, IPIP, ESP, AH, SCTP, OSPF, SWIPE, TLSP, Compaq_PEE и т. д. На западе можно встретить иное разделение, в котором выделяют пять типов DDoS-атак: TCP, HTTP, UDP, ICMP, другие.

Подобное разделение позволяет выявлять тенденции, какие протоколы больше подвергаются паразитному трафику, а какие меньше. На основе этих тенденций корректируются стратегии защиты веб-ресурсов, разрабатываются новые алгоритмы фильтрации паразитного трафика.

2. Классификация по механизму воздействия.

Авторы работы [9] приводят разделение DDoS-атак по механизму воздействия на три большие группы. Первая группа — атаки, направленные на переполнение канала связи, то есть забивание полосы пропускания. Интенсивность данного вида атак измеряется в битах в секунду. По сути, это различные типы флуда. Флуд используется с целью создания мощного потока запросов (пакетов данных), который займет собой всю выделенную ресурсу-жертве полосу трафика. Данные атаки воздействуют на канальный

уровень модели OSI. В качестве способа защиты можно применять фильтрацию паразитного трафика на уровне центра обработки данных или специализированных сервисов защиты. Остатки паразитного трафика могут быть удалены с применением аппаратной защиты. В топ-5 данной группы атак входят DNS-амплификация, фрагментированный UDP-флуд, ICMP-флуд, NTP-амплификация, NTP-флуд.

Вторая группа атак использует уязвимости стека сетевых протоколов, воздействуя на сетевой и транспортный уровни модели OSI. DDoS-атаки этой группы фальсифицируют пакеты, которыми засоряют ресурсы оборудования. Таким образом, система становится перегруженной и не способной обработать полезный трафик. Интенсивность атаки измеряется в пакетах в секунду. Эффективно обезопасить систему или ресурс от таких атак можно аппаратной защитой — применяются специальные устройства, отфильтровывающие паразитный трафик. В топ-5 данной группы атак входят: SYN-флуд, IP-null атака, атака поддельными TCP-сессиями, TCP-null атака, атаки с модификациями поля TOS.

Третья группа — это DDoS-атаки, ориентированные на прикладной уровень модели OSI. Они используют уязвимости программного обеспечения и приводят к выходу из приложения или сразу из операционной системы. Интенсивность атаки измеряется в запросах в секунду. Отразить атаки на прикладной уровень довольно сложно, так как они узконаправленные и способны при минимальных затратах ресурсов атакующего создать большие проблемы оборудованию жертвы. Для защиты от атак третьей группы используют программные алгоритмы, направленные на анализ запросов и создания на его основе правил для брандмауэра. В топ-5 данной группы атак входят HTTP-флуд, атака с целью отказа приложения, HTTP-флуд одиночными запросами, атака фрагментированными HTTP-пакетами, HTTP-флуд одиночными сессиями.

SYN-флуд: характеристика и реализация

Среди DDoS-атак особо следует выделить SYN-флуд атаки, так как они представляют наибольшую опасность для работоспособности системы или ресурса. DDoS-атаки типа SYN (или SYN flood-атаки) — атаки, использующие уязвимости стека сетевых протоколов. Во время SYN-флуда на атакуемый сервер с большой скоростью посылаются SYN-запросы, содержащие поддельный IP-адрес источника. SYN-флуд поражает сервер, занимая всю память таблицы соединений (Transmission Control Block (TCB), которую используют для обработки и хранения входящих пакетов. Это вызывает критическое падение производительности и отказ в обслуживании как следствие.

В работе [9] поясняется, что SYN-флуд атаки используют «тройное рукопожатие», которое присуще TCP-соединению, чтобы задействовать все ресурсы атакуемой машины и сделать ее недоступной для окружающих. При реализации SYN-флуда злоумышленник посылает запросы на TCP-соединение с бо-

лее высокой скоростью, чем скорость, с которой ресурс-жертва способна их обрабатывать. Вследствие этого сетевой канал забивается, и конечное оборудование становится недоступным.

Рассмотрим этот процесс подробнее. Существует правило «трехстороннего рукопожатия» (рис. 1). По данному правилу клиент отправляет серверу пакет, содержащий SYN-флаг. Таким образом, сервер уведомляют о начале сеанса. В ответ сервер отправляет SYN-ACK пакет, тем самым подтверждая успешное подключение клиента и открывая обратный порт соединения. Теперь сервер должен получить от клиента ACK-пакет, подтверждающий обратный сеанс и завершающий «рукопожатие».

При реализации SYN-флуда (рис. 2) атакующий отправляет множественные пакеты SYN, не подтверждая обратные запросы сервера, вследствие чего сервер вынужден ожидать подтверждения сеанса, сохраняя соединение в памяти, которая ограничена. При заполнении стека подключения на сервере легитимные клиенты не могут открыть соединение с сервером, ввиду чего сервис стано-

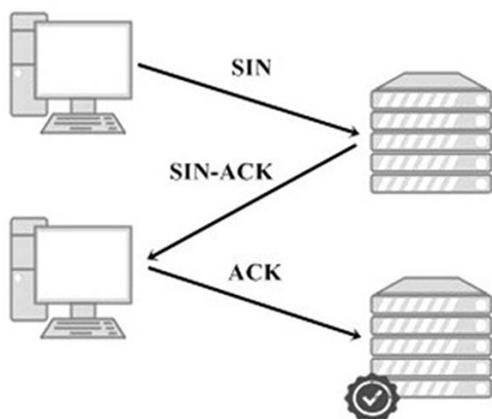


Рис. 1. Схема «трехстороннего рукопожатия»

Fig. 1. The scheme of «three-way handshake»

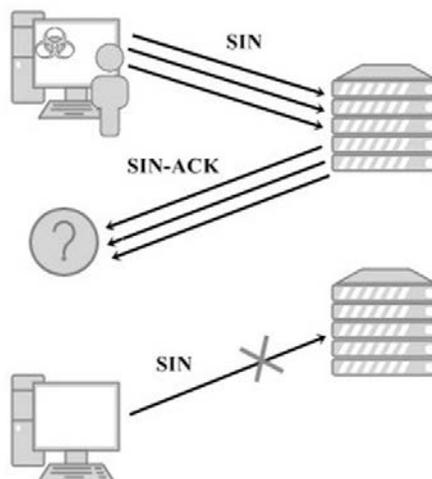


Рис. 2. Схема реализации SYN-флуд атаки

Fig. 2. The scheme of implementation of SYN-flood attack

Таблица 1. Собранные данные о количестве SYN-флуд атак SYN и DDoS-атак длительностью до 4 ч.

Table 1. Collected data on the number of SYN-flood attacks and DDoS-attacks lasting up to 4 hours

Год	Период	SYN-флуд атаки, %	DDoS-атаки длительностью до 4 ч., %
2016	I квартал	54,9	67,8
	II квартал	76	59,8
	III квартал	81,03	68,98
	IV квартал	75,33	67,42
2017	I квартал	48,07	82,21
	II квартал	53,26	85,93
	III квартал	60,4	76,09
	IV квартал	55,63	76,76
2018	I квартал	57,3	80,73

вится недоступным. Говоря проще, при SYN-флуд атаке «крукопожатие» не может быть завершено. Потенциальное соединение помещается в очередь, из которой выйдет лишь через 75 сек. Хакеры пользуются этим свойством, отправляя жертве множество SYN-пакетов каждые 10 сек., что полностью исчерпывает ресурсы системы.

По данным исследований «Лаборатории Касперского» [10–18] с 2016 г., большую часть DDoS-атак составляют атаки типа SYN (от 54 до 82% ежеквартально), а наиболее распространенными по длительности являются DDoS-атаки длительностью до 4 ч. включительно (от 59 до 86% ежеквартально).

Далее построим и проанализируем временные ряды SYN-флуд атак и DDoS-атак длительностью до 4 ч., на основе которых будет предложен прогноз SYN-флуд атак на второе полугодие 2018 г., а также первое полугодие 2019 г.

Анализ временных рядов SYN-флуд атак и DDoS-атак длительностью до 4 ч.

Для анализа статистические данные «Лаборатории Касперского» о количестве SYN-флуд атак и DDoS-атак длительностью до 4 часов были сведены в табл. 1.

Выполним анализ временного ряда DDoS-атак типа SYN:

1. Проведем корреляционный анализ.
2. Рассчитаем индекс сезонности.

Лаг автокорреляции определим как $n/2$, где n — общее число периодов в исследовании, в данном случае $n = 9$. Значения автокорреляции временного ряда DDoS-атак типа SYN представлены в табл. 2.

Из таблицы 2 видно, что значение корреляции между последовательными значениями ряда невысокое, следовательно, тренд отсутствует. Корреляция между исходным рядом и сдвинутым на пять позиций заметная — 0,6334. Следовательно, в данных присутствует сезонность и период сезонности равен пяти кварталам. Расчет и анализ индекса

Таблица 2. Автокорреляция для временного ряда SYN-флуд атак

Table 2. Autocorrelation for the time series of SYN-flood attacks

Лаг	Автокорреляция
x-0	1
x-1	0,3718
x-2	-0,2514
x-3	-0,4335
x-4	0,3907
x-5	0,6334

сезонности для DDoS-атак типа SYN представлен в табл. 3.

Таблица 3. Индексы сезонности для SYN-флуд атак

Table 3. Seasonal indices for SYN-flood attacks

Период \ Год наблюдений	2016	2017	2018	Индекс сезонности
I квартал	54,9	48,07	57,3	1,1409
II квартал	76	53,26	—	0,9201
III квартал	81,03	60,4	—	1,0068
IV квартал	75,33	55,63	—	0,9322

В случае если индекс сезонности превышает единицу, присутствует влияние сезонного фактора в сторону увеличения уровней динамического ряда, в противном случае сезонный фактор вызывает снижение уровней динамического ряда [19]. Анализ показал, что в I и III кварталах ожидается наибольшее количество SYN-флуд атак. Аналогично выполним анализ временного ряда DDoS-атак длительностью до 4 ч. Результаты автокорреляции приведены в табл. 4.

Таблица 4. Автокорреляция для временного ряда DDoS-атак длительностью до 4 ч.

Table 4. Autocorrelation for time series of DDoS-attacks lasting up to 4 hours

Лаг	Автокорреляция
x-0	1
x-1	0,5502
x-2	0,2582
x-3	0,0101
x-4	-0,3254
x-5	0,4628

Из таблицы видно, что значение корреляции между последовательными значениями ряда высокое, следовательно, тренд присут-

ствует. Корреляция между исходным рядом и сдвинутым на пять позиций умеренная — 0,4628. Следовательно, в данных может присутствовать сезонность. Рассчитаем и проанализируем индекс сезонности (табл. 5).

Таблица 5. Индексы сезонности для DDoS-атак длительностью до 4 ч.

Table 5. Seasonal indices for DDoS-attacks lasting up to 4 hours

Период \ Год наблюдений	2016	2017	2018	Индекс сезонности
I квартал	67,8	82,21	80,73	1,3864
II квартал	59,8	85,93	—	0,8756
III квартал	68,98	76,09	—	0,8717
IV квартал	67,42	76,76	—	0,8663

Анализ показал, что в I квартале следует ожидать максимальное количество атак длительностью до 4 ч.

Для выявления корреляционной зависимости рассчитаем взаимную корреляционную функцию, допуская запаздывание длительности атак по отношению к количеству SYN-флуд атак. Результаты кросс-корреляции приведены в табл. 6.

Таблица 6. Кросс-корреляция ряда атак типа SYN и ряда атак длительностью до 4 ч.

Table 6. Cross-correlation of a number of SYN-flood attacks and a number of attacks lasting up to 4 hours

Лаг	Кросс-корреляция
x-0	-0,75855
x-1	-0,29303
x-2	0,28405
x-3	0,54873
x-4	-0,29211
x-5	-0,91739

Как видно, корреляционная связь между переменными ряда атак типа SYN и ряда

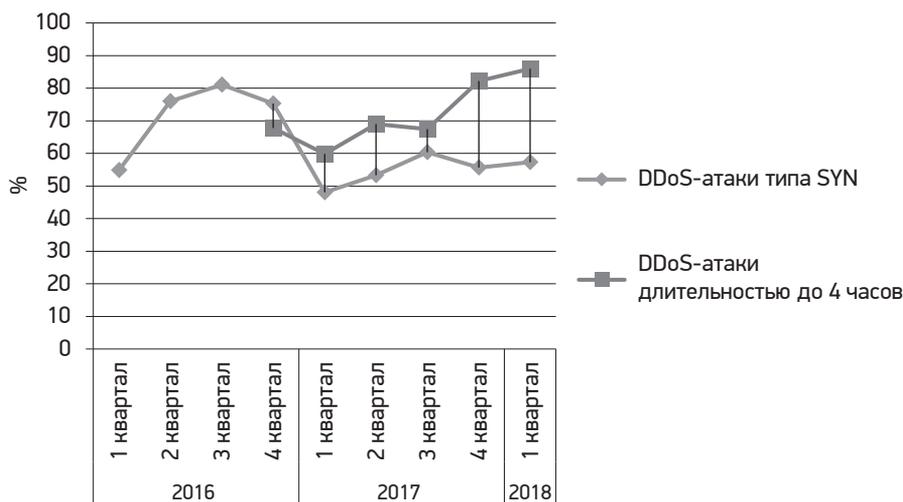


Рис. 3. Тенденция временных рядов с учетом лага
 Fig. 3. The trend of time series taking into account the lag

длительности атак умеренно сильна при временном запаздывании на три квартала. Тенденция временных рядов с учетом лага в три квартала для временного ряда длительности атак представлена на рис. 3.

Корреляционный анализ показал, что временные ряды имеют схожую тенденцию. Связь между SYN-флуд атаками и атаками длительностью до 4 ч. умеренно сильная, не критичная, прослеживается при запаздывании длительности на три временных лага (в нашем случае — три квартала).

Прогнозирование SYN-флуд атак по методу простого экспоненциального сглаживания

При методе простого экспоненциального сглаживания применяется следующая формула экспоненциального сглаживания [20]:

$$S_t = \begin{cases} Y_t, & t = 1 \\ \alpha Y_t + (1 - \alpha)S_{t-1}, & t > 1 \end{cases}, \quad (1)$$

где t — период расчета, S_t — сглаженное значение (экспоненциальное среднее — это прогнозная величина на период $t + 1$), Y_t — значение временного ряда на период t , α — коэффициент сглаживания.

Коэффициент сглаживания выбирается субъективно из диапазона $\alpha \in (0,1)$. Существует подход к выявлению оптимального коэффициента по уже известным данным.

Найдем оптимальный коэффициент сглаживания α . Значения α разобьем в сетку с шагом 0.1 и посчитаем в процентах среднюю абсолютную ошибку между сглаженными прогнозными значениями (S_t) и реальными наблюдениями (Y_{t+1}) при $\alpha = \overline{0.1, 0.9}$. Затем выберем значение коэффициента, при котором ошибка минимальная.

Посчитаем среднюю абсолютную ошибку при $\alpha = 0.1$. Для этого следует посчитать сглаженные значения по формуле (1). Как первое сглаженное значение авторы [20] предлагают использовать первое наблюдение ($S_1 = Y_1 = 54.9$).

При $t = \overline{2,9}$ получится:

$$\begin{aligned} S_2 &= \alpha Y_2 + (1 - \alpha)S_1 \approx 0.1 \times 76 + 0.9 \times 54.9 \approx 57.01 \\ S_3 &= \alpha Y_3 + (1 - \alpha)S_2 \approx 0.1 \times 81.03 + 0.9 \times 57.01 \approx 57.41 \\ &\dots \\ S_8 &= \alpha Y_8 + (1 - \alpha)S_7 \approx 0.1 \times 55.63 + 0.9 \times 59.2 \approx 58.84 \end{aligned}$$

Подсчет ошибок представлен в табл. 7:
Средняя абсолютная ошибка в процентах равна:

$$MAPE = \frac{1}{N} \sum_{t=1}^N \frac{|Y_{t+1} - S_t|}{Y_{t+1}} \cdot 100\% \approx \frac{1.289}{8} \cdot 100\% \approx 16.1\%.$$

Аналогично посчитаем сумму отклонений при $\alpha = 0.2, 0.9$. Результаты представлены в табл. 8.

Из результатов сделаем вывод, что наименьшая ошибка получается при $\alpha = 0.8$, а следовательно, такое значение коэффициента обеспечит прогноз лучше, чем другие значения диапазона [0.1, 0.9] с шагом 0.1.

Посчитаем прогнозную величину количества DDoS-атак на II квартал 2018 г. с помощью формулы (1):

$$\begin{aligned} S_2 &= \alpha Y_2 + (1-\alpha)S_1 \approx 0.8 \times 76 + 0.2 \times 54.9 \approx 71.78 \\ S_3 &= \alpha Y_3 + (1-\alpha)S_2 \approx 0.8 \times 81.03 + 0.2 \times 71.78 \approx 79.18 \\ &\vdots \\ S_9 &= \alpha Y_9 + (1-\alpha)S_8 \approx 0.8 \times 55.63 + 0.2 \times 56.3 \approx 57.1 \end{aligned}$$

Полученные сглаженные значения представлены в табл. 6.

Следовательно, высчитанное методом простого экспоненциального сглаживания количество SYN-флуд атак во II квартале 2018 г. составит 57,1%. Проведем аналогичные вычисления для прогноза на III и IV кварталы 2018 г., I и II кварталы 2019 г.

Таким образом, во II квартале 2018 г. ожидается 57,1% DDoS-атак типа SYN, в III и IV кварталах 2018 г. их количество составит, соответственно, 55,35% и 57,12%, соответственно. В I квартале 2019 г. ожидается 58,73% SYN-флуд атак, во II квартале 2019 г. — 57,08%.

Таблица 7. Подсчет ошибок при коэффициенте сглаживания $\alpha = 0.1$

Table 7. Counting errors with a smoothing factor $\alpha = 0.1$

Период, t	1	2	3	4	5	6	7	8
Количество атак, %, Y_{t+1}	76	81,03	75,33	48,07	53,26	60,4	55,63	57,3
Прогноз, %, S_t	54,9	57,01	59,41	61	59,71	59,07	59,2	58,84
$ Y_{t+1} - S_t $	21,1	24	16	12,9	6,45	1,3	3,57	1,54
$ Y_{t+1} - S_t / Y_{t+1}$	0,278	0,3	0,21	0,269	0,121	0,02	0,064	0,027

Таблица 8. MAPE при разных значениях коэффициента сглаживания α

Table 8. MAPE for different values of the smoothing factor α

Значение α	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9
MAPE (%)	16,1	17,16	17,23	16,9	16,3	15,83	15,73	15,4	15,9

Таблица 9. Сглаженные значения при $\alpha = 0.8$

Table 9. Smoothed values for $\alpha = 0.8$

Период, t	1	2	3	4	5	6	7	8	9
Сглаженное значение, S_t	54,9	71,78	79,18	76,1	53,68	53,34	58,99	56,3	57,1

Заключение

Исследование SYN-флуд атак показало, что данный тип атак наиболее опасен: действуя на сетевом уровне и забивая сетевой канал паразитным трафиком, SYN-флуд атака выводит из строя конечное оборудование. Проведенный корреляционный анализ временного ряда SYN-флуд атак показал, что им присуща сезонность. Для DDoS-атак длительностью до 4 ч. также выявлена сезонность. Связь между SYN-флуд атаками и атаками длительностью до 4 ч. умеренно сильная, не критичная, прослеживается при запаздывании длительности на три временных лага (в нашем случае — три квартала).

На основе проведенного исследования предложены следующие механизмы защиты от SYN- SYN-флуд атак:

1) Timeout. Полуоткрытые соединения по прошествии некоторого времени выбрасываются из буфера. При истощении буфера запросы клиентов на подключение будут проходить с вероятностью $C1/C2$, где $C1$ — количество SYN-пакетов от клиента, $C2$ — количество SYN-пакетов от всех остальных (включая атакующего).

2) Безлимитный буфер полуоткрытых соединений. При нагрузке на канал атакующего 100 Мб/сек и timeout около минуты очередь полуоткрытых соединений будет занимать примерно 1 Гб памяти, что для крупных серверов не критично. Побочный эффект — атакуемый сервер отвечает трафиком, в 3 раза большим, чем трафик атакующего (происходит процесс DDoS с умножением в 4 раза), что может привести к истощению пропускной способности канала. Однако при невозможности истощить ширину канала защита от атаки будет абсолютной, ни одно клиентское соединение не будет отвергнуто.

3) Очистка ранних полуоткрытых соединений. При переполнении буфера из него удаляется самое раннее полуоткрытое соединение. Побочный эффект — если при атаке буфер заполняется за время t , то клиент не смо-

жет подключиться во время атаки, если время подтверждения соединения больше t — его запрос тоже будет выброшен. Увеличивая размер буфера, защиту можно свести к предыдущему варианту.

4) SYN COOKIE. После истощения буфера информация, не помещающаяся в буфер, отправляется клиенту, который якобы запросил ее. Если клиент — настоящий, то он возвращает информацию обратно, если поддельный — она теряется, причем механизм реализован в рамках RFC по TCP, т.е. его поддерживают и клиенты, не знакомые с этой технологией. Операционная система с SYN COOKIE, независимо от размера буфера полуоткрытых соединений, совершенно неуязвима для SYN-флуд атак. Побочный эффект — запрет «больших окон».

Предложенное в статье исследование позволит повысить уровень информационной безопасности веб-ресурсов, предотвратить попытки злоумышленников реализации SYN-флуд атак, обеспечив надежную и адекватную защиту веб-ресурсов.

Список литературы

1. Харитонов В. С., Черяпкин Д. П. DDoS-атака: классификация и особенности // Постулат. 2016. №12 (14). С. 45.
2. Бондаренко М. С. Обзор методов и инструментов для реализации распределенных атак отказа в обслуживании // Вестник Воронежского института высоких технологий. 2017. №4 (23). С. 59–63.
3. Жуков Ю. В. Основы веб-хакинга: нападение и защита. СПб.: Питер, 2012. — 208 с.
4. Бирюков А. А. Информационная безопасность: защита и нападение. М.: ДМК Пресс, 2012. — 474 с.
5. Тумбинская М. В. Модель защищенной информационной системы интернет-банкинга // Прикладная информатика. 2015. Т. 10. №5. С. 62–72.
6. Тумбинская М. В. Защита информации в социальных сетях от социоинженерных атак злоумышленника // Прикладная информатика. 2017. Т. 12. №3 (69). С. 88–102.
7. Тумбинская М. В. Информационная поддержка при обеспечении защищенности систем интернет-банкинга // Национальные интересы: приоритеты и безопасность. 2015. Т. 11. №15 (300). С. 48–58.
8. Стюгин М. А. Повышение защищенности информационных систем на основе технологий защиты

- от исследования // Прикладная информатика. 2017. Т. 12. № 3 (69). С. 103–115.
9. Фролов С. Г., Демин А. Ю. Типы DDoS-атак, методы профилактики и защиты от них // III Международная научная конференция «Информационные технологии в науке, управлении, социальной сфере и медицине». Издательство: Национальный исследовательский Томский политехнический университет (Томск), 2016. С. 76–78.
 10. Лаборатория Касперского. DDoS-атаки в первом квартале 2016 года. URL: <https://securelist.ru/kaspersky-ddos-intelligence-report-for-q1-2016/28429/>.
 11. Лаборатория Касперского. DDoS-атаки во втором квартале 2016 года. URL: <https://securelist.ru/kaspersky-ddos-intelligence-report-for-q2-2016/28961/>.
 12. Купреев О., Штрошнайдер Й., Халимоненко А. DDoS-атаки в третьем квартале 2016 года. URL: <https://securelist.ru/kaspersky-ddos-intelligence-report-for-q3-2016/29506/>.
 13. Халимоненко А., Штрошнайдер Й., Купреев О. DDoS-атаки в четвертом квартале 2016 года. URL: <https://securelist.ru/ddos-attacks-in-q4-2016/30152/>.
 14. Халимоненко А., Купреев О. DDoS-атаки в первом квартале 2017 года. URL: <https://securelist.ru/ddos-attacks-in-q1-2017/30631/>.
 15. Халимоненко А., Купреев О., Ибрагимов Т. DDoS-атаки во втором квартале 2017 года. URL: <https://securelist.ru/ddos-attacks-in-q2-2017/79062/>.
 16. Халимоненко А., Купреев О., Ильганаев К. DDoS-атаки в третьем квартале 2017 года. URL: <https://securelist.ru/ddos-attacks-in-q3-2017/87918/>.
 17. Халимоненко А., Купреев О., Ильганаев К. DDoS-атаки в четвертом квартале 2017 года. URL: <https://securelist.ru/ddos-attacks-in-q4-2017/88505/>.
 18. Халимоненко А., Купреев О., Бадовская Е. DDoS-атаки в первом квартале 2018. URL: <https://securelist.ru/ddos-report-in-q1-2018/89700/>.
 19. Подкорытова О. А., Соколов М. В. Анализ временных рядов: учебное пособие для бакалавриата и магистратуры. М.: Издательство Юрайт, 2018. — 267 с.
 20. Тарджумян А. А. Прогнозирование по методам простого и двойного экспоненциального сглаживания // Молодежный научно-технический вестник. 2015. №3. С. 39.
 3. Zhukov Yu. V. *Osnovy veb-khakinga: napadenie i zashchita* [The basics of web hacking: attack and defense]. Saint-Petersburg, SPb Publ., 2012, 208 p.
 4. Biryukov A. A. *Information security: protection and attack*. Moscow, DMK Press Publ., 2012, 474 p.
 5. Tumbinskaya M. V. The model of the protected information system of Internet banking. *Prikladnaya informatika* — Journal of Applied Informatics, 2015, vol. 10, no. 5, pp. 62–72.
 6. Tumbinskaya M. V. Protection of information in social networks from social attack of an attacker. *Prikladnaya informatika* — Journal of Applied Informatics, 2017, vol. 12, no. 3 (69), pp. 88–102.
 7. Tumbinskaya M. V. *Informatsionnaya podderzhka pri obespechenii zashchishchennosti sistem internet-bankinga* [Information support while ensuring the security of Internet banking systems]. *Natsional'nye interesy: priority i bezopasnost'* — National Interests: Priorities and Security, 2015, vol. 11, no. 15 (300), pp. 48–58.
 8. Styugin M. A. Increase the security of information systems based on technology protection from research. *Prikladnaya informatika* — Journal of Applied Informatics, 2017, vol. 12, no. 3 (69), pp. 103–115.
 9. Frolov S. G., Demin A. Yu. *Tipy DDoS-atak, metody profilaktiki i zashchity ot nikh* [Types of DDoS-attacks, methods of prevention and protection from them]. *Materialy III Mezhdunarodnoi nauchnoi konferentsii «Informacionnye tekhnologii v nauke, upravlenii, social'noi sfere i medicine»* [Proc. 3rd Sci. Conf. «Information Technologies in Science, Management, Social Sphere and Medicine»]. Tomsk, 2016, pp. 76–78.
 10. Laboratoriya Kasperskogo. *DDoS-ataki v pervom kvartale 2016 goda* [DDoS-attacks in the first quarter of 2016]. Available at: <https://securelist.ru/kaspersky-ddos-intelligence-report-for-q1-2016/28429/> (accessed 18.07.2018).
 11. Laboratoriya Kasperskogo. *DDoS-ataki vo vtorom kvartale 2016 goda* [DDoS-attacks in the second quarter of 2016]. Available at: <https://securelist.ru/kaspersky-ddos-intelligence-report-for-q2-2016/28961/> (accessed 18.07.2018).
 12. Kupreev O., Shtroshnaider I., Khalimonenko A. *DDoS-ataki v tret'em kvartale 2016 goda* [DDoS-attacks in the third quarter of 2016]. Available at: <https://securelist.ru/kaspersky-ddos-intelligence-report-for-q3-2016/29506/> (accessed 18.07.2018).
 13. Khalimonenko A., Shtroshnaider I., Kupreev O. *DDoS-ataki v chetvertom kvartale 2016 goda* [DDoS-attacks in the fourth quarter of 2016]. Available at: <https://securelist.ru/ddos-attacks-in-q4-2016/30152/> (accessed 18.07.2018).
 14. Khalimonenko A., Kupreev O. *DDoS-ataki v pervom kvartale 2017 goda* [DDoS-attacks in the first quarter of 2017]. Available at: <https://securelist.ru/ddos-attacks-in-q1-2017/30631/> (accessed 18.07.2018).

References

1. Kharitonov V. S., Cheryapkin D. P. *DDoS-ataka: klasifikatsiya i osobennosti* [DDoS attack: classification and features]. *Postulat*, 2016, no. 12 (14), p. 45.
2. Bondarenko M. S. *Obzor metodov i instrumentov dlya realizatsii raspredelennykh atak otказа v obsluzhivanii* [Overview of methods and tools for implementing distributed denial of service attacks]. *Vestnik Voronezhskogo instituta vysokikh tekhnologii*, 2017, no. 4 (23), pp. 59–63.

15. Khalimonenko A., Kupreev O., Ibragimov T. *DDoS-ataki vo vtorom kvartale 2017 goda* [DDoS-attacks in the second quarter of 2017]. Available at: <https://securelist.ru/ddos-attacks-in-q2-2017/79062/> (accessed 18.07.2018).
16. Khalimonenko A., Kupreev O., Il'ganaev K. *DDoS-ataki v tret'em kvartale 2017 goda* [DDoS-attacks in the third quarter of 2017]. Available at: <https://securelist.ru/ddos-attacks-in-q3-2017/87918/> (accessed 18.07.2018).
17. Khalimonenko A., Kupreev O., Il'ganaev K. *DDoS-ataki v chetvertom kvartale 2017 goda* [DDoS-attacks in the fourth quarter of 2017]. Available at: <https://securelist.ru/ddos-attacks-in-q4-2017/88505/> (accessed 18.07.2018).
18. Khalimonenko A., Kupreev O., Badovskaya E. *DDoS-ataki v pervom kvartale 2018 goda* [DDoS-attacks in the first quarter of 2018]. Available at: <https://securelist.ru/ddos-report-in-q1-2018/89700/> (accessed 18.07.2018).
19. Podkorytova O. A., Sokolov M. V. *Analiz vremennykh ryadov: uchebnoe posobie dlya bakalavriata i magistratury* [Time series analysis: a textbook for undergraduate and graduate programs]. Moscow, Yurait Publ., 2018, 267 p.
20. Tardzhumanyan A. A. *Prognozirovaniye po metodam prostogo i dvoynogo eksponentsial'nogo sglazhivaniya* [Forecasting by methods of simple and double exponential smoothing]. *Molodezhnyi nauchno-tehnicheskii vestnik*, 2015, no. 3, p. 39.

S. Barmina, Kazan national technical research university named after A. N. Tupolev — KAI, Kazan, Russia, molibdenbora@yandex.ru

F. Tadzhibaeva, Kazan national technical research university named after A. N. Tupolev — KAI, Kazan, Russia, frida.t.1465@gmail.com

M. Tumbinskaya, Kazan national technical research university named after A. N. Tupolev — KAI, Kazan, Russia, tumbinskaya@inbox.ru

Correlation analysis and forecasting of SYN-flood attacks

DDoS-attacks are included in the top-10 network attacks and lead to serious failures in the work of web-resources. In the paper DDoS-attacks, their classification and methods of protection are considered. Particular attention is paid to the most common type of DDoS-attacks – SYN-flood attacks, correlation analysis of their time series and forecasting. In the paper, the correlation analysis of the time series of SYN-flood attacks is implemented, the coefficient of data autocorrelation and seasonal indices are determined. The forecasting of SYN-flood attacks for the coming quarters of 2018 and 2019 was carried out using simple exponential smoothing. During the investigation, it was revealed that protection against DDoS-attacks should be implemented taking into account the mechanism of DDoS-attack. Depending on it, you should use hardware, software or mixed protection. Investigation of SYN-flood attacks showed that this type of attacks is the most dangerous: acting at the network level and clogging the network channel with parasitic traffic, SYN-flood attack destroys the end equipment. The correlation analysis of the time series of SYN-flood attacks showed that they are seasonally characterized: the greatest number of attacks is expected in the first and third quarters of 2018 and 2019. For DDoS-attacks lasting up to 4 hours, seasonality in the first quarter of the calendar year was also revealed, which means that in the first quarter of 2019 the greatest number of attacks of a given duration should be expected. The relation between SYN-flood attacks and attacks of up to 4 hours is moderately strong, not critical, can be traced when the delay is 3 lag time (in our case, 3 quarters). According to the forecast made using the simple exponential smoothing method, 57.1% of DDoS attacks of the SYN type are expected in the second quarter of 2018, in the III and IV quarters of 2018 their number will be respectively 55.35% and 57.12%. In the first quarter of 2019, 58.73% of SYN-flood attacks are expected, in the II quarter of 2019 – 57.08%.

Keywords: DDoS-attacks, SYN-flood, forecasting, web-resources, information protection

About authors: S. Barmina, *student*; F. Tadzhibaeva, *student*; M. Tumbinskaya, *PhD in Technique*

For citation: Barmina S., Tadzhibaeva F., Tumbinskaya M. Correlation analysis and forecasting of SYN-flood attacks. *Prikladnaya informatika* — Journal of Applied Informatics, 2018, vol. 13, no. 4 (76), pp. 93–102 (in Russian).

Е. А. Федорова, докт. экон. наук, профессор, Финансовый университет при Правительстве Российской Федерации, НИУ ВШЭ, г. Москва, ecolena@mail.ru

О. Ю. Рогов, младший научный сотрудник, Филиал ФГУП «ГосНИИАС» «ЦОД», г. Москва, fintech@gmx.ch

Ф. Ю. Федоров, аспирант, Финансовый университет при Правительстве Российской Федерации, г. Москва, fedorovfedor92@mail.ru

Модели прогнозирования банкротства предприятий: алгоритм ансамбля классификаторов¹

Цель исследования — построение ансамбля классификаторов для прогнозирования банкротства российских предприятий. Эмпирическая база исследования включает 713 торговых компании (334 — банкроты). На основе количественных характеристик ROC-кривых и показателей прогностической способности моделей отбираются наиболее эффективные алгоритмы, формирующие ансамбль классификаторов. Доказана эффективность применения ансамбля классификаторов на основе голосования (точность метода превышает точность других алгоритмов классификации — метод случайных лесов, нейронная сеть, метод опорных векторов, логистическая регрессия и др.). Показано, что добавление макроэкономических факторов улучшает прогностическую способность почти всех методов до 8%.

Ключевые слова: банкротство, классификатор, машинное обучение, прогнозирование, информационные системы, комбинированный алгоритм.

Введение

Финансовые организации, менеджеры инвестиционных фондов, правительства стран и игроки финансовых рынков стремятся разработать модель для эффективной оценки возможности риска дефолта контрагента. Теория и практика современных исследований по теме банкротства компаний направлена на улучшение предиктивной силы моделей банкротства с использованием различных подходов к вычислениям. Методы машинного обучения, применяемые в наиболее актуальных задачах прикладной

математики, позволяют классифицировать объекты и определять значимые признаки (features), отличающие различные группы объектов друг от друга. В настоящее время особого внимания заслуживают интенсивно развивающиеся методы машинного обучения и инструменты, основанные на искусственном интеллекте, которые показывают высокие результаты в задачах оценки кредитных рисков. Поскольку риск неплатежеспособности схож с задачей распознавания образов по признакам, алгоритмы машинного обучения могут быть применены для классификации компаний [1, 2]. Таким образом, происходит развитие традиционно сложившихся моделей, основанных на таких более простых статистических оценках, как дискриминантный анализ или логистическая

¹ Статья подготовлена по результатам исследований, выполненных за счет бюджетных средств по государственному заданию Финансового университета 2017 г.

регрессия. Так, новые альтернативные методы динамично развиваются, предлагается широкий спектр подходов к анализу и решению поставленной задачи. Среди этих подходов, к примеру, метод опорных векторов (Support Vector Machines), предложенный [3]. Другие методы машинного обучения с широкой областью применения включают методы бустинга и бэггинга. Также динамично развиваются методы, использующие искусственные нейронные сети.

Следующим этапом развития методов машинного обучения является формирование комбинаций алгоритмов. Авторы [4] использовали метод ансамблей классификаторов (а именно, бустинг, бэггинг, стэкинг) вместе с базовыми моделями — логистической регрессией, деревьями решений, нейронными сетями и методом опорных векторов. Ряд исследований также выявил сильные и слабые стороны методов в применении к задачам классификации в зависимости от области использования [5–9].

Цель настоящего исследования — построение ансамбля классификаторов с наивысшей прогностической способностью для прогнозирования банкротства российских предприятий на примере отрасли торговли.

Программный комплекс

Существующие коммерческие программные решения для целей бинарной классификации включают встроенные пакеты анализа данных с поддержкой кода на различных языках (C++, C#, Java, Python). С помощью готовых программных решений модели могут быть развернуты и масштабированы в виде веб-службы за короткое время с использованием облачных сервисов. Среди наиболее эффективных систем машинного обучения можно отметить следующие:

Google Cloud Machine Learning — облачный сервис анализа данных, отличительной особенностью которого является использование самых современных фреймворков ма-

шинного обучения и моделирования (PyTorch, xgboost, Keras). Акцент в платформе сделан на упрощение и автоматизацию процесса генерации признаков для обучения моделей. Применяется автоматическая предварительная обработка данных на сервере Google Cloud, модуль HyperTune осуществляет поддержку при выборе параметров моделирования в каждом отдельном алгоритме.

Microsoft Azure ML — студия для разработки и развертывания моделей обладает рядом преимуществ: графический веб-интерфейс позволяет осуществлять работу с рабочей станции или портативного компьютера. Вся вычислительная мощность размещается в облаке Azure, а для развертывания веб-сервисов можно воспользоваться системами балансировки нагрузки. В платформе уже реализованы узлы вычислений для прогнозирования или классификации, включая параметры настройки отдельных моделей, что существенно сокращает подготовительную работу и позволяет сконцентрировать усилия непосредственно на аналитике. Также предусмотрено использование дополнительных модулей на языке Python и R. Настройка оптимизации параметров для отдельных моделей может осуществляться как автоматически, так и задаваться пользователем. В таблице 1 представлено сравнение доступных функций машинного обучения в используемых средах разработки.

В настоящей работе используется среда Azure ML Studio с использованием *Intel Data Analytics Acceleration Library (Intel DAAL)*. Intel DAAL применяется для формирования ансамблей классификаторов методом голосования, построенных в Azure ML Studio, визуализация метрик для построенных моделей, включая ROC-кривые и показатели точности.

Построение алгоритма

На первом этапе делим выборку на обучающую и тестовую. Для обучения базовых классификаторов и ансамбля выборка

Таблица 1. Сравнение доступных функций машинного обучения в используемых средах разработки

Table 1. A comparison of the available machine learning methods in the considered environments

Платформа	Google Cloud	Azure ML	Intel DAAL
Доступные функции	Предустановленные библиотеки. Обучение моделей Составление прогнозов. Статистика. Визуализация	Предустановленные и добавляемые библиотеки. Обучение моделей. Составление прогнозов. Статистика. Шаблоны алгоритмов. Визуализация	Предустановленные библиотеки. Статистика. Поддержка C++. Шаблоны алгоритмов

делится по объему данных на обучающую (30%) и тестовую (70%).

На втором этапе нашего исследования выполняется определение точности стандартных методов на эмпирической выборке показателей.

Для каждого из двух наборов данных выполняется обучение алгоритмов и тестирование с последующим сравнением. В качестве базовых алгоритмов используются: случайный лес (random forest), деревья решений (decision trees), деревья решений (CHAID,

CRT, C5, QUEST), линейный дискриминантный анализ (LDA), методы опорных векторов (LSVM, SVM), нейронные сети (многослойные (MLP) и радиальные RBF-сети) (рис. 1).

Затем на основе ROC-кривых (1) и точности прогнозирования (5) на всех выборках выбираются наиболее эффективные методы для формирования ансамбля классификаторов. Графики ROC позволяют оценить качество в задачах двухклассовой классификации. Графики демонстрируют зависимость True Positive Rate (TPR), чувствительности алгоритма от False Positive Rate, (1 — FPR, специфичность алгоритма) при изменении пороговых значений. В точке (0,0) порог (threshold) минимален, минимальны значения и FPR, TPR. Идеальный случай классификации — кривая графика будет проходить через точку (0,1). Значение площади под кривой I для каждой построенной ROC-кривой дает возможность оценить модели, полученные по обучающей выборке, а затем — на тестовой.

Площадь определяется как доля пар объектов («1» и «0»), которые использует алгоритм верно упорядочил при расчете. Чем ближе значение I к «1», тем более точной принимается модель классификации.

$$I = \sum_{i=1}^A |f_i - f_{i-1}| \cdot \frac{t_i + t_{i-1}}{2}. \quad (1)$$

Третий этап — отбор базовых классификаторов в ансамбль на основе следующих методов: голосованием (2), голосованием со взве-

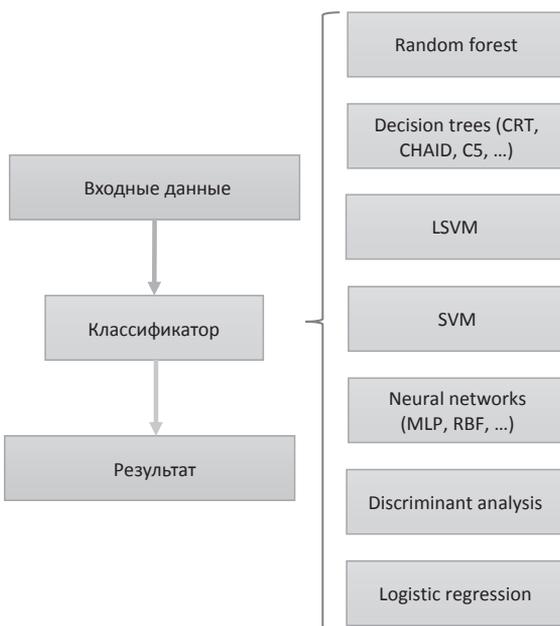


Рис. 1. Набор базовых классификаторов

Fig. 1. A schematic view of the individual classifiers grouped by algorithm types

шенными доверительными вероятностями (3) и интервальных оценок (4) и применение ансамбля классификаторов.

В общем случае для категорий $K = \{\pm 1\}$ и моделей M (базовых классификаторов) справедливы следующие уравнения:

$$\sum_{m=1}^M d_{m,j} = \max_{k=1}^K \left(\sum_{m=1}^M d_{m,k} \right); \quad (2)$$

$$\sum_{m=1}^M p_{m,j} d_{m,j} = \max_{k=1}^K p_{m,k} \left(\sum_{m=1}^M d_{m,k} \right); \quad (3)$$

$$\max_{m=1}^M (p_{m,j}) = \max_{k=1}^K \max_{m=1}^M (p_{m,k}); \quad (4)$$

где $d_{m,k}$ — признак принадлежности к категории.

Базовые классификаторы могут быть ранжированы в соответствии с наибольшим выигрышем в точности или по площади под кривыми ROC. Предпочтение отдается алгоритму, демонстрирующему наибольшую точность и максимальное значение площади под кривой ROC. На основании вычисления

площади под кривой ROC выбираются модели ансамбля классификаторов, использующихся для повторного определения категорий принадлежности компаний к группе банкротов. Максимальный выигрыш $W\%$ определяется как процент верно определенных компаний-банкротов:

$$W\% = \frac{N}{n} \cdot 100\%. \quad (5)$$

Принимаем во внимание характеристики ROC-кривых и расчетной точности и классифицируем модель как успешную. Общая схема решения задачи классификации банкротства компаний представлена на рис. 2.

Результаты проведения эксперимента

В исследовании использовались показатели финансовой отчетности о 713 компаниях раздела G по ОКВЭД (334 — банкроты). Данные отражали сведения об организациях, бухгалтерскую отчетность, показатели, характеризующие ее финансовое состояние, арбитражные данные и сведения о процеду-

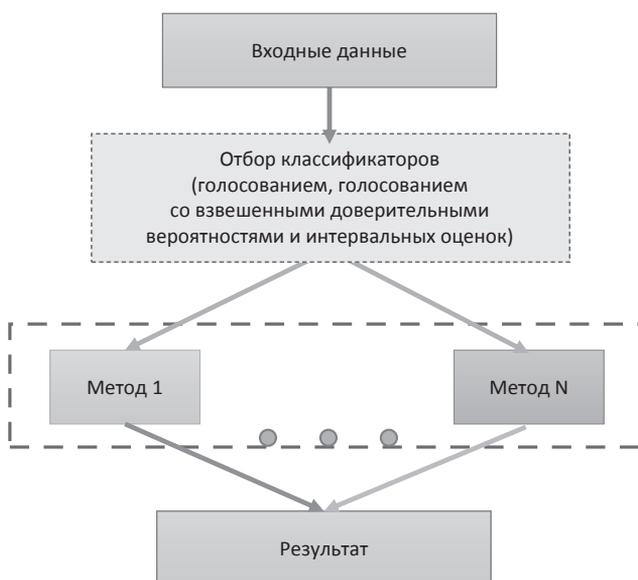


Рис. 2. Формирование ансамбля классификаторов

Fig. 2. A schematic view of the ensemble of classifiers formation procedure

рах финансового оздоровления и банкротства (при наличии). Все необходимые данные были собраны при помощи информационных систем «СПАРК» и «Ruslana», для всех предприятий рассчитаны 12 показателей финансовой устойчивости, ликвидности и платежеспособности, рентабельности, деловой активности. Собраны показатели, характеризующие внешнее влияние среды: ВВП, индекс потребительских цен, уровень безработицы, ключевая ставка ЦБ РФ, курс

доллара США и евро, индекс ММВБ и его прирост.

На первом этапе исследуемая выборка разделена на обучающую (503 объекта) и тестовую (210 объектов). На втором этапе отобраны методы, обладающие наибольшей точностью и площадью под ROC-кривыми, представленные в табл. 2.

В результате отбора простым голосованием по точности и площади под кривой ROC (параметр AUC) были выбраны алгоритмы

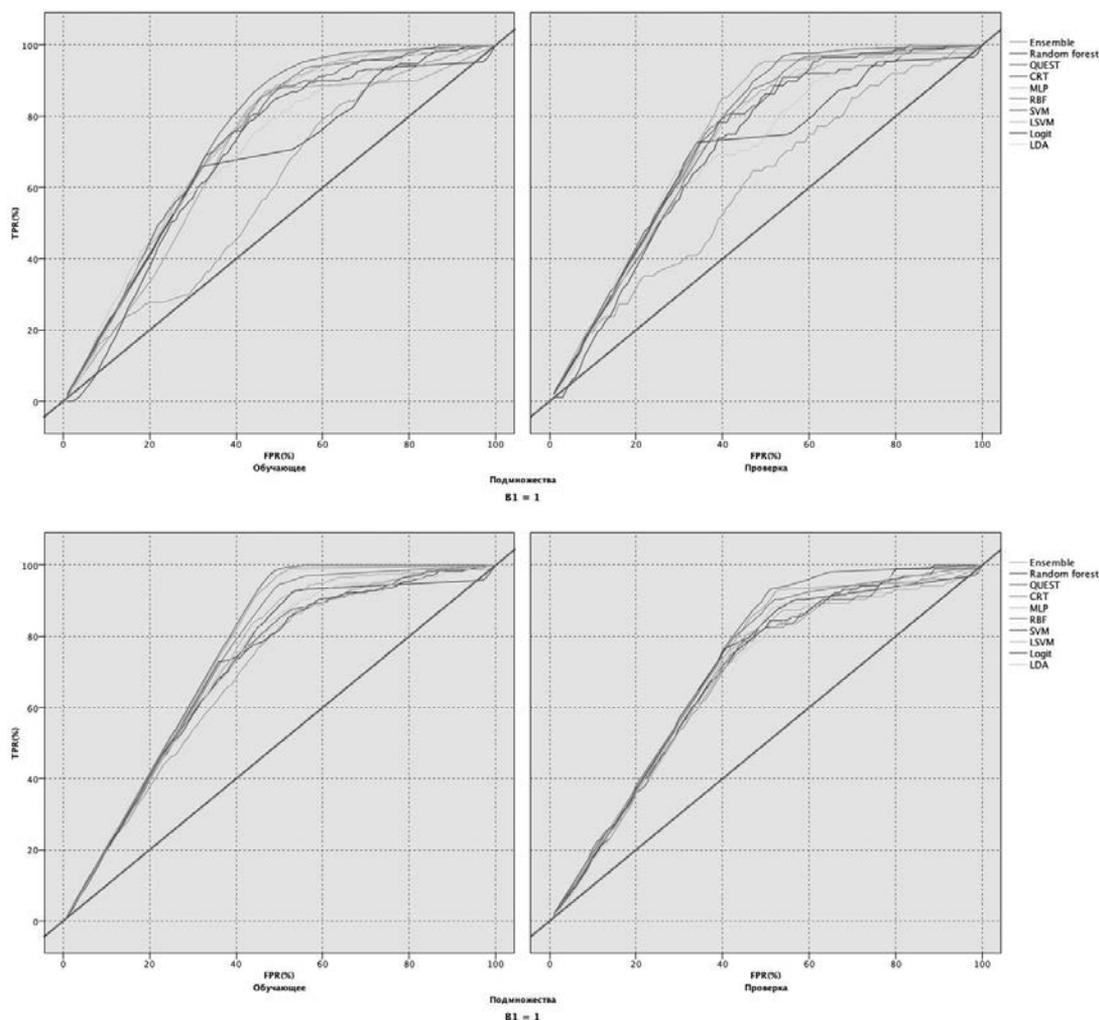


Рис. 3. Сравнение ROC-характеристик трех ансамблей и базовых методов классификации на базе торговых компаний с учетом внешних параметров

Fig. 3. A comparison of ROC-curves for the 3 ensembles and individual classification methods for the trading companies set (with the external parameters included in training)

Таблица 2. Сравнение точности прогнозирования и площади под ROC-кривыми (AUC) для исследуемых методов (тестовая выборка)

Table 2. A comparison of the precision metric and the Area Under ROC-curves (AUC) for the methods (test sample)

Алгоритм	Выборка	Точность, %	AUC
C5	Внешние и внутренние факторы	92,381	0,966
Random forest	Внешние и внутренние факторы	91,429	0,973
QUEST	Внешние и внутренние факторы	90,476	0,957
Logit	Внешние и внутренние факторы	90,000	0,949
CRT	Внешние и внутренние факторы	88,095	0,911
LSVM	Внешние и внутренние факторы	87,619	0,878
CHAID	Внешние и внутренние факторы	87,143	0,934
MLP	Внешние и внутренние факторы	86,667	0,932
SVM	Внешние и внутренние факторы	79,048	0,891
LDA	Внешние и внутренние факторы	78,571	0,876
C5	Внутренние факторы	88,095	0,911
CHAID	Внутренние факторы	85,714	0,915
Random forest	Внутренние факторы	85,238	0,928
QUEST	Внутренние факторы	82,857	0,905
CRT	Внутренние факторы	82,857	0,848
MLP	Внутренние факторы	81,905	0,881
Logit	Внутренние факторы	80,476	0,894
LSVM	Внутренние факторы	78,571	0,797
SVM	Внутренние факторы	75,714	0,774
LDA	Внутренние факторы	69,524	0,749

Таблица 3. Сравнение эффективности ансамблей классификаторов и исследуемых методов (для тестовой выборки)

Table 3. A comparison of the ensembles efficiency for the two data sets (test set)

Модель	Внутренние и внешние факторы		Внутренние факторы	
	Точность, %	AUC	Точность, %	AUC
Ensemble	91,42857	0,975	86,349	0,93
Ensemble 2	89,04762	0,954	85,555	0,914
Ensemble 3	83,17460333	0,883	86,349	0,914
Random forest	91,42857	0,969	85,238	0,924
QUEST	87,61905	0,878	78,571	0,864
CRT	88,09524	0,911	82,857	0,848
MLP	86,66667	0,932	81,904	0,881
SVM	79,04762	0,891	75,714	0,774
CHAID	90,47619	0,957	82,857	0,905
Logit	90	0,949	80,476	0,894
LDA	78,57143	0,876	69,523	0,749

деревьев решений Random forest, C5 и CRT. Для набора данных с внутренними и внешними переменными и для набора только внутренних переменных наилучший результат показал алгоритм C5 с уровнем точности 92,381% и 88,095%, соответственно. Значение AUC для C5 ниже (0,966), чем у Random forest (0,973), Logit (0,949) и QUEST (0,957) для выборки с внешними и внутренними параметрами. Однако алгоритм C5 устойчиво показывает высокую точность по сравнению с другими методами.

На третьем этапе были построены классификаторы с помощью простого голосования («Ensemble»), интервальных оценок («Ensemble 2») или голосования со взвешенными доверительными вероятностями («Ensemble 3»), представленными на рис. 3 для внутренних и внешних переменных (*a, б*) и только внутренних переменных (*в, г*) и табл. 3.

По сравнению с базовыми классификаторами ансамбль классификаторов («Ensemble») демонстрирует более устойчивую работу как на выборке с внешними переменными, так и без них. Данное обстоятельство особенно важно, поскольку алгоритм считается тем эффективнее, чем с большей точностью он предсказывает банкротство.

Наилучшую способность к прогнозированию банкротства показал ансамбль классификаторов, состоящий из деревьев решений и их модификаций. Для выборки с учетом внешних переменных показатели всех базовых классификаторов показывают результат выше на 6–8%. При этом более сложные в реализации методы, такие как многослойные нейронные сети, метод опорных векторов и деревья решений, показывают более высокую стабильность, выражающуюся в небольшом относительном увеличении точности при использовании выборки с внешними показателями.

Устойчивость алгоритма к изменению объема выборки, качества и количества параметров позволяет сделать вывод о том, что

выбранный ансамбль с высокой степенью достоверности (на уровне до 90%) прогнозирует риски неплатежеспособности компаний, в отличие от отдельных базовых классификаторов.

Заключение

Сравнение ансамблей классификаторов, составленных на основе различных базовых методов, позволяет сделать вывод о качестве моделей. Алгоритм «Ensemble», сформированный по результатам простого голосования из модификаций деревьев решений (деревья решений C5, случайный лес (RF), деревья решений CRT), показал наилучший результат как по точности, так и по кривым ROC (см. рис. 3). При составлении ансамбля с помощью интервальных оценок («Ensemble 2») или голосованием со взвешенными доверительными вероятностями («Ensemble 3») прослеживается тенденция к понижению точности — как показали результаты для «Ensemble 2, 3». Данные метрик моделей позволяют сделать вывод о высокой стабильности алгоритмов деревьев решений на смешанной выборке из категориальных признаков и числовых значений.

Полученные результаты исследования могут быть использованы финансовыми аналитиками, сотрудниками банков и инвестиционных компаний для оценки банкротства или кредитоспособности заемщиков.

Список литературы / References

1. Kruppa J., Schwarz A., Arminger G., Ziegler A. Consumer credit risk: Individual probability estimates using machine learning. *Expert Systems with Applications*, 2013, vol. 40 (13), pp. 5125–5131.
2. Pal R., Kupka K., Aneja A., Militky J. Business health characterization: A hybrid regression and support vector machine analysis. *Expert Systems with Applications*, 2016, vol. 49, pp. 48–59.
3. Drucker H., Cortes C., Jackel L. D., LeCun Y., Vapnik V., Boosting and Other Machine Learning Algorithms. *Machine Learning Proceedings 1994*, pp. 53–61.

4. Wang G., Hao J., Jiang H., A comparative assessment of ensemble learning for credit scoring. *Expert Systems with Applications*, 2011, vol. 38 (1), pp. 223–230.
5. Fedorova E., Gilenko E., Dovzhenko S., Bankruptcy prediction for Russian companies: Application of combined classifiers. *Expert Systems with Applications*, 2013, vol. 40 (18), pp. 7285–7293.
6. Cleofas-Sánchez L., García V., Marqués A., Sánchez J. S. Financial distress prediction using the hybrid associative memory with translation. *Applied Soft Computing*, 2016, vol. 44, pp. 144–152.
7. Heo J., Yong Yang J., AdaBoost based bankruptcy forecasting of Korean construction company. *Applied Soft Computing*, 2014, vol. 24, pp. 494–499.
8. Kim M., Kang J., Kim J., Reliable fault diagnosis for incipient low-speed bearings using fault feature analysis based on a binary bat algorithm. *Information Sciences*, 2015, vol. 294, pp. 423–438.
9. Gerlein E. A., McGinnity M., Belatreche A., Coleman S., Evaluating machine learning classification for financial trading: An empirical approach. *Expert Systems with Applications*, 2016, vol. 54, pp. 193–207.

E. Fedorova, Financial University under the Government of the Russian Federation, HSE, ecolena@mail.ru

O. Rogov, Branch of FGUP «GosNIIAS» «Data Center», fintech@gmx.ch

F. Fedorov, Financial University under the Government of the Russian Federation, fedorovfedor92@mail.ru

Models of corporate bankruptcy prediction with the ensemble of classifiers algorithm¹

Classification problems represent the group of machine learning methods where each instance is associated with a certain category or label. An individual classifier like Neural Networks, or Decision Trees is conventionally trained on a pre-marked or processed data set. Depending on the parameters distributions the data sets may feature issues when all the indicators are not learned efficiently by such a classifier, and this results in an inconsistent performance on the test sets. Ensemble classifiers denote a set of individual classifiers algorithms that are simultaneously trained in a classification problem. The paper aim is twofold. We present an ensemble of classifiers approach with a high predictive power for the Russian trade-related companies bankruptcy prediction. At the first stage we split the data into a train set (70%) and a test set (30%). At the second stage the precision of standard algorithms is measured as applied to the empirical indicators of the data. The algorithms are trained and tested, and then compared via the performance metrics. The standard algorithms include: random forest, decision trees and the modifications: the chi-square automatic interaction detection (CHAID), classification and regression trees (CRT, C5), Quick, Unbiased, Efficient, Statistical Tree (QUEST), discriminant analysis LDA, support vector algorithms (LSVM, SVM), neural networks (multilayer and radial). Based on the ROC-curve metrics and the prediction ability of the algorithms we select the most efficient methods that form the ensemble of classifiers algorithm. The empirical data set included 713 trade companies (334 — known bankrupts). The results feature the efficiency of the ensemble of classifiers algorithms based on the simple voting (the precision metric outperforms the one of the other individual algorithms, e.g. random forest, SVM, Logit). We also show that including the macroeconomic factors improves the prediction power of almost all studied algorithms by at least 8%. Given that, more sophisticated variations of the classifiers such as multilayer neural networks and random forests demonstrate higher precision and recall with the external variables employed in the training process.

Keyword: bankruptcy prediction, classifier ensemble, machine learning, information systems, combined algorithm.

About authors: E. Fedorova, *Dr of Economics, Professor*; O. Rogov, *junior researcher*; F. Fedorov, *Postgraduate student*

For citation: Fedorova E., Rogov O., Fedorov F. Models of corporate bankruptcy prediction with the ensemble of classifiers algorithm. *Prikladnaya Informatika — Journal of Applied Informatics*, 2018, vol. 13, no. 4 (76), pp. 103–110 (in Russian).

¹ The authors acknowledge the support provided by the Financial University under the Government of the Russian Federation through budget resources assigned in 2017.

М. И. Дли, докт. техн. наук, зам. директора филиала Национального исследовательского университета «МЭИ» в г. Смоленске, midli@mail.ru

О. В. Булыгина, канд. экон. наук, доцент филиала Национального исследовательского университета «МЭИ» в г. Смоленске, baguzova_ov@mail.ru

П. Ю. Козлов, канд. техн. наук, ассистент филиала Национального исследовательского университета «МЭИ» в г. Смоленске, originaldod@gmail.com

Формирование структуры интеллектуальной системы анализа и рубрицирования неструктурированной текстовой информации в различных ситуациях¹

Анализ электронных текстовых документов, написанных на естественном языке, является одной из важнейших задач, реализуемых в системах автоматизированного анализа лингвистической информации. Известно, что такие документы могут характеризоваться различными параметрами: размер, наличие структуры, частота упоминания ключевых слов и т. п. Проведенный анализ показал невозможность построения единой модели для рубрицирования неструктурированных текстовых документов в различных ситуациях. Для решения указанной проблемы предложен мультимодельный подход к рубрицированию, отличающийся комбинированным использованием интеллектуальных и вероятностно-статистических методов анализа текстовых документов. Конкретная модель выбирается с использованием нечетко-логических алгоритмов на основе предложенных характеристик. Его применение позволит повысить точность отнесения электронных текстовых документов к конкретным рубрикам с учетом их специфики и различных целей практического применения в организации.

Ключевые слова: электронный неструктурированный текстовый документ, рубрицирование, мультимодельный подход, растущие пирамидальные сети, нечетко-логические алгоритмы.

Введение

В настоящее время отмечается активный рост объема текстовой информации в электронной форме как в сети Интернет, так и в глобальных и локальных хранилищах данных и знаний [2, 3].

Проблема обработки электронной текстовой информации относится к числу приоритетных задач, решить которую необходимо для достижения цели перехода к цифровой экономике, сформулированной в Стратегии развития информационного общества в Российской Федерации на 2017–2030 годы. В программе «Цифровая экономика», утвержденной Правительством в июле 2017 г., рассматривается в качестве одной из важнейших

¹ Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта №18-01-00558.

задач «улучшение доступности и качества государственных услуг для граждан».

Разновидностью подобной текстовой информации являются обращения граждан и организаций, передаваемые посредством электронных коммуникаций в муниципальные и федеральные органы государственной власти. Из-за возрастания количества таких сообщений для повышения оперативности и снижения трудоемкости их обработки используется широкий набор средств компьютерного анализа.

Известно, что повышение результативности автоматизированной обработки электронных текстовых документов зависит от точности определения адресата, т. е. решения задачи рубрицирования.

С учетом данной задачи значительная часть интерфейсов систем обработки обращений различных министерств и ведомств обеспечивает возможность самостоятельного выбора рубрик самими обращающимися. Однако в большей части случаев заявитель не обладает необходимым объемом информации о компетенциях различных подразделений или их обращения затрагивают сферу ответственности нескольких структур.

В результате возникает актуальная задача разработки нового подхода к автоматизированной рубрикации поступающих электронных текстовых документов с использованием различных методов, связанных с анализом текстов на естественном языке.

Понятие электронного текстового документа

Очевидно, что выбор метода автоматизированного рубрицирования электронных текстовых документов напрямую зависит от вида поступающего документа.

Ключевым критерием классификации электронных текстовых документов является их структурированность. Так, по данному критерию принято выделять следующие виды документов: структурированные,

слабоструктурированные и неструктурированные [17].

Структурированные электронные текстовые документы имеют фиксированную разметку. Обычно они хранятся в различных базах данных (реляционных, объектных или многомерных) или файлах создаваемых табличных процессоров (например, *Microsoft Excel*).

Слабоструктурированные электронные текстовые документы обладают следующими отличительными признаками [6]:

- документ имеет некоторую разметку;
- содержимое документа разбито внутренним форматированием;
- документ состоит из фрагментов, представляющих собой либо некоторое значение, либо атрибут данных;
- во внутренней разметке документов отсутствуют формальные признаки, указывающие на то, что есть значение данных, а что есть атрибут данных.

Примерами слабоструктурированных электронных текстовых документов являются анкеты, налоговые декларации, страховые формы, прайс-листы, транспортные накладные, типовые договоры, счета.

Документы такого рода целесообразно представлять в виде совокупности объектов с последующей возможной идентификацией текстовых фрагментов как атрибутов и значений данных.

Перечислим основные признаки неструктурированных электронных текстовых документов (ЭНТД) [7, 8]:

- текст написан на естественном языке;
- отсутствие четко определенной структуры;
- невозможно автоматическое выделение структур однозначным образом.

Примерами неструктурированных электронных текстовых документов являются метаданные, сообщения электронной почты, веб-страницы, книги, журналы и иные документы, созданные с помощью текстовых процессоров.

Обычно неструктурированные текстовые электронные документы представляются пользователю с помощью браузеров либо текстовых редакторов (например, *Microsoft Word*).

Как показывают исследования, ежегодно доля неструктурированных электронных текстовых документов растет намного интенсивнее по сравнению с слабоструктурированными и структурированными документами.

С точки зрения объема и с учетом возможности применения различных методов анализа неструктурированные текстовые электронные документы можно разделить на очень короткие, короткие и длинные.

Очень короткие документы — это ЭНТД, объем содержащейся информации в которых не позволяет автоматизировать их рубрицирование с использованием вероятностно-статистических или экспертных методов анализа.

Короткие текстовые документы — это ЭНТД, написанные на естественном языке и содержащие информацию в лингвистической или цифровой форме. Однако их малый объем не позволяет применять известные процедуры вероятностно-статистического анализа текстов, но в то же время допускает использование экспертной информации, полученной в результате комплексирования знаний лингвистов и специалистов в рассматриваемых предметных областях.

Длинные документы — это ЭНТД, допускающие их автоматизированное рубрицирование с использованием различных вероятностно-статистических методов.

Анализ методов рубрицирования неструктурированных ЭНТД

Задачу рубрицирования как разновидность задачи анализа неструктурированных электронных текстовых документов в общем можно сформулировать следующим образом.

Пусть имеется множество объектов (ЭНТД) D и множество классов (рубрик) $H = \{h_i \mid i = 1 \dots N_h\}$, состоящее из N_h клас-

сов. Каждый класс h_i представлен некоторым описанием P_i , имеющим определенную внутреннюю структуру.

Тогда процедура рубрицирования f заключается в выполнении преобразований над объектом $d \in D$, после которых делается вывод либо о принадлежности d к классу h_i , либо о невозможности классификации d в заданных условиях.

Математически модель рубрикатора неструктурированных электронных текстовых документов можно представить следующим образом:

$$R = \langle D, H, P, R_h, f \rangle,$$

где D — множество объектов; H — множество классов (рубрик); P — множество описаний; R_h — отношение на множествах H и P ; f — операция рубрицирования.

В настоящее время при построении в информационных системах анализа электронных текстовых документов, написанных на естественном языке, может использоваться достаточно широкий методический аппарат классификации объектов различного типа.

Для выбора инструментария рубрицирования ЭНТД необходим тщательный анализ особенностей электронных текстовых сообщений, поступающих в различные муниципальные и федеральные органы государственной власти.

Проведенный анализ показал, что такие электронные текстовые документы с точки зрения возможности их автоматизированного анализа обладают следующим набором специфических особенностей [12]:

- в большинстве случаев имеют небольшой размер, что затрудняет его вероятностно-статистический анализ;
- отсутствие структуризации (специальной разметки и полей для компьютерной обработки), что усложняет реализацию процедуры извлечения информации;
- наличие грамматических и синтаксических ошибок в тексте документа, что приво-

дит к необходимости осуществления дополнительных этапов обработки;

- нестационарность тезауруса (состава и важности слов), обусловленная частым изменением нормативной правовой базы и внутренних стандартов организаций, что приводит к необходимости использования процедур динамического выделения (кластеризации) рубрик.

Указанные особенности свидетельствуют о невозможности применения традиционных методов анализа неструктурированных электронных текстовых документов в явном виде.

В этой связи для решения задачи рубрицирования ЭНТД целесообразно использование алгоритмов интеллектуального анализа данных, которое предполагает обучение модели рубрикатора, что подразумевает частичное или полное формирование H , P , R_n и f на основе некоторых априорных данных с заранее известными классами (рубриками).

С точки зрения наличия этапа обучения в модели рубрицирования ЭНТД все методы интеллектуального анализа, применимые для решения данной задачи, можно условно разделить на две группы: с наличием или отсутствием этапа обучения.

Методы рубрицирования ЭНТД с «учителем» предполагают наличие полной информации о рубричном поле (число рубрик и их характеристики). В данном случае в качестве «учителя» используется подготовленная выборка ЭНТД (обучающее множество) с известными степенями принадлежности к той или иной рубрике.

Сегодня методы рубрицирования «с учителем» достаточно широко используются в информационных системах анализа текстовых документов и подходят для решения большого количества возникающих задач. Однако данные методы требуют большого количества предварительной информации для обучения модели рубрицирования.

Методы рубрицирования «без учителя» анализируют коллекцию ЭНТД с целью от-

несения их к рубрикам таким образом, чтобы в каждую рубрику попали документы, наиболее близкие с точки зрения выбранной метрики. В данных методах отсутствует необходимость в первоначальном обучении используемых моделей, так как заранее неизвестны характеристики рубричного поля текстовых документов.

В общем случае алгоритмы рубрицирования «без учителя» (алгоритмы кластеризации) осуществляют группировку ЭНТД в некотором пространстве признаков (параметров документа) с последующей интерпретацией результата. Кластеризация ЭНТД основывается на гипотезе о том, что близкие по смыслу текстовые документы релевантны одним и тем же запросам и выделенным рубрикам.

В работах [9, 10, 18] рассмотрен широкий класс различных модификаций алгоритмов рубрицирования ЭНТД на основе решающих правил, применения искусственных нейронных сетей, растущих пирамидальных сетей и вероятностно-статистических методов.

Как показывает проведенный анализ указанных научных работ, ни один из предлагаемых в них математических и инструментальных методов не позволяет эффективно рубрицировать ЭНТД во всех возможных ситуациях.

В этой связи для решения указанной научной задачи целесообразно использовать несколько алгоритмов интеллектуального анализа, обеспечивающих эффективное рубрицирование ЭНТД с учетом специфики каждого документа.

Критерии выбора метода рубрицирования ЭНТД

Реализация предлагаемого мультимодельного метода рубрицирования ЭНТД предполагает формирование набора критериев, определяющих условия применимости конкретного математического метода интеллектуального анализа текстовых документов.

Так, при выборе математического метода рубрицирования ЭНТД предлагается учитывать следующие признаки:

1. Степень пересечения тезаурусов рубрик, которая зависит от числа уникальных слов для всех рубрик. Наиболее трудным условием рубрицирования является наличие взаимосвязанных рубрик.

Степень пересечения тезаурусов рубрик H_i и H_j определяется следующим образом:

$$K_{\cap} = \frac{1}{J} \cdot \sum_{j=1}^J \sum_{i=1}^J \frac{Count(H_i \cap H_j)}{M_j},$$

где K_{\cap} — степень пересечения тезаурусов рубрик; J — общее количество рубрик; $Count(H_i \cap H_j)$ — количество совпадающих значимых слов в тезаурусах рубрик H_i и H_j ; M_j — общее число значимых слов в тезаурусе рубрики H_j .

2. Объем накопленной статистической информации, который определяет возможность построения конкретной модели рубрицирования ЭНТД.

3. Характеристика тезауруса рубрик (определяет изменчивость состава и степени влияния значимых слов их словарей). Так, некоторые модели рубрицирования не могут применяться в условиях динамически изменяющихся тезаурусов из-за сложности переобучения и перестроения их структуры.

4. Размер ЭНТД, определяемый на основе количества используемых уникальных значимых слов.

В таблице 1 дана характеристика описанных признаков классификации ЭНТД, которые необходимо учитывать при выборе метода рубрицирования ЭНТД.

Типовые ситуации и методы их рубрицирования ЭНТД

В соответствии с указанными в табл. 1 признаками классификации ЭНТД выделены шесть типовых ситуаций, определяющих условия анализа текстовых документов, которые легли в основу выбора конкретного математического метода рубрицирования.

Таблица 1. Дополнительные признаки классификации ЭНТД

Table 1. Дополнительные признаки классификации ЭНТД

Признак	Значение	Описание
Степень пересечения тезаурусов рубрик	Незначительная	$K_{\cap} \in [0; 0,15)$
	Средняя	$K_{\cap} \in [0,15; 0,4)$
	Существенная	$K_{\cap} \in [0,4; 1]$
Объем накопленной статистической информации	Недостаточный	Объем обучающей выборки менее 5000 наблюдений
	Достаточный	Объем обучающей выборки более 5000 наблюдений
Характеристика тезауруса рубрик	Статичный	Состав и степень влияния значимых слов в словарях не изменяется
	Динамичный	Состав и степень влияния значимых слов в словарях изменяется
Размер ЭНТД	Малый	Документ содержит менее 10% уникальных значимых слов
	Средний	Документ содержит 10–20% уникальных значимых слов
	Большой	Документ содержит более 20% уникальных значимых слов

Ситуация 1: анализ малых / средних ЭНТД при средней / существенной степени пересечения рубрик и недостаточности статистических данных.

Предлагаемый метод: нечеткие пирамидальные сети, осуществляющие классификацию объектов «без учителя» в условиях недостатка статистических данных. В отличие от исходного алгоритма [4, 16], данная модификация [14, 22] позволяет учитывать степень связанности рубрик на основе использования нечеткого логического алгоритма, что повышает точность рубрицирования ЭНТД.

Нечеткая пирамидальная сеть представляет собой ациклический ориентированный граф, состоящий из рецепторов (характеристик поступающих ЭНТД), концепторов (промежуточных вершин) и выходов (степень принадлежности к конкретной рубрике).

Отличительной особенностью данной модификации пирамидальных сетей является возможность оценки степени влияния одних факторов на другие (силу связи) и степень их важности в процессе поиска единственного решения (значимость вершины) в условиях квазистатистики [13].

Поступающий ЭНТД представляется в виде множества значимых слов, которые сопоставляются с рецепторами сети. Процесс распознавания предполагает восходящий просмотр сети на основе расчета трех показателей для каждого исхода (кумулятивной возможности возникновения, суммарной значимости и числа возбужденных рецепторов). Исходы (рубрики) выбираются с помощью заданного решающего правила.

Ситуация 2: анализ средних ЭНТД при незначительной / средней степени пересечения рубрик и недостаточности статистических данных.

Предлагаемый метод: модель на основе весовых коэффициентов, учитывающая, помимо статистических характеристик, экспертную информацию в процессе рубрицирования ЭНТД. Указанная модель, в отличие от известных [1, 19], позволяет учитывать

степень значимости слов в ЭНТД в зависимости от появления новых значимых событий, оказывающих влияние на тезаурусы рубрик.

Для построения модели формируется набор ЭНТД с заранее определенными рубриками. Каждый ЭНТД представляется в виде множества значимых слов, при этом каждому слову назначается начальный весовой коэффициент, показывающий его степень соответствия рубрике, к которой относится ЭНТД. Далее проводится подстройка весовых коэффициентов с использованием элементов обучающей выборки, и на выходе формируются словари рубрик.

Процедура применения данной модели начинается с расчета показателя, характеризующего степень соответствия поступающих ЭНТД каждой рубрике. В результате выбираются те рубрики, степень принадлежности к которым является максимальной.

Ситуация 3: анализ средних ЭНТД при средней / значительной степени пересечения рубрик и недостаточности статистических данных.

Предлагаемый метод: нечеткое дерево решений, которое за счет использования дополнительной экспертной информации позволяет повысить точность и оперативность рубрицирования. Данная модель, в отличие от существующих [5], оценивает синтаксические связи и роли слов, а также производит нечеткую оценку различий между документами в n -мерном пространстве признаков текстов при построении и использовании нечеткого дерева решений для отнесения документа к конкретным рубрикам в условиях взаимозависимости их тезаурусов.

Построение нечеткого дерева решений основано на анализе степени пересечений словарей рубрик, а также расстояний между рубриками в n -мерном пространстве признаков.

На вход нечеткого дерева решений поступает ЭНТД в виде множества синтаксических подмножеств значимых слов. Для определения степени принадлежности ЭНТД к рубрикам от корневого узла до листа данное

подмножество сопоставляется с узлами нечеткого дерева решений. Для каждого слова из синтаксических подмножеств высчитывается степень его значимости для группы рубрик. Дополнительно вводится показатель, характеризующий степень соответствия ЭНТД рубрикам, который определяется на основе дополнения относительного евклидова расстояния между нечеткими множествами вида. Считается, что ЭНТД относится к той рубрике, степень принадлежности к которой является максимальной.

Ситуация 4: анализ малых ЭНТД при незначительной степени пересечения рубрик и достаточном объеме и качестве статистических данных (для обучения гибридных нечетких моделей [11, 15, 21]).

Предлагаемый метод: нейронечеткий классификатор на основе известного подхода [20], но с использованием экспертной информации для определения значимости ключевых слов при формализации и последующего рубрицирования ЭНТД.

Каскадная модель на основе нейронечеткого классификатора представляет собой множество частных моделей (трехслойных гибридных нейронечетких сетей), используемых для формирования степени принадлежности ЭНТД к каждой выделенной рубрике.

На вход каскадной нейронечеткой модели поступает множество ЭНТД, которое модифицируется предварительными этапами и синтаксическим парсером в множество из синтаксических подмножеств значимых слов ЭНТД. Для каждого синтаксического подмножества вычисляется оценка его степени близости к каждой рубрике. Полученные оценки подаются на входы нейронечеткой модели для каждой рубрики. Выходы со всех нейронечетких моделей поступают на анализатор, который позволяет определить рубрику, к которой относится ЭНТД.

Ситуация 5: анализ больших ЭНТД при незначительной степени пересечения рубрик и достаточном объеме и качестве статистических данных.

Предлагаемый метод: Байесов классификатор, использующий процедуру отнесения документа к рубрике на основе формулы Байеса для условной вероятности.

Входной ЭНТД d представляется в виде последовательности терминов $\{w_k\}$. Каждая рубрика h_i характеризуется безусловной вероятностью $P(h_i)$ отнесения к ней ЭНТД

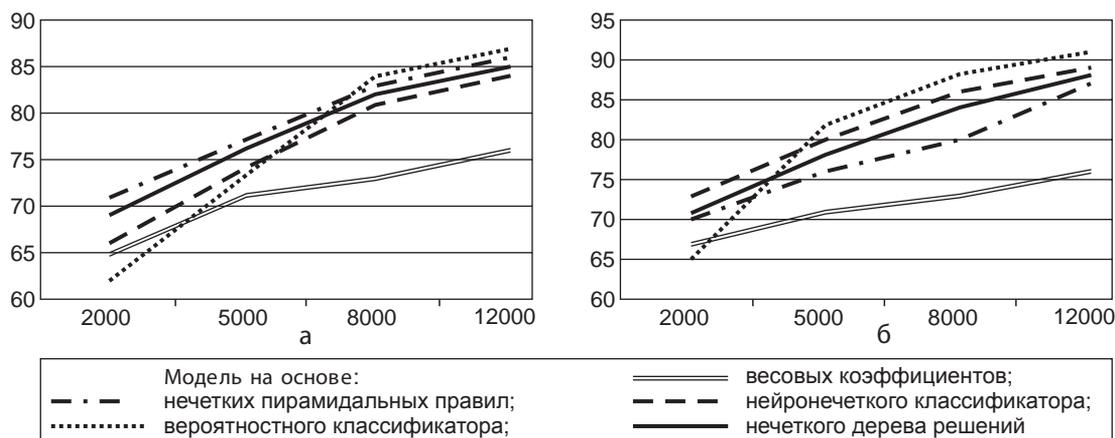


Рис. 1. Графики зависимостей точности рубрицирования от объема обучающей выборки при использовании моделей: *а* — в условиях взаимосвязанных рубрик; *б* — в условиях невязанных рубрик

Fig. 1. Graph of the dependence of the rubrication accuracy on the volume of the training sample when using different rubrication models in the condition of interrelated (*a*) and non-interrelated (*b*) rubrics

и условной вероятностью $P(w/h_i)$ встретить термин w в ЭНТД d при условии выбора рубрики h_i . Тогда под вероятностью $P(d|hi)$ понимается вероятность того, что ЭНТД будет классифицирован при условии выбора рубрики h_i .

Ситуация б: анализ больших ЭНТД при средней / существенной степени пересечения рубрик и достаточном объеме и качестве статистических данных.

Предлагаемый метод: метод голосования классификаторов, позволяющий снизить вероятность ошибок при рубрицировании ЭНТД. Данный метод предполагает взвешенное агрегирование решений, полученных пятью вышеописанными моделями рубрицирования, для получения окончательной классификации.

На рисунке 1 представлены графики зависимостей точности рубрицирования от объема обучающей выборки при их использовании описанных моделей в разных условиях.

Процедура реализации мультимодельного подхода к рубрицированию ЭНТД

Анализ типовых ситуаций и предлагаемых математических методов рубрицирования ЭНТД доказывает невозможность применения единой модели, что позволяет обосновать целесообразность создания мультимодельного подхода к решению данной научной задачи.

На рисунке 2 приведена процедура организации процесса автоматизированного рубрицирования ЭНТД на основе применения предложенного мультимодельного подхода в различных типовых условиях (нотация IDEF0).

Важнейшим этапом реализации рассматриваемого мультимодельного метода является формирование рубричного поля, которое заключается в описании элементов множества рубрик с помощью таких характеристик, как степень соответствия слова рубрике,

частота встречаемости слова и порог употребления слова.

Реализация предлагаемого мультимодельного подхода к рубрицированию предполагает анализ множества ЭНТД, в котором каждый документ представляется в виде набора значимых слов.

На начальном этапе каждый ЭНТД проходит регистрацию и представляется в виде лингвистической XML-разметки с использованием совокупности тегов (дескрипторов).

После регистрации документ анализируется с целью выделения в его текстовой части в форме лингвистической разметки слов, предложений, абзацев и т. д. (блок «Сегментатор»).

Следует отметить сложность реализации процедуры сегментации, а также высокую степень влияния ошибок на итоговые результаты автоматизированного рубрицирования ЭНТД, что, в первую очередь, характерно для процедур рубрицирования коротких ЭНТД.

После лингвистической разметки осуществляется морфологический анализ ЭНТД с выделением лексических характеристик слов и морфем (наименьших имеющих смысл языковых единиц).

Стандартные морфологические характеристики по всему тезаурусу хранятся в морфологической базе, заполняемой из нескольких источников, важнейшим из которых является набор текстовых документов СинТагРус (Национальный корпус русского языка, содержащий размеченные тексты).

Результаты морфологического анализа ЭНТД сохраняются в виде отдельной лингвистической группы в исходном документе. Это означает, что помимо морфологической информации, приписанной каждому слову текста, для каждого предложения задана его синтаксическая структура.

Следующим этапом является выбор конкретной модели рубрицирования ЭНТД на основе выявления приведенных типовых ситуаций. При этом выбор конкретной модели осуществляется на основе идентификации соот-

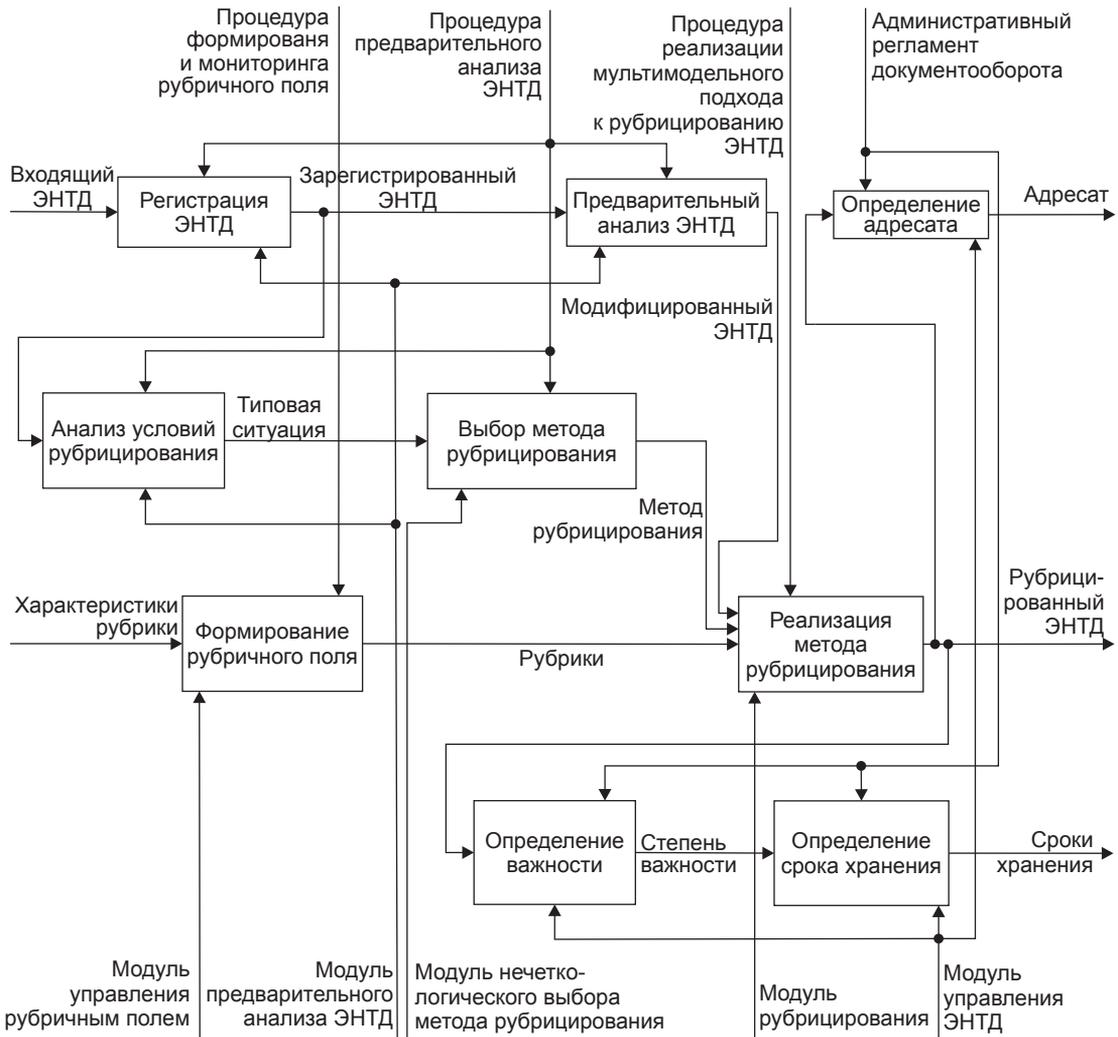


Рис. 2. Процедура организации процесса автоматизированного рубрицирования ЭНТД

Fig. 2. Procedure for organizing the automated rubrication of the EUTD

ветствующей типовой ситуации с помощью применения базы нечетких продукционных правил (П1-П6).

В связи с тем, что значения характеристик ЭНТД, приведенных в табл. 1 (размер ЭНТД L_k , степень пересечения тезаурусов рубрик K_{\cap} , достаточность статистических данных V_{st}), определяются экспертной информацией, то для реализации процедуры выбора конкретной модели автоматизированной рубрики ЭНТД M_j из набора моделей предлагается использовать базу нечетких продукционных правил следующего вида:

$IF (V_{st} \text{ is } V_{st}^a) \text{ and } (K_{\cap} \text{ is } K_{\cap}^b) \text{ and } (L_k \text{ is } L_k^c) \text{ THEN } (M \text{ is } M_j),$
 $a \in \{insignificant, significant\},$
 $b \in \{insignificant, medium, significant\},$
 $c \in \{short, medium, large\}, j \in \{1, 2, 3, 4, 5, 6\}.$

Предлагаемая процедура реализации указанного метода включает в себя этап определения сроков хранения актуальных документов в общей базе системы, которые зависят от степени важности документа.

Перед записью в базу проанализированных документов определяется срок хранения

документа по его степени важности, граничные значения которой определяются экспертами в зависимости от типа документа и существующих административных регламентов документооборота. По истечении срока хранения ЭНТД больше не используются при мо-

нитинге рубричного поля и не учитываются при пересчете статистических характеристик. На рисунке 3 приведен алгоритм обработки и анализа ЭНТД при использовании мультимодельного подхода для автоматизированного рубрицирования.

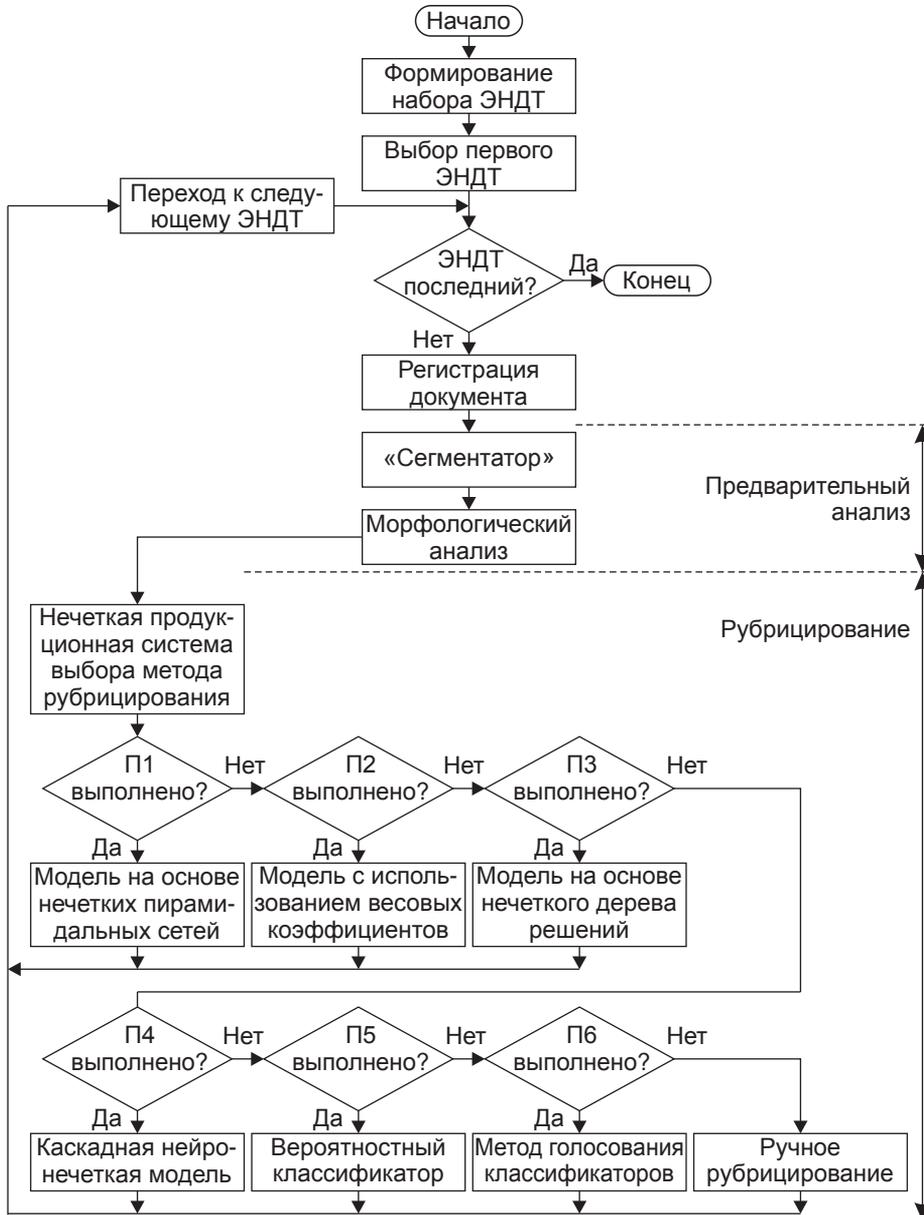


Рис. 3. Алгоритм мультимодельного подхода к рубрицированию ЭНТД

Fig. 3. Algorithm of the multimodel approach to the EUTD rubrication

Заключение

В заключение вкратце перечислим полученные научные результаты.

Проведен анализ различных методов автоматизированного анализа текстов для решения задачи рубрицирования неструктурированных электронных документов, показавший невозможность построения единой модели ввиду многообразия условий его проведения, которые различаются в зависимости от значений выделенного набора характеристик (степени пересечения тезаурусов рубрик, объема накопленной статистической информации, изменчивости тезауруса рубрик, числа уникальных значимых слов в документе).

Предложен концептуально новый мультимодельный подход к рубрицированию неструктурированных документов, основанный на комплексном использовании интеллектуальных и вероятностно-статистических методов анализа текстов. Выбор конкретного метода осуществляется с использованием базы нечетких продукционных правил на основе идентификации одной из шести типовых ситуаций, построенных на основе комбинации приведенных характеристик.

Предложена процедура организации автоматизированного рубрицирования неструктурированных электронных текстовых документов с описанием информационных потоков между различными этапами анализа текстов, а также приведен алгоритм непосредственной реализации предложенного мультимодельного подхода.

Предложенный мультимодельный подход к автоматизированному рубрицированию электронных неструктурированных документов позволяет повысить точность отнесения к конкретным рубрикам текстовых документов с учетом специфики их предметной области и различного объема накопленных статистических данных.

Список литературы

1. *Bevainyte A., Butenas L.* Document classification using weighted ontology // *Materials Physics and Mechanics*. 2010. No. 9. P. 246–250.
2. *Borisov V., Dli M., Zernov M., Fedulov A.* Method of time series analysis using scenarios // *International Journal of Applied Engineering Research*. 2016. No. 11 (21). P. 10536–10539.
3. *Dli M., Ofitserov A., Stoianova O., Fedulov A.* Complex model for project dynamics prediction // *International Journal of Applied Engineering Research*. 2016. No. 11 (22). P. 11046–11049.
4. *Gladun V., Vashchenko N.* Analytical processes in pyramidal networks // *International journal on information theories & applications*. 2000. Vol. 7. P. 103–109.
5. *Quinlan J.* Induction of decision trees // *Machine Learning*. 1998. Vol. 1. No. 1. P. 81–106.
6. *Sebastiani F.* Machine learning in automated text categorization // *ACM Computing Surveys*. 2002. Vol. 34. No. 1. P. 1–47.
7. *Singh. S.* Dynamic Pattern Recognition for Temporal Data // *Proceedings of the 5th European Congress on Intelligent Techniques and Soft Computing (EUFIT '97, Aachen, Germany, September 8–11, 1997)*. 1997. Vol. 3. P. 1993–1997.
8. *Witten I., Frank E.* *Data Mining: Practical Machine Learning Tools and Techniques* (2nd edition). Amsterdam: Morgan Kaufmann Series in Data Management Systems, Elsevier Inc., 2005. — 560 p.
9. *Андреев А. М., Березкин Д. В., Сюзев В. В., Шабанов В. И.* Модели и методы автоматической классификации текстовых документов // *Вестник МГТУ им. Н. Э. Баумана. Сер. Приборостроение*. 2003. №4. С. 64–94.
10. *Багузова (Булыгина) О. В., Головинская М. В., Чичерова Е. Ю.* Анализ особенностей и выбор инструментов диагностики состояния сложных социально-экономических объектов промышленности // *Научное обозрение*. 2014. №3. С. 165–168.
11. *Бояринов Ю. Г., Стоянова О. В., Дли М. И.* Применение нейро-нечеткого метода группового учета аргументов для построения моделей социально-экономических систем // *Программные продукты и системы*. 2006. №3. С. 20–23.
12. *Борисов В. В., Дли М. И., Козлов П. Ю.* Анализ и мониторинг рубрицирования электронных текстовых документов // *Вестник МЭИ*. 2018. №4. С. 121–127.
13. *Булыгина О. В., Емельянов А. А., Емельянова Н. З., Кукушкин А. А.* Системный анализ в управлении / Под ред. А. А. Емельянова. М.: ФОРУМ: ИНФРА-М, 2017. — 450 с.
14. *Булыгина О. В., Окунев Б. В.* Нечетко-сетевые инструменты анализа перспективности проектов

- по развитию информационно-телекоммуникационной инфраструктуры // *Нейрокомпьютеры*. 2016. № 7. С. 15–20.
15. Гимаров В. А., Дли М. И. Нейросетевой алгоритм классификации сложных объектов // *Программные продукты и системы*. 2004. № 4. С. 51–56.
 16. Гладун В. П. Растущие пирамидальные сети — организация памяти интеллектуальных сетей // *Штучный интеллект*. 2003. № 3. С. 70–77.
 17. Заболеева-Зотова А. В., Петровский А. Б., Орлова Ю. А., Шитова Т. А. Автоматизированный анализ тематики текстов новостей // *International Journal «Information Content and Processing»*. 2016. Vol. 3. No. 3. P. 288–299.
 18. Козлов П. Ю. Методы автоматизированного анализа коротких неструктурированных текстовых документов // *Программные продукты и системы*. 2017. № 1. С. 100–105.
 19. Козлов П. Ю. Сравнение частотного и весового алгоритмов автоматического анализа документов // *Научное обозрение*. 2015. № 14. С. 245–250.
 20. Круглов В. В., Дли М. И., Голунов Р. Ю. Нечеткая логика и искусственные нейронные сети. М.: Наука, Физматлит, 2001. — 201 с.
 21. Мешалкин В. П., Дли М. И., Гимаров В. А. Динамическая классификация сложных технологических систем. Методы, алгоритмы и практические результаты. М.: Физматлит, 2006. — 344 с.
 22. Булыгина О. В. Анализ реализуемости инновационных проектов по созданию наукоемкой продукции: алгоритмы и инструменты // *Прикладная информатика*. 2016. Т. 11. № 4 (64). С. 87–102.
 7. Singh. S. Dynamic Pattern Recognition for Temporal Data. Proceedings of the 5th European Congress on Intelligent Techniques and Soft Computing (EUFIT '97, Aachen, Germany, September 8–11, 1997), 1997, vol. 3, pp. 1993–1997.
 8. Witten I., Frank E. *Data Mining: Practical Machine Learning Tools and Techniques* (2nd edition). Amsterdam, Morgan Kaufmann Series in Data Management Systems, Elsevier Inc., 2005, 560 p.
 9. Andreev A. M., Berezkin D. V., Syuzev V. V., Shabanov V. I. *Modeli i metody avtomaticheskoi klassifikatsii tekstovyykh dokumentov* [Models and methods of automatic classification of text documents]. *Herald of the Bauman Moscow State Technical University. Series Instrument Engineering*, 2003, no. 4, pp. 64–94 (in Russian).
 10. Baguzova (Bulygina) O. V., Golovinskaya M. V., Chicherova E. Yu. Analysis of specific features and choice of tools for diagnosing the condition of complex socio-economic industrial objects. *Nauchnoe obozrenie* — Science Review, 2014, no. 3, pp. 165–168 (in Russian).
 11. Boyarinov Yu. G., Stoyanova O. V., Dli M. I. *Primeniye neuro-nechetkogo metoda gruppovogo ucheta argumentov dlya postroeniya modelei sotsial'no-ekonomicheskikh sistem* [Application of the neuro-fuzzy method of group accounting of arguments for constructing models of socio-economic systems]. *Programmnye produkty i sistemy*, 2006, no. 3, pp. 20–23 (in Russian).
 12. Borisov V. V., Dli M. I., Kozlov P. Yu. Analysis and monitoring of the rubrication of electronic text documents. *Vestnik MEI*, 2018, no. 4, pp. 121–127 (in Russian).
 13. Bulygina O. V., Emel'yanov A. A., Emel'yanova N. Z., Kukushkin A. A. *Sistemnyi analiz v upravlenii / Pod red. A. A. Emel'yanova* [System analysis in the management. Edited by A. A. Emelyanov]. Moscow, FORUM, INFRA-M Publ., 2017. 450 p. (in Russian)
 14. Bulygina O. V., Okunev B. V. Creating fuzzy network tools to analyze prospects of projects of information and telecommunication infrastructure development. *Neirokomp'yutery*, 2016, no. 7, pp. 15–20 (in Russian).
 15. Gimarov V. A., Dli M. I. *Neiroseteivoi algoritm klassifikatsii slozhnykh ob'ektov* [Neural network algorithm of complex object classification]. *Programmnye produkty i sistemy* (Software & Systems), 2004, no. 4, pp. 51–56 (in Russian).
 16. Gladun V. P. *Rastushchie piramidal'nye seti — organizatsiya pamyati intellektual'nykh setei* [Growing pyramidal network — the organization of the memory of intelligent networks]. *Shtuchnyi intellekt* — Artificial Intelligence, 2003, no. 3, pp. 70–77 (in Russian).
 17. Zaboileeva-Zotova A. V., Petrovskii A. B., Orlova Yu. A., Shitova T. A. Automated analysis of news

References

1. Bevainyte A., Butenas L. Document classification using weighted ontology. *Materials Physics and Mechanics*, 2010, no. 9, pp. 246–250.
2. Borisov V., Dli M., Zernov M., Fedulov A. Method of time series analysis using scenarios. *International Journal of Applied Engineering Research*, 2016, no. 11 (21), pp. 10536–10539.
3. Dli M., Ofitserov A., Stoianova O., Fedulov A. Complex model for project dynamics prediction. *International Journal of Applied Engineering Research*, 2016, no. 11 (22), pp. 11046–11049.
4. Gladun V., Vashchenko N. Analytical processes in pyramidal networks. *International journal on information theories & applications*, 2000, vol. 7, pp. 103–109.
5. Quinlan J. Induction of decision trees. *Machine Learning*, 1998, vol. 1, no. 1, pp. 81–106.
6. Sebastiani F. Machine learning in automated text categorization. *ACM Computing Surveys*, 2002, vol. 34, no. 1, pp. 1–47.

- text topics. International Journal «Information Content and Processing», 2016, vol. 3, no. 3, pp. 288–299 (in Russian).
18. Kozlov P. Yu. Methods of automated analysis of short unstructured text documents. *Programmnye produkty i sistemy* (Software & Systems), 2017, no. 1, pp. 100–105 (in Russian).
 19. Kozlov P. Yu. Compare the frequency and weighting algorithms for automatic analysis of documents. *Nauchnoe obozrenie* — Science Review, 2015, no. 14, pp. 245–250 (in Russian).
 20. Kruglov V. V., Dli M. I., Golunov R. Yu. *Nechetkaya logika i iskusstvennye neyronnye seti* [Fuzzy logic and artificial neural networks]. Moscow, Nauka, Fizmatlit Publ., 2001, 201 p. (in Russian).
 21. Meshalkin V. P., Dli M. I., Gimarov V. A. *Dinamicheskaya klassifikatsiya slozhnykh tekhnologicheskikh sistem. Metody, algoritmy i prakticheskie rezul'taty* [Dynamic classification of complex technological systems. Methods, algorithms and practical results]. Moscow, Fizmatlit Publ., 2006, 344 p. (in Russian).
 22. Bulygina O. V. Analysis of the feasibility of innovative projects for creating high technology products: the algorithms and instruments. *Prikladnaya Informatika* — Journal of Applied Informatics, 2016, vol. 11, no. 4 (64), pp. 87–102 (in Russian).

M. Dli, The Branch of National Research University «MPEI» in Smolensk, Russia, midli@mail.ru

O. Bulygina, The Branch of National Research University «MPEI» in Smolensk, Russia, baguzova_ov@mail.ru

P. Kozlov, The Branch of National Research University «MPEI» in Smolensk, Russia, originaldod@gmail.com

Formation of the structure of the intellectual system of analyzing and rubricating unstructured text information in different situations¹

The analysis of electronic text documents written in natural language is one of the most important tasks implementing in systems of automated analyzing linguistic information. Today the most complicated problem is analyzing unstructured text documents coming to various organizations and authorities through the electronic communications. The increasing volume of such documents leads to the need to rubricate incoming messages, i.e. to solve the classification task.

The analysis of the scientific works in this field has showed the impossibility of constructing a unified model for rubricating unstructured electronic text documents in various situations. The main reasons are the lack of statistical data, the dynamism of the thesaurus and the small size of the incoming document.

To solve this problem, we propose a multimodel approach to the rubrication that is characterized by the combined use of intellectual and probabilistic-statistical methods of the text document analysis. The choice of a specific model is carried out using fuzzy logic algorithms based on the proposed characteristics (the size of document, the degree of rubric thesaurus intersection, the frequency of meaningful keywords, etc.).

The implementation of the proposed multimodel approach will improve the accuracy of attributing unstructured electronic text documents to concrete rubrics taking into account their specificity and various objectives of practical application in the organization.

Keywords: electronic unstructured text documents, rubrication, multimodel approach, growing pyramidal networks, fuzzy logic algorithms.

About authors:

M. Dli, *Dr of Engineering, Professor;*

O. Bulygina, *PhD in Economics, Associate Professor;*

P. Kozlov, *PhD in Engineering, Assistant*

For citation:

Dli M., Bulygina O., Kozlov P. Formation of the structure of the intellectual system of analyzing and rubricating unstructured text information in different situations. *Prikladnaya informatika* — Journal of Applied Informatics, 2018, vol. 13, no. 4 (64), pp. 111–123 (in Russian).

¹ This work was supported in part by grants RFBR № 18-01-00558.

Л. А. Рейнгольд, канд. техн. наук, консультант ООО «ДИАВЕР», г. Москва, leonidrein@gmail.com
Е. А. Рейнгольд, канд. экон. наук, руководитель группы АО «МКД Партнер», г. Москва, l_r@mail.ru

Структурирование информации для целей автоматизации — концептуальные аспекты

Развитие информационных технологий, ИТ-инфраструктуры радикально меняет общество, менеджмент, государственное управление. Необходима понятная для всех заинтересованных сторон общая система понятий в области ИТ, обеспечивающая согласованный взгляд на происходящие изменения. Востребована система терминов, позволяющая структурировать и комплексно анализировать происходящие изменения. В статье рассматривается базовая система понятий, включающая термины: условия, объект, субъект, ситуация, деятельность объекта. Предложенный подход обеспечит структурную полноту описания, решение задач семантической интероперабельности и целеполагания при использовании информационных технологий на всех этапах их жизненного цикла, послужит концептуальной основой при разработке и поиске применений новых информационных технологий.

Ключевые слова: виртуализация, семантическая интероперабельность, цифровизация, интернет вещей, условия, ситуация, ситуационная осведомленность, интегрированные средства представления знаний, анализ и визуализация информации ИТ-инфраструктуры.

Введение

В современном обществе происходят инфраструктурные изменения, касающиеся всех сторон жизни. Это во многом обусловлено новым уровнем развития информационных технологий (ИТ), которые внедряются во все области жизни — в экономику, управление государством, а также используются в быту. Нет таких областей деятельности, где не происходило бы сейчас принципиальных изменений, связанных с повсеместным внедрением электронных устройств.

В государстве приняты нормативные документы по развитию цифровой экономики и Индустрии 4.0 [1, 2], внедрению Интернета

вещей. Выполнение этих работ требует развития системы знаний с целью своевременного выявления проблем и их эффективного решения [3, 4]. Явления, связанные с внедрением ИТ, требуют обобщения, позволяющего обеспечить понимание возникающих проблем, их решение, обеспечить комплексное понимание происходящих в инфраструктуре общества изменений. В частности, требуется формирование необходимых нормативных и правовых ограничений, позволяющих выявлять и проактивно предотвращать возможные нарушения в сфере ИТ [5].

В основе обобщенного анализа новых тенденций должен быть понятийный аппарат, который понимался бы единообразно всеми участниками процессов внедрения ИТ. В основе такого понятийного аппарата могут быть, казалось бы, очевидные, понятные и часто употребляемые,

¹ Работа выполнена при частичной поддержке грантов РФФИ № 16-07-00860, № 17-07-00762. Автор выражает благодарность РФФИ за поддержку научных исследований.

но не имеющие четкого определения понятия: объект, субъект, ситуация, условия и др. Эти термины, несмотря на некоторую размытость и различия определений, в целом единообразно воспринимаются людьми. Предложенные понятия, с одной стороны, должны быть общеупотребительными, а с другой, объединены в единую систему, не вызывая отторжения и противоречия с имеющимися у большинства людей представлениями. Они должны послужить одной из основ для формализованного представления недостаточно формализованных предметных областей и решения слабоструктурированных задач.

В конкретной предметной области такая система понятий должна помочь формированию некоторого концептуального каркаса для отражения деятельности человека в условиях ускоряющегося процесса изменения инфраструктуры общества. Такая система должна быть удобна для использования всеми типами пользователей информационных технологий. Применение единой системы понятий для решения практических задач будет способствовать обеспечению семантической интероперабельности, взаимопониманию во взаимодействии людей, использующих ИТ-инфраструктуру.

В статье рассматривается такая система понятий. Она может применяться для описания особенностей технологий, свойств вещей, их взаимодействия, а также деятельности человека в формирующейся ИТ-инфраструктуре. Предложенная система имеет междисциплинарный характер и может применяться для решения различных научных, технических и повседневных задач. На этой основе может выполняться решение задач моделирования и целеполагания, создание интегрированных средств для представления знаний в различных предметных областях, а также технологических решений для представления знаний в целях структурирования и интеллектуализации обработки информации о деятельности человека.

Объект и субъект применительно к деятельности

Существует проблема структурирования деятельности человека. Действительность, которая его окружает, состоит из взаимодействующих в различных ситуациях объектов.

Под объектом будем понимать любую обособленную от других вещь, явление, живой организм, который целесообразно выделять, рассматривать отдельно от других объектов. Во многих случаях объекты в той или иной мере автономны, могут перемещаться относительно друг друга, имеют некоторые собственные границы.

Объекты взаимодействуют, то есть влияют друг на друга. Например, воздействовать различными способами друг на друга могут расположенные рядом объекты. Это может быть механическое воздействие, обмен информацией и др.

Во взаимодействии можно выделить некоторые эпизоды, в которых общая структура и порядок этого взаимодействия сохраняется. Назовем такие эпизоды ситуациями взаимодействия объектов или просто «ситуациями». Один такой эпизод может быть похож на другой, то есть ситуации повторяются с той или иной степенью подобия. В каждой ситуации объекты, а значит, и их свойства, изменяются в результате влияния этих объектов друг на друга.

Некоторые объекты имеют цели во взаимодействии. Под целью понимается ожидаемый объектом результат своих действий в ситуации. В одной ситуации могут участвовать несколько объектов, имеющих собственные цели. Каждый из таких объектов «видит» ситуацию с точки зрения своих целей и стремится действовать так, чтобы эти цели осуществить.

Каждую ситуацию возможно рассматривать с точки зрения того из заинтересованных объектов, который в данном случае для нас существенен и интересен для изучения.

Назовем объект, с точки зрения которого мы смотрим на ситуацию, субъектом деятель-

ности в ситуации, а далее для краткости будем говорить просто «субъект».

Человек как субъект в ситуации преследует некоторую цель и достигает ее, выполняя некоторую последовательность шагов. И цели, и пути их достижения осознаны и «запрограммированы» в различной степени. Например, некоторые люди ведут себя в ситуации спонтанно, а кто-то четко планирует каждый шаг в своей последовательности действий.

Понятие субъекта может относиться как к живым объектам, так и к техническим устройствам, которые имеют собственный алгоритм, логику поведения и могут влиять на ситуацию в собственных целях. Цели технического устройства могут быть неочевидны и являются выражением целей создателей этого устройства.

Живые объекты могут быть одушевленными, то есть иметь способность к пониманию ситуации (в той или иной степени) и ее значения для себя. Например, иметь возможность формировать и использовать в процессе деятельности в ситуации новые способы поведения, формируемые в своих интересах.

Неодушевленные объекты имеют, как правило, наследуемые способы поведения или заданную извне программу, если речь идет о технических устройствах.

Понятия одушевленности, понимания собственной сущности объектом и влияния этого понимания на поведение, являются сложными и нуждаются в дополнительном изучении. Это концептуальный и технический, а также философский, а для кого-то даже религиозный вопрос.

Люди — наиболее сложный тип субъектов, которые имеют разнообразные и многоуровневые системы целеполагания, основывающийся на биологической основе.

Специфическими субъектами являются группы людей. Каждая такая группа имеет объединяющие ее признаки и, оказываясь в подобных ситуациях, действует согласо-

ванно с обусловленным этими общими признаками образом с использованием сформированных тем или иным способом шаблонов поведения. В то же время члены группы сохраняют автономность и могут менять признаки, по которым присоединяются к группе, и в связи с этим переходить в другие группы.

Группы могут формироваться по различным общим признакам: профессиональному, этническому, религиозному и др. Такая группа имеет некоторые механизмы координации поведения своих членов: обсуждает собственные цели и способы поведения, формулирует их, формирует присущие группе стереотипы, создает иерархию и органы управления и консолидирует на этой основе свое поведение в ситуации для достижения общей цели.

Общая стратегия поведения каждого субъекта формируется на основе его целостного представления об окружающей среде, ситуационной осведомленности. Используется собственный и чужой опыт поведения в соответствующей ситуации. Можно сказать, что субъект стремится удовлетворить во взаимодействии потребности, которые у него сформированы ранее и каким-то образом выражены, конкретизированы применительно к рассматриваемой ситуации.

Деятельность субъекта можно рассматривать в виде последовательности ситуаций, представленных на нескольких уровнях абстракции. Например, жизнь человека в обществе может быть представлена в виде иерархии взаимосвязанных ситуаций, некоторой многомерной и многоуровневой «матрешки» из элементов взаимодействия различного уровня. Измерения и иерархические уровни вложения в иерархию ситуаций связаны между собой. Ситуации верхнего уровня могут включать несколько более конкретных ситуаций следующего уровня.

Объекты могут быть классифицированы различным образом. В контексте решения ИТ-задач интересно деление объектов на реальные и виртуальные — смоделированные тем или иным образом и отображае-

мые человеку удобным способом: от реалистичного отображения объекта до информационного выражения в рамках той или иной знаковой системы. Отдельная тема для рассмотрения — частичная виртуализация материальных объектов за счет внедрения технологий Интернета вещей. Традиционные вещи при этом получают новые свойства за счет встраивания в них возможностей по обработке и обмену информацией, существенно изменяющих их свойства.

Другой классификацией является деление объектов на естественные и измененные или заново созданные человеком — искусственные объекты.

Коротко рассмотрим соотношение между понятиями вещь и объект в контексте анализируемой системы понятий. Понятие «вещь» более подходит к неодушевленным объектам, имеющим материальный, реальный характер.

Из сказанного выше следует, что субъект — это объект, имеющий собственные интересы, которые нужно учитывать в рассмотрении. Условимся также считать, что одушевленные объекты — это люди, а неодушевленные — вещи. Живые организмы обычно не принято называть вещами, поэтому при необходимости будем называть их живыми объектами.

Потребности выражаются в интересах субъекта, которые лежат в основе целеполагания этого субъекта, его представлений о состояниях, которых необходимо достичь в будущем.

Для живых организмов целеполагание обусловлено биологическими причинами и сформировано эволюционно. Потребности человека как субъекта сформированы более сложно: биологически и социально. В технические системы в той или иной форме встроена проекция потребностей людей-субъектов, которые создают эти системы и ими управляют.

Потребности субъекта определяют его целеполагание — выбор желаемых состояний окружения в рамках имеющихся условий

и выбираемые пути достижения поставленных целей. Социально-экономическое целеполагание для групп людей формируется условиями кооперации, требованиями к совместной деятельности людей, социализацией и обучением. Целеполагание групп также определяется требованиями управления сложными техническими объектами, такими как корабль, самолет, система связи и пр.

Можно сказать, что субъекты — это объекты, имеющие собственное целеполагание, то есть целеустремленные системы. Чаще всего под субъектом будем понимать человека, действия которого в некоторой ситуации нас интересуют.

Возникает вопрос: техническая система, имеющая встроенное целеполагание, — это объект или субъект? Можно принять, что это субъект, опосредованно реализующий целеполагание своего субъекта — создателя, человека. Подробное рассмотрение целеустремленных систем — отдельная задача, попытка решения которой предпринята в [6]. Правда, это рассмотрение слабо связано с ИТ-тематикой, поскольку книга выпущена тогда, когда вычислительная техника использовалась только в производственных задачах, а компьютерные устройства «для всех» и коммуникационная ИТ-инфраструктура отсутствовали.

Субъект в той или иной степени отражает, понимает, «осознает» свое окружение и может влиять на это окружение. А оно, в свою очередь, может влиять на него. Это взаимодействие может быть отражено характеристиками объектов. Они описывают структуру и свойства объектов, в том числе взаимосвязи и взаимозависимости между ними.

Характеристики в некоторой ситуации всегда описываются с точки зрения того или иного субъекта. Причем каждый из субъектов может иметь свою модель ситуации и соответственно собственный набор ее характеристик. Поэтому если нас интересует поведение нескольких субъектов, то нужно рассматривать ситуацию последовательно с точки

зрения каждого значимого субъекта, а при необходимости строить модель их взаимодействия.

У субъекта для целенаправленного действия в ситуации в той или иной форме имеется отражение ситуации и некоторая модель взаимодействия с существенными в этой ситуации объектами. Объект соответствующим образом структурируется, и осуществляется взаимодействие с ним.

На основе изложенного рассмотрим понятие ситуации более детально. Выше отмечалось, что ситуацией мы считаем характерный, типичный, повторяющийся случай в поведении объекта. На концептуальной схеме (рис. 1) приведена структура ситуации.

Как правило, структура ситуации для заданной предметной области на выбранном уровне обобщения сохраняется, меняются только значения существенных характеристик — некоторых определенных в процессе структурирования переменных ситуации.

Ситуация включает:

- **Объекты.** Это все действующие участники, вещи, информационные элементы, которые могут влиять на условия в ситуации, существуют самостоятельно, представляют самостоятельную ценность, могут участвовать в нескольких ситуациях.

- **Субъект** — объект, с точки зрения которого мы смотрим на ситуацию. Как правило, для него важны условия в ситуации и достижение определенного результата.

- **Показатели (характеристики),** описывающие ситуацию и участие в них объектов, состояние условий.

Как отмечено выше, структура описания ситуации принимается нами постоянной, меняются только переменные, характеризующие эту ситуацию. Если существенно изменились объект, субъект, значения условий и структура взаимодействия — значит, произошел переход к следующей ситуации во взаимодействии.

Жизнь (жизненный цикл субъекта) — последовательность взаимосвязанных ситуаций, в которых он участвует, действует. В этой последовательности ситуаций реализуется целеполагание субъекта в рамках ограничений, характеризуемых условиями, предоставляемыми окружающими объектами. Как правило, можно определить начальную и конечную ситуацию для жизненного цикла каждого объекта.

В более сложной модели ситуаций в каждый рассматриваемый момент времени субъект включен в некоторое множество ситуаций взаимодействия — некоторое ситуационное

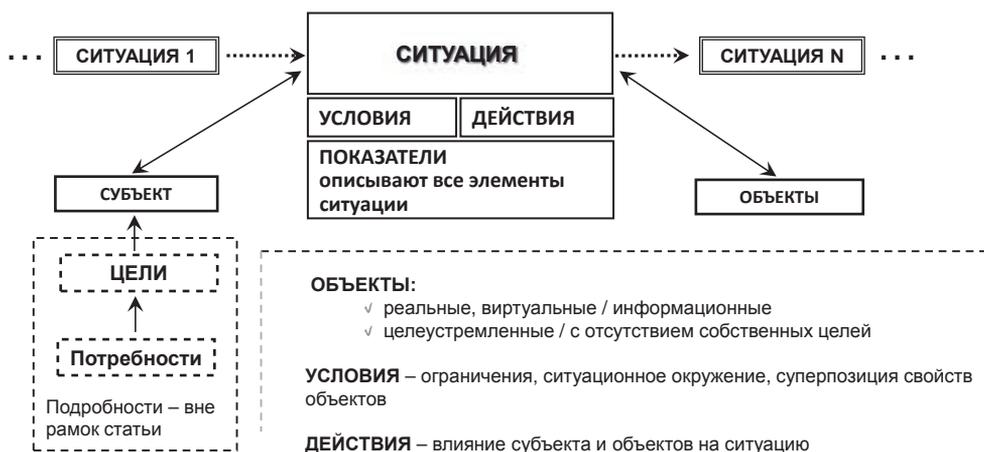


Рис. 1. Структура деятельности субъекта в ситуации

Fig. 1. Structure of the subject's activity in the situation

окружение, в котором реализуется его поведение. То есть субъект включен в несколько параллельных потоков ситуаций. Простой пример — человек, как правило, участвует в следующих потоках ситуаций: работа, домашние дела, хобби и пр.

Ситуации, подобные друг другу, можно сравнивать между собой, если они и входящие в них объекты описываются некоторыми общими существенными характеристиками, позволяющими из сравнения их значений получать полезную информацию. В частности, одну из ситуаций можно выбирать в качестве прототипа для изучения и планирования другой подобной ситуации. Например, все потоки ситуаций «работа» можно сравнивать по характеристикам, описывающим рабочее время субъекта. Такими характеристиками могут быть доля времени, которую субъект посвящает работе, уровень сложности выполняемой работы и пр.

Ситуации, подобные друг другу, возможно рассматривать на различных уровнях обобщения. Общая ситуация может состоять из нескольких более конкретных ситуаций, выполняющихся последовательно или параллельно. На каждом уровне обобщения в некоторой деятельности целесообразна своя детализация. В более конкретной ситуации субъект ставит более конкретные цели и задает более конкретные критерии достижения заданных этими целями результатов.

Поведение субъекта направлено на изменение характеристик ситуации в соответствии со сформированным ранее целеполаганием. Планирование поведения субъекта для достижения заданной цели в сложной ситуации часто означает выстраивание в рамках этой ситуации последовательности следующего уровня ситуаций, обеспечивающего в результате достижение некоторого целевого состояния объекта.

В каждой ситуации объекты выступают для субъекта как некоторые условия — ограничения или источники его возможностей.

Рассмотрим взаимосвязь условий и ситуации более подробно.

Условия и ситуации во взаимодействии объектов

При взаимодействии с любым объектом, любой вещь субъект использует в ситуации нужные ему особенности, которые могут быть отражены в свойствах этого объекта. Назовем особенности вещей, существенных для их использования условиями. Система условий в некоторой ситуации возникает как результат взаимодействия свойств окружающих человека объектов, вещей и в целом из особенностей внешней среды, окружающей человека. На взаимодействие с конкретным объектом влияет также наличие у субъекта информации о нем, понимание правил его использования и ситуации, в которой он находится, а также другие факторы.

То есть поведение субъекта в ситуации ограничено условиями, в которых он находится. Для субъекта условия — ограничения в деятельности, ограничение возможностей. Востребован способ структурирования условий, позволяющий их классифицировать и оценивать последствия их изменения.

Выделим наиболее общие, характерные для произвольного объекта условия, применимые в любой возможной ситуации. Эти общие условия в каждом случае получают конкретное содержание и детализацию. Назовем их условиями-аксиомами, поскольку они принимаются нами без доказательства. Они являются основой для простой и понятной классификации окружения субъекта в ситуации, основанной на ее существенных свойствах.

Условия-аксиомы характеризуют любой объект и позволяют комплексно рассмотреть его функциональность применительно к удовлетворяемой объектом в ситуации потребности. Использование условий-аксиом позволяет предложить принципы детализации функциональности любого объекта, вещи.

Необходимость этих аксиом очевидна, однако достаточность нуждается в дополнительном обосновании.

Отметим, что в каждой ситуации субъекта окружает некоторое множество объектов, каждый из которых, в свою очередь, может быть описан собственной детализацией условий-аксиом. В таблице 1 приведены аксиомы-условия, присущие каждому объекту в любой ситуации.

На основе условий-аксиом возможно структурирование свойств объектов «сверху вниз». В частности, этот подход может применяться в процессе человеко-машинного взаимодействия и согласования семантики свойств объектов с унифицированным репозитарием свойств вещей, а также онтологий, построенных на их основе, отражающих

особенности использования объектов в различных ситуациях.

Как известно [7, с. 17], схемы классификации могут быть моноиерархическими и полииерархическими. Полииерархические классификации имеют множественное наследование, то есть каждая позиция классификации может иметь несколько родителей. В нашем случае каждое конкретное свойство вещи описывается полииерархической классификацией, т. е. точкой в многомерном пространстве, образованном иерархической классификацией условий. Вершиной каждого измерения является одна из условий-аксиом.

То есть верхним уровнем такой многомерной и многоуровневой классификации и являются аксиомы-условия, детализация кото-

Таблица 1. Условия-аксиомы и их содержание

Table 1. Conditions-axioms and their content

Название условия	Содержание
Качество	Характеризует особенности функциональности используемого объекта, его отличия от других подобных объектов, применяемых в соответствующей ситуации. Условие используется в ситуации для отражения существа влияния объекта на ситуацию
Пространство	Отражает особенности, связанные с размерами, формой, размещением в пространстве, территориальным местоположением объекта. Объект ограничен окружающим его пространством. Для виртуальных объектов это виртуальная среда, с которой они взаимодействуют
Время	Характеризует динамику использования и изменения объекта: быстроту выполнения функций, ограничения по времени использования и др. Все процессы развиваются во времени. Любой объект существует в последовательности ситуаций в некотором временном промежутке, ограниченном его жизненным циклом
Эквивалент	Характеризует все виды затрат, связанные с получением и эксплуатацией объекта, в частности его цена и стоимостные параметры всех элементов его жизненного цикла. Условия, создаваемые объектами, приходится создавать и затем распределять между субъектами-потребителями. Поэтому между субъектами в ситуации вырабатываются механизмы эквивалентного обмена условиями. Это деньги за временное использование, «владение» объектом и пр. Чтобы владелец смог получить за свой объект эквивалент, этот объект должен быть чем-то полезен другим субъектам
Информация	Все, что известно об объекте, — источнике условий, в том числе то, что нужно знать о свойствах объекта в интересующей ситуации. Эти сведения всегда имплицитно присутствуют у объекта, и они должны быть достаточны для его использования. Другими словами, субъект должен уметь пользоваться объектом в соответствующей ситуации, знать его свойства и др. В целом аксиома информации-условия об объекте — это знания, необходимые для его применения во всех интересующих субъекта ситуациях. Если объект способен отражать свое окружение в ситуации, то он накапливает и использует опыт об этом окружении в своем взаимодействии. Т.е. субъект строит собственную модель окружения и своего места в нем, чтобы увеличить успешность взаимодействия

рых тем более конкретна, чем более конкретные типы и экземпляры вещей она описывает.

Использование условий-аксиом позволит создать некоторый фреймворк, типовую структуру, описывающую условия для субъекта в ситуации любой интересующей предметной области. Такая структура позволит упорядочить свойства вещей на различных уровнях абстракции и связать эти уровни между собой. Она может применяться также в онтологиях, описывающих различные по степени общности ситуации.

Наличие общих семантических оснований в иерархии конкретных условий позволит реализовать единообразное структурирование подобных друг другу вещей, используемых в различных предметных областях. То есть предлагаемый подход может способствовать решению вопросов обеспечения семантической интероперабельности на верхних уровнях обобщения свойств объектов. Существующая нормативная база [8], на взгляд авторов, нуждается в концептуальной проработке этого понятия. На основе общих концептуальных положений возможно решение проблем выбора показателей и измерение уровня семантической интероперабельности [9].

Нужно учитывать принцип суперпозиции условий — все условия действуют одновременно. Их конкретные проявления имеют собственную структуру и являются ограничениями для субъекта в ситуации в каждый момент времени.

Для описания условий и их применения в ИТ-инфраструктуре могут быть разработаны механизмы получения структурированных свойств объектов с использованием имеющихся описаний этих свойств. Необходимо также применение знаний пользователей информационных систем, работающих с информацией об объектах в интересующих ситуациях для формализации и получения значенных свойств объектов.

Классификация свойств в соответствии с предлагаемой концепцией может приме-

няться для формального представления виртуальных объектов в различных практических приложениях. Результаты структурирования таких объектов могут применяться для работы пользователей в ситуациях, требующих использования компьютерной техники с обеспечением требований семантической интероперабельности.

Рассматриваемый подход, в частности, позволяет представить свойства объектов Интернета вещей в иерархической форме. В результате появится понимание возможных направлений развития, которое будет совершенствоваться в процессе применения унифицированного представления мира вещей для решения различных практических задач.

Унифицированное описание свойств вещей является актуальной задачей в связи с развитием и удешевлением всех компонентов, необходимых для формирования и функционирования Интернета вещей. Как следствие, упрощается создание вещей с новыми, в том числе виртуальными, свойствами, поскольку все большее количество вещей взаимодействует между собой и с людьми с использованием ИТ-инфраструктуры. Более подробно применение рассматриваемого подхода к структурированию условий для решения задач Интернета вещей будет рассмотрено ниже.

Особенность внедрения ИТ в настоящее время — виртуализация окружающей человека инфраструктуры. Многие значимые свойства объектов, действующих в ситуации, реализуются программным путем. Вещи могут осуществлять коммуникации между собой, что также придает им новые свойства. В то же время виртуальный мир может материализоваться — например, на основе 3D-моделей строятся реальные объекты с использованием аддитивных технологий, которые становятся все более доступными и требуют все меньшего участия человека.

Виртуальные свойства объектов не видны пользователю, а при необходимости могут

отображаться различным образом. Окружающий человека мир становится более изменчивым и часто менее наглядным, поэтому и актуальна выработка подходов, позволяющих структурировать его на различных уровнях абстракции. Отображение объектов и их влияние на ситуацию должно быть доступно для понимания всеми заинтересованными категориями пользователей — от научных работников и технических специалистов до обычных пользователей ИТ.

Такое универсальное структурирование в том числе должно решать задачи обеспечения интероперабельности на различных уровнях — от самого общего до решения конкретных задач. Оно должно способствовать созданию концептуально совместимого программного обеспечения и оборудования.

Рассмотренный подход к представлению условий деятельности объектов в ситуациях может быть применен в качестве «метауровня» для изучения поведения человека в современных условиях, обеспечения интероперабельности создаваемых элементов инфраструктуры и технологий и может употребляться как одна из концептуальных основ в процессе проектирования новых информационных технологий и их использования. Этот подход может быть применен и к детализации описания ИТ-метафор — пока не вполне осознанно используемой языковой конструкции, комплексно отражающей на метауровне особенности ИТ. Более подробно роль метафор в обеспечении семантической интероперабельности применительно к ИТ рассмотрена в работах [10, 11].

Всеобщая коммуникационная связность современного мира и возрастающая способность окружающих нас вещей к регистрации и анализу информации влияет на потребности и условия деятельности людей в окружающем мире.

В связи с этим рассмотрим более подробно применение предложенных выше решений, связанных со структурированием объектной среды, для решения проблем формализованного представления и интероперабельности применительно к одному из наиболее быстро развивающихся направлений в ИТ — Интернету вещей.

Интернету вещей.

Проблемы структурирования и интероперабельности в Интернете вещей

Одним из наиболее значимых факторов развития ИТ-инфраструктуры в настоящее время является Интернет вещей. Он находит применение в системах производства и управления, повседневной жизни людей, изменяя функциональность привычных вещей. Объекты Интернета вещей изменчивы и способны адаптироваться в соответствии с меняющимися требованиями к внешнему окружению. При этом новая функциональность вещей не видна человеку, кроме того она может существенно изменяться перепрограммированием устройства в процессе эксплуатации и изменением его информационных связей.

В процессе внедрения Интернета вещей должны быть решены в том числе следующие концептуальные вопросы:

- разработка методологии и практических решений, обеспечивающих описание функциональности вещи;
- появление средств оценки и формализованного описания степени виртуализации вещи, прогнозирования направлений ее развития, оценки тенденций изменения вещи со временем;
- предложение методов формирования системы показателей оценки качества функционирования вещи, существенных с точки зрения каждого субъекта, принимающего участие в ее жизненном цикле;
- разработка способов оценки социально-экономической эффективности вещи, определяющих целесообразность ее использования и видоизменения.

Использование Интернета вещей требует решения ряда следующих научных проблем:

- разработка и научное осмысление системы понятий, необходимых в процессе автоматизации задач, связанных с использованием Интернета вещей;

- структурированное описание вещей как возможностей и ограничений во взаимодействии с использованием условий-аксиом и их детализации применительно к различным ситуациям;

- концептуальные проблемы визуализации свойств вещей: разработка, модификация, управление, применение пользователями. Необходимо учитывать, что у каждого элемента Интернета вещей практически всегда имеется два «лица»: материальная, видимая вещь, а также ее виртуальное отображение в некотором интерфейсе (например, в виде встроеного сайта вещи), моделирующее и позволяющее мониторить и настраивать все значимые для применения, в том числе неявные, свойства вещи;

- концептуальные аспекты, связанные со стандартизацией описания объектов Интернета вещей;

- моделирование объектов Интернета вещей и их связей как с традиционной видимой инфраструктурой, так и информационного взаимодействия с другими вещами и людьми.

В процессе внедрения Интернета вещей актуальна разработка технологий, обеспечивающих полное, совместимое, понятное представление смысловой информации о них всем заинтересованным пользователям таких «умных» и коммуницирующих вещей. Конечному пользователю необходимо включаться в инфраструктуру Интернета вещей, а также стать источником собственного опыта жизни в этом технологизированном мире для других.

В условиях появления Интернета вещей пользователь должен получить простую и наглядную среду для описания окружающего его мира вещей, совместимую с описанием традиционных вещей. Взаимодействующие системы должны «понимать» как друг друга, так и требования обслуживаемых ими лю-

дей. То есть наряду с технической должна обеспечиваться семантическая интероперабельность как во взаимодействии человека с окружающими вещами, так и при совместном функционировании интернет-вещей. Это позволит на следующем уровне развития мира вещей встроить в него механизмы, обеспечивающие развитие функциональности и самоорганизацию предметного окружения общества.

Изменения в мире вещей должны выявляться ИТ-инфраструктурой и осуществляться автоматическая настройка необходимых свойств вещей, а также коммуникаций между ними с учетом технических и нормативно-правовых ограничений.

Подвижность и изменчивость как вещей, так и ситуаций, в которых они применяются в условиях Интернета вещей, требует дополнительных средств, помогающих пользователю ориентироваться в них. Интернет вещей включен в мир обычных вещей, поэтому рассматривать все вещи необходимо как единую инфраструктуру в совместимой системе понятий. Семантическая интероперабельность должна касаться всех существенных вещей, поскольку они совместно формируют условия, определяющие поведение пользователя в каждой ситуации.

Характерной тенденцией является увеличение в обычных вещах «виртуальной составляющей» по мере встраивания в них компьютерных и коммуникационных устройств. Современная инфраструктура вещей меняется в двух встречных направлениях (рис. 2). Одно направление — изменение в сторону виртуализации: реализация дополнительной функциональности встроеными в вещь устройствами обработки информации. Второе направление — там, где это требуется, внедряются в практику средства преобразования виртуальных вещей в реальные. Например, востребованные в рамках концепции Индустрии 4.0 [12, 13] аддитивные технологии обеспечивают материализацию виртуальных вещей, создание кото-

рых другими средствами было трудоемким, неудобным или излишне сложным. В частности, такие технологии начинают широко использоваться в промышленности и строительстве [14–15].

На рисунке 2 приводится схема, иллюстрирующая взаимный переход реальной и виртуальной составляющих в мире Интернета вещей. В зависимости от условий и ситуаций будет происходить взаимный переход между виртуальной и материальной составляющими вещей. Если требуемые условия целесообразнее обеспечить виртуальной частью вещи, то в реализации ее функциональности будет возрастать доля «виртуальности», в противном случае виртуальные свойства будут материализовываться. Разумеется, это касается не всех свойств вещи, а тех, которые могут быть реализованы как материально, так и виртуально.

Предлагается любой объект Интернета вещей рассматривать комплексно, учитывая его виртуальную и реальную составляющую. Любая вещь находится между «вырожденными» случаями, когда вещь является полностью реальной или виртуальной. Полностью реальными являются объекты, полученные из естественной среды и не имеющие встроенных механизмов, изменяющих их функциональность, а виртуальными — объекты, не имеющие независимого материального воплощения, например компьютерные файлы, программное обеспечение.

Гипотеза о наличии в каждой вещи реальной и виртуальной составляющих предполагает, что с развитием технологий преобладающей является тенденция замещения реальной составляющей вещи виртуальной, то есть

имеется тенденция к повышению виртуальности вещей.

Граница реального и виртуального в ситуационном окружении смещается в сторону увеличения виртуальной составляющей. Это проиллюстрировано на рис. 3 — граница (линия) раздела реального и виртуального в окружающей среде смещается вверх. Эта тенденция сопровождается формированием метасистем, поддерживающих новую функциональность виртуального мира — интероперабельность свойств вещей, их коммуникации, механизмы управления этой инфраструктурой.

Можно говорить, что виртуальную составляющую могут включать все типы объектов в ситуации. Это субъекты, часть деятельности которых в повседневной жизни и трудовой деятельности смещается в виртуальную сферу: группы людей, которые все более включены в социальные сети, виртуальные взаиморасчеты и другие виды виртуальной деятельности; вещи, получающие все более значимую виртуальную составляющую. При этом реализация условий-аксиом в виртуализированных ситуациях имеет некоторые особенности. Например, условия-аксиомы пространства и времени функционируют там по-другому относительно реального мира. Участники виртуализированной ситуации часто не имеют информации, где находятся другие объекты и когда они выполняли свою деятельность в ситуации. Это приводит к трудностям в поведении субъектов и к новым рискам в их деятельности.

Новые возможности в обработке данных приводят к возникновению условий для получения дополнительной функцио-

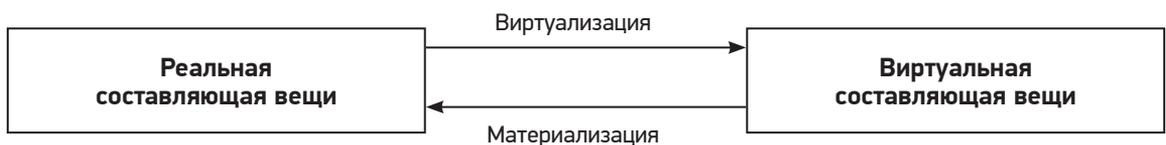


Рис. 2. Тенденции виртуализации и материализации вещей в современных условиях

Fig. 2. Trends of virtualization and materialization of things in modern conditions



Рис. 3. Схема взаимодействия вещи с естественным и интеллектуальным окружением в некоторой ситуации

Fig. 3. Scheme of interaction of things with natural and intellectual environment in some situation

нальности вещей. Во многих случаях объекты Интернета вещей являются субъектами ситуации с внешним и встроенным целеполаганием. Примером может служить транспорт.

В современном автомобиле функции электромеханической системы управления двигателем автомобиля практически замещены встроенным компьютером, обеспечивающим дополнительную эффективность работы двигателя. То же во все большей степени касается тормозной системы, а также некоторых других функций управления традиционно выполняемым человеком-оператором.

В обозримой перспективе прогнозируется полное замещение функциональности водителя автомобиля средствами автоматизации. То есть система управления автомобилем становится самостоятельным субъектом в ситуациях дорожного движения. Этот субъект автоматически распознает предметное окружение автомобиля, осуществляет коммуникации с внешними объектами в ситуации, используя инфраструктуру Интернета вещей. Таким образом, виртуальная составляющая вещи становится самостоятельным субъек-

том, обеспечивающую без участия человека по его поручению и «от его имени» полную адаптацию свойств вещи к состоянию внешнего окружения.

Можно сказать, что свойства внешних систем с использованием коммуникационной инфраструктуры в некоторой степени имманентно встраиваются в интернет-вещь, дополняют ее. С использованием коммуникаций M2M появляется возможность использовать взаимное функциональное дополнение вещей. Так, уже сейчас наличие динамической информации о дорожной обстановке оказывает воздействие на функционирование всех транспортных средств одновременно, безотносительно к тому насколько высок уровень автоматизации конкретного транспортного средства. Транспортная система в целом адаптируется к среде, отражаемой навигаторами, имеющимися у большинства участников движения, а средства управления дорожным движением во все большей степени действуют на основе анализа получаемой в автоматическом режиме информации о ситуационном окружении на обслуживаемом участке.

Таким образом, в современных условиях субъект вынужден учитывать как реальную, так и виртуальную составляющие своего предметного окружения. Граница между реальной и виртуальной составляющими пока смещается преимущественно в сторону виртуальной составляющей вещей, становящейся все более интеллектуальной, получающей дополнительные способности к адаптации.

Следствием смещения свойств вещей в виртуальную сферу является замещение реальных условий виртуальными. При этом качество вещей, отражаемое в рассмотренных условиях-аксиомах, качеством-условием становится выше, поведение вещей более предсказуемым. Пространственные ограничения, т. е. ограничения, связанные с пространством-условием во взаимодействии, связанные с взаимным пространственным размещением вещей, исчезают. Однако комплексное восприятие ситуационной осведомленности субъектом при недостаточном внимании к проблеме может усложниться.

Проявления времени-условия также трансформируются: вещи выполняют свои функции с более высокой скоростью, изготавливаются и модернизируются быстрее. Эквивалент-условие при внедрении интернет-вещей также изменяется: вещи становятся менее ресурсоемкими, более доступными, их изготовление, тиражирование, модернизация, эксплуатация требуют меньших затрат.

Реализация информации-условия переходит на новый уровень — сведения о полезных свойствах вещей, их обслуживании, приобретении и возможностях продажи ненужных вещей становятся более доступными. Во многих случаях субъекту для получения и обработки информации в ситуации не требуется непосредственного участия других людей. Вещи становятся более понятными в применении, например, они могут сами «рассказывать» пользователю о своих возможностях.

Рассмотренные выше концептуальные положения позволяют реализовать новые прак-

тические решения для построения моделей данных и знаний. В процессе дальнейших исследований возможно получение средств структурирования и анализа данных, методов работы со знаниями, отражающими социально-экономические аспекты информатизации и компьютеризации общества.

В целом предлагаемые решения позволяют в значительной степени реализовать мягкий системный подход [16], позволяющий учесть неформальные и трудноформализуемые аспекты деятельности человека в условиях меняющейся ИТ-инфраструктуры общества.

Заключение

В условиях радикального изменения социально-экономической инфраструктуры, связанной с внедрением новых информационных технологий, важное значение имеет выработка подходов к структурированию информации в процессах взаимодействия всех участников общественной жизни — государства, бизнеса, обычных граждан.

Возникают новые тенденции, которые выражаются в появлении новых терминов, технических устройств. Изменения комплексно представлены в виде ИТ-метафор, которые в понятной для всех заинтересованных лиц форме в понятных, броских, часто афористичных терминах отражают сущность происходящих изменений, выражают знания о происходящих в сфере ИТ изменениях и определяют направления развития. В различном контексте встречаются метафоры: Цифровая экономика, Интернет вещей, Индустрия 4.0 и другие.

Происходит, с одной стороны, виртуализация окружающей нас среды, а с другой, — внедряются технологии, позволяющие материализовать виртуальный объект любой формы из наиболее удобного материала.

В работе рассмотрены концептуальные вопросы, позволяющие помочь в структурировании и формализации перечисленных слож-

ных явлений, а также сделать эти явления более понятными всем заинтересованным лицам. Эту задачу призвана решать система понятий для рассмотрения функциональности вещей на самом общем уровне с использованием общеупотребительных терминов, в частности: объекта, субъекта, ситуации применительно к использованию их в контексте структурирования ИТ-инфраструктуры. Предложено также понятие условий-аксиом, позволяющих описать функциональность вещей на самом общем уровне, а затем детализировать условия для конкретных ситуаций.

Такой поход позволяет обеспечивать семантическую совместимость при рассмотрении ИТ-инфраструктуры и эффективно решать задачи представления знаний о ней, моделирования, анализа функционирования и управления, получать формализованные знания об ИТ-инфраструктуре на «метауровне». Возможно осуществление разработки и поиска применений новых информационных технологий, решение практических задач по их совершенствованию, получению новых средств анализа и визуализации информации от естественного и интеллектуального окружения человека.

Список литературы

1. Программа «Цифровая экономика Российской Федерации». Утверждена распоряжением Правительства Российской Федерации от 28.07.2017 № 1632-р. URL: <http://government.ru/docs/28653/> (дата обращения: 29.06.2018).
2. Распоряжение Правительства Российской Федерации от 23.03.2018 № 482-р «О плане мероприятий («дорожной карте») по совершенствованию законодательства и устранению административных барьеров в целях обеспечения реализации Национальной технологической инициативы по направлению «Технет» (передовые производственные технологии)». URL: <http://assets.fea.ru/uploads/fea/nti/docs/gasporgazyhenie-dk-23032018-482.pdf> (дата обращения: 29.06.2018).
3. Волков А. И., Клименко С. В., Рейнгольд Л. А. Внедрение Интернета вещей и проблемы развития системы знаний // Труды Международной конференции и Школы по физико-технической информатике СРТ2016, 10–17 мая 2014 г., Ларнака, Республика Кипр; 01–03 июля 2015, Царьград, Московская область. Изд. ИФТИ, Москва-Протвино, 2015. Труды Международной научной конференции. Москва-Протвино: ИФТИ, 2015–2016. С. 104–112.
4. Рейнгольд Л. А. Концептуальные аспекты структурирования информации для целей автоматизации // Ситуационные центры и информационно-аналитические системы класса 4i для задач мониторинга и безопасности (SCVRT2017). Труды Международной научной конференции. Москва-Протвино, 2017. С. 44–53.
5. Мантуров: отсутствие нормативной базы мешает цифровизации российских предприятий. URL: <https://rns.online/it-and-media/Manturov-otsutstvie-normativnoi-bazi-meshaet-tsifrovizatsii-rossiiskih-predpriyatii--2017-07-11> (дата обращения: 29.06.2018).
6. Аюф Р., Эмери Ф. О целеустремленных системах. М.: Советское радио, 1974. — 272 с.
7. FIBO conceptual methodology, Further exploration of conceptual Modeling Principles FDTF Day 2, 17 Sept 2014. С. 17 URL: http://www.edmcouncil.org/semanticsrepository/docs/fibo-bco-conceptual-modeling_v0.2.pptx (дата обращения: 29.06.2018).
8. ГОСТ Р 55062–2012 Информационные технологии (ИТ). Системы промышленной автоматизации и их интеграция. Интероперабельность. Основные положения. М.: Стандартинформ, 2014. — 12 с.
9. Рейнгольд Л. А., Волков А. И., Конайгородский А. Н., Пустозеров Е. Ю. Семантическая интероперабельность в решении финансовых задач и способы ее измерения // Прикладная информатика. 2016. № 4 (64). С. 115–134.
10. Рейнгольд Л. А. Метафоры информационных технологий в современном обществе // Прикладная информатика. 2017. Т. 12. № 6 (72). С. 24–39.
11. Рейнгольд Л. А. Информационно-технологические метафоры в геоинформатике // Труды Международной Школы-семинара Московского физико-технического института (университета) Института физико-технической информатики. SCRT1516, 21–24 ноября 2016, Царьград, Московская область, Россия. Изд. ИФТИ, Москва-Протвино, 2016. С. 157–162.
12. Пуха Ю. Индустриальная революция 4.0. Анализ PwC UN report World Population Ageing 1950–2050. Октябрь 2017. URL: <https://www.pwc.ru/ru/assets/pdf/industry-4-0-pwc.pdf> (дата обращения: 29.06.2018) (in Russian).
13. Industrie 4.0. The future of industry — Made in Germany. URL: <https://www.plattform-i40.de/I40/Navigation/EN/Industrie40/WhatIsIndustrie40/what-is-industrie40.html> (дата обращения: 29.06.2018).

14. 3D-принтер для строительства домов как бизнес. URL: <http://p-business.ru/3d-printer-dlya-stroitelstva-domov-kak-biznes> (дата обращения: 29.06.2018).
 15. Погорельский А. Российский инженер разработал 3D-принтер для строительства домов. URL: <http://realty.rbc.ru/articles/06/11/2015/562949998072349.shtml> (дата обращения: 29.06.2018).
 15. Голованов В. В Амстердаме скоро появится первый мост, напечатанный на 3D-принтере. URL: <https://geektimes.ru/post/251894>.
 16. Ledington P. Soft systems methodology — core concepts. San Francisco, 2014. P. 53.

References

1. *Programma «Tsifrovaya ekonomika Rossiiskoi Federatsii»* [The program «Digital Economy of the Russian Federation»]. Approved by government decree of July 28, 2017, no. 1632-p. Available at: <http://government.ru/docs/28653> (accessed 29.06.2018).
 2. *Rasporyazhenie Pravitel'stva RF, 23 march 2018, no. 482-r «O plane meropriyatiy («dorozhnoi karte») po sovershenstvovaniyu zakonodatel'stva i ustraneniyu administrativnykh bar'erov v tselyakh obespecheniya realizatsii Natsional'noi tekhnologicheskoi initsiativy po napravleniyu «Tekhnet» (peredovye proizvodstvennye tekhnologii)»*. Available at: <http://assets.fea.ru/uploads/fea/nti/docs/rasporyazhenie-dk-23032018-482.pdf> (accessed 29.06.2018).
 3. Volkov A. I., Klimenko S. V., Reingol'd L. A. *Vnedrenie Interneta veshchei i problemy razvitiya sistemy znaniy* [The introduction of the Internet of things and the problems of the development of the knowledge system]. *Trudy Mezhdunarodnoi konferentsii i Shkoly po fiziko-tekhnicheskoi informatike CPT2016, 10–17 maya 2014 g., Larnaka, Respublika Kipr; 01–03 iyulya 2015, Tsar'grad, Moskovskaya oblast'* [Proc. of the International Conference and School of Physical and Technical Informatics CPT2016, May 10–17, 2014, Larnaca, Republic of Cyprus; 01–03 July 2015, Tsargrad, Moscow region]. Izd. IFTI, Moskva-Protvino, 2015, pp. 104–112.
 4. Reingol'd L. A. *Kontseptual'nye aspekty strukturirovaniya informatsii dlya tselei avtomatizatsii* [Conceptual aspects of information structuring for automation purposes]. *Situatsionnye tsentry i informatsionno-analiticheskie sistemy klassa 4i dlya zadach monitoringa i bezopasnosti (SCVRT2017)*. *Trudy Mezhdunarodnoi nauchnoi konferentsii*. IFTI, Moskva-Protvino, 2017, pp. 44–53.
 5. *Manturov: otsutstvie normativnoi bazy meshaet tsifrovizatsii rossiiskikh predpriyatii* [Manturov: the lack of a regulatory framework prevents the digitalization of Russian enterprises]. Available at: [\[and-media/Manturov-otsutstvie-normativnoi-bazy-meshaet-tsifrovizatsii-rossiiskikh-predpriyatii--2017-07-11\]\(https://rms.online/it-and-media/Manturov-otsutstvie-normativnoi-bazy-meshaet-tsifrovizatsii-rossiiskikh-predpriyatii--2017-07-11\) \(accessed 29.06.2018\).
 6. Akoff R., Emeri F. *O tselestremlynykh sistemakh* \[About purposeful systems\]. Moscow, Sovetskoe radio Publ., 1974, 272 p.
 7. *FIBO conceptual methodology, Further exploration of conceptual Modeling Principles FDTF Day 2, 17 Sept 2014, p. 17*. Available at: \[http://www.edmcouncil.org/semanticsrepository/docs/fibo-bco-conceptual-modeling_v0.2.pptx\]\(http://www.edmcouncil.org/semanticsrepository/docs/fibo-bco-conceptual-modeling_v0.2.pptx\) \(accessed 29.06.2018\).
 8. *GOST 55062–2012. Informatsionnye tekhnologii. Sistemy promyshlennoi avtomatizatsii i ikh integratsiya. Interoperabel'nost'. Osnovnye polozheniya* \[State Standard 55062–2012. Information technologies. Industrial automation systems and integration. Interoperability. Basic principles\]. Moscow, Standartinform, 2014, 12 p.
 9. Reingol'd L. A., Volkov A. I., Kopaigorodskii A. N., Pustozarov E. Yu. Semantic interoperability in solving financial problems and ways to measure it. *Prikladnaya informatika — Journal of Applied Informatics*, 2016, vol. 11, no. 4 \(64\), pp. 115–134 \(in Russian\).
 10. Reingol'd L. A. Metaphors of Information Technologies in Modern Society\]. *Prikladnaya informatika — Journal of Applied Informatics*, 2017, vol. 12, no. 6 \(72\), pp. 24–39 \(in Russian\).
 11. Reingol'd L. A. *Informatsionno-tekhnologicheskie metafory v geoinformatike* \[Information and technological metaphors in geoinformatics\]. *Trudy Mezhdunarodnoi Shkoly-seminara Moskovskogo fiziko-tekhnicheskogo instituta \(universiteta\) Instituta fiziko-tekhnicheskoi informatiki. SCRT1516* \[Proc. of the International School-Seminar of the Moscow Institute of Physics and Technology \(University\) of the Institute of Physical and Technical Informatics. SCRT1516\], 21–24 November 2016, Tsar'Grad, Moskovskaya oblast', Rossiya. Izd. IFTI, Moskva-Protvino, 2016, pp. 157–162.
 12. Pukha Yu. *Industrial'naya revolyutsiya 4.0. Analiz PwC UN report World Population Ageing 1950–2050* \[Industrial Revolution 4.0. Analysis PwC UN report World Population Ageing 1950–2050. October 2017\]. October 2017. Available at: <https://www.pwc.ru/ru/assets/pdf/industry-4-0-pwc.pdf> \(accessed: 29.06.2018\) \(in Russian\).
 13. *Industrie 4.0. The future of industry — Made in Germany*. Available at: <https://www.plattform-i40.de/I40/Navigation/EN/Industrie40/WhatIsIndustrie40/what-is-industrie40.html> \(accessed 29.06.2018\).
 14. *3D printer dlya stroitel'stva domov kak biznes* \[3D printer for building houses as a business\]. Available at: \[\\[138 \\]\]\(http://p-business.ru/3d-printer-</p>
</div>
<div data-bbox=\)](https://rms.online/it-</p>
</div>
<div data-bbox=)

- dlya-stroitelstva-domov-kak-biznes) (accessed 29.06.2018).
15. Pogorel'skii A. *Rossiiskii inzhener razrabotal 3D-printer dlya stroitel'stva domov* [Russian engineer has developed a 3D printer for building houses]. Available at: <http://realty.rbc.ru/articles/06/11/2015/562949998072349.shtml> (accessed 29.06.2018).
 16. Golovanov V. *V Amsterdame skoro poyavitsya pervyi most, napechatannyi na 3D-printere* [In Amsterdam, soon to appear the first bridge, printed on a 3D printer]. Available at: <https://geektimes.ru/post/251894> (accessed 29.06.2018).
 17. Ledington P. *Soft systems methodology — core concepts*. San Francisco, 2014, p. 53.

L. Reingold, LLC DIAVER, Moscow, Russia, leonidrein@gmail.com

E. Reingold, LLC MCD PARTNERS, Moscow, Russia, L_r@mail.ru

Structuring information for automation-conceptual aspects¹

The development of information technologies and its infrastructure is radically changing society, management and public administration. However, there is no clear for all stakeholders common system of concepts in the field of it, providing mutual understanding between them, a coherent view of the changes. Therefore, it is necessary to build a basic system of concepts that allows to structure and analyze the changes in a complex way. The article considers such a system of concepts, including terms: conditions, object, subject, situation, object activity and others in connection with the trend of virtualization of the properties of the object environment surrounding a person. This allows us to structure the technical and socio-economic phenomena described by the metaphors of the Digital economy, the Internet of things, Industry 4.0 and others. The proposed approach will provide structural completeness of the description of situations, contribute to the solution of problems of semantic interoperability and goal-setting. Such an approach can be applied in the process of analyzing the functionality of information technologies, in the development and support of information technologies throughout the life cycle, can be a conceptual basis for the development of new information technologies and in the search for their new applications.

Keywords: virtualization, semantic interoperability, digitalization, Internet of things, conditions, situation, situational awareness, integrated means of knowledge representation, analysis and visualization of IT infrastructure information.

About authors:

L. Reingold, *PhD in Technique*;

E. Reingold, *PhD in Economics*

For citation:

Reingold L., Reingold E. Structuring information for automation-conceptual aspects. *Prikladnaya informatika* — Journal of Applied Informatics, 2018, vol. 13, no. 4 (76), pp. 124–139 (in Russian).

¹ This work was supported in part by grants RFBR № 16-07-00860, № 17-07-00762.

Правила оформления материалов

Шаблон оформления рукописи¹

<И. О. Фамилия автора>, <ученая степень>, <ученое звание>, <организация>, <город>, <емэйл>

...

<Название статьи на русском языке>

<Аннотация на русском языке объемом 300–500 знаков>.

Ключевые слова (не менее пяти): <перечислить через запятую>.

Введение

<Вводная часть статьи>

<Подзаголовок_1 основной части>

<Текст первого раздела основной части>

...

<Подзаголовок_2 основной части>

<Текст второго раздела основной части>



Рис. 1. <Название рисунка на русском языке>

Fig. 1. <Название рисунка на английском языке>²

...

¹ При подготовке использованы методические указания эксперта БД «SCOPUS» О. В. Кирилловой (ред. 2013 г.).

² По всем рисункам должны быть предоставлены **исходники** (скриншоты — выполненные с **высоким разрешением** (не менее 300 dpi), векторные рисунки (схемы) — в векторном формате, **доступном для редактирования**, диаграммы — в формате Excel).

Таблица 1. <Название таблицы на русском языке>

Table 1. <Название таблицы на английском языке>

...

Листинг 1. <Название программного текста на русском языке>

Listing 1. <Название программного текста на английском языке>

```
switch (wParam)
{
case IDOK: // Кнопка «Настрой»
GetDlgItemText (hwnd, DM_C001, str, 80); //
Ввод из окна 001
break;
case IDCANCEL: // Кнопка «Пропуск»
EndDialog (hwnd, FALSE);
break; default: return false;
}
...

```

...

$$F2 = L(N, P_2, L) = \sum_{m=0}^{L-1} \left[C_N^m P_2^{N-m} (1-P_2)^m m \right]^3 \text{ (<№ формулы>)}$$

<Подзаголовок_п основной части>

<Текст последнего раздела основной части>.

³ Формулы, включающие специализированные символы, набираются только с помощью внешних редакторов **Microsoft Equation** либо **Math Type**. Использование встроенного редактора формул Word 2007 и более высоких версий не допускается. Отдельные математические выражения в составе текста, не требующие специальной символики, могут быть набраны строковым редактором (клавиатурный ввод) с использованием режима «Вставка символа». Однако выражение не может быть набрано частично во внешнем редакторе формул, а частично строковым редактором.

Заключение

<Текст заключительной части>.

Список литературы

<Список источников на русском языке>.

References

<Список источников на латинице — транслитерация плюс перевод на английский язык>.

<И. Фамилия автора на английском языке>,
<организация>, <город>, <страна>, <емэйл⁴>.

<Название статьи на английском языке>

<Аннотация на английском языке объемом порядка 200 слов>.

Keywords: <ключевые слова на английском языке>.

About authors:

<И. Фамилия автора на английском языке>,
<ученая степень>, <ученое звание>

For citation: <И. Фамилия автора на английском языке>, <Название статьи на английском языке>. *Prikladnaya informatika* — Journal of Applied Informatics, 2015, vol. 10, no. x (xx), pp. xx–xx (in Russian).

Приложение 1**Примеры оформления русскоязычных источников в References⁵**

Приведены некоторые примеры описания источников в References и даны комментарии к ним. Обращаем внимание на то, что структура описания источника, а также служебные символы и сокращения отличаются от принятых для русскоязычного списка литературы.

Статья из бумажного журнала

Zagurenko A. G., Korotovskikh V. A., Kolesnikov A. A., Timonov A. V., Kardymon D. V. *Tekhniko-ekonomicheskaya optimizatsiya dizaina*

⁴ Согласно стандартам SCOPUS, в данном блоке указывается только та информация, которая приведена в настоящем шаблоне. Ученые степени и звания авторов следует указывать в отдельном блоке после ключевых слов.

⁵ Порядок следования источников в References совпадает с русскоязычной версией списка литературы.

*gidrorazryva plasta*⁶ [Techno-economic optimization of the design of hydraulic fracturing]. *Neftyanoe khozyaistvo* — Oil Industry⁷, 2008, no. 11, pp. 54–57.

Kharlamova T. L. *Motivatsionnye osnovy effektivnoy raboty predpriyatiya* [Motivational basis for the effective work of an enterprise]. *Ekonomika i upravlenie*, 2006, no. 3, pp. 100–102.

Keyno P. P., Siluyanov A. V. Design and implementation of a declarative web-interface modeling language interpreter on a high-performance distributed systems. *Prikladnaya informatika* — Journal of Applied Informatics, 2015, vol. 10, no. 1, pp. 15–25 (in Russian)⁸.

Статья из электронного журнала

Kontorovich A. E., Korzhubaev A. G., Eder L. V. *Mineral'nye resursy Rossii. Ekonomika i upravlenie* [Forecast of global energy supply: Techniques, quantitative assessments, and practical conclusions], 2006, no. 5. Available at: <http://www.vipstd.ru/gim/content/view/90/278/> (accessed 22.05.2012).

Статья из сборника трудов

Astakhov M. V., Tagantsev T. V. *Experimentalnoe issledovanie prochnosti coedinenij «stal-kompozit»* [Experimental study of the strength of joints «steel-composite»]. *Trudy MGTU Matematicheskoe modelirovanie slozhnykh tekhnicheskikh sistem* [Proc. of the Bauman MSTU «Mathematical Modeling of Complex Technical Systems»], 2006, no. 593, pp. 125–130.

⁶ Если цитируемая работа не имеет англоязычного названия, для его транслитерирования следует использовать сайт www.translit.net. На этом сайте необходимо предварительно выбрать вариант транслитерирования «BSI» — Британский институт стандартов.

⁷ Если цитируемый журнал имеет официальное двуязычное название (публикуемое в его печатной версии), указываются через дефис русский и английский варианты; в противном случае — только транслитерация (см. следующий пример). То же относится к названиям книжных издательств (см. примеры далее).

⁸ Если в цитируемой публикации уже представлено ее англоязычное название (что в последние годы является требованием ВАК для научных журналов), его и следует указать, не переводя самостоятельно. Давать транслитерацию названия статьи в этом случае не нужно, а в конце описания источника указывается в скобках (in Russian) — если статья написана на русском языке.

Материалы конференции

Usmanov T. S., Gusmanov A. A., Mullagalin I. Z., Muhametshina R. Ju., Chervyakova A. N., Sveshnikov A. V. *Osobennosti proektirovaniya razrabotki mestorozhdeniy s primeneniem gidrorazryva plasta* [Features of the design of field development with the use of hydraulic fracturing]. *Trudy 6 Mezhdunarodnogo Simpoziuma «Novye resursosberegayushchie tekhnologii nedropol'zovaniya i povysheniya neftegazootdachi»* [Proc. 6th Int. Technol. Symp. «New energy saving subsoil technologies and the increasing of the oil and gas impact»]. Moscow, 2007, pp. 267–272.

Книга

Kashnikov Y. A., Ashikhmin S. G. *Mekhanika Gornykh Porod Pri Razrabotke Mestorozhdeniy glevodorodnogo Syr'ya* [Rock Mechanics In The Development Of Hydrocarbon Deposits]. Moscow, ООО «Nedra-Biznestsentr» Publ., 2007. 486 p.

Lindorf L. S., Mamikonians L. G., eds. *Ekspluatatsiya turbogeneratorov s neposredstvennym okhlazhdeniem* [Operation of turbine generators with direct cooling]. Moscow, Energiia Publ., 1972. 352 p.

Kanevskaya R. D. *Matematicheskoe modelirovanie gidrodinamicheskikh protsessov razrabotki mestorozhdenii uglevodorodov* [Mathematical modeling of hydrodynamic processes of hydrocarbon deposit development]. Izhevsk, 2002. 140 p.

Izvekov V. I., Serikhin N. A., Abramov A. I. *Proektirovanie turbogeneratorov* [Design of turbo-generators]. Moscow, MEI Publ., 2005, 440 p.

Latyshev V. N. *Tribologiya rezaniya. Kn. 1: Friksionnye protsessy pri rezanie metallov* [Tribology of Cutting, vol. 1: Frictional Processes in Metal Cutting], Ivanovo, Ivanovskii Gos. Univ., 2009.

Переводная книга

Timoshenko S. P., Young D. H., Weaver W. *Vibration problems in engineering*. 4th ed. N. Y., Wiley, 1974. 521 p. (Russ. ed.: Timoshenko S. P., Iang D. Kh., Uiver U. *Kolebaniia v inzhenerenom dele*. Moscow, Mashinostroenie Publ⁹, 1985. 472 p.).

Brooking A., Jones P., Cox F. *Expert systems. Principles and case studies*. Chapman and Hall, 1984. 231 p. (Russ. ed.: Bruking A., Dzhons P., Koks F.

Ekspertnye sistemy. Printsipy raboty i primery. Moscow, Radio i sviaz' Publ., 1987. 224 p.).

Описание учебника или другого материала с указанием «под редакцией»

Arkhipov G. I., Sadovnichiy V. A., Chubarikov V. N. *Lekcii po matematicheskomu analizu: Uchebnik dlya universitetov i ped. vuzov* [Lectures on Mathematical Analysis: A Textbook for universities and pedagogical institutes]. Ed. by V. A. Sadovnichiy. Moscow, Visshaya shkola Publ., 1999, 695 p.

Диссертация

Grigor'ev Iu. A. *Razrabotka nauchnykh osnov proektirovaniia arkhitektury raspredelennykh sistem obrabotki dannykh. Diss. dokt. tekhn. nauk* [Development of scientific bases of architectural design of distributed data processing systems. Dr. eng. sci. diss.]. Moscow, Bauman MSTU Publ., 1996. 243 p.

ГОСТ

GOST 8.586.5–2005. Metodika vypolneniia izmerenii. Izmerenie raskhoda i kolichestva zhidkosti i gazov s pomoshch'iu standartnykh suzhaiushchikh ustroistv [State Standard 8.586.5–2005. Method of measurement. Measurement of flow rate and volume of liquids and gases by means of orifice devices]. Moscow, Standartinform Publ., 2007. 10 p.

Патент

Palkin M. V. e. a. *Sposob orientirovaniia po krenu letatel'nogo apparata s opticheskoi golovkoi samonavedeniia* [The way to orient on the roll of aircraft with optical homing head]. Patent RF¹⁰, no. 2280590, 2006.

Федеральный закон

RF Federal Law «On Protection of Consumers'Rights» of February 07, 1992 2300–1 (as amended by Federal Law of January 09, 1996 2 FZ, December 17, 1999 212 FZ) (in Russian).

¹⁰ Обратите внимание, что при цитировании документа, не являющегося опубликованной печатной работой (статьей, книгой, материалами конференции, сборником трудов), его статус (*патент, сертификат, инструкция и т. п.*) не транслитерируется, а указывается только на английском языке. То же касается законодательных материалов (*см. следующий пример*).

⁹ После наименования издательств необходимо указывать «Publ.».

ДИСТАНЦИОННОЕ ОБРАЗОВАНИЕ
В ВЕДУЩЕЙ ШКОЛЕ БИЗНЕСА «СИНЕРГИЯ»

MBA ONLINE

СПЕЦИАЛИЗАЦИИ ПРОГРАММЫ MBA ONLINE:

- Стратегический менеджмент
- Финансовый менеджмент
- Управление человеческими ресурсами
- Маркетинг и продажи
- Налоговый консалтинг
- MBA Менеджмент в фитнес-индустрии

КОНТАКТЫ

105318 | Москва | Измайловский Вал, дом 2
+7 (495) 545-43-14 | info@sbs.edu.ru | sbs.edu.ru

Подписка-2019

Журнал «Прикладная информатика» выходит 6 раз в год:
Февраль Апрель Июнь Август Октябрь Декабрь

Подписка через редакцию

Стоимость подписки на 2019 год:

1 номер	3 номера	6 номеров
1190 руб.	3570 руб.	7140 руб.

Подписку можно оформить с любого месяца.

Тел./факс: +7 (495) 987-43-74 (доб. 33-04)

Руководитель службы маркетинга А. А. Розанцев
тел. +7 (495) 987-43-74 (доб. 33-04), e-mail: arozantsev@synergy.ru

Подписка на почте

По каталогу агентства «Роспечать»	индекс 20497
По объединенному каталогу «Пресса России»	индекс 88059
По каталогу российской прессы «Почта России»	индекс 14241

Доставка осуществляется заказной бандеролью с уведомлением.

Электронный выпуск, а также отдельные статьи журнала можно приобрести на сайтах www.appliedinformatics.ru, www.elibrary.ru и www.litres.ru.

К оплате принимаются все виды электронных платежей, банковские карты.
Возможна также оплата с помощью SMS.

Учредитель и издатель

Негосударственное образовательное частное учреждение высшего образования
«Московский финансово-промышленный университет «Синергия»
Свидетельство о регистрации ПИИ № ФС 77-57869 от 25.04.2014 г.

Выдано Федеральной службой по надзору в сфере связи, информационных технологий
и массовых коммуникаций (Роскомнадзор)

Литературный редактор *А. К. Наумко*

Верстка, дизайн макета *Б. В. Зипунов*

Адрес редакции и издателя

129090, Москва, ул. Мещанская, д. 9/14, стр.1 (юрид.)
125190, Россия, Москва, Планетная ул., д. 36, оф. 301, 302
Тел.: +7 (495) 987-43-74 (доб. 33-04)
e-mail: appliedinformaticsjournal@gmail.com; www.appliedinformatics.ru

Наши реквизиты

Университет «Синергия»
ИНН 7729152149
КПП 770201001
ОГРН 1037700232558
Р/с 40703810338180120073
БИК 044525225
К/с 3010181040000000225
Сбербанк России (ПАО) Вернадское ОСБ

При перепечатке и цитировании материалов ссылка на журнал «Прикладная информатика» обязательна.
Редакция не несет ответственности за достоверность информации, опубликованной в рекламных объявлениях.
Мнения авторов и редакции могут не совпадать.

© Университет «Синергия»
Подписано в печать: 17.08.2018

Тираж 1000 экз.

Отпечатано в ООО «САМ ПОЛИГРАФИСТ»
129090, г. Москва, Протопоповский пер., д. 6