СИСТЕМЫ И СРЕДСТВА ИНФОРМАТИКИ

**Научный журнал Российской академии наук
(издается под руководством Отделения нанотехнологий
и информационных технологий РАН)**

Издается с 1989 года
Журнал выходит ежеквартально

**Учредители:
Российская академия наук
Федеральный исследовательский центр
«Информатика и управление» Российской академии наук**

РЕДАКЦИОННЫЙ СОВЕТ

академик РАН И. А. Соколов — председатель Редакционного совета
академик РАН Г. И. Савин
академик РАН А. Л. Стемпковский
член-корреспондент РАН Ю. Б. Зубарев
профессор Ш. Долев (S. Dolev, Beer-Sheva, Israel)
профессор Ю. Кабанов (Yu. Kabanov, Besancon, France)
профессор М. Никулин (M. Nikulin, Bordeaux, France)
профессор В. Ротарь (V. Rotar, San-Diego, USA)
профессор М. Финкельштейн (M. Finkelstein, Rostok, Germany)

РЕДАКЦИОННАЯ КОЛЛЕГИЯ

академик РАН И. А. Соколов — главный редактор
профессор, д.ф.-м.н. С. Я. Шоргин — заместитель главного редактора
д.т.н. В. Н. Захаров
проф., д.т.н. В. Д. Ильин
проф., д.ф.-м.н. Л. А. Калиниченко
д.т.н. В. А. Козмидиади
проф., д.т.н. К. К. Колин
проф., д.ф.-м.н. В. Ю. Королев
проф., д.г.-м.н. Р. Б. Сейфуль-Мулюков
проф., д.т.н. И. Н. Синицын
к.т.н. А. В. Филин
к.ф.-м.н. С. А. Христочевский

Редакция

профессор, д.г.-м.н. Р. Б. Сейфуль-Мулюков
к.ф.-м.н. Е. Н. Арутюнов
С. Н. Стригина (ответственный секретарь)

© Федеральный исследовательский центр «Информатика
и управление» Российской академии наук, 2015

Журнал входит в систему Российского индекса научного цитирования (РИНЦ):
http://elibrary.ru/title_about.asp?id=28980
Журнал включен в базу данных CrossRef (систему DOI — Digital Object Identifier),
в базу данных Ulrich's periodicals directory
и в информационную систему «Общероссийский математический портал Math-Net.Ru»

Журнал реферируется в «Реферативном журнале» ВИНИТИ
и в системе Google Scholar

Журнал «Системы и средства информатики»
включен в «Перечень российских рецензируемых журналов,
в которых должны быть опубликованы основные научные результаты диссертаций
на соискание ученых степеней доктора и кандидата наук», утвержденный ВАК

http://www.ipiran.ru/journal/collected

СИСТЕМЫ И СРЕДСТВА ИНФОРМАТИКИ

Том 25 № 1 Год 2015

СОДЕРЖАНИЕ

Моделирование грид и облачных сервисов как важный этап их разработки
В. В. Кореньков, А. В. Нечаевский, Г. А. Ососков, Д. И. Пряхина, В. В. Трофимов, А. В. Ужинский **4**

Комбинирование тезаурусных и корпусных знаний для извлечения оценочных слов
Н. В. Лукашевич, И. И. Четверкин **20**

Многокритериальный метод выявления нечетких дубликатов в потоке текстовых сообщений
А. М. Андреев, Д. В. Березкин, И. А. Козлов, К. В. Симаков **34**

Генерация тестового набора на основе потока управления
Н. В. Воинов, П. Д. Дробинцев, В. П. Котляров, И. В. Никифоров, И. А. Селин **54**

Анализ покрытия UCM-модели тестовыми сценариями
Н. В. Воинов, П. Д. Дробинцев, В. П. Котляров, И. В. Никифоров **74**

Обобщенный табличный LL-анализ
С. В. Григорьев, А. К. Рагозина **89**

Синтез устойчивых линейных фильтров и экстраполяторов Пугачёва для стохастических систем с мультипликативными широкополосными шумами
И. Н. Синицын, Э. Р. Корепанов **108**

О сходимости случайных сумм независимых случайных векторов к многомерным обобщенным дисперсионным гамма-распределениям
А. Ю. Корчагин **127**

СИСТЕМЫ И СРЕДСТВА ИНФОРМАТИКИ

Том 25 № 1 Год 2015

СОДЕРЖАНИЕ

Прогнозирование объемов железнодорожных грузоперевозок по парам веток
Р. К. Газизуллина, М. М. Стенина, В. В. Стрижов **142**

Некоторые подходы к формированию нормативно-технической базы единого информационного пространства России в части информационных ресурсов
А. А. Зацаринный, Э. В. Киселев **155**

Технология предотвращения дублирования библиографических описаний в базе данных научных публикаций БИАС ИПИ РАН
М. Ю. Заикин, В. С. Долгополов, О. Л. Обухова, И. В. Соловьев **168**

Построение системы управления потоками пациентов
Г. Я. Илюшин, В. И. Лиманский **186**

Информационные ресурсы контрастивных лингвистических исследований: типологические базы данных
М. Г. Кружков **198**

О вкладе академика И. А. Мизина в теорию и практику создания отечественных информационно-телекоммуникационных систем: к 80-летию со дня рождения
И. А. Соколов, А. А. Зацаринный, В. Н. Захаров **213**

Об авторах **230**

Правила подготовки рукописей статей **233**

Requirements for manuscripts **237**

МОДЕЛИРОВАНИЕ ГРИД И ОБЛАЧНЫХ СЕРВИСОВ КАК ВАЖНЫЙ ЭТАП ИХ РАЗРАБОТКИ*

***В. В. Кореньков*[1], *А. В. Нечаевский*[2], *Г. А. Ососков*[3], *Д. И. Пряхина*[4], *В. В. Трофимов*[5], *А. В. Ужинский*[6]**

Аннотация: Представлена новая система моделирования грид и облачных сервисов, ориентированная на повышение эффективности разработки системы хранения и обработки данных в ВЦ ЛИТ ОИЯИ, входящего в инфраструктуру WLCG в качестве центра уровней Tier1 и Tier2. Описаны предлагаемые принципы моделирования и их реализация в программном комплексе SyMSim, где реализован подход с учетом качества работы уже функционирующей системы при проектировании ее дальнейшего развития за счет объединения самой программы моделирования с системой мониторинга реального (или модельного) грид-облачного сервиса через специальную базу данных. Приведен пример применения программы для моделирования общей облачной структуры, которая может быть также использована и вне рамок физического эксперимента.

Ключевые слова: имитационная модель; распределенное хранение данных; облачные вычислительные центры; большие данные; оптимизация; мониторинг

DOI: 10.14357/08696527150101

1 Введение

Стремительный прогресс компьютерных технологий, программных средств и взрывное развитие глобального информационного пространства, возникшего с появлением Интернета, объединившего между собой компьютерные сети во всемирную систему передачи информации с помощью информационно-вычислительных ресурсов, — все это ознаменовало вступление человечества в новую эру больших данных (Big Data). В качестве одного из наиболее впечатляющих

*Статья рекомендована к публикации в журнале Программным комитетом конференции «Электронные библиотеки: перспективные методы и технологии, электронные коллекции» (“Digital Libraries: Advanced Methods and Technologies,” RCDL-2014), г. Дубна, 13–16 октября 2014 г. Работа выполнена при поддержке РФФИ (проект 14-07-00215).

[1]Объединенный институт ядерных исследований, korenkov@cv.jinr.ru

[2]Объединенный институт ядерных исследований, nechav@jinr.ru

[3]Объединенный институт ядерных исследований, ososkov@jinr.ru

[4]Объединенный институт ядерных исследований, pry-darya@yandex.ru

[5]Объединенный институт ядерных исследований, tvv@jinr.ru

[6]Объединенный институт ядерных исследований, zalexandr@list.ru

примеров больших данных приводят потоки экспериментальных данных физики высоких энергий, поступающие с Большого адронного коллайдера (БАК) в ЦЕРНе [1]. За время первого запуска БАК в 2012 г. четыре экспериментальные установки на нем ALICE, ATLAS, CMS и LHCb выдавали каждую секунду один петабайт (10^8 байт) данных. Запомнить такое количество данных невозможно ни на какой из современных вычислительных систем. Поэтому после сверхбыстрой сложной электронной предобработки, оставлявшей только одно полезное физическое событие из 10 тысяч, выполнялся их анализ в ЦЕРНовском компьютерном центре обработки из многих тысяч процессоров. После этого анализа оставался только 1% событий, возможно содержащих искомый физический феномен. Но даже после такого радикального сокращения потока экспериментальных данных в миллион раз для этих четырех больших экспериментов требовалось хранить поступающие в год 25 петабайт данных в специальных роботизированных ленточных хранилищах, так как копии этих данных подлежат передаче в сотни физических центров в 36 странах мира для более тщательного анализа на сотнях тысячах компьютеров, объединенных во Всемирную сеть распределенных вычислений — Worldwide LHC Computing Grid (WLCG). Ежедневно в WLCG обрабатываются полтора миллиона заданий, на что одному даже самому мощному современному компьютеру потребовалось бы 600 лет. Сравнение БАК по общим объемам перерабатываемых в 2012 г. данных с тем, что выполняется в социальных сетях, поисковых системах, разных отраслях бизнеса, медицины, климатических прогнозов и др. наглядно показывает, что исследования в ЦЕРНе идут в условиях больших данных.

На торжестве 4 июля 2012 г. по поводу получения ЦЕРНом Нобелевской премии за открытие бозона Хиггса директор ЦЕРНа Рольф Хойер прямо назвал грид-технологии одним из трех столпов успеха (наряду с ускорителем LHC (Large Hadron Collider) и физическими установками). Этот успех также подтверждает, что ЦЕРН, войдя в эру больших данных, является одним из примеров (наряду с созданием в ЦЕРНе WWW — Всемирной паутины), когда разработки в области физики частиц начинают влиять на исследования в других научных областях.

Следует отметить, что в 2015 г. предстоит вторичный запуск БАК после его существенной модернизации, когда поток данных возрастет в 2,5 раза при удвоении времени на их обработку. В своих планах на такое развитие компьютинга в ЦЕРНе после 2015 г., которое смогло бы обеспечить потенциально новую физику, помимо значительного увеличения вычислительных мощностей и ресурсов хранения данных, совершенствования программных средств анализа и моделирования, активного использования распределенных параллельных вычислений предлагается повысить эффективность распределенных систем вычислений на базе WLCG путем синтеза грид и облачных технологий [2].

Такие же планы характерны для ведущих физических центров мира, в том числе и для ОИЯИ.

Жесткая структура системы грид создавалась для интеграции уже существующих аппаратных и программных ресурсов, зафиксированных в системе,

в то время как облачная структура распределенных вычислений оказывается более гибкой, используя виртуальные кластеры. Как было показано в недавно защищенной в ОИЯИ диссертации Н. А. Кутовского, включение в грид с его жесткой структурой, ориентированной на интеграцию уже существующих процессорных и программных ресурсов, облачных структур, более гибких за счет использования виртуальных кластеров из виртуальных вычислителей, позволяет сократить время решения широкого круга задач в области физики высоких энергий и повысить эффективность использования ресурсов [3].

Примером уже имеющейся технологии, реализующей подобный синтез для работы с большими данными, является система PanDA (Production and Distributed Analysis — обработка данных и распределенный анализ) эксперимента ATLAS на LHC. Сегодня PanDA рассматривается как возможная система для российского мегапроекта NICA в Объединенном институте ядерных исследований (г. Дубна) [4].

Следует подчеркнуть, что в силу своей сложности и высокой стоимости разработка таких сложнейших грид-облачных систем сбора, передачи и распределенной обработки сверхбольших объемов информации требует больших предварительных исследований по выбору их оптимальной структуры с учетом стоимости и предполагаемых ресурсов и загрузки. Подобные исследования должны основываться на тщательном моделировании как потока заданий с учетом их типов и статистических данных о распределении времени их поступления и требуемых компьютерных ресурсов для их выполнения, так и состава моделируемой грид-структуры.

Настоящая работа посвящена разрабатываемой в ЛИТ ОИЯИ новой системе моделирования грид и облачных сервисов, ориентированной на повышение эффективности их разработки путем учета качества работы уже функционирующей системы в прогнозах на ее дальнейшее развитие. Это предлагается сделать за счет объединения самой программы моделирования с системой мониторинга реального (или модельного) грид-облачного сервиса через специальную базу данных, осуществляющую сбор и статистический анализ по вычислению распределений данных мониторинга, используемых затем для динамической коррекции параметров моделирования.

2 Вычислительный центр ЛИТ ОИЯИ и его специфика

Обсуждаемая работа выполняется сотрудниками ВЦ ЛИТ ОИЯИ [5]. Вычислительный центр ЛИТ имеет ряд особенностей. Динамика параметров центра приведена в табл. 1. Если рассматривать с точки зрения вычислительной мощности, или объемов хранения информации, или числа пользователей, то ВЦ нельзя отнести к большим вычислительным установкам, таким как Центр вычислений ЦЕРНа [6]. Отличием ВЦ ЛИТ от многих других центров, ориентированных на обработку научной информации, является разнообразие архитектур

Таблица 1 Параметры вычислительного центра ЛИТ ОИЯИ по годам

Компоненты ресурсного уровня	2010	2011	2012	2013	2014	2015
Наращивание производительности ЦИВК: CPUkSI2K	2500	4000	4000	7000	7000	10 000
Наращивание дисковой системы хранения, ТБ	1200	2000	2000	4000	4000	8 000
Наращивание системы массовой памяти, ТБ	500	2000	2000	5000	5000	10 000

вычислительных ресурсов, типов потоков задач и требований пользователей. При этом наращивание мощностей ВЦ демонстрирует хороший рост.

Следует указать, что изменения архитектуры и потоков задач выходят за рамки сложившихся в IT типовых центров обработки данных (ЦОД). За последние 3 года в состав центра вошли облачный кластер, GPU-ферма, разворачивается ЦОД уровня Tier1 WLCG [7], при этом сохраняется сайт Tier2 [8], что является нестандартным решением.

Приведенные выше факты говорят о том, что сотрудники центра имеют большой опыт проектирования, эксплуатации и развития вычислительных установок. На основании этого опыта можно утверждать, что процесс эксплуатации и развития требует создания специализированного инструмента, который позволит предсказывать последствия принимаемых технических решений. Этот подход не противоречит накопленному опыту проектирования ЦОД, отраженному в стандартах [9]. В то же время разработка единого перспективного плана развития центра распределенных вычислений с использованием грид-облачных технологий вряд ли может быть решена в данном случае на основании стандартов, применяемых для отраслевых центров или ориентированных на определенный сегмент рынка, в силу нестандартности задач, которые решает ВЦ. Требуется новый инструмент для анализа проектных решений на основании вычислительного эксперимента, в котором моделируются и структура центра, и предполагаемые загрузки. Такой подход является обычным для отраслей, где создаются и эксплуатируются сложные дорогостоящие или опасные в эксплуатации объекты. В рамках этого подхода параллельно с объектом на всех этапах его существования создается модель, математическая, электронная или программная, которая вместе с объектом модернизируется. В практике эксплуатации ВЦ ЛИТ такой подход предлагается впервые.

3 Подход к моделированию

Предлагаемый подход состоит в имитационном моделировании, объединенном с мониторингом процессов прохождения задач и передачи данных. Эффективность этой концепции состоит в том, что техническое решение проверяется

на модели прежде, чем обсуждается его фактическая реализация. Развернутое обоснование предлагаемого подхода, а также описание функций мониторинга, используемых при моделировании, дано в [10], поэтому отметим здесь, что модель должна рассматриваться как неотъемлемая часть системы обработки данных, а данные мониторинга — как входные для моделирования. Это позволит принимать обоснованные проектные решения при развитии системы.

Центральным компонентом такой процедуры принятия решения по развитию вычислительной установки является имитационная модель вычислительной структуры, в которую в качестве входной поступает информация, накапливаемая в ходе мониторинга существующей установки в специальной базе данных, где она модифицируется в соответствии с планами развития моделируемой грид-облачной структуры.

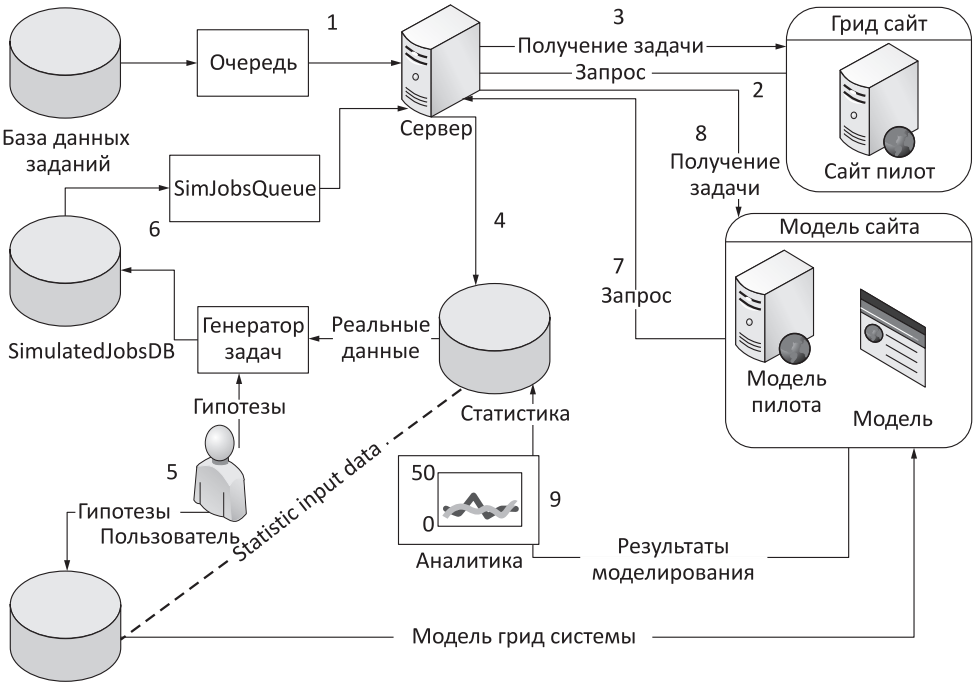


Рис. 1 Схема программы SyMSim, предназначенной для моделирования системы распределенных вычислений с учетом данных мониторинга этой системы: Мониторинг: 1 — отправка заданий на сервер; 2 — запрос задания; 3 — отправка задания на исполнение; 4 — запись статистики. Моделирование: 5 — модель системы с учетом реальных данных мониторинга, генерация входных данных для модели; 6 — имитация потока заданий; 7 — запрос задания; 8 — отправка задания на исполнение; 9 — анализ результатов работы, запись статистики

Схема программы SyMSim (Synthesis of Monitorng and Simulation — Синтез Мониторинга и Моделирования), реализующей идею синтеза процессов мониторинга и моделирования, представлена на рис. 1.

4 Область моделирования

Объектами моделирования являются вычислительные установки, предназначенные для обработки потоков информации объемом до десятков петабайтов в год, поступающей с детекторов, которые производят ускорители заряженных частиц — как существующие LHC-CMS [11], LHC-Atlas [12], так и находящиеся в процессе создания или проектирования FAIR-PANDA [13], BES-III [14], NICA-MPD [15]. Как отмечалось выше, опыт мировых физических экспериментальных центров показывает, что единственно возможным способом хранения объемов информации, производимых такими детекторами, является использование роботизированных библиотек. Данные затем обрабатываются на фермах, включающих тысячи процессоров. Как правило, структуры обработки данных выполняются в парадигме грид и включают в себя центры с облачной архитектурой. Специфика данных заключается в том, что поток может быть разбит на события, каждое из которых обрабатывается независимо от остальных.

Набор программных средств модели создается в течение нескольких лет. Если на первом этапе рассматривалось моделирование систем, построенных в парадигме грид, то в настоящее время проект сосредоточен на моделировании облачных структур, интегрированных в глобальные грид-структуры.

Как это принято при имитационном моделировании потока дискретных событий, происходящие в модели события привязываются к внутреннему времени. Результатом работы модели является величина времени обработки пакета и понимание того, как структура вычислительной установки и производительность отдельных ее частей влияют на это время. Другой вопрос, на который получают ответ при моделировании: какими резервами обладает вычислительная установка, т. е. какова верхняя граница интенсивности потока задач или данных и какие компоненты установки оказывают на нее существенное влияние?

Объекты модели включают задачи, процессоры, файлы, ленты, дисковые хранилища, линии передачи данных, роботизированные библиотеки. Список событий, происходящих с объектами, включает: поступление задачи в очередь, извлечение задачи из очереди, занятие или освобождение вычислительного ресурса, передачу файла, извлечение ленты из хранилища, ее монтирование, чтение/запись на ленту и др. Модель построена в виде программы имитации потока дискретных событий на основе библиотеки GridSim [16].

Архитектура имитационной модели была адаптирована из распространяемого по лицензии LGPL (Lesser General Public License) кода проекта Alea [17].

Отличительной особенностью предлагаемой системы моделирования является ее интеграция с системами мониторинга вычислительного процесса. В частности, используется мониторинг системы запуска заданий PanDA [18].

5 Компоненты модели, архитектура программы, входные и выходные данные

Имитационная модель состоит из блоков, состав и связи которых показаны на рис. 2.

Представленная структура в главных чертах отражает задачи, решаемые при накоплении информации с ее параллельной обработкой, и соответствует структуре центра уровня Tier1.

Объекты реализуются в виде классов JAVA, при этом пользователь только задает внешнее описание их количества и связей между ними, но не изменяет собственно текст программы. В этом существенное отличие рассматриваемой программы от набора библиотек, которые требуют от пользователя программных усилий.

Средство описания вычислительной структуры реализовано как база данных с веб-интерфейсом. Описанию присваивается идентификатор, который пользователь должен указать в параметрах запуска модели. Модель считывает информацию из базы и строит в памяти описание вычислительной структуры. Другими данными являются характеристики потока задач, которые подлежат обработке, и информация. Блок описания потока заданий — набор утилит, который позволяет статистически проанализировать результаты мониторинга

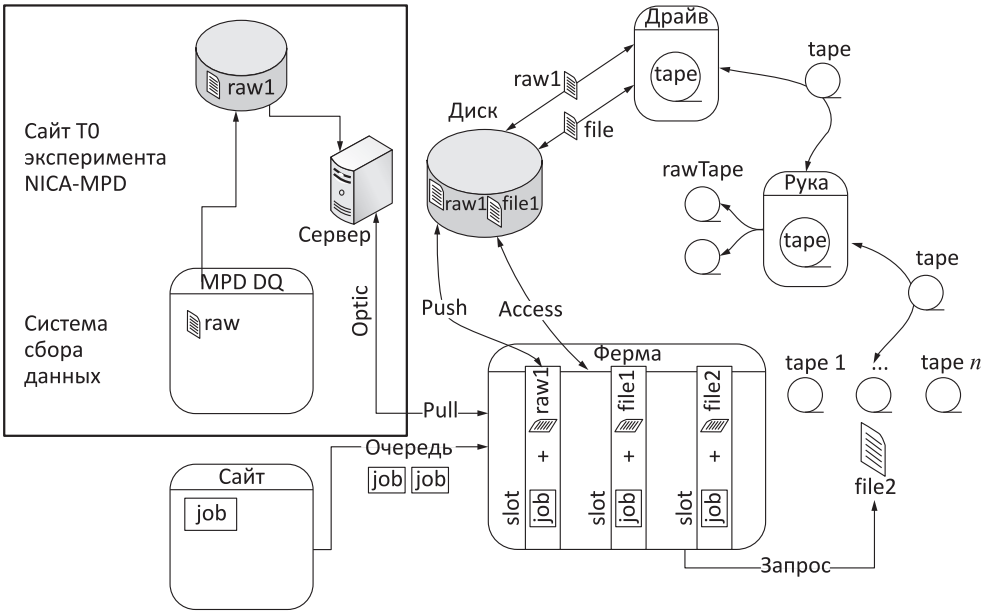


Рис. 2 Схема прохождения заданий через SyMSim

и сформировать поток заданий, аналогичный или отличающийся от проанализированного на управляемое воздействие пользователя, в виде упорядоченной по времени последовательности записей в базе данных. Для обеспечения совместимости с системами мониторинга формат базы данных был выбран совпадающим с PanDA — системой управления потоком задач эксперимента Atlas. Такая совместимость дает возможность использовать результаты мониторинга потока без изменения его параметров.

Существует множество систем управления, выбор PanDA обусловлен тем, что эта система в последнее время рассматривается как перспективная для нескольких новых или модернизируемых экспериментов. Для того чтобы не генерировать поток заданий всякий раз, когда необходимо изменить его интенсивность без измерения распределения типов задач, во входных параметрах программы определен параметр компрессии потока. На этот коэффициент умножается «временно́е расстояние» между поступающими в систему задачами. Аналогичный способ используется для задания интенсивности потока информации.

6 Внутренняя структура программы моделирования

Ядром модели является планировщик — модуль, который в программе отвечает за прием заданий на обработку и «запуск» их на процессорах. Планировщик управляет очередями, сохраняет информацию о занятых и свободных в данный момент процессорах. Текст планировщика заимствован из проекта Alea с существенными модификациями и сокращением функций.

Другой существенный по объему набор классов описывает миграцию информации в системе. Этот набор был создан как расширение библиотеки GridSim. При его создании авторы отказались от детализации процесса передачи информации на уровне пакетов в пользу моделирования последовательности событий манипуляции с файлами: запрос–открытие–передача–закрытие. При этом рассчитывается уровень нагрузки на среду передачи, а не время поступления пакета на узел.

Результатом работы программы моделирования служит файл (или последовательность записей в базе данных, отражающая все события, происходящие в системе). К ним относится, например, поступление задания, начало и конец обработки задания, начало и конец передачи файла, манипуляции с лентами и т. д. Все события описываются в едином формате. Запись привязывается к внутреннему времени.

По ходу развития проекта из текста модели были удалены все модули сбора статистики. Необходимость такого подхода выяснилась после того, как авторы стали анализировать результаты моделирования проекта NICA-MPD [10, 18]. Поскольку невозможно априори сформулировать весь список параметров, подлежащих анализу, целесообразно сосредоточиться на получении информации о событиях, связанных с прохождением каждого задания через вычислительную структуру, а статистические закономерности этого процесса анализируются

и визуализируются отдельно. В данном случае для этого были использованы таблицы Excel. Для большего удобства планируется расширить проект отдельным модулем, предоставляющим доступ к результатам моделирования через веб-интерфейс.

Перед проектировщиками вычислительных установок естественным образом встает вопрос, как обеспечить масштабируемость, т. е. способность установки сохранять работоспособность при увеличении потоков данных и задач. В SyMSim настройка топологии установки и характеристик ее узлов конфигурируется через внешние параметры, что позволяет легко настраивать модель под различные варианты потоков. Наиболее простой параметр при настройке — это количество счетных узлов. Однако утверждать, что скорость обработки потока заданий будет линейно зависеть от него, неверно. На скорость будут оказывать влияние также пропускная способность каналов и производительность оборудования, на котором хранятся исходные данные. Возможность учета этих факторов заложена в SyMSim, но требует дальнейшего развития. Например, в программе невозможно учесть деградацию скорости передачи при увеличении загрузки канала. Поскольку зависимость эта нелинейная, то вопрос требует расширения функции SyMSim.

При создании модели надо также обращать внимание на предполагаемый срок службы оборудования. Например, счетные узлы могут заменяться в течение нескольких лет, а роботизированные библиотеки используются несколько пятилетий. Следовательно, на моделировании процессов в них должен быть сделан особый акцент.

Другая проблема, которая стоит перед разработчиками, — это обеспечение надежности функционирования. Как правило, такие задачи решаются резервированием оборудования и моделированием его отказов. В облачных структурах вопрос резервирования решается за счет использования механизма виртуализации, когда процесс может быть перенесен на реплику оборудования без существенных затрат. В грид-структурах для обеспечения вычислительного процесса используются десятки служб, отказ любой из них приводит к невозможности запуска заданий, по крайней мере, от отдельных виртуальных организаций. Отказ может происходить как по причине сбоя оборудования, так и по причине ошибки программного обеспечения. Поэтому в перспективе в SyMSim для моделирования грид-структур необходимо добавить классы, описывающие грид-службы как обеспечивающие процесс вычислений в целом, и моделировать отказ этих узлов как программно-аппаратный сбой.

7 Пример принятия решения на основе результатов моделирования, выполненного в SyMSim

Для иллюстрации преимуществ, которые дает моделирование при проектировании систем обработки, проанализируем параметры, заданные для системы

Таблица 2 Параметры оборудования в модели

Центр обработки	Количество ядер, шт.	Буфер, ТБ	Количество драйвов, шт.	Количество RAW-драйвов, шт.
Tier0	500	5	0	0
Tier1	500	5	12	2

Таблица 3 Параметры роботизированной библиотеки

Параметр	Значение	Единица измерения
Емкость общая	5	ПБ
Тип накопителя	LTO6	
Емкость картриджа	3	ТБ
Скорость записи	200	МБ/с
Время монтирования ленты	20	с
Время подачи ленты в накопитель	6	с

сбора и хранения данных детектора частиц. Исходные данные возьмем из предложения по концептуальному дизайну NICA-MPD [15]. Эта установка является типичной для класса спектрометров ядерной физики. Рассмотрим процесс передачи и сохранения «сырых данных», которые традиционно именуются RAW-файлами. Пусть центр обработки и хранения состоит из двух уровней: Tier0 и Tier1. Уровень Tier0 предназначен только для приема информации с детектора и ее временного хранения. Уровень Tier1 обеспечивает длительное хранения информации и одновременно ее обработку. Параметры оборудования приведены в табл. 2.

Кроме этого центр Tier1 оснащен роботизированной библиотекой [19] с параметрами, приведенными в табл. 3.

Для расчета используем значения потока информации из табл. 4.

В пересчете на час объем информации — 1750 ГБ. При таком потоке данных необходимо 3 ленточных драйва, чтобы не допустить переполнения буфера. Эти расчеты не учитывают возможной компрессии сохраняемых данных,

Таблица 4 Параметры потока данных

Параметр	Значение	Единица измерения
Скорость поступления информации с детектора	4,7	ГБ/с
Длительность периода набора данных	120	дней в году
Эффективность сбора данных	60	%
Полный объем исходной информации	8,4	ПБ/год
Объем записи за сутки	4,6	ТБ
Размер файла RAW	6	ГБ
Файлов записано в час	32	

Таблица 5 Зависимость времени прохождения потоков заданий от интенсивности сбора данных

Эксперимент	Уровень загрузки, %	Время выполнения, с
1	10	21 000
2	20	20 821
3	40	20 470
4	60	20 460

которая может быть значительной. Расчеты начнем с потока данных в 10% от максимального и будем считать, что компрессии не производится.

Файлы RAW должны быть записаны на ленты роботизированной библиотеки. Параллельно этому процессу на Tier1 выполняются задания обработки. Будем анализировать поток из 1000 заданий. Каждое задание требует единственный файл, копия которого может находиться на дисковом массиве или на ленте. В последнем случае он должен быть загружен на диск автоматически, когда к нему обращается задание. Копии файлов на дисковых накопителях могут стираться сборщиком мусора, если к ним долгое время нет обращений. Такой алгоритм работы с файлами реализован в системе dCache [20].

Проанализируем уровень загрузки пулов Tier0 и Tier1, а также нагрузку на устройство смены лент библиотеки. Для этого моделируем процесс записи файлов на ленты в условиях, когда интенсивность сбора данных возрастает от 10% до 60% максимальной интенсивности. Время прохождения потока из 1000 заданий указано в табл. 5.

При заданной конфигурации центров Tier0 и Tier1 загрузка и запись файлов не оказывают влияния на процесс выполнения. Причина этого становится понятна, если проанализировать загрузку устройства смены лент.

Интенсивность использования устройства смены лент вычисляется как отношение числа загрузок за период времени к максимально возможному за этот период.

Результат моделирования приведен на рис. 3 в процентах от максимальной загрузки.

На графике рис. 3 видно, что загрузки с увеличением интенсивности потока информации растут, но запас значительный. Теперь исследуем количество свободного места в буферной памяти Tier0. Рассмотрим интенсивность потока в 10% и 40%.

График на рис. 4 показывает, что система справляется с потоком данных, если он небольшой, и уже при потоках в 40% от максимума начинаются задержки в передаче. Они связаны с тем, что сборщик мусора Tier0 не может удалить файлы, которые еще не переданы на Tier1 и не записаны на ленты. Следовательно, надо анализировать цепочку передачи, включая пропускную способность каналов, количество накопителей на лентах и величину дискового буфера. Решением

Рис. 3 Результат моделирования загрузки ленточного робота в зависимости от интенсивности потока: *1* — 10%; *2* — 20%, *3* — 30%, *4* — 40%

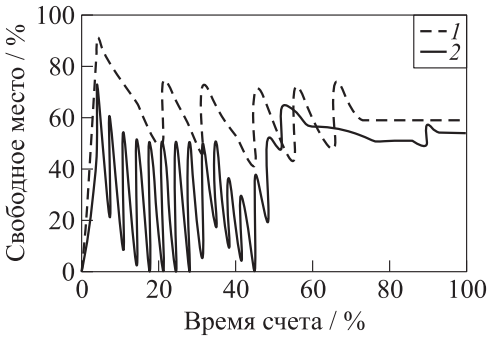


Рис. 4 Наличие свободного места в буферной памяти в процентах к максимуму в зависимости от интенсивности потока: *1* — 10%, *2* — 40%

проблемы может стать как увеличение количества ресурсов, так и изменение процедуры записи. В частности, возможно, следует отказаться от заложенной в TDR процедуры и перейти к прямой записи данных на ленту.

Приведенный пример наглядно демонстрирует, что ширина канала записи RAW-файлов недостаточна для сохранения всей информации, ожидаемой с детектора. Несмотря на то что данный пример является достаточно тривиальным, он демонстрирует, что моделирование может ответить на вопрос, какой параметр окажет существенное влияние на процесс сохранения информации. Для более глубокого анализа процессов сбора и хранения данных на конкретной установке необходимо проводить заранее спланированные вычислительные эксперименты. Эта тема выходит далеко за рамки данной работы.

Разумеется, невозможно принять решение о конфигурации системы хранения и обработки данных только на основании вычисленных в процессе моделирования параметров. Для оптимального выбора необходимо учитывать множество факторов, таких как надежность, располагаемые финансовые ресурсы. Но приведенный пример демонстрирует, как с помощью SyMSim можно снять фактор неопределенности, который касается параметров процессов обработки.

8 Заключение

Разработка грид-облачных систем сбора, передачи и распределенной обработки информации требует тщательного моделирования, которое будет эффективным, только если оно будет учитывать качество работы уже функционирующей системы в прогнозах на ее дальнейшее развитие. Новизна подхода, предложенного авторами, состоит в соединении процессов моделирования и мониторинга в рамках одного проекта.

Предложенный подход к моделированию и анализу вычислительных грид-облачных структур инфраструктуры эксперимента NICA основан на учете данных мониторинга, используемых затем для динамической коррекции параметров моделирования.

Дальнейшее развитие системы предполагает внесение дополнений для моделирования оригинальных алгоритмов назначения пулов dCache. Структура моделирующей программы достаточно общая, чтобы в дальнейшем заменять упрощения, допущенные на предварительном этапе, на более реальные условия. Также необходимо провести полномасштабные испытания модели с целью выявления ее ошибок.

Литература

1. Компьютинг в ЦЕРНе. http://home.web.cern.ch/about/computing.
2. *Hernández J. M.* On behalf of the Spanish LHC Computing community // Perspectives on LHC Computing. — Santiago de Compostela: Jornadas CPAN, 2013.
3. *Кутовский Н. А.* Развитие методов построения грид-сред и систем облачных вычислений для задач физики высоких энергий: Автореф. дисс. . . . канд. физ.-мат. наук. — Дубна: ОИЯИ, 2014. 20 с.
4. *Климентов А., Ваняшин А., Кореньков В.* За большими данными следит ПАНДА // Суперкомпьютеры, 2013. № 3(15). С. 56–61.
5. Лаборатория информационных технологий ОИЯИ. http://lit.jinr.ru.
6. Guide to CERN Computing Center. http://malandes.web.cern.ch/malandes/cc.html.
7. *Астахов Н. С., Белов С. Д., Горбунов И. Н., Дмитриенко П. В., Долбилов А. Г., Жильцов В. Е., Кореньков В. В., Мицын В. В., Стриж Т. А., Тихоненко Е. А., Трофимов В. В., Шматов С. В.* Автоматизированная система уровня Tier-1 обработки данных эксперимента CMS // Информационные технологии и вычислительные системы, 2013. № 4. С. 27–36.
8. *Chudoba J., Elias M., Fiala L., Horky J., Kouba T., Kundrat J., Lokajicek M., Schovancova J., Svec J., Belov S., Dmitrienko P., Dolbilov A., Korenkov V., Mitsyn V., Oleynik D., Petrosyan A., Russakovich N., Tikhonenko E., Trofimov V., Uzhinsky A., Zhiltsov V.* JINR (Dubna) and Prague Tier2 sites: Common experience in the WLCG grid infrastructure // Письма в ЭЧАЯ, 2013. Т. 10. № 3(180). С. 458–467.
9. Стандарты ЦОД. http://www.rvip.ru/1065/document1070.shtml#10.
10. *Кореньков В. В., Нечаевский А. В., Ососков Г. А., Пряхина Д. И., Трофимов В. В., Ужинский А. В.* Моделирование грид-облачных сервисов проекта NICA как средство повышения эффективности их разработки // Компьютерные исследования и моделирование, 2014. Т. 6. № 5. С. 635–642.
11. CMS detector. http://home.web.cern.ch/about/experiments/cms.
12. ATLAS detector. http://home.web.cern.ch/about/experiments/atlas.
13. Веб-портал проекта PANDA. http://www-panda.gsi.de.
14. Веб-портал проекта BESIII. http://bes3.ihep.ac.cn.

15. *Сисакян А. Н., Сорин А. С., Кекелидзе В. Д.* Многоцелевой детектор — MPD — для изучения столкновений тяжелых ионов на ускорителе NICA: Концептуальный дизайн-проект. Версия 1.4. http://nica.jinr.ru/files/CDR_MPD/MPD_CDR_ru.pdf.
16. GridSim: A grid simulation toolkit for resource modelling and application scheduling for parallel and distributed computing. http://www.gridbus.org/gridsim.
17. *Klusácek D., Matyska L., Rudová H.* Alea — grid scheduling simulation environment // Parallel processing and applied mathematics / Eds. R. Wyrzykowski, J. Dongarra, K. Karczewski, J. Wasniewski. Lecture notes in computer science ser. — Springer, 2008. Vol. 4967. P. 1029–1038.
18. *Кореньков В. В., Нечаевский А. В., Трофимов В. В.* Модель системы off-line обработки данных эксперимента НИКА // Системный анализ в науке и образовании: Электронный журнал, 2012. Вып. 4. http://www.sanse.ru/archive/26.
19. Ленточная библиотека IBM System Storage TS3500 Tape Library. http://www.ibm.com/ru/servers/storage/tape/ts3500.
20. Веб-портал проекта dCache. http://www.dcache.org.

Поступила в редакцию 30.12.14

GRID AND CLOUD SERVICES SIMULATION AS AN IMPORTANT STEP OF THEIR DEVELOPMENT

V. V. Korenkov, A. V. Nechaevskiy, G. A. Ososkov, D. I. Pryakhina, V. V. Trofimov, and A. V. Uzhinskiy

Joint Institute for Nuclear Research, 6 Joliot-Curie Str., Dubna, Moscow Region 141980, Russian Federation

Abstract: A new system for grid and cloud services simulation is described. It is focused on improving the efficiency of grid-cloud systems development by using work quality indicators of a real system to design and predict its evolution. For these purposes, the simulation program is combined with a real monitoring system of a grid-cloud service through a special database. The simulation principles and their implementation in the SyMSim software package are described. An example of using the program to simulate a general cloud structure is given.

Keywords: simulation; distributed data storage; cloud computing; Big Data; optimization; monitoring

DOI: 10.14357/08696527150101

Acknowledgments

The research was supported by the Russian Foundation for Basic Research (grant No. 14-07-00215).

References

1. Komp'yuting v TsERNe [Computing in CERN]. Available at: http://home.web.cern.ch/about/computing (accessed November 17, 2014).
2. Hernández, J. M. 2013. *Hernández J. M.* On behalf of the Spanish LHC Computing community. *Perspectives on LHC Computing*. Santiago de Compostela: Jornadas CPAN.
3. Kutovskiy, N. A. 2014. Razvitie metodov postroeniya grid-sred i sistem oblachnykh vychisleniy dlya zadach fiziki vysokikh energiy [Development of methods for constructing grid environments and cloud computing systems for high energy physics]. PhD Thesis. Dubna: Joint Institute of Nuclear Reseach. 20 p.
4. Klimentov, A., A. Vanyashin, and V. Koren'kov. 2013. Za bol'shimi dannymi sledit PANDA [PANDA watches the Big Data]. *Superkomp'yutery* [Supercomputers] 3(15):56–61.
5. Laboratoriya Informatsionnykh Tekhnologiy [Laboratory of Information Technologies]. Available at: http://lit.jinr.ru (accessed November 18, 2014).
6. Guide to CERN Computing Center. Available at: http://malandes.web.cern.ch/malandes/cc.html (accessed November 25, 2014).
7. Astakhov, N. S., S. D. Belov, I. N. Gorbunov, P. V. Dmitrienko, A. G. Dolbilov, V. E. Zhil'tsov, V. V. Koren'kov, V. V. Mitzyn, T. A. Strizh, E. A. Tikhonenko, V. V. Trofimov, and S. V. Shmatov. 2013. Avtomatizirovannaya sistema urovnya Tier-1 obrabotki dannykh eksperimenta SMS [Automated system for Tier-1 data CMS experiment]. *Informatsionnye Tekhnologii i Vychislitel'nye Sistemy* [Information Technology and Computer Systems] 4:27–36.
8. Chudoba, J., M. Elias, L. Fiala, J. Horky, T. Kouba, J. Kundrat, M. Lokajicek, J. Schovancova, J. Svec, S. Belov, P. Dmitrienko, A. Dolbilov, V. Korenkov, V. Mitsyn, D. Oleynik, A. Petrosyan, N. Russakovich, E. Tikhonenko, V. Trofimov, A. Uzhinsky, and V. Zhiltsov. 2013. JINR (Dubna) and Prague Tier2 sites: Common experience in the WLCG grid infrastructure. *Pis'ma v EChAYa* [Pepan Letters] 10(3):458–467.
9. Standarty TsOD [Standards for Data Centers]. Available at: http://www.rvip.ru/1065/document1070.shtml#10 (accessed November 10, 2014).
10. Koren'kov, V. V., A. V. Nechaevsky, G. A. Ososkov, D. I. Pryakhina, V. V. Trofimov, and A. V. Uzhinskiy. 2014. Modelirovanie grid-oblachnykh servisov proekta NICA kak sredstvo povysheniya effektivnosti ikh razrabotki [NICA grid-cloud services simulation as a means of improving their development]. *Komp'yuternye issledovaniya i modelirovanie* [Computer Research and Modeling] 6(5):635–642.
11. CMS detector. Available at: http://home.web.cern.ch/about/experiments/cms (accessed November 11, 2014).
12. ATLAS detector. Available at: http://home.web.cern.ch/about/experiments/atlas (accessed November 27, 2014).
13. PANDA web-page. Available at: http://www-panda.gsi.de/ (accessed November 14, 2014).
14. BES-III web-page. Available at: http://bes3.ihep.ac.cn/ (accessed November 14, 2014).

15. Sissakian, A., and A. Sorin. 2014. Multi-MPD detector for the study of heavy-ion collisions at the accelerator NICA (Conceptual design project). v. 1.4. Available at: http://nica.jinr.ru/files/CDR_MPD/MPD_CDR_ru.pdf (accessed November 17, 2014).
16. GridSim: A grid simulation toolkit for resource modelling and application scheduling For parallel and distributed computing. Available at: http://www.gridbus.org/gridsim/ (accessed November 20, 2014).
17. Klusácek, D., L. Matyska, and H. Rudová. 2008. Alea — grid scheduling simulation environment. *Parallel processing and applied mathematics*. Eds. R. Wyrzykowski, J. Dongarra, K. Karczewski, and J. Wasniewski. Lecture notes in computer science ser. Springer. 4967:1029–1038.
18. Koren'kov, V., A. Nechaevsky, and V. Trofimov. 2012. Model' sistemy off-line obrabotki dannykh eksperimenta NIKA [Offline data processing simulation for NICA]. *Sistemnyy Analiz v Nauke i Obrazovanii* [System Analysis in Science and Education]. Available at: http://sanse.ru/download/146 (accessed November 25, 2014).
19. IBM System Storage TS3500 Tape Library. Available at: http://www.ibm.com/ru/servers/storage/tape/ts3500/ (accessed November 17, 2014).
20. dCache. web-page. Available at: http://www.dcache.org (accessed November 10, 2014).

Received December 30, 2014

Contributors

Korenkov Vladimir V. (b. 1953) — Doctor of Science in technology, professor, Head of Department, Dubna University; Director of Laboratory, Joint Institute for Nuclear Research, 6 Joliot-Curie, Dubna, Moscow Region 141980, Russian Federation; korenkov@cv.jinr.ru

Nechaevskiy Andrey V. (b. 1982) — software engineer, Joint Institute for Nuclear Research, 6 Joliot-Curie, Dubna, Moscow Region 141980, Russian Federation; nechav@jinr.ru

Ososkov Gennadij A. (b. 1931) — Doctor of Science in physics and mathematics, professor, principal scientist, Joint Institute for Nuclear Research, 6 Joliot-Curie, Dubna, Moscow Region 141980, Russian Federation; ososkov@jinr.ru

Pryakhina Dar'ja I. (b. 1992) — software engineer, Joint Institute for Nuclear Research, 6 Joliot-Curie, Dubna, Moscow Region 141980, Russian Federation; pry-darya@yandex.ru

Trofimov Vladimir V. (b. 1955) — leading programmer, Joint Institute for Nuclear Research, 6 Joliot-Curie, Dubna, Moscow Region 141980, Russian Federation; tvv@jinr.ru

Uzhinskiy Aleksandr V. (b. 1983) — Candidate of Science (PhD) in technology, leading programmer, Laboratory of Information Technologies, Joint Institute for Nuclear Research, 6 Joliot-Curie, Dubna, Moscow Region 141980, Russian Federation; zalexandr@list.ru

КОМБИНИРОВАНИЕ ТЕЗАУРУСНЫХ И КОРПУСНЫХ ЗНАНИЙ ДЛЯ ИЗВЛЕЧЕНИЯ ОЦЕНОЧНЫХ СЛОВ*

***Н. В. Лукашевич*[1], *И. И. Четверкин*[2]**

Аннотация: Описан комбинированный подход к извлечению оценочных слов в заданной предметной области. На первом этапе упорядоченный список оценочных слов извлекается из текстов предметной области на основе методов машинного обучения. На втором этапе полученный список уточняется за счет привлечения лексических знаний, описанных в тезаурусе русского языка. Данная комбинированная модель была обучена на предметной области отзывов о фильмах, а затем была перенесена на другие предметные области. Кроме того, на основе объединения оценочных списков данных предметных областей был получен улучшенный список оценочных слов в широкой области товаров и услуг ProductSentiRus+.

Ключевые слова: анализ тональности текстов; автоматическая обработка текстов; отзывы пользователей; тезаурус русского языка

DOI: 10.14357/08696527150102

1 Введение

Автоматический анализ тональности текстов на естественном языке, подразумевающий автоматическое определение отношения автора (положительное, отрицательное) к теме текста, является одним из активно развивающихся направлений в сфере автоматической обработки текстов.

Важной задачей в рамках этого направления является задача автоматизации создания и улучшения качества словарей оценочных слов и выражений. В настоящее время для многих языков создаются такого рода ресурсы оценочной лексики [1, 2]. Для русского языка опубликован и свободно доступен лишь один автоматический порожденный словарь оценочных слов и выражений ProductSentiRus [3].

Кроме того, анализ тональности текстов в предметной области, например отзывов пользователей некоторого продукта, требует адаптации систем анализа тональности к текстам этой предметной области [4]. Так, некоторые слова могут

*Статья рекомендована к опубликованию в журнале Программным комитетом конференции «Электронные библиотеки: перспективные методы и технологии, электронные коллекции» (“Digital Libraries: Advenced Methods and Technologies,” RCDL-2014),г. Дубна, 13–16 октября 2014 г. Работа выполнена при частичной поддержке РФФИ (проект 14-07-00682).

[1]НИВЦ Московского государственного университета им. М. В. Ломоносова, louk_nat@mail.ru

[2]НИВЦ Московского государственного университета им. М. В. Ломоносова, ilia201@yandex.ru

терять свою полярность в конкретных областях. Например, такое слово, как *зло*, в предметной области отзывов о фильмах обычно относится к сюжету фильма и ничего не говорит об отношении пользователя к данному фильму.

Другие слова могут приобрести некоторую тональность в конкретной предметной области. Например, слово *киношный* может иметь негативную полярность, передавая значение «далекий от реальной жизни». Другой пример — слово *атмосферный*, которое в области отзывов о фильмах не имеет отношения к атмосфере планет, а несет положительную тональность создания хорошего настроения, чувства. Данное значение слова атмосферный является относительно новым в русском языке, не описано ни в каких русских толковых словарях.

Автоматическое извлечение оценочных слов может осуществляться на основе текстовых корпусов или словарных ресурсов (толковых словарей, тезаурусов и т. п.). В этой статье описывается комбинированный подход к извлечению словаря оценочной лексики. На первом шаге словарь оценочной лексики создается на основе извлечения признаков слов в нескольких текстовых корпусах и последующего применения методов машинного обучения для комбинирования этих признаков. Словарь представлен как список слов, упорядоченный по мере снижения вероятности слова быть оценочным. На этом этапе можно извлечь оценочные слова, которые отсутствуют в словарях или имеют тональность только в текущей предметной области. Оценочные слова извлекаются без их положительной или отрицательной оценки, поскольку этот этап рассматривается как первый шаг в порождении оценочного словаря предметной области.

На втором этапе производится улучшение качества полученного списка оценочных слов на основе сопоставления его с недавно опубликованным тезаурусом русского языка РуТез. Модель извлечения оценочных слов на основе корпусов и тезауруса настраивается на области отзывов о фильмах. Затем построенная модель используется для порождения оценочных словарей в четырех других областях. Наконец, созданные словари оценочных слов из пяти областей комбинируются, чтобы породить единый словарь оценочных слов в широкой области товаров и услуг (ProductSentiRus+).

Данная статья имеет следующую структуру. Во втором разделе рассматриваются методы порождения словарей оценочной лексики. В третьем разделе кратко представлена структура тезауруса русского языка РуТез, предназначенного для автоматической обработки текстов. Четвертый раздел представляет подход к извлечению оценочной лексики в заданной предметной области. Пятый раздел рассматривает порождение обобщенного словаря оценочной лексики в области товаров и услуг ProductSentiRus+. Полученный словарь тестируется в задаче классификации твитов на объективные и субъективные.

2 Близкие работы

Существуют два подхода к созданию словарей оценочной лексики: один основан на текстовых корпусах, а другой — на словарных ресурсах.

Корпусные методы используют совместную встречаемость слов [5–7] или появление их в заданных контекстах или лексико-синтаксических шаблонах [8]. Современные корпусные методы обрабатывают большие объемы данных, например миллионы веб-страниц [6] или сотни тысяч отзывов пользователей [9].

Словарные методы извлекают нужную информацию из существующих словарей или тезаурусов и обычно начинают свою работу с некоторого начального множества известных оценочных слов (*плохой, хороший, отличный, отвратительный* и др.). В [10] описывается ресурс SentiWordNet, который был получен в результате автоматической разметки тезауруса английского языка WordNet [11] и в котором каждому синсету WordNet (набору синонимов) сопоставляется три числовых веса, отражающих долю позитивности, негативности и нейтральности слов синсета. Таким образом, различные значения одного и того же слова, входящие в разные синсеты, могут иметь различные соотношения весов полярности.

В [12] авторы исследуют различные подходы с частичным обучением для разметки полярности слов в графе лексических отношений WordNet. В частности, применяются такие методы, как MinCut, Randomized MinCut, Label Propagation [13]; в работе показано, что алгоритм Label Propagation достигает лучших результатов. Эти и подобные алгоритмы на графах используются и в сочетании с обработкой информации, извлекаемой из текстовых корпусов.

Во многих исследованиях словари оценочной лексики для заданной предметной области создаются на основе корпусных подходов с использованием различных типов распространения разметки, исходящей из начального множества известных оценочных слов [9]. Серьезной проблемой таких подходов является определение подходящего размера и состава начального множества слов, что может зависеть от конкретной предметной области.

В данном исследовании создается оценочный словарь для заданной предметной области на основе нескольких текстовых корпусов. Затем полученный словарь улучшается на основе разметки понятий тезауруса с использованием для всех предметных областей минимального изначально заданного множества оценочных слов, состоящего из слов *плохой* и *хороший*.

3 Лингвистическая онтология РуТез

В данном исследовании используется тезаурус русского языка РуТез [14]. РуТез — это так называемая лингвистическая онтология, т. е. онтология, подавляющее большинство понятий которой основано на значениях реально существующих языковых выражений (слов и словосочетаний). РуТез создавался вручную и в течение долгого времени использовался в различных проектах автоматической обработки текстов и информационного поиска. Теперь РуТез доступен для использования в исследовательских целях. Опубликованная версия РуТез содержит около 100 тысяч слов и выражений (http://www. labinform. ru/pub/ruthes/index.htm).

Тезаурус РуТез отличается от тезаурусов типа WordNet тем, что его структура представляет собой единую иерархическую сеть понятий, в которой представлены значения разных частей речи (существительных, прилагательных, глаголов). Каждое понятие в тезаурусе имеет уникальное однозначное имя. Понятия соединяются между собой несколькими типами отношений. Каждому понятию соответствует совокупность текстовых выражений, которыми данное понятие может выражаться в текстах (текстовые входы понятия). Текстовые входы включают в себя как отдельные слова, так и большое количество словосочетаний. Особенностью тезауруса РуТез является и то, что он включает термины так называемой общественно-политической области, широкой предметной области жизни современного общества, включающей такие подобласти, как политика, экономика, право, спорт и др.

Многозначные слова описаны в РуТез так же, как и в тезаурусе WordNet, а именно отнесены к разным понятиям тезауруса. Например, в текущей версии тезауруса РуТез слово *пресный* приписано трем понятиям:

- ПРЕСНАЯ ВОДА;
- ПРЕСНЫЙ, БЕЗВКУСНЫЙ;
- ПРЕСНЫЙ (НЕИНТЕРЕСНЫЙ).

Первое понятие нейтрально и не относится к области отзывов о фильмах. Второе понятие негативно, но также не относится к этой области. Последнее понятие негативно и релевантно отзывам о фильмах.

4 Извлечение оценочных словарей на основе комбинированных знаний

Алгоритм извлечения оценочных слов в заданной предметной области подразделяется на два этапа, которые будут рассмотрены ниже. Модель извлечения слов строится в предметной области отзывов о фильмах, а затем переносится в другие предметные области.

4.1 Этап 1. Извлечение оценочного словаря предметной области из текстовой коллекции

На первом этапе для извлечения оценочных слов в каждой предметной области используются три текстовые коллекции.

Основной коллекцией является область отзывов пользователей в заданной предметной области (*целевая коллекция* — DOM), большинству отзывов приписана числовая оценка, сообщающая отношение пользователя к фильму, обсуждаемому в данном отзыве. Такая коллекция содержит значительное количество как оценочных слов, так и терминов предметной области (*актер, режиссер, кастинг* и др.).

Таблица 1 Количество документов по предметным областям, использовавшихся в экспериментах

Область	Количество отзывов	Количество описаний
Фильмы	28 773	17 680
Книги	23 883	22 321
Игры	7 928	1 853
Цифровые камеры	10 208	920
Мобильные телефоны	30 620	890

Для данного эксперимента использовались коллекции отзывов, собранные с онлайновых сервисов imhonet.ru и market.yandex.ru в пяти предметных областях: фильмы, книги, компьютерные игры, мобильные телефоны и цифровые камеры (табл. 1).

В качестве второй коллекции, относящейся к предметной области, выбрана область описаний объектов отзыва (*коллекция описаний* — DESCR). Для области фильмов и книг это описания их сюжетов, для отзывов о технических устройствах — описания этих технических устройств. В этой коллекции содержатся небольшая доля оценочных слов и множество слов, относящихся к предметной области. Таким образом, такой корпус может использоваться как контрастный корпус к коллекции отзывов (см. табл. 1).

В качестве другого контрастного корпуса используется коллекция новостей (*новостная коллекция* — NEWS) — 1 млн документов. В этом корпусе содержится относительно низкая доля как оценочных, так и предметно-ориентированных слов из отзывов.

На основе этих коллекций для каждого знаменательного слова w (существительного, прилагательного, глагола, наречия), встречающегося в коллекции отзывов, вычисляется совокупность признаков, описывающих «поведение» слова в вышеперечисленных коллекциях.

Первой группой признаков являются **частотные признаки**, вычисленные по каждой из текстовых коллекций. Такие признаки включают:

- общие частоты встречаемости слов в разных коллекциях (tf_w);
- документные частоты (df_w) — число документов, в которых употреблялось слово w;
- частота употребления слов с большой буквы (предполагается, что оценочные слова обычно не являются именами собственными);
- частота встречаемости слова со словами-операторами, усиливающими или меняющими тональность слова на противоположную, например *не очень*.

Признаки второй группы — **контрастные признаки** — считаются по разным корпусам. В их состав входит признак Странность [15] и известный в информационном поиске признак tf.idf. Оба типа признаков призваны зафиксировать

более высокую долю оценочных слов в целевом корпусе DOM по сравнению с контрастными корпусами.

Признак Странность вычисляется на двух корпусах, один из которых — корпус отзывов DOM, а второй — контрастный: DESCR или NEWS.

$$\text{Странность} = \frac{w_s/t_s}{w_g/t_g},$$

где w_s — частота употребления слова в корпусе s; t_s — общее количество слов в корпусе s; w_g — частота употребления слова в контрастном корпусе g; t_g — общее количество слов в контрастном корпусе g. Вместо количества употреблений слова в коллекции (tf_w) для вычисления данного признака может быть использована и документная частота df_w.

Таким образом, данный типовой признак порождает три конкретных признака $\text{Странность}_{\text{df}}$ на парах корпусов DOM–DESCR, DOM–NEWS и $\text{Странность}_{\text{tf}}$ на паре корпусов DOM–DESCR.

Признак tf.idf обычно используется для вычисления веса слова в документе при информационном поиске: tf — это обычно частота упоминания слова в отдельном документе, idf — это обратная документная частота df упоминания слов в текстовой коллекции, в которой производится поиск.

В описываемых экспериментах используется tf_w — частота слова в целевой коллекции DOM, а df_w вычисляется по контрастным коллекциям.

В литературе встречаются различные формулы для вычисления tf.idf. В данной работе использовалась формула из [16]:

$$\text{tf.idf} = \beta + (1-\beta)\cdot \text{tf}\cdot \text{idf},$$

где

$$\text{tf}(l) = \frac{\text{freq}(l)}{\text{freq}(l) + 0{,}5 + 1{,}5\text{dl}/\text{avg_dl}}; \quad \text{idf}(l) = \frac{\log\left((|c| + 0{,}5)/\text{df}(l)\right)}{\log(|c|+1)},$$

где $\text{freq}(l)$ — частота вхождения слова l в целевую коллекцию; dl — количество словоупотреблений в целевой коллекции; avg_dl — средняя длина документа; $\text{df}(l)$ — количество документов в контрастной коллекции, в которых встречается слово l; $\beta = 0{,}4$; $|c|$ — общее число документов в контрастной коллекции.

Поскольку оценочные словари создаются на основе отзывов пользователей, которые обычно сопровождаются числовой оценкой в баллах, то также возможно использовать эти данные для создания отдельной группы **признаков на основе оценок пользователей**.

В состав этих признаков входит **признак отклонения от средней оценки**. Он вычисляется как разница между средней оценкой отзывов, в которых упоминалось слово, и средней оценкой отзыва:

$$\mathrm{dev}\,(l) = \left| \frac{\sum_{i=1}^{n} m_i k_i}{k} - \frac{\sum_{i=1}^{n} m_i}{n} \right|, \qquad \sum_{i=1}^{n} k_i = k\,,$$

где l — рассматриваемое слово; n — общее число отзывов в коллекции; m_i — оценка i-го отзыва; k_i — частота упоминания слова в i-м отзыве (может быть равной 0); k — частота слова в коллекции.

Второй признак в этой группе — это **дисперсия оценки слова**. Этот признак показывает, насколько значительно различаются оценки отзывов, в которых упоминается слово. Предполагается, что если слово упоминается в различных отзывах с похожими оценками, то это, скорее всего, оценочное слово:

$$\mathrm{var}\,(l) = \frac{\sum_{i=1}^{n} m_i^2 k_i}{k} - \left(\frac{\sum_{i=1}^{n} m_i k_i}{k} \right)^2, \qquad \sum_{i=1}^{n} k_i = k\,.$$

Отдельную группу признаков составляют **лингвистические признаки**, включающие бинарные признаки части речи, многозначности части речи для слова, признак наличия приставки из предопределенного списка приставок, часто встречающихся в составе оценочных слов (*не*, *без*, *бес* и др.).

Для обучения алгоритма были размечены все слова с частотой встречаемости более 3 в коллекции отзывов о фильмах. Разметка была сделана двумя разметчиками. Если возникало несогласие между разметчиками, то решение принималось на основе совместного обсуждения. В результате было размечено 4079 тональных слов. Лучшее качество в задаче классификации на два класса (оценочное слово vs. неоценочное слово) было достигнуто на основе набора классификаторов Logistic Regression, LogitBoost и Random Forest из программного пакета WEKA. Для этого вероятности оценочности слов, выдаваемые отдельными классификаторами, усреднялись.

Результатом применения описанного метода стал список слов, упорядоченный по оцениваемой вероятности их оценочности — *оценочным весам*. Этот алгоритм располагает оценочные слова ближе к началу списка и присваивает им высокие оценочные веса, а нейтральные слова с низкими оценочными весами — ближе к концу списка.

В итоге для отзывов о фильмах был получен список из более чем 18 тыс. слов. На первых позициях списка оказались слова: *ненапряжный*, *несмешной*, *неподражаемый*, *невнятный*, *неправдоподобный*, *неинтересный*, *нежный*, *отличнейший*, *неспешный*, *бездарный* и др.

Слово *атмосферный* располагается на достаточно высокой 830-й позиции в порожденном списке. Очевидные оценочные прилагательные в области отзывов о фильмах *безвкусный* и *пресный* занимают куда более высокие позиции: 139-ю и 193-ю. Но образованные от них существительные *пресность*, *безвкусие*,

Таблица 2 Точность оценочных списков в предметных областях на уровне 100 и 1000 первых слов

Предметная область	Базовый уровень P@100	+РуТез P@100	Базовый уровень P@1000	+РуТез P@1000
Фильмы	99%	100%	81,5%	85,5%
Книги	99%	100%	86,0%	86,2%
Игры	97%	100%	72,2%	73,1%
Цифровые камеры	85%	92%	65,8%	66,3%
Мобильные телефоны	85%	97%	73,2%	78,6%
Обобщенная область продуктов	100%	100%	90,5%	95,2%

безвкусность вообще отсутствуют в списке из-за своей малой частоты употребления в отзывах, а *безвкусица* занимает 1515-ю позицию. Поэтому видно, что учет тезаурусных связей может улучшить упорядочение оценочных слов в списке.

Предложенная модель была перенесена на другие четыре области отзывов (книги, игры, цифровые камеры и мобильные телефоны) без дополнительных затрат на обучение. Качество извлеченных оценочных словарей представлено в табл. 2 в колонках под названием Базовый уровень. Представлено два показателя точности: точность на уровне первых 100 слов в списке (Базовый уровень P@100) и точность на уровне первых 1000 слов в списке (Базовый уровень P@1000).

4.2 Улучшение оценочного словаря предметной области на основе тезауруса РуТез

Для улучшения качества порождаемого списка оценочных слов применяется тезаурус русского языка РуТез. На вход алгоритма поступает список слов, упорядоченных по мере вероятности их оценочности. Список получен на основе применения метода машинного обучения, описанного в предыдущем подразделе.

Слова, содержащиеся в списке, имеют различный статус относительно тезауруса РуТез. Часть слов не содержится в тезаурусе, например три из наиболее вероятных оценочных слов в области фильмов не содержатся в тезаурусе. Вторая часть слов описана в тезаурусе и ровно в тех значениях, в которых слова употребляются в отзывах. Последняя часть слов (таких как слово *атмосферный*) представлена в тезаурусе однако в отзывах о фильмах и книгах эти слова употребляются в другом значении и имеют другую тональность. Таким образом, описания слов в тезаурусе РуТез должны использоваться так, чтобы не ухудшить исходный список.

Основная идея алгоритма уточнения оценочного лексикона — это автоматическое разделение понятий тезауруса на оценочные и нейтральные и дальнейшее использование полученной разметки для переранжирования списка оценочных слов. Этот процесс, в противоположность вышеупомянутому алгоритму Label

Propagation, регулируется посредством ранее полученных весов оценочности слов.

Обозначим полученный на входе лексикон, в котором слова упорядочены по их оценочному весу (sw), как W_D. Сначала алгоритм формирует два множества тезаурусных понятий, используя слова, расположенные на обоих концах списка: L_s — понятия, которые предполагаются оценочными, и L_n — нейтральные понятия. Для этого вычисляются средние оценочные веса понятий csw по их текстовым входам из списка W_D. Затем алгоритм вносит в L_s понятия с высоким средним весом ($\text{csw}_s > \text{csw}_{s0}$), также в это множество вносятся понятия, соответствующие словам *плохой* и *хороший*. Понятия с низким средним весом ($\text{csw}_n < \text{csw}_{n0}$) добавляются к множеству нейтральных понятий L_n, которое формируется без всяких дополнительных предопределенных понятий. В экспериментах были получены следующие значения: $\text{csw}_{s0} = 0{,}85$ и $\text{csw}_{n0} = 0{,}05$.

Далее каждое множество (L_s и L_n) итеративно пополняется понятиями на основе проверки двух условий: порогов для каждого множества на оценочные веса понятий и числа прямых отношений в тезаурусе между понятием и множествами.

Формально L_s и L_n итеративно набираются, как показано в псевдокоде Алгоритм 1.

Алгоритм 1

Input: список понятий W_D с весами csw
Output: L_s, L_n
$L_s = \{C_{\text{bad}}, C_{\text{good}}\} \cup C_{\text{high}}$, $C_{\text{high}} = \{C_i : \text{csw}(C_i) > 0.85\}$,
$L_n = L_n \cup C_{\text{low}}$, $C_{\text{low}} = \{C_i : \text{csw}(C_i) < 0.05\}$
$\theta = 0.1$, Nlink $= 3$, $L_{s_\text{iter}} = L_s$, $L_{n_\text{iter}} = L_n$
while $\theta < 0.6$
  for $C \in \text{Adj}(L_s)$
   if $\text{nlink}(C, L_s) > \text{Nlink}\&\&\text{csw}(C) > 0.7 - \theta$
    then Include (C, L_s);
  end

  for $C \in \text{Adj}(L_n)$
   if $\text{nlink}(C, L_n) > \text{Nlink}\&\&\text{csw}(C) < \theta$
    then Include (C, L_n);
  end

  if $Ln == L_{n_\text{iter}}\&\&L_s == L_{s_\text{iter}}$
    then Nlink $=$ Nlink $- 1$;
  if Nlink $== 0$
    then $\theta = \theta + 0.05$, Nlink $= 3$;
  $L_n = L_{n_\text{iter}}$, $L_s = L_{s_\text{iter}}$
end

Алгоритм использует следующую нотацию:

- $Adj(L)$ — это множество понятий, связанных непосредственными отношениями с понятиями из множества L;
- $nlink(C, L)$ — это функция, возвращающая число прямых тезаурусных отношений между понятием C и множеством L.

На последнем шаге веса sw всех слов, соответствующих понятиям из множества L_s, модифицируются умножением их на множитель k_1 ($k_1 > 1$), а веса sw всех слов, относящихся к понятиям из L_n, домножаются на множитель k_2 ($0 < k_2 < 1$). Оценочный список слов переупорядочивается в соответствии с новыми весами.

В результате работы алгоритма для всех предметных областей было получено улучшение качества оценочных словарей (см. табл. 2, колонки под названием «+РуТез»).

5 Улучшение словаря обобщенной области продуктов и услуг

Несмотря на существующие различия между словарями оценочной лексики в конкретных предметных областях, можно предположить, что словари оценочной лексики для широкой области продуктов и услуг имеют и достаточно много общего между собой. Так, объединением оценочных словарей нескольких предметных областей в единый словарь был создан словарь ProductSentiRus[1].

В данном исследовании обновленные оценочные словари конкретных предметных областей также суммируются посредством формулы, которая повышает рейтинг слов, встречающихся в оценочных словарях разных предметных областей и имеющих высокий оценочный вес в каждой из них.

Объединенный вес оценочного слова считается по следующей формуле:

$$R(w) = \max\left(\underset{d\in D}{\mathrm{prob}_d(w)}\right) \sum_{d\in D} \frac{1}{|D|}\left(1 - \frac{\mathrm{pos}_d(w)}{|d|}\right),$$

где D — набор предметных областей (в данном случае пять предметных областей); d — это упорядоченный список оценочных слов в конкретной предметной области; $|d|$ — это общее количество слов в данном списке. Функции $\mathrm{prob}_d(w)$ и $\mathrm{pos}_d(w)$ отражают оценочный вес и позицию слова в списке d.

Точность на уровне 1000 P@1000 списка ProductSentiRus была оценена как 90,5%. Комбинация улучшенных списков оценочных слов в словарь ProductSentiRus+ дает 95,2% в терминах P@1000 (см. табл. 2).

Для оценки качества ProductSentiRus+ в задаче анализа тональности из него были извлечены первые 5000 оценочных слов (то же самое количество, что и из предыдущей версии).

[1] http://www.cir.ru/SentiLexicon/ProductSentiRus.txt.

Таблица 3 Качество классификации твитов на объективные и субъективные

Словарь оценочных слов	Точность классификации	Полнота классификации	F_1-мера
Словарь оценочных слов, автоматически порожденный по твитам [5]	—	—	61,0
ProductSentiRus (по данным [5])	—	—	61,0
ProductSentiRus+	58,5	84,7	**69,2**

Оценка качества списка базировалась на тестовой коллекции TEST, описанной в работе [5], которая включала 2000 твитов на русском языке. Предполагалось, что ProductSentiRus+ содержит значительный объем оценочных слов, употребляемых в Интернете для оценки объектов и событий. Твит рассматривался как субъективный, если в нем содержалось хотя бы одно слово из заданного на входе словаря. Таблица 3 показывает, что качество определения (F_1-мера) оценочных твитов на основе словаря ProductSentiRus+ превышает качество, полученное на основе других словарных ресурсов.

6 Заключение

В статье описан комбинированный подход к извлечению оценочных словарей в разных предметных областях. На первом этапе для создания оценочного словаря используется несколько текстовых коллекций, из которых извлекаются признаки слов, и создается модель извлечения слов на основе методов машинного обучения. На втором этапе упорядоченный список оценочных слов уточняется на основе информации о лексических отношениях между словами, описанных в тезаурусе русского языка РуТез.

Комбинированная модель извлечения оценочных слов была применена к нескольким предметным областям, получено улучшение качества оценочного списка для этих предметных областей.

Порожденные списки оценочных слов были сведены в единый список оценочных слов для продуктов и услуг, в результате был получен новый список оценочных слов ProductSentiRus+, который представляет собой улучшенную версию ранее опубликованного списка ProductSentiRus. Данный подход к извлечению слов может быть применен для других языков и использовать другие тезаурусные ресурсы.

Литература

1. *Mihalcea R., Banea C., Wiebe J.* Learning multilingual subjective language via cross-lingual projections // 45th Annual Meeting of the Association of Computational Linguistics (ACL-2007) Proceedings. — Prague, Czech Republic, 2007. P. 976–983.
2. *Steinberger J., Lenkova P., Ebrahim M., Ehrmann M., Hurriyetogly A., Kabadjov M., Steinberger R., Tanev H., Zavarella V., Vazquez S.* Creating sentiment

dictionaries via triangulation // 2nd Workshop on Computational Approaches to Subjectivity and Sentiment Analysis (ACL-HLT-2011) Proceedings. — Portland, Oregon, USA, 2011. P. 28–36.

3. *Chetviorkin I., Loukachevitch N.* Extraction of Russian sentiment lexicon for product meta-domain // 24th Conference (International) on Computational Linguistics (COLING-2012) Proceedings. — Mumbai, India, 2012. P. 593–610.
4. *Choi Y., Cardie C.* Adapting a polarity lexicon using integer linear programming for domain-specific sentiment classification // Conference on Empirical Methods in Natural Language Processing (EMNLP-2009) Proceedings. — Edinburgh, Scotland, U.K., 2009. Vol. 2. P. 590–598.
5. *Hatzivassiloglou V., McKeown K.* Predicting the semantic orientation of adjectives // 35th Annual Meeting of the Association for Computational Linguistics (ACL-97) Proceedings. — Madrid, Spain, 1997. P. 174–181.
6. *Velikovich L., Blair-Goldensohn S., Hannan K., McDonald R.* The viability of web-derived polarity lexicons // 11th Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-2010) Proceedings. — Los Angeles, CA, USA, 2010. P. 777–785.
7. *Volkova S., Wilson Th., Yarowsky D.* Exploring sentiment in social media: Bootstrapping subjectivity clues from multilingual twitter streams // 51st Annual Meeting of the Association of Computational Linguistics (ACL-13) Proceedings. — Sofia, Bulgaria, 2013. P. 505–510.
8. *Feng S., Kang J. S. , Kuznetsova P., Choi Y.* Connotation Lexicon: A dash of sentiment beneath the surface meaning // 51th Annual Meeting of the Association for Computational Linguistics (ACL-2013) Proceedings. — Sofia, Bulgaria, 2013. P. 1774–1784.
9. *Lau R., Lai C., Bruza P., Wong K.-F.* Leveraging web 2.0 data for scalable semi-supervised learning of domain-specific sentiment lexicons // 20th ACM Conference (International) on Information and Knowledge Management (CIKM-2011) Proceedings. — New York, NY, USA: ACM, 2011. P. 2457–2460.
10. *Esuli A., Sebastiani F.* Senti-wordnet: A publicly available lexical resource for opinion mining // 5th Conference on Language Resources and Evaluation (LREC-2006) Proceedings. — Genoa, Italy, 2006. P. 417–422.
11. *Fellbaum Ch.* WordNet: An Electronic Lexical Database. — Cambridge, MA, USA: MIT Press, 1998.
12. *Rao D., Ravichandran D.* Semi-supervised polarity lexicon induction // 12th Conference of the European Chapter of the ACL (EACL-2009) Proceedings. — Athens, Greece, 2009. P. 675–682.
13. *Zhu X., Ghahramani Z.* Learning from labeled and unlabeled data with label propagation. — Pittsburg: Carnegie Mellon University, 2002. Technical Report CMU-CALD-02-107.
14. *Loukachevitch N., Dobrov B.* RuThes Linguistic Ontology vs. Russian Wordnets // 7th Global Wordnet Conference (GWC-2014) Proceedings. — Tartu, Estonia, 2014. P. 154–162.
15. *Ahmad K., Gillam L., Tostevin L.* University of Surrey participation in Trec8: Weirdness indexing for logical documents extrapolation and retrieval // 8th Text Retrieval Conference (TREC-8) Proceedings. — Gaithersburg, MD, USA: NIST, 1999. P. 717–724.

16. *Callan J. P., Croft W. B., Harding S. M.* The INQUERY retrieval system // 3rd Conference (International) on Database and Expert System Applications (DEXA-92) Proceedings. — New York: Springer Verlag, 1992. P. 78–93.

Поступила в редакцию 20.01.15

COMBINING CORPUS AND THESAURUS INFORMATION FOR EXTRACTING SENTIMENT WORDS

N. V. Loukachevitch and I. I. Chetviorkin

Research Computing Center, M. V. Lomonosov Moscow State University, 4 Leninskie Gory, Moscow 119991, Russian Federation

Abstract: The paper describes a combined approach to extraction of a domain-specific sentiment lexicon. At first, an initial version of a domain-specific lexicon is obtained by application of a supervised model. At the second stage, the ordered list of sentiment words is refined using the thesaurus information. This combined model is applied to several domains and at last, the domain-specific sentiment lexicons are united to create an improved version of the Russian sentiment lexicon in the generalized domain of products.

Keywords: sentiment analysis; domain adaptation; natural language processing; thesaurus

DOI: 10.14357/08696527150102

Acknowledgments

The work was partly supported by the Russian Foundation for Basic Research (project 14-07-00682).

References

1. Mihalcea, R., C. Banea, and J. Wiebe. 2007. Learning multilingual subjective language via cross-lingual projections. *45th Annual Meeting of the Association of Computational Linguistics ACL-2007 Proceedings*. Prague, Czech Republic. 976–983.
2. Steinberger, J., P. Lenkova, M. Ebrahim, M. Ehrmann, A. Hurriyetogly, M. Kabadjov, R. Steinberger, H. Tanev, V. Zavarella, and S. Vazquez. 2011. Creating sentiment dictionaries via triangulation. *2nd Workshop on Computational Approaches to Subjectivity and Sentiment Analysis, ACL-HLT-2011 Proceedings*. Oregon. 28–36.
3. Chetviorkin, I., and N. Loukachevitch. 2012. Extraction of Russian sentiment lexicon for product meta-domain. *COLING-2012 Proceedings*. Mumbai. 593–610.
4. Choi, Y., and C. Cardie. 2009. Adapting a polarity lexicon using integer linear programming for domain-specific sentiment classification. *Conference on Empirical Methods in Natural Language Processing EMNLP-2009 Proceedings*. Edinburgh. 2:590–598.

5. Hatzivassiloglou, V., and K. McKeown. 1997. Predicting the semantic orientation of adjectives. *35th Annual Meeting of the Association for Computational Linguistics, ACL-1 997 Proceedings*. 174–181.
6. Velikovich, L., S. Blair-Goldensohn, K. Hannan, and R. McDonald. 2010. The viability of web-derived polarity lexicons. *NAACL-2010 Proceedings*. Los Angeles, CA. 777–785.
7. Volkova, S., Th. Wilson, and D. Yarowsky. 2013. Exploring sentiment in social media: Bootstrapping subjectivity clues from multilingual twitter streams. *51st Annual Meeting of the Association of Computational Linguistics, ACL-13 Proceedings*. Sofia, Bulgaria. 505–510.
8. Feng, S., J. S. Kang, P. Kuznetsova, and Y. Choi. 2013. Connotation Lexicon: A dash of sentiment beneath the surface meaning. *51th Annual Meeting of the Association for Computational Linguistics ACL-2013 Proceedings*. Sofia, Bulgaria. 1774–1784.
9. Lau, R., C. Lai, P. Bruza, and K.-F. Wong. 2011. Leveraging web 2.0 data for scalable semi-supervised learning of domain-specific sentiment lexicons. *20th ACM Conference (International) on Information and Knowledge Management CIKM-2011 Proceedings*. New York, N.Y.: ACM. 2457–2460.
10. Esuli, A., and F. Sebastiani. 2006. SentiWordnet: A publicly available lexical resource for opinion mining. *LREC-2006 Proceedings*. Genoa. 417–422.
11. Fellbaum, Ch. 1998. *WordNet: An Electronic Lexical Database*. Cambridge, MA: MIT Press.
12. Rao, D., and D. Ravichandran. 2009. Semi-supervised polarity lexicon induction. *12th Conference of the European Chapter of the ACL, EACL-2009 Proceedings*. Athens. 675–682.
13. Zhu, X., and Z. Ghahramani. 2002. Learning from labeled and unlabeled data with label propagation. Pittsburg: Carnegie Mellon University. Technical Report CMU-CALD-02-107.
14. Loukachevitch, N., and B. Dobrov. 2014, RuThes Linguistic Ontology vs. Russian Wordnets. *Global Wordnet Conference GWC-2014 Proceedings*. Tartu, Estonia. 154–162.
15. Ahmad, K., Gillam L., and L. Tostevin. 1999. University of Surrey participation in Trec8: Weirdness indexing for logical documents extrapolation and retrieval. *8th Text Retrieval Conference TREC-8 Proceedings*. Gaithersburg, MD: NIST. 717–724.
16. Callan, J. P., W. B. Croft, and S. M. Harding. 1992. The INQUERY retrieval system. *3rd Conference (International) on Database and Expert Systems Applications DEXA-92 Proceedings*. New York: Springer Verlag. 78–93.

Received January 20, 2015

Contributors

Loukachevitch Natalia V. (b. 1964) — Candidate of science (PhD) in physics and mathematics, leading scientist, Research Computing Center, M. V. Lomonosov Moscow State University, 4 Leninskie Gory, Moscow 119991, Russian Federation; louk_nat@mail.ru

Chetviorkin Ilia I. (b. 1988) — programmer, Research Computing Center, M. V. Lomonosov Moscow State University, 119991, Moscow, 4 Leninskie Gory, Moscow 119991, Russian Federation; ilia2010@yandex.ru

МНОГОКРИТЕРИАЛЬНЫЙ МЕТОД ВЫЯВЛЕНИЯ НЕЧЕТКИХ ДУБЛИКАТОВ В ПОТОКЕ ТЕКСТОВЫХ СООБЩЕНИЙ*

А. М. Андреев[1], *Д. В. Березкин*[2], *И. А. Козлов*[3], *К. В. Симаков*[4]

Аннотация: Рассмотрена задача обнаружения нечетких дубликатов в потоке текстовых сообщений. Предложена модель документа, имеющая возможность гибкой настройки на различные предметные области. Представлен многокритериальный метод выявления дублирующихся документов на основе бинарной классификации с помощью метода опорных векторов. Предложен способ обеспечения высокого быстродействия метода посредством предварительного отбора кандидатов в дубликаты. Проведена экспериментальная оценка предложенного метода, демонстрирующая его практическую применимость.

Ключевые слова: обнаружение нечетких дубликатов; мера близости; бинарная классификация

DOI: 10.14357/08696527150103

1 Введение

В настоящее время во многих предметных областях существует потребность в формировании больших текстовых коллекций. При этом производится сбор текстовой информации из открытых интернет-источников, а также специализированных ресурсов. Основной областью использования создаваемых таким образом хранилищ документов является интеллектуальная обработка текстов, которую, как правило, можно отнести к классу Text Mining.

С ростом количества разнообразных источников данных в сети Интернет (новостные сайты, блоги, социальные сети) все более серьезной проблемой становится дублирование информации. Сообщения, публикуемые одним источником, зачастую многократно перепечатываются другими (в исходном виде или с небольшими изменениями). В результате при выполнении автоматического сбора

*Статья рекомендована к публикации в журнале Программным комитетом конференции «Электронные библиотеки: перспективные методы и технологии, электронные коллекции» (“Digital Libraries: Advanced Methods and Technologies,” RCDL-2014), г. Дубна, 13–16 октября 2014 г.

[1]Московский государственный технический университет им. Н. Э. Баумана, arkandreev@gmail.com

[2]Московский государственный технический университет им. Н. Э. Баумана, dmitryb2007@yandex.ru

[3]Московский государственный технический университет им. Н. Э. Баумана, kozlovilya89@gmail.com

[4]Московский государственный технический университет им. Н. Э. Баумана, skv@ixlab.ru

документов из многочисленных источников в формируемой текстовой коллекции накапливаются идентичные или близкие по содержанию документы — дубликаты. В некоторых задачах наличие дубликатов должно учитываться — например, при определении значимости сообщений [1]. Но в большинстве случаев попадание таких документов в коллекцию снижает ее качество [2].

В данной статье рассматривается решение задачи обнаружения дубликатов в потоке текстовых сообщений. Особое внимание уделяется обеспечению возможности использования разработанного метода для обработки документов из различных предметных областей.

2 Постановка задачи

2.1 Функционирование системы сбора и обработки новостной информации

В работе [3] авторами было предложено решение задачи качественного автоматического сбора новостных данных из интернет-источников, предполагающего извлечение с веб-страницы текста новости, а также сопутствующих метаданных, включающих название, дату публикации, автора новости и др. Текстовые данные, извлеченные с веб-сайтов, подвергаются последующей обработке различными методами интеллектуального анализа, такими как автоматическая классификация и кластеризация документов, извлечение знаний и фактов из естественно-языковых текстов и прогноз развития ситуаций.

Для эффективного применения этих методов необходимо обеспечить качество анализируемой коллекции документов, в частности — устранить из нее дубликаты. Поэтому в систему сбора требуется встроить подсистему, выполняющую их оперативное обнаружение и удаление (рис. 1). Она должна сравнивать каждый загружаемый документ с полученными ранее новостями и удалять его, если он является дубликатом одной из них.

2.2 Особенности решаемой задачи

Если проблема обнаружения и удаления полных дублей тривиальна, то при необходимости распознавать нечеткие дубликаты (т. е. документы, имеющие различный текст, но близкие по содержанию) возникают значительные сложности.

В связи с явлением частичного дублирования в работе [4] предлагается понятие информативной необходимости элемента высказывания: «Всякий частично-дублетный элемент, если он отвечает коммуникативной задаче высказывания, признается информативно-необходимым в той же мере, что и прочие, недублетные элементы речи: такой элемент информативно необходим постольку, поскольку вносит свой уникальный, неповторимый в других элементах, вклад в суммарную информацию, передаваемую высказыванием». Таким образом, задача обнаружения нечетких дубликатов состоит в распознавании и удалении сообщений, не являющихся информативно-необходимыми.

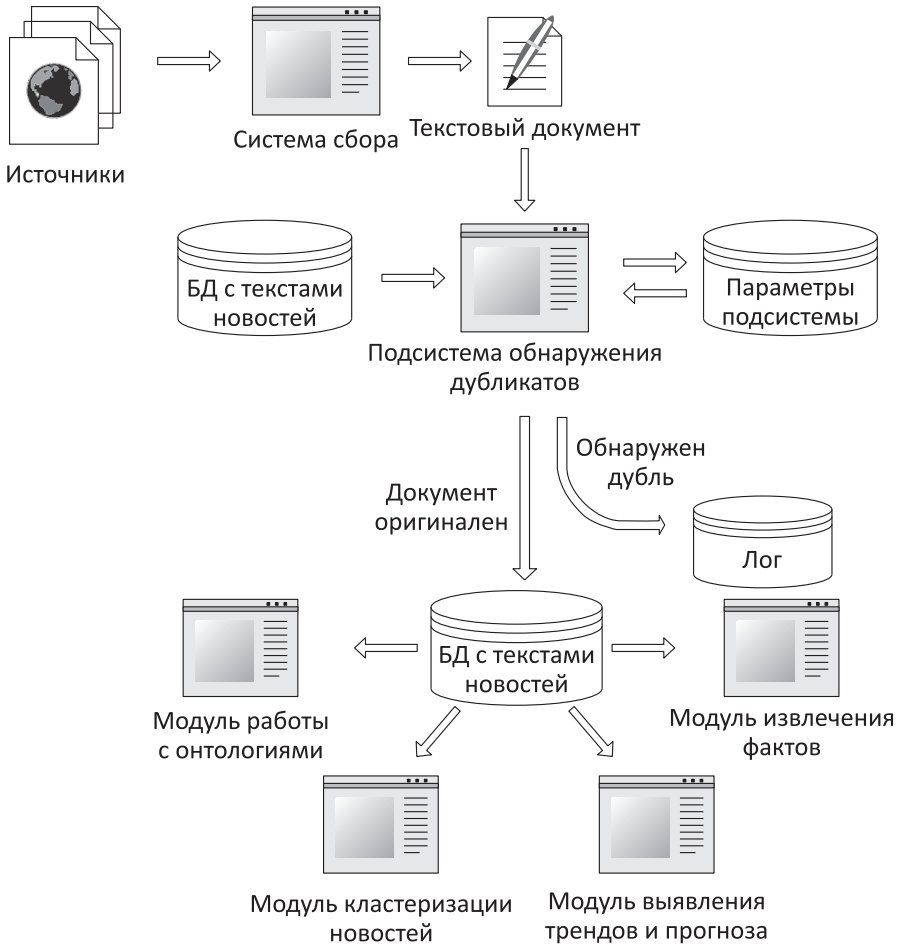


Рис. 1 Место подсистемы обнаружения дубликатов в системе автоматизированного сбора и анализа новостной информации

Установить информативную необходимость и ценность некоторого высказывания можно только с учетом соответствующего ситуативного контекста. Так, при ручной проверке документов эксперт, определяя наличие или отсутствие дублирования, принимает решение с учетом предметной области и характера документов, составляющих анализируемую коллекцию. Например, при работе с юридическими документами особое внимание должно уделяться метаданным — для текстов такого типа два документа с практически идентичным содержанием, но различающимися названиями и датами публикации не могут считаться дублями. Другой подход используется при анализе экономических новостей,

например сводок о состоянии фондового рынка, — дубликатами не должны признаваться документы, имеющие одно и то же название и одинаковый текст, но различающиеся числовыми данными (значениями курсов валют).

Таким образом, в разных случаях эксперт сравнивает документы с точки зрения различных критериев (близость текстового содержания, сходство названий, разница во времени публикации), т. е. использует различные модели распознавания дубликатов в зависимости от предметной области и контекста коммуникационных сообщений. Поэтому не представляется возможным выработать единую систему правил для обнаружения нечеткого дублирования сразу для всех случаев. Следовательно, разрабатываемая подсистема должна иметь возможность гибкой настройки на разные предметные области, что позволит ей моделировать деятельность эксперта по распознаванию дублей с использованием различных моделей. Поскольку система сбора выполняет извлечение документов из множества источников, которые относятся к различным предметным областям, необходимо обеспечить возможность работы с несколькими моделями распознавания дубликатов одновременно.

Еще одной проблемой, с которой приходится сталкиваться при решении задачи устранения дубликатов, является большой объем обрабатываемых данных. Наличие в коллекции сотен тысяч документов делает весьма трудоемким анализ каждого нового сообщения путем сравнения его с каждым из ранее загруженных. Эта проблема может быть решена с помощью приближенных методов, но их применение ведет к снижению качества обнаружения дубликатов, т. е. уменьшению точности и полноты [2]. При разработке предлагаемого подхода решалась задача совмещения высокого качества и низкой вычислительной сложности проверки документов.

3 Обзор методов обнаружения дублей

3.1 Однокритериальные методы

Большинство методов, используемых при выявлении дублирования сообщений, основано на общей идее — попарном сопоставлении документов с использованием выбранной меры близости. При этом документы d_i и d_j считаются дубликатами, если значение этой меры для них превосходит заданную пороговую величину: $\mathrm{sim}\,(d_i, d_j) > \mathrm{Tr}$. Методы, предлагаемые исследователями, различаются способом представления документов и используемыми мерами близости.

Наиболее эффективным с точки зрения вычислительной сложности являются представление документа одним числовым значением — «сигнатурой» — и определение дублирования сообщений на основании совпадения их сигнатур. В качестве сигнатур используются:

– контрольная сумма строки, полученной из нескольких сцепленных в алфавитном порядке слов документа с наибольшими значениями весов (для расчета весов используются методы TF (term frequency), TF-IDF (inverse document frequency) и OptFreq (optimal frequency)) [2];

– контрольная сумма строки, полученной из нескольких наиболее длинных или значимых (т. е. состоящих из слов с наибольшим суммарным значением весов) предложений документа [2];

– контрольная сумма строки, полученной из слов текста, принадлежащих заданному словарю (содержащему слова со средними значениями IDF) [5];

– битовый вектор, каждый элемент которого соответствует слову из заданного словаря и равен 1, если частота встречаемости этого слова в документе превышает пороговое значение, и 0 в противном случае [6].

Другой популярный подход состоит в представлении документа множеством элементов. Близость множеств, соответствующих двум сообщениям, рассчитывается, как правило, с помощью коэффициента Жаккара. В качестве элементов множеств используются:

– предложения документа [7];

– ключевые термы (слова, наиболее значимые с точки зрения лингвостатистических критериев) [1];

– шинглы (всевозможные последовательности фиксированной длины k, состоящие из соседних слов документа) [8];

– N-граммы (всевозможные последовательности, состоящие из N соседних символов документа) [9].

Широко применяются векторные модели документов, в которых каждое сообщение представляется вектором в многомерном признаковом пространстве. В качестве элементов вектора обычно используются слова, встречающиеся в текстах коллекции [10]. При этом значениями элементов являются веса соответствующих слов, отражающие их значимость для документа. В качестве признаков могут применяться и другие характеристики текстов, например частота появления различных пар символов или частота появления тех или иных частей речи [11]. Для сравнения векторов часто используется косинусная мера или ее модификации [10].

Существует несколько методов, в которых элементами вектора являются сигнатуры, полученные различными способами. При этом для сравнения векторов используется расстояние Хэмминга. Так, в [12] предложена модификация сигнатурного метода [5], предполагающая использование нескольких словарей, для каждого из которых вычисляется своя сигнатура. Другим примером этого подхода служит метод, развивающий идею алгоритма шинглов [8, 13]. Он состоит в применении к элементам множества шинглов различных хэш-функций и выборе для каждой из них шингла, минимизирующего ее значение. Из выбранных шинглов формируются группы, именуемые «супершинглами», которые и являются элементами вектора, представляющего документ.

Наконец, в [14] предлагается использовать в качестве модели документа упорядоченную последовательность символов. Для сравнения в этом случае используются редакционные расстояния, такие как расстояние Левенштейна.

3.2 Многокритериальные методы

В отличие от вышеперечисленных подходов методы, описанные в [14, 15], основаны на оценке схожести документов с точки зрения нескольких критериев одновременно. Это связано с тем, что анализируемые данные в этих работах состоят из множества полей и для каждого поля значение близости записей рассчитывается отдельно (при этом для разных полей могут использоваться различные меры близости). В результате пара записей представляется вектором, каждый элемент которого соответствует схожести соответствующих полей документов. Пары, отмеченные в обучающей выборке как «дубликаты» или «недубликаты», представлены двумя кластерами точек многомерного пространства. Таким образом, задача обнаружения дублей сводится к отнесению вектора, соответствующего анализируемой паре документов, к одному из этих кластеров. В [14] для решения этой задачи используется метод опорных векторов, в [15] выбирается кластер, центроид которого находится ближе к вектору.

4 Предложенный подход к обнаружению дубликатов

В целях обеспечения возможности применения разрабатываемого подхода к обработке сообщений из разных предметных областей модель документа должна предусматривать возможность сравнения сообщений по различным признакам (см. п. 2.2). Окончательное же решение должно приниматься на основе анализа пары документов с точки зрения всех критериев. Исходя из этого, удобно представить пару документов (d_i, d_j) вектором, элементами которого являются результаты сравнения документов по соответствующим признакам:

$$\rho_{i,j} = \left(\rho^1_{i,j}, \rho^2_{i,j}, \ldots, \rho^k_{i,j}\right),$$

где $\rho^k_{i,j}$ характеризует сходство документов d_i и d_j по k-му критерию. На основе этого вектора принимается решение о наличии или отсутствии дублирования. Для этого необходимо задать функцию, выполняющую интерпретацию $\rho_{i,j}$, т. е. определяющую его в один из двух классов, один из которых означает наличие дублирования (обозначим его M_+), другой — отсутствие (M_-):

$$D(\rho_{i,j}) = \begin{cases} 1, & \rho_{i,j} \in M_+; \\ 0, & \rho_{i,j} \in M_-. \end{cases} \tag{1}$$

С учетом выбранного подхода процесс обнаружения дублей можно разбить на следующие этапы (рис. 2):

(1) построение модели документа, отражающей характеристики новостного сообщения с точки зрения каждого из выбранных критериев;
(2) сравнение моделей двух документов и получение вектора $\rho_{i,j}$;
(3) интерпретация вектора с помощью решающей функции $D(\rho_{i,j})$.

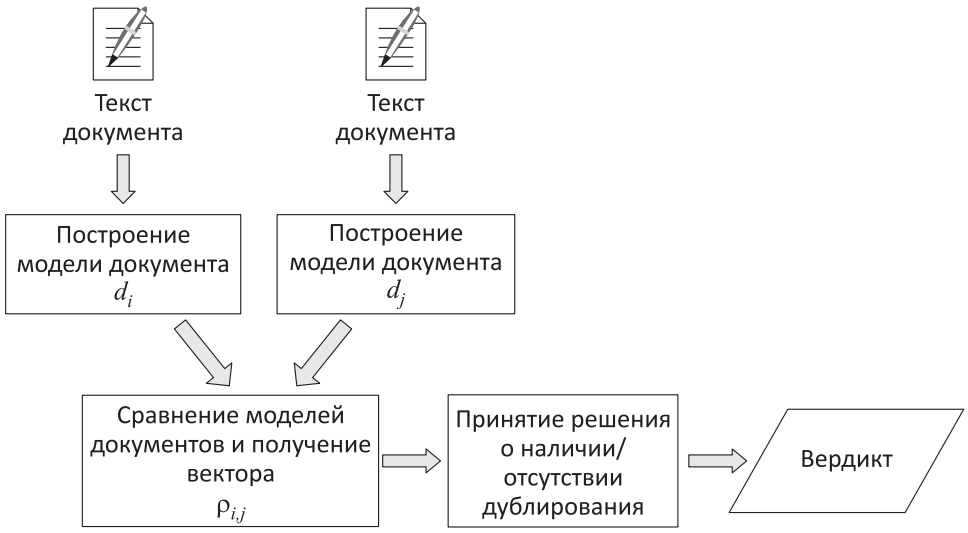


Рис. 2 Этапы обнаружения дубликатов

5 Модель документа

Важным этапом решения задачи является выбор критериев для сравнения документов. Набор критериев должен быть достаточно выразительным, чтобы обеспечивать возможность гибкой настройки модели в соответствии со спецификой различных предметных областей. Для определения набора критериев было проведено исследование выборки текстовых документов, автоматически собираемых из различных интернет-источников.

В качестве источников были выбраны 35 сайтов по различной тематике: основные новостные сайты, публикующие материалы общественно-политической и экономической тематики, официальные сайты органов государственной власти РФ, некоторые сайты органов законодательной и исполнительной власти субъектов РФ. Такой набор сайтов позволил охватить значительное число тем и типов информационных сообщений (ленты новостей, аналитические статьи и обзоры, документы правового характера, документы, содержащие финансово-экономические показатели).

Одна тысяча двести пар документов были проанализированы экспертами вручную. В результате выполненного анализа были выявлены следующие критерии, характеризующие модель распознавания дубликатов.

5.1 Содержание текста

В большинстве случаев определяющее значение имеет близость текстов. Прежде всего это характерно для общественно-политических новостей, вклад

которых в суммарную информацию, передаваемую новостным потоком, определяется оригинальностью их текстового содержания.

Для обеспечения возможности сравнения составов слов документов модель включает векторное представление текста сообщения

$$d_i^w = \left(w_i^1, w_i^2, \ldots, w_i^{N^w}\right) ,$$

где N^w — общее количество различных слов во всех документах; w_i^j — вес j-го слова в i-м документе. Если слово не встречается в документе, его вес равен нулю. Для остальных слов вес рассчитывается по методу TF-IDF.

Векторное представление позволяет использовать для сравнения текстов простые алгебраические методы, такие как косинусная мера. Она оценивает близость векторов на основании значения косинуса угла между ними. При использовании неотрицательных весов слов косинусная мера принимает значения в интервале $[0,\ 1]$, поэтому в качестве оценки различия векторов используется значение $\rho_{i,j}^w = 1 - \mathrm{sim}_{\cos}\left(d_i^w, d_j^w\right)$.

5.2 Содержание заголовка

Отдельно при принятии решения учитываются заголовки, причем их роль значительно варьируется в зависимости от предметной области. При перепечатке политических новостей с одного сайта на другой нередко изменяется только заголовок, причем иногда — весьма существенно. В таком случае наличие дублирования может быть обнаружено на основе близости остального текста новостей. Однако для сообщений другого типа (например, правового характера) различие заголовков имеет решающее значение, и такие документы не должны признаваться дубликатами, несмотря на близкое содержание.

В модели документа заголовок текста d_i^t представляется аналогично основному тексту, а определение близости заголовков $\rho_{i,j}^t$ выполняется идентично сравнению содержания документов.

5.3 Предложения и абзацы

Помимо оценки близости текстового содержания документов в целом эксперт обращает особое внимание на наличие в сообщениях идентичных структурных элементов текстов — предложений и абзацев. Это связано с тем, что при перепечатке некоторым источником ранее опубликованного документа многие из этих элементов переносятся в текст-дубликат без изменений.

Для учета этих критериев документ представляется множеством своих предложений

$$d_i^c = \left\{c_i^1, c_i^2, \ldots, c_i^{N_i^c}\right\} ,$$

где N_i^c — количество предложений в i-м документе. Вес предложения $\omega(c)$ определяется как сумма весов составляющих его слов.

При выборе меры близости нужно учесть, что она должна быть несимметричной: если один из документов получен из другого путем удаления нескольких предложений, то лишь одно сообщение пары содержит дополнительную информацию относительно второго, но не наоборот. Для учета этого требования в качестве меры сходства была выбрана взвешенная мера включения:

$$\mathrm{sim}^c_{\mathrm{inc}}\left(d^c_i, d^c_j\right) = \frac{\sum\limits_{c \in |d^c_i \cap d^c_j|} \omega(c)}{\sum\limits_{c \in d^c_i} \omega(c)}.$$

Для оценки различия документов с точки зрения предложений используются значения $\rho^c_{i,j} = 1 - \mathrm{sim}^c_{\mathrm{inc}}\left(d^c_i, d^c_j\right)$ и $\rho^c_{j,i} = 1 - \mathrm{sim}^c_{\mathrm{inc}}\left(d^c_j, d^c_i\right)$.

Построение модели множества абзацев d^p_i и определение мер различия документов с точки зрения абзацев $\rho^p_{i,j}$ и $\rho^p_{j,i}$ выполняется аналогично.

5.4 Числовые данные

Кроме документов, содержащих только текстовую информацию, часто встречаются и те, которые включают числовые данные. В ряде случаев даже незначительное изменение этих данных может существенно повлиять на содержание сообщения. Примером таких документов являются новостные сообщения из области экономики (новости о состоянии фондового рынка) или спорта (сообщения о результатах соревнований). Такие документы не должны признаваться дубликатами даже при полном совпадении их текста.

Для учета этого признака сообщения представляются множествами чисел, извлеченных из текста:

$$d^n_i = \left\{ n^1_i, n^2_i, \ldots, n^{N^{n_d}_i}_i \right\},$$

где $N^{n_d}_i$ — количество различных чисел в i-м документе. Для выполнения оценки сходства таких наборов также используется мера включения, однако элементы сравниваемых множеств не являются взвешенными, а потому учитывается лишь количество одинаковых и различающихся числовых значений:

$$\mathrm{sim}^n_{\mathrm{inc}}\left(d^n_i, d^n_j\right) = \frac{|d^n_i \cap d^n_j|}{|d^n_i|}.$$

Отличие одного документа от другого определяется как $\rho^n_{i,j} = 1 - \mathrm{sim}^n_{\mathrm{inc}}\left(d^n_i, d^n_j\right)$.

Для некоторых предметных областей важен не только состав набора чисел, но и порядок их следования в тексте документа. Это относится, в частности, к спортивным новостям, где разные последовательности одних и тех же чисел

могут соответствовать различным результатам соревнований (например, два сета в теннисном матче, завершившиеся со счетом 6:4 и 4:6). В таком случае для представления сообщения используется кортеж $d_i^n = \left(n_i^1, n_i^2, \ldots, n_i^{N_i^{n_a}}\right)$, где $N_i^{n_a}$ — общее количество чисел в i-м документе, а для сравнения документов — расстояние Дамерау–Левенштейна $\rho_{i,j}^n = \mathrm{dist}_{\mathrm{DL}}\left(d_i^n, d_j^n\right)$.

5.5 Фотографии и ссылки

Информация в сообщении может быть представлена не только текстом или числовыми данными, но и различными объектами, включенными в текст документа — фотографиями, видеороликами, ссылками на сторонние источники. Присутствие в документе дополнительных фото- и видеоматериалов значительно повышает вероятность его оригинальности, а наличие ссылок, напротив, может говорить о заимствовании информации из первоисточников. Поэтому в модель включаются компоненты, отражающие количество присутствующих в тексте сообщения фотографий d_i^{im} и ссылок d_i^h. Различие документов с точки зрения этих критериев определяется как разность соответствующих значений:

$$\rho_{i,j}^{\mathrm{im}} = d_i^{\mathrm{im}} - d_j^{\mathrm{im}}; \quad \rho_{i,j}^h = d_i^h - d_j^h.$$

5.6 Дата и время публикации

Существенным фактором, влияющим на принятие решения о наличии или отсутствии дублирования, является разница во времени публикации сообщений. Так, при дублировании новостных статей перепечатыванию обычно подвергаются свежие новости, недавно опубликованные на сайте первоисточника. С увеличением интервала между моментами появления документов в сети вероятность дублирования быстро убывает.

В модели эта характеристика сообщения представлена посредством POSIX-времени момента публикации (которое определяется как количество секунд, прошедших с полуночи 1 января 1970 г. до момента, когда документ был опубликован источником): d_i^{dt}. Различие между документами определяется как разность между моментами публикации сообщений в секундах:

$$\rho_{i,j}^{\mathrm{dt}} = d_i^{\mathrm{dt}} - d_j^{\mathrm{dt}}.$$

5.7 Авторитетность источника

При анализе документов эксперты обращают внимание на источники сообщений. Статья из авторитетного источника (который обычно публикует оригинальные материалы) имеет существенно меньшую вероятность быть признанной

дубликатом, чем документ, полученный из источника, регулярно занимающегося перепечаткой чужих сообщений.

В модель документа включается компонент, отражающий авторитетность источника, опубликовавшего это сообщение: $d_i^a \in [0, 1]$. Это значение может быть задано экспертом вручную или получено на основе обучающей выборки как отношение числа оригинальных документов, поступивших от источника, к общему числу опубликованных им сообщений.

Таким образом, модель документа представляет собой совокупность компонентов, характеризующих сообщение с точки зрения различных критериев:

$$d_i = \left(d_i^w, d_i^t, d_i^n, d_i^c, d_i^p, d_i^{\mathrm{im}}, d_i^h, d_i^{\mathrm{dt}}, d_i^a\right) .$$

6 Метод обнаружения дубликатов

6.1 Интерпретация результата сравнения

После получения моделей d_i и d_j двух документов необходимо сравнить их и вынести решение о том, являются ли документы нечеткими дубликатами. Для этого выполняется анализ схожести сообщений по каждому из критериев. Результатом сравнения документов с точки зрения некоторого критерия k является значение $\rho_{i,j}^k$. Выполнив попарно сравнение компонентов моделей для каждого из критериев, получим вектор

$$\rho_{i,j} = \left(\rho_{i,j}^w, \rho_{i,j}^t, \rho_{i,j}^n, \rho_{i,j}^c, \rho_{i,j}^p, \rho_{i,j}^{\mathrm{im}}, \rho_{i,j}^h, \rho_{i,j}^{\mathrm{dt}}, \rho_{i,j}^a\right) .$$

Ввиду вышеуказанной несимметричности мер сходства, используемых для некоторых критериев, результатом сравнения двух документов являются два различных вектора $\rho_{i,j}$ и $\rho_{j,i}$, характеризующие степень отличия первого и второго сообщения друг от друга. Каждый из этих векторов интерпретируется с помощью функции $D(\rho)$ (1).

Для интерпретации результата сравнения необходимо решить задачу бинарной классификации, т. е. отнести вектор к классу M_+ или M_-. Для настройки параметров классификатора используется обучающая выборка — набор векторов, каждый из которых снабжен меткой $m \in \{M_+, M_-\}$, обозначающей класс, к которому принадлежит этот вектор.

Задача бинарной классификации состоит в том, чтобы для вновь поступившего на исследование вектора $\rho = \left(\rho^1, \rho^2, \ldots, \rho^K\right)$ определить класс, к которому он принадлежит, т. е. значение m. Для ее решения в данной работе использовался метод опорных векторов (SVM — support vector machine). Этот метод основан на построении в K-мерном пространстве $(K-1)$-мерной гиперплоскости, разделяющей объекты классов M_+ и M_-. В зависимости от расположения вектора ρ относительно этой гиперплоскости выполняется его отнесение к одному из классов.

Возможны следующие результаты интерпретации:

- $D(\rho_{i,j}) = D(\rho_{j,i}) = 0$. Оба документа признаются оригинальными;
- $D(\rho_{i,j}) \neq D(\rho_{j,i})$. Один из документов является оригиналом, а второй — дублем;
- $D(\rho_{i,j}) = D(\rho_{j,i}) = 1$. Оба документа являются дублями друг относительно друга, т. е. ни один из них не содержит оригинальных данных относительно другого.

На основе полученных результатов принимается решение о дальнейших действиях в отношении документов. Так, в разработанной системе решалась задача проверки документов в момент их поступления от источника, при этом загружаемые сообщения сравнивались на предмет дублирования с документами, уже загруженными в базу. Поэтому интерес представлял лишь один из результатов интерпретации — является ли загружаемый документ дубликатом ранее полученного сообщения. Однако при обработке готовой коллекции документов с целью обнаружения и устранения дубликатов важно выявить все пары сообщений, в которых имеет место дублирование, для чего требуется использовать оба результата.

6.2 Метод предварительного отбора кандидатов

Предложенный метод выявления нечетких дубликатов подразумевает, что каждое новое сообщение подвергается сравнению со всеми ранее загруженными и при каждом сравнении выполняется расчет близости документов по множеству критериев. На практике напрямую применять такой подход не представляется возможным из-за его высокой вычислительной сложности. Для увеличения быстродействия метода необходимо исключать из рассмотрения пары новостей, которые слишком сильно отличаются друг от друга. С этой целью вышеописанный метод предваряется процедурой отбора пар документов, являющихся потенциальными дубликатами:

$$F(d_i, d_j) = \begin{cases} 1, & \text{если документы — потенциальные дубликаты;} \\ 0, & \text{если документы не являются дубликатами.} \end{cases}$$

Пары, признанные процедурой кандидатами в дубликаты, должны проверяться основным методом, остальные же сообщения далее не рассматриваются (рис. 3).

Эта процедура, по сути, также решает задачу обнаружения дубликатов, причем основными требованиями, предъявляемыми к ней, являются минимальная вычислительная сложность, максимальная полнота, а также высокая эффективность (под эффективностью понимается отношение числа пар документов, отброшенных на этапе предварительного отбора, к исходному числу пар).

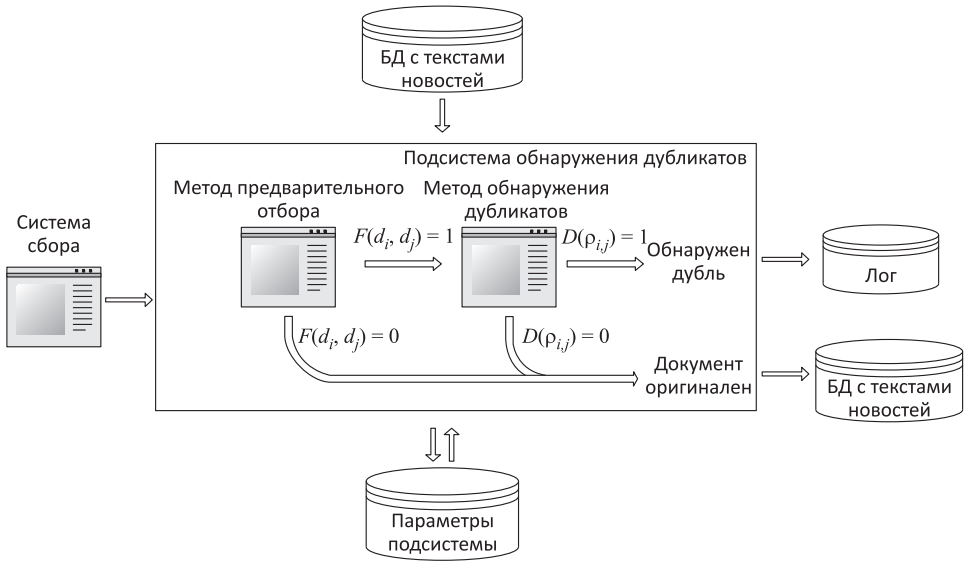


Рис. 3 Двухэтапный подход к обнаружению дубликатов

Ввиду наличия множеств взвешенных слов документов было решено положить в основу процедуры сравнение наборов наиболее значимых слов: $s = \{s^1, s^2, \ldots, s^A\}$, где A — мощность сравниваемых наборов, определяемая опытным путем. При этом документы признаются кандидатами в дубликаты, если мощность пересечения их наборов не меньше заданного порогового значения:

$$F(d_i, d_j) = \begin{cases} 1, & |s_i \cap s_j| \geq T; \\ 0, & |s_i \cap s_j| < T. \end{cases}$$

Мощность набора A и пороговое значение $T \in [0; A]$ являются настраиваемыми параметрами, выбираемыми исходя из желаемых показателей полноты и эффективности.

7 Экспериментальная проверка метода

В рамках данной работы были проведены эксперименты, направленные на анализ качества работы разработанной подсистемы обнаружения дубликатов, реализующей предложенный метод. Все эксперименты проводились на ПЭВМ со следующими основными параметрами: процессор Intel Core 2 Duo 2,2 ГГц, объем ОЗУ — 2 ГБ.

Целью первого эксперимента была оценка качества метода предварительного отбора потенциальных дубликатов в зависимости от параметров A и T на при-

Оценка качества метода отбора кандидатов

Конфигурация	A	T	Распознанные дубликаты	Ложные срабатывания	Пропущенные дубликаты	Полнота	Точность	Эффективность
1	2	2	352	482	324	52,1%	42,2%	99,99%
2	6	2	625	4231	51	92,4%	12,8%	99,98%
3	8	2	651	8724	25	96,3%	6,9%	99,96%

мере анализа общественно-политических новостей. Тестирование проводилось в течение суток. В качестве входных данных использовались 1 427 документов, извлеченных с 20 новостных сайтов. Каждый из документов подвергался сравнению с 17 835 загруженными ранее сообщениями. В общей сложности было обработано 25 450 545 пар документов, из которых 676 были дубликатами.

В таблице приведены результаты работы метода предварительного отбора при использовании трех конфигураций параметров процедуры.

Использование более мягких критериев фильтрации (конфигурация 3) предсказуемо приводит к увеличению полноты распознаваемых дубликатов и к уменьшению эффективности процедуры по сравнению с жесткими настройками (конфигурация 1). Исходя из требований к процедуре, оптимальной была признана конфигурация 3: сравнению должны подвергаться 8 наиболее значимых слов документа, при совпадении 2 и более документы считаются кандидатами в дубликаты. Такие параметры позволяют получить высокую полноту, равную 96,3%, при этом дальнейшая проверка основным методом требуется лишь для 0,04% от общего числа пар.

При таких настойках дубликаты составляют лишь 6,9% пар, прошедших отбор. Столь низкая точность процедуры доказывает необходимость дополнительного анализа отобранных пар с помощью основного метода.

В рамках второго эксперимента выполнялась оценка качества основного метода. Для тестирования использовались 26 036 документов, извлеченных с 20 новостных сайтов. С помощью метода фильтрации было отобрано 2650 пар-кандидатов, каждая из которых была проанализирована экспертами на предмет наличия дублирования. Часть пар использовалась для обучения, на остальных выполнялось тестирование метода. Целью эксперимента было определение зависимости показателей качества (точности, полноты и F-меры) от мощности обучающей выборки и от учитываемых критериев. Полученные зависимости представлены на рис. 4.

Как видно из рис. 4, при использовании 400 обучающих примеров происходит насыщение и с дальнейшим увеличением обучающей выборки качество работы метода не улучшается. Таким образом, для обучения системы достаточно 400 пар документов, размеченных экспертами.

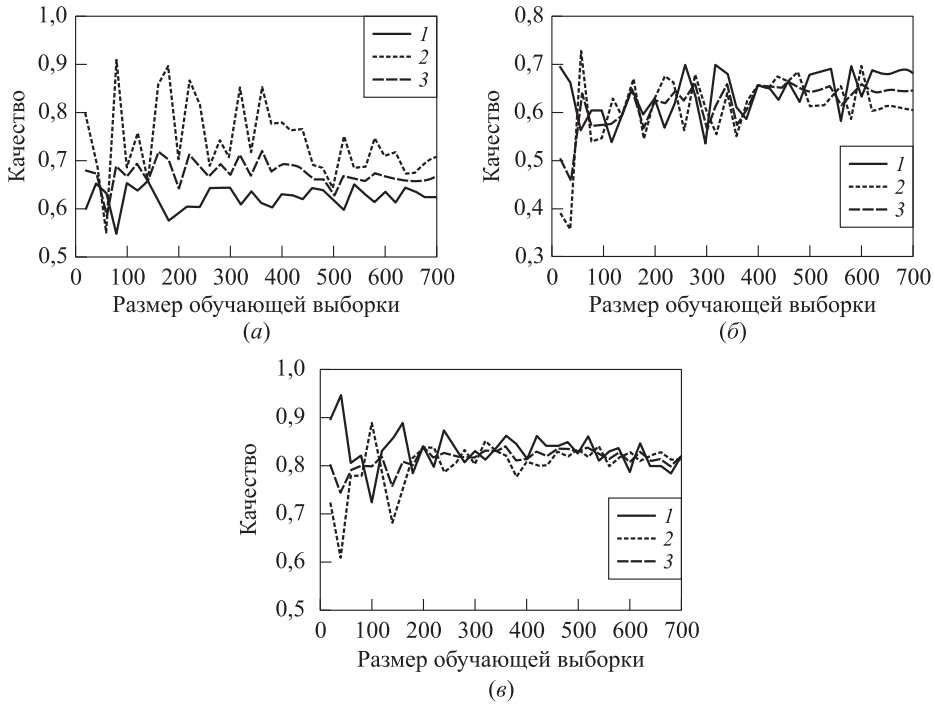


Рис. 4 Зависимость точности (*1*), полноты (*2*) и F-меры (*3*) от мощности обучающей выборки при использовании различных наборов критериев: (*а*) использование близости составов слов; (*б*) использование схожести абзацев; (*в*) использование всех критериев

Проведенный эксперимент доказывает целесообразность многокритериального сравнения документов: при использовании всех критериев достигаются более высокие показатели качества (F-мера в зоне насыщения равна 0,82), чем при анализе документов только с точки зрения слов (0,67) или абзацев (0,64).

При более подробном анализе примеров, на которых метод совершает ошибку, оказалось, что в большинстве случаев (76%) ошибка связана с человеческим фактором: каждый эксперт, участвовавший в подготовке обучающей и тестовой выборок, имеет свое представление о том, какие документы являются информативно необходимыми, а какие — нет, что приводит к конфликтам в суждениях экспертов. Лишь в 24% случаев ошибочность решения, принятого методом, не вызывала сомнений.

Проведенный эксперимент также показал, что среднее время, затрачиваемое на анализ пары документов основным методом, составляет 5 мс. С учетом высокой эффективности процедуры фильтрации это означает, что система способна обрабатывать 10 000 документов в час при наличии в базе 200 000 ранее загруженных сообщений, с которыми требуется сравнивать новые документы.

8 Направления дальнейших исследований

Помимо обнаружения и устранения дубликатов, предлагаемый метод может быть использован для решения других задач интеллектуального анализа текстов. При соответствующей настройке набора учитываемых критериев и порога близости документов, необходимого для вынесения решения о наличии дублирования, метод может быть применен для решения общей задачи обнаружения документов, близких по содержанию к заданному. Это позволит, в частности, выполнять формирование подборок тематически близких документов, а также сообщений, которые с большой долей вероятности связаны с общим событием.

Еще одним перспективным направлением развития метода является снабжение его возможностью не только обнаружения наличия или отсутствия дублирования, но и выделения в тексте близких по содержанию сообщений фрагментов с оригинальной (недублированной) информацией.

9 Заключение

В работе предложен метод обнаружения и устранения нечеткого дублирования в потоке текстовых сообщений. В его основе лежит отнесение пар документов к классу дубликатов или недубликатов с помощью метода опорных векторов.

Предлагаемый метод обладает высокой гибкостью благодаря возможности его настройки для обработки сообщений из различных предметных областей. Это достигается посредством включения в модель документа компонентов, отражающих критерии, которыми руководствуются эксперты при анализе текстовых коллекций вручную.

Для обеспечения низкой вычислительной сложности предложена процедура отбора пар-кандидатов на основе сравнения наборов наиболее значимых термов документов. Это позволяет применять основной метод лишь к парам документов, прошедшим отбор.

Представленный метод был апробирован при решении задачи анализа потока текстовых сообщений, загружаемых из открытых интернет-источников, с целью устранения документов, являющихся дубликатами ранее загруженных материалов.

Литература

1. *Ландэ Д. В., Дармохвал А. Т., Морозов А. Ю.* Подход к выявлению дублирования сообщений в новостных информационных потоках // Электронные библиотеки: перспективные методы и технологии, электронные коллекции: Тр. 8-й Всеросс. науч. конф. (RCDL'2006). — Ярославль: ЯрГУ им. П. Г. Демидова, 2006. С. 115–119.
2. *Зеленков Ю. Г., Сегалович И. В.* Сравнительный анализ методов определения нечетких дубликатов для Web-документов // Электронные библиотеки: перспективные методы и технологии, электронные коллекции: Тр. 9-й Всеросс. науч. конф.

(RCDL'2007). — Переславль-Залесский: Университет города Переславля, 2007. С. 166–174.

3. *Андреев А. М., Березкин Д. В., Козлов И. А., Симаков К. В.* Метод обнаружения изменений структуры веб-сайтов в системе сбора новостной информации // Электронные библиотеки: перспективные методы и технологии, электронные коллекции: Тр. 14-й Всеросс. науч. конф. (RCDL'2012). — Переславль-Залесский: Университет города Переславля, 2012. С. 124–133.
4. *Блох М. Я.* Теоретические основы грамматики: Учебник. — 2-е изд., испр. — М.: Высшая школа, 2000. 160 с.
5. *Chowdhury A., Frieder O., Grossman D., McCabe M.* Collection statistics for fast duplicate document detection // ACM Trans. Inform. Syst. (TOIS), 2002. Vol. 20. No. 2. P. 171–191.
6. *Ilyinsky S., Kuzmin M., Melkov A., Segalovich I.* An efficient method to detect duplicates of Web documents with the use of inverted index // 11th World Wide Web Conference (International) (WWW'2002) Proceedings. — Honolulu, Hawaii, USA: ACM Press, 2002. http://www2002.org/CDROM/poster/187/index.html.
7. *Brin S., Davis J., Garcia-Molina H.* Copy detection mechanisms for digital documents // ACM SIGMOD Record, 1995. Vol. 24. No. 2. P. 398–409.
8. *Broder A.* Algorithms for duplicate documents. http://www.cs.princeton.edu/courses/archive/spr05/cos598E/bib/Princeton.pdf.
9. *Manber U.* Finding similar files in a large file system // USENIX Winter 1994 Technical Conference Proceedings. — Berkeley, CA, USA: USENIX Association, 1994. P. 1–10.
10. *Shivakumar N., Garcia-Molina H.* SCAM: A copy detection mechanism for digital documents // 2nd Conference (International) in Theory and Practice of Digital Libraries (DL'95) Proceedings. — Austin, TX, USA, 1995. http://ilpubs.stanford.edu:8090/95/1/1995-28.pdf.
11. *Шевелёв О. Г.* Методы автоматической классификации текстов на естественном языке. — Томск: ТМЛ-Пресс, 2007. 144 с.
12. *Kołcz A., Chowdhury A., Alspector J.* Improved robustness of signature-based near-replica detection via lexicon randomization // 10th ACM SIGKDD Conference (International) on Knowledge Discovery and Data Mining Proceedings. — ACM, 2004. P. 605–610.
13. *Fetterly D., Manasse M., Najork M., Wiener J.* A large-scale study of the evolution of web pages // 12th Conference (International) on World Wide Web (WWW'03) Proceedings. — New York, NY, USA: ACM Press, 2003. P. 669–678.
14. *Bilenko M., Mooney R. J.* Adaptive duplicate detection using learnable string similarity measures // 9th ACM SIGKDD Conference (International) on Knowledge Discovery and Data Mining Proceedings. — ACM, 2003. P. 39–48.
15. *Князева А. А., Турчановский И. Ю., Колобов О. С.* Выявление дубликатов в библиографических базах данных // Электронные библиотеки: перспективные методы и технологии, электронные коллекции: Тр. 15-й Всеросс. науч. конф. (RCDL'2013). — Ярославль: ЯрГУ им. П. Г. Демидова, 2013. С. 207–213.

Поступила в редакцию 30.12.14

MULTICRITERIA METHOD FOR DETECTING NEAR-DUPLICATES IN A STREAM OF TEXT MESSAGES

A. Andreev, D. Berezkin, I. Kozlov, and K. Simakov

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

Abstract: The problem of near-duplicate detection in a stream of text messages is considered. A model of a text document and a multicriteria duplicate identification method is proposed. The model provides flexible adjustment for different domains. The method is based on binary classification using support vector machine. The paper also provides a method of candidates prefiltration in order to ensure high efficiency of the approach. Several experiments with data obtained from a stream of news articles were carried out. The results show feasibility of the suggested approach.

Keywords: near-duplicate detection; similarity measure; binary classification

DOI: 10.14357/08696527150103

References

1. Lande, D. V., A. T. Darmokhval, and A. Yu. Morozov. 2006. Podkhod k vyyavleniyu dublirovaniya soobshcheniy v novostnykh informatsionnykh potokakh [The approach to duplication detection in news information flows]. *Elektronnye biblioteki: Perspektivnye metody i tekhnologii, elektronnye kollektsii: Tr. 8-y Vseross. konf. (RCDL2006)* [Digital Libraries: Advanced Methods and Technologies, Digital Collections: 8th All-Russian Scientific Conference RCDL-2006 Proceedings]. Suzdal. 115–119.
2. Zelenkov, Yu. G., and I. V. Segalovich. 2007. Sravnitel'nyy analiz metodov opredeleniya nechetkikh dublikatov dlya Web-dokumentov [Comparative analysis of near-duplicate detection methods of Web documents]. *Elektronnye biblioteki: Perspektivnye metody i tekhnologii, elektronnye kollektsii: Tr. 9-y Vseross. nauch. konf. (RCDL'2007)* [Digital Libraries: Advanced Methods and Technologies, Digital Collections: 9th All-Russian Scientific Conference RCDL'2007 Proceedings]. Pereslavl'-Zalessky. 166–174.
3. Andreev, A. M., D. V. Berezkin, I. A. Kozlov, and K. V. Simakov. 2012. Metod obnaruzheniya izmeneniy struktury veb-saytov v sisteme sbora novostnoy informatsii [The method of detecting structure changes of news web sites]. *Elektronnye biblioteki: Perspektivnye metody i tekhnologii, elektronnye kollektsii: Tr. 14-y Vseross. nauch. konf. (RCDL2012)* [Digital Libraries: Advanced Methods and Technologies, Digital Collections: 14th All-Russian Scientific Conference RCDL2012 Proceedings]. Pereslavl'-Zalessky. 124–133.
4. Blokh, M. Ya. 2000. *Teoreticheskie osnovy grammatiki* [Theoretical Foundations of Grammar]. 2nd ed. Moscow: Vysshaya Shkola. 160 p.

5. Chowdhury, A., O. Frieder, D. Grossman, and M. McCabe. 2002. Collection statistics for fast duplicate document detection. *ACM Trans. Inform. Syst. (TOIS)* 20(2):171–191.
6. Ilyinsky, S., M. Kuzmin, A. Melkov, and I. Segalovich. 2002. An efficient method to detect duplicates of Web documents with the use of inverted index. *11th World Wide Web Conference (International) (WWW'2002) Proceedings*. Honolulu, Hawaii: ACM Press. Available at: http://www2002.org/CDROM/poster/187/index.html (accessed April 1, 2015).
7. Brin, S., J. Davis, and H. Garcia-Molina. 1995. Copy detection mechanisms for digital documents. *ACM SIGMOD Record* 24(2):398–409.
8. Broder, A. Algorithms for duplicate documents. Available at: http://www.cs.princeton.edu/courses/archive/spr05/cos598E/bib/Princeton.pdf (accessed December 17, 2014).
9. Manber, U. 1994. Finding similar files in a large file system. *USENIX Winter 1994 Technical Conference Proceedings*. Berkeley, CA, USA: USENIX Association. 1–10.
10. Shivakumar, N., and H. Garcia-Molina. 1995. SCAM: A copy detection mechanism for digital documents. *2nd Conference in Theory and Practice of Digital Libraries (DL'95) Proceedings.* Available at: http://ilpubs.stanford.edu:8090/95/1/1995-28.pdf (accessed April 1, 2015).
11. Shevelev, O. G. 2007. *Metody avtomaticheskoy klassifikatsii tekstov na estestvennom yazyke* [Methods of automatic classification of natural language texts]. Tomsk: TML-Press. 144 p.
12. Kołcz, A., A. Chowdhury, and J. Alspector. 2004. Improved robustness of signature-based near-replica detection via lexicon randomization. *10th ACM SIGKDD Conference (International) on Knowledge Discovery and Data Mining Proceedings*. 605–610.
13. Fetterly, D., M. Manasse, M. Najork, and J. Wiener. 2003. A large-scale study of the evolution of web pages. *12th Conference (International) on World Wide Web (WWW'03) Proceedings*. New York, NY: ACN Press. 669–678.
14. Bilenko, M., and R. J. Mooney. 2003. Adaptive duplicate detection using learnable string similarity measures. *9th ACM SIGKDD Conference (International) on Knowledge Discovery and Data Mining Proceedings*. 39–48.
15. Knyazeva, A. A., I. Yu. Turchanovskiy, and O. S. Kolobov. 2013. Vyyavlenie dublikatov v bibliograficheskikh bazakh dannykh [Identification of duplicates in bibliographic databases]. *Elektronnye biblioteki: Perspektivnye metody i tekhnologii, elektronnye kollektsii: Tr. 15-y Vseross. nauch. konf. (RCDL2013)* [Digital Libraries: Advanced Methods and Technologies, Digital Collections: 15th All-Russian Scientific Conference RCDL2013 Proceedings]. Yaroslavl. 207–213.

Received December 30, 2014

Contributors

Andreev Ark M. (b. 1943) — Candidate of Science (PhD) in technology, Associate professor, Bauman Moscow State Technical University, 5 Baumanskaya 2nd Str., Moscow 105005, Russian Federation; arkandreev@gmail.com

Berezkin Dmitry V. (b. 1966) — Candidate of Science (PhD) in technology, senior scientist, Bauman Moscow State Technical University, 5 Baumanskaya 2nd Str., Moscow 105005, Russian Federation; dmitryb2007@yandex.ru

Kozlov Ilya A. (b. 1989) — PhD student, Bauman Moscow State Technical University, 5 Baumanskaya 2nd Str., Moscow 105005, Russian Federation; kozlovilya89@gmail.com

Simakov Konstantin V. (b. 1980) — Candidate of Science (PhD) in technology, senior scientist, Bauman Moscow State Technical University, 5 Baumanskaya 2nd Str., Moscow 105005, Russian Federation; skv@ixlab.ru

ГЕНЕРАЦИЯ ТЕСТОВОГО НАБОРА НА ОСНОВЕ ПОТОКА УПРАВЛЕНИЯ*

***Н. В. Воинов*[1], *П. Д. Дробинцев*[2], *В. П. Котляров*[3], *И. В. Никифоров*[4], *И. А. Селин*[5]**

Аннотация: Статья посвящена подходу к генерации тестового набора в соответствии с широко известными структурными критериями покрытия на основе использования модели потока управления. Подход основан на автоматизированной генерации тестов, использующей символьную верификацию. К отличительным особенностям подхода можно отнести сокращение количества генерируемых тестов за счет учета данных о потоке управления и уменьшение пространства состояний, подлежащих обходу при генерации тестового набора инструментом верификации. В статье представлены основные идеи подхода, описана формальная модель потока управления и инструментальные средства для работы с ней, а также приведены результаты применения подхода в нескольких проектах по разработке промышленного программного обеспечения (ПО).

Ключевые слова: автоматизация тестирования; формальная модель; критерии покрытия

DOI: 10.14357/08696527150104

1 Введение

Сложность разработки и тестирования современного ПО приводит к необходимости создания новых подходов, позволяющих полностью или частично автоматизировать данный процесс. Одним из наиболее активно развивающихся направлений стало использование технологий верификации для обеспечения качества ПО. При этом инструмент верификации часто используется для проверки

*Статья рекомендована к публикации в журнале Программным комитетом конференции «Tools & Methods of Program Analysis» («Инструменты и методы анализа программ», TMPA-2014), г. Кострома, 14–15 ноября 2014 г.

[1]Санкт-Петербургский государственный политехнический университет, Voinov@ics2.ecd.spbstu.ru

[2]Санкт-Петербургский государственный политехнический университет, drob@ics2.ecd.spbstu.ru

[3]Санкт-Петербургский государственный политехнический университет, vpk@ics2.ecd.spbstu.ru

[4]Санкт-Петербургский государственный политехнический университет, igor.nikiforovv@gmail.com

[5]Санкт-Петербургский государственный политехнический университет, ivanselin93@gmail.com

свойств модели программы, заданной на каком-либо формальном языке. В силу того, что при данном подходе проверяется не реализация приложения, а только его модель, становится очевидным, что одной лишь проверки на модели недостаточно, а необходимо также проведение тестирования на основе результатов верификации. Такой подход носит название тестирования на основе моделей. Однако подобное тестирование сопряжено со следующими сложностями:

– мощность инструментов верификации позволяет получить огромное количество сценариев (трасс), описывающих поведение системы. Полученные трассы используются для генерации тестов. Однако использование этого множества ограничено возможностями по прогону тестов на тестируемом приложении, и если в случае использования инструмента верификации получение 100 000 трасс не является проблемой, то прогон такого количества тестов невозможен в силу ограниченности времени, отведенного на цикл тестирования;

– при использовании инструмента верификации для генерации тестов программных систем возникает взрыв числа состояний, подлежащих обходу при получении тестового сценария. Это обусловлено тем, что состояние формируется как совокупность значений всех переменных, используемых в тестируемом приложении (их количество может быть огромным), поэтому, обеспечивая перебор значений, инструмент верификации сталкивается с необходимостью рассмотрения огромного количества состояний.

Данная работа посвящена рассмотрению подхода, позволяющего решить вышеуказанные проблемы за счет использования модели потока управления, описанной на высокоуровневом графическом языке моделирования. Предлагаемый подход позволяет выделить подмножество тестов в соответствии с различными критериями тестирования, а также существенно сократить пространство состояний при работе верификатора.

2 Формальные модели

Индустрия промышленного производства программных продуктов в настоящее время достигла таких масштабов, что для обеспечения соответствующего качества уже недостаточно традиционных способов ручного тестирования: индустрия требует как высокого качества тестирования, так и достаточного уровня его автоматизации. Тестирование на основе моделей способно предоставить приемлемый уровень качества и автоматизации.

Создание любой системы начинается с построения модели и изучения ее свойств. Модели, которые описывают структуру системы, часто называют архитектурными, а модели, которые описывают поведение системы — поведенческими. Модель — формализованное описание структуры и поведения системы [1]. Модель может описываться в терминах состояния системы, входных

воздействий на нее, конечных состояний, потоков данных и потоков управления, возвращаемых системой результатов и т. д.

Большое число ошибок при составлении требований возникает из-за неоднозначной интерпретации их заказчиком и разработчиком. Чтобы избежать таких ошибок, используют формальные спецификации. Формальная спецификация представляет собой законченное описание модели системы и требований к ее поведению в терминах той или иной формальной нотации. Для описания характеристик системы можно использовать несколько моделей в рамках нескольких формализмов. Обычно чем более абстрактной является нотация моделирования, тем больше трудностей возникает при автоматизации тестирования программы на основе модели/спецификации, описанной в этой нотации. Одни нотации и языки больше ориентированы на доступность и прозрачность описания, другие — на последующий анализ и трансляцию, в частности трансляцию спецификации в тест.

Одной из серьезных проблем при создании формальной модели системы на основе неформализованных исходных требований является их неоднозначная интерпретация заказчиком и разработчиком. На начальном этапе разработки ПО как у заказчика, так и у исполнителей может отсутствовать четкое представление об архитектуре будущей системы, есть лишь общее представление о ее поведении. Но уже на начальном этапе дизайна необходимо находить возможные ошибки и несоответствия требованиям. Большинство инструментов тестирования на основе моделей используют формальные нотации достаточно низкого уровня. Такие нотации, как Specification Description Language (SDL) [2], Unified Modeling Language (UML) [3], конечные автоматы и др., вынуждают пользователя сразу разрабатывать поведение системы на достаточно детальном уровне. Как показывает практика, использование подобных нотаций для контроля заказчиком этапа дизайна проектируемой системы оказывается неудобным, а детализация — слишком низкоуровневой (избыточной).

Решением данной проблемы является использование более высокоуровневой формальной нотации, которая бы стала связующим звеном между исходными требованиями и формальной моделью, с которой непосредственно работают методы верификации и тестирования на основе моделей [4].

3 Методы создания тестовых сценариев на основе модели

В настоящее время методы тестирования на основе моделей используют множество типов моделей и техник, которые можно условно разделить на три группы [5].

3.1 Методы проверки согласованности автоматов и систем переходов

Такие методы относятся к одному из трех типов в зависимости от используемых моделей.

– *Обычные и расширенные автоматы* [6, 7]. Методы построения тестов на основе конечных автоматов наиболее глубоко разработаны, известны их точные ограничения и гарантии полноты выполняемых проверок. Методы, использующие расширенные автоматы, сводят их к обычным, но применяют более детальные критерии покрытия, основанные на использовании данных в расширенных автоматах. Наиболее известными инструментами создания тестов на основе таких моделей являются BZ-TT [7, 8], использующий модели, описанные на языке B [9], и GOTCHA-TC Beans [10, 11], или UML State charts в рамках набора инструментов Telelogic [12].

– *Системы переходов*. Такие методы чаще используются при тестировании распределенных систем, поскольку моделирование таких систем с помощью конечных автоматов очень трудоемко. Большинство этих методов не определяют практически применимых критериев полноты и не дают конечных тестовых наборов для реальных систем, поэтому использующие их инструменты опираются на те или иные эвристики для обеспечения конечности набора тестов. Первые методы построения тестов по LTS (Label Transition Systems) моделям были разработаны в работах Бринксмы [13] и Тритманса [14]. Наиболее известны из инструментов, созданных на основе результатов Тритсманса, TorX [15] и TGV [16]. Помимо обычных систем переходов используются и временны́е (расширенные с помощью таймеров), построение тестов на их основе возможно с использованием инструмента UPAAL [17]. Примерами инструментальной поддержки таких методов служат инструменты технологии UniTESK [18, 19], созданной в ИСП РАН, и инструмент SpecExplorer [20, 21], разработанный в Microsoft Research.

3.2 Методы построения тестов на основе формального анализа свойств программного обеспечения

Данные методы используют формальный анализ для классификации тестовых ситуаций и нацеленной генерации тестов.

– *Методы на основе классов эквивалентности* [22–24]. В таких методах тестовые ситуации выбираются как представители классов эквивалентности, задаваемых критерием покрытия. Часто за основу разбиения выбирается используемый критерий покрытия, а ситуации, которые соответствуют одному покрываемому элементу в рамках этого критерия, считают эквивалентными. Далее тесты строятся так, чтобы в каждом классе эквивалентности был хотя бы один тест. К данному классу методов относится, например, метод функциональных диаграмм Майерса [25].

– *Методы на основе дедуктивного анализа* (см., например, [26]). В этих методах выбираемые тестовые ситуации соответствуют особым случаям в дедуктивном анализе свойств тестируемой системы. Метод дедуктивного анализа (theorem proving) — проверка того, что набор утверждений, представляющий

спецификацию, формально выводится из реализации и, быть может, каких-то гипотез о поведении окружения системы, сформулированных в том же формализме, что и реализация [3].

3.3 Методы построения тестов с помощью символьного выполнения (symbolic execution)

Обычно подобные методы используют символьное описание пути, содержащее наборы предикатов. Данный путь проходит во время выполнения теста по коду программы (или пути по проверяемым формальным спецификациям). Такое описание позволяет выбирать новые тестовые ситуации так, чтобы они покрывали другие пути, и строить тесты с помощью техник разрешения ограничений (constraint solving) [27, 28]. Символьное выполнение в комбинации с конкретизацией тестовых данных используется и для построения тестов, нацеленных на типичные дефекты, такие как использование неинициализированных объектов, тупики или гонки параллельных потоков [29].

Более детальная информация о методах тестирования на основе моделей содержится в [30, 31].

Следует отметить, что наиболее развиты методы тестирования на основе моделей первой из перечисленных выше групп. Поддерживающие их инструменты все шире используются в промышленных проектах. Методы построения тестов на основе проверки моделей и на основе символьного выполнения активно развиваются и реализуются в инструментальных средствах.

4 Предлагаемый подход

Рассмотрим более детально этапы работы методов тестирования на основе модели с применением методов верификации. При использовании верификации на первом этапе проверяются свойства разработанной формальной модели ПО и на базе полученных результатов вносятся корректировки в требования и программный код. По сути, происходит формальный анализ заданных свойств ПО, описанный в предыдущем разделе. В процессе анализа большинство средств верификации предоставляет возможность получения так называемого контрпримера, который описывает поведение системы, приводящее к нарушению заданного пользователем свойства. Данный контрпример полезно использовать при тестировании для проверки того факта, что внесенные в код системы на основе результатов верификации изменения привели к исправлению ее поведения и заданное свойство больше не нарушается при функционировании ПО. Тестирование с помощью контрпримеров позволяет получить тесты, проверяющие достаточно узкий класс поведений и не обеспечивающие полноты покрытия. Таким образом, данные тесты рациональнее всего использовать как дополнение к регрессионному тестированию, проводимому после внесения в код системы исправлений, вызванных анализом нарушения ее свойств.

Альтернативным подходом к получению тестов является их генерация в процессе обхода дерева поведения системы, при этом для сокращения числа генерируемых тестов часто используются методы символьной верификации, позволяющие получать в параметрах теста не конкретные значения переменных, а их допустимые диапазоны. Одной из основных задач при использовании подобной генерации является задача определения критериев, которым должны соответствовать тесты для обеспечения необходимого покрытия.

В рассматриваемом в рамках работы подходе используется генерация тестов посредством обхода дерева поведения системы на основе модели потока управления с использованием методов символьной верификации и ограничением числа тестов структурными критериями.

На втором этапе проводится тестирование кода с помощью тестов, полученных на предыдущем этапе. Для данного этапа характерными задачами являются конкретизация параметров теста в случае использования символьной верификации, обеспечение взаимодействия теста с приложением и анализ результатов выполнения тестов.

Предлагаемый подход условно можно разделить на следующие составные части:

- ручная формализация модели в высокоуровневой графической нотации. Предлагается использовать язык UCM, позволяющий формально определить потоки управления в модели и в то же время являющийся достаточно высокоуровневым языком, доступным для понимания специалистами из предметных областей разрабатываемого ПО;
- верификация модели и ее корректировка. На данном этапе используется система верификации VRS (Verification Requirements System) [32], позволяющая проводить верификацию с использованием метода проверки на модели и символьной верификации;
- генерация тестовых сценариев по модели. На основе обхода дерева поведения проводится генерация символьных тестов в соответствии с определенным пользователем структурным критерием. Тесты генерируются на языке MSC (Message Sequence Charts) [33], представляющем собой диаграммы обмена последовательностями сообщений;
- генерация исполняемых тестов. Разделяется на два этапа: на первом, подготовительном, происходит конкретизация тестов соответствующими значениями переменных, на втором происходит генерация тестов с использованием системы тестирования TAT [32] на целевой язык;
- исполнение тестов и оценка результатов. Данный этап связан с исполнением тестов и анализом полученных логов, представленных в формате MSC.

Реализация описанного подхода приведена на рис. 1 в виде технологической цепочки, описывающей совместное использование инструментов VRS и TAT (Thematic Apperception Test). В рамках цепочки можно выделить следующие основные этапы:

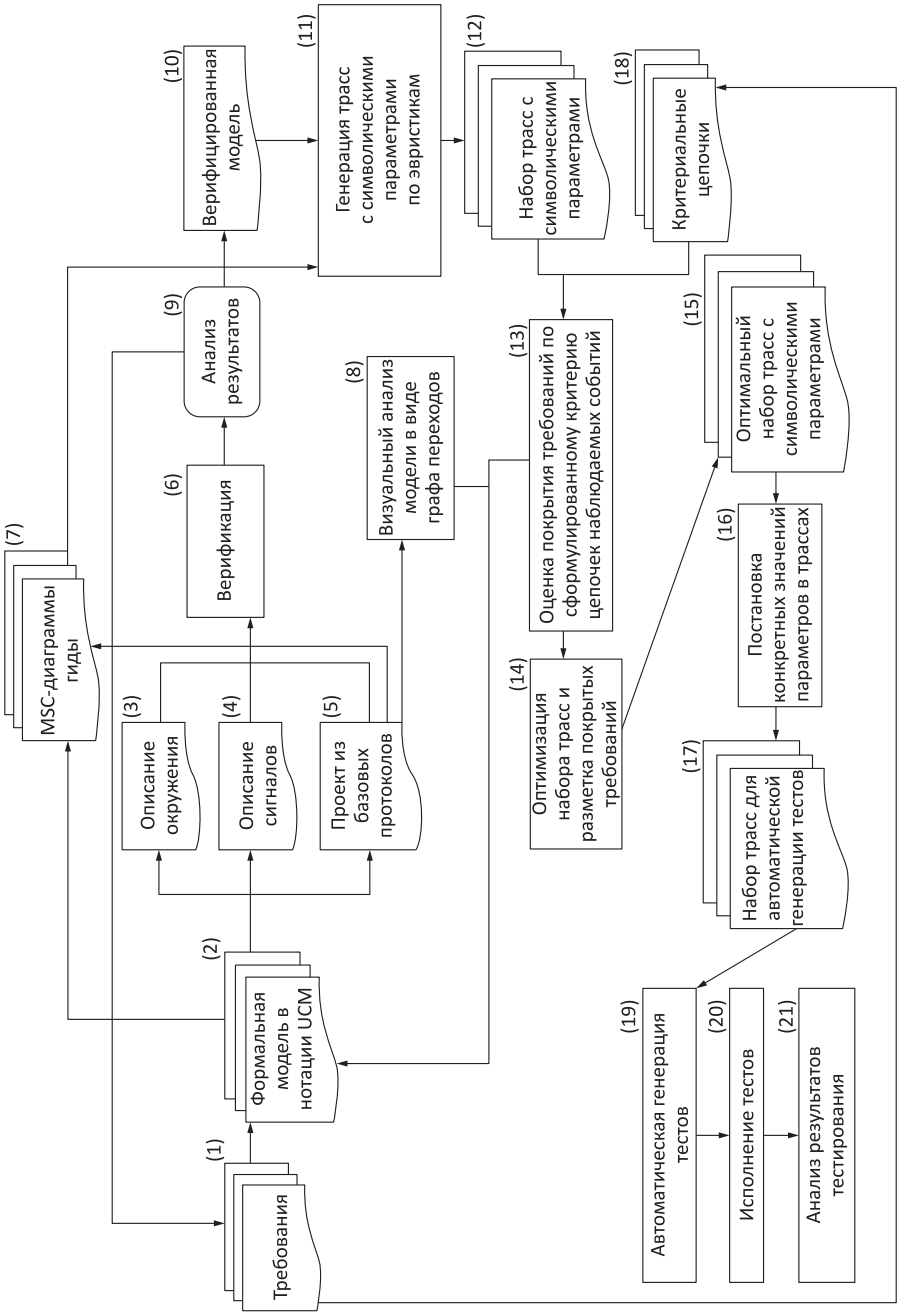


Рис. 1 Технологическая цепочка предлагаемого подхода

1. По исходным требованиям (1) вручную создается формальная модель системы в нотации UCM (2).
2. С помощью инструмента UCM2BP [2] производится трансляция UCM-проекта в набор базовых протоколов [32] (элементарных MSC-диаграмм, каждая из которых описывает законченное действие в пространстве поведения модели) (5), файлы описания сигналов (4) и окружения (3).
3. Средствами VRS осуществляется верификация модели (6).
4. В случае нахождения ошибок или неточностей в модели (например, недетерминированного поведения, тупиковых или недостижимых состояний и т. п.) требования исправляют, что приводит к изменениям в формальной модели.
5. С помощью инструмента UCM2MSC производится генерация MSC-диаграмм (7), которые транслируются в гиды [34], пригодные для инструмента VRS.
6. По верифицированной модели (10) и гидам (7) генерируются трассы (12).
7. Если трассы содержали символические параметры, осуществляется подстановка конкретных значений (16).
8. Средствами TAT выполняется генерация тестов (19) по набору трасс (17) и исполнение тестов (20) с последующим анализом результатов тестирования (21).

Для сокращения пространства состояний анализируемой модели будущего ПО предлагается использование гидов. Гиды определяет те точки поведения системы, которые наиболее интересны пользователю, и таким образом система верификации получает подсказку, дающую информацию о том, через какие точки поведения должна пройти символьная трасса. Одним из наиболее эффективных методов автоматического определения гидов является их генерация на основе структурных критериев покрытия, например критерия ветвей [35]. При этом UCM-модель системы может быть использована как модель потока управления. Данные методы, позволяющие осуществлять автоматическую генерацию гидов, реализуются на 5-м шаге описанной технологической цепочки, а именно в инструменте UCM2MSC.

Как было указано выше, генерация гидов производится по формальной модели системы в нотации UCM, но так как генерация тестовых трасс производится в верификаторе VRS, то гиды реализуются в терминах модели базовых протоколов. Для этого используется инструмент UCM2BP, который отвечает за трансляцию UCM-нотации в нотацию базовых протоколов. Рассмотрим особенности использования нотации UCM для реализации модели потока управления.

5 Модель потока управления в нотации UCM

Использование UCM для создания формальной модели поведения системы подразумевает описание последовательности действий, которые может произво-

дить система в ответ на внешние воздействия пользователей или систем окружения. Варианты использования отражают не только функциональность системы с точки зрения описания ее архитектуры, но и определяют особенности потока управления, лежащего в основе будущего приложения. Явное представление вариантов (режимов функционирования) поведения приложения привносит важную информацию в процесс разработки программных систем [36]:

- заполняет промежуток между словесным описанием требований и детальным дизайном системы;
- позволяет разработать архитектуру системы на высоком уровне абстракции, а также задать поведение системы на полученной архитектуре;
- помогает разработчику предсказывать поведение сложных систем;
- предоставляет удобную нотацию, позволяющую изображать на диаграмме параллельные процессы, таймеры, точки прерывания, аспекты.

Дизайн системы в нотации UCM представляет собой набор взаимодействующих между собой диаграмм. В свою очередь, каждая из диаграмм сосредоточена на описании взаимодействия компонентов (агентов, процессов системы), объектов, наблюдателей и подсистем. Каждый компонент или подсистема содержит элементы ответственности (responsibilities), соответствующие событиям в системе, а также упорядоченную последовательность их возникновения, которая, по сути, является описанием потока управления. Таким образом, совокупность компонентов и диаграмм дает пользователю наглядное представление о поведении системы и взаимодействии между ее компонентами. Разработка UCM-диаграмм осуществляется с использованием графического редактора UCMNavigator [37].

Формализация поведения системы дополняется использованием метаданных [2], которые могут быть добавлены практически ко всем элементам UCM-диаграммы. Метаданные представляют собой набор данных формата: «Маркер» – «Текстовая информация заданного типа», в рамках которого происходит определение операций с переменными модели, а также прием и передача сигналов для обмена с окружением.

6 Инструментальная реализация

В технологической цепочке VRS/TAT за генерацию тестовых трасс отвечает инструмент VRS. Технология VRS использует нотацию базовых протоколов, что не обеспечивает UCMNavigator.

Предлагаемый подход использует инструмент UCM2BP, отвечающий за автоматическую трансляцию UCM-проекта в нотацию базовых протоколов. UCM2BP позволяет получить набор базовых протоколов, которые используются в качестве шагов MSC-сценария.

UCM-диаграмма представляет собой структурированный граф. Задача обхода UCM-диаграммы и построения MSC-сценариев сводится к обходу графа.

При увеличении размера UCM-проекта число возможных сценариев растет экспоненциально, что приводит к так называемой проблеме «взрыва состояний». Для решения данной проблемы в разработанном подходе используются:

- структурный критерий покрытия: критерий ветвей;
- гиды с глубиной.

6.1 Критерий покрытия

Современные методы тестирования обычно оценивают полноту покрытия тестовым набором по таким критериям, как число выполненных операторов программы, ветвей, путей, проверенных значений данных, покрытие граничных значений функции, переходов между состояниями и т. д. В разработанном подходе для автоматической генерации набора MSC-сценариев используется структурный критерий ветвей.

Структурные критерии покрытия обеспечивают оценку полноты тестирования и основаны на структуре тестируемой системы. В основе структурных критериев полноты лежит простая идея: если ошибка находится в какой-то конструкции кода или компоненте тестируемой системы, то тест, инициировавший выполнение этой конструкции или компоненты, скорее всего, сможет ее обнаружить.

Условие критерия тестирования ветвей (критерий P2 [35]) — набор тестов в совокупности должен обеспечить прохождение каждой ветви не менее одного раза. Это достаточно сильный и при этом экономичный критерий, поскольку множество ветвей тестируемой системы конечно и не так уж велико. UCM-нотация позволяет осмысленно выделить ветви на диаграмме — участки пути между операторами ветвлений (OrFork, OrJoin, AndFork, AndJoin, Waiting place, Timer), а также стартовыми (StartPoint) и конечными (EndPoint) UCM-элементами.

Выбор критерия покрытия по ветвям позволяет производить генерацию конечного числа тестовых сценариев, причем количество полученных сценариев меньше или равно общему числу ветвей всей системы. Такой подход позволяет добиться адекватности набора тестов и хорошей скорости генерации тестовых сценариев.

6.2 Использование гидов

Использование гидов позволяет пользователю задавать направление поиска трасс для тестовых сценариев при обходе пространства поведения модели. Метод подразумевает наложение ограничений на размер тестового сценария, что дает возможность проверить его допустимость. Критерии покрытия формулируют дополнительные ограничения на поиск, отсекая ветви поведения модели, не удовлетворяющие тестовому сценарию. Подробное описание концепции и реализации метода гидов приведено в [34].

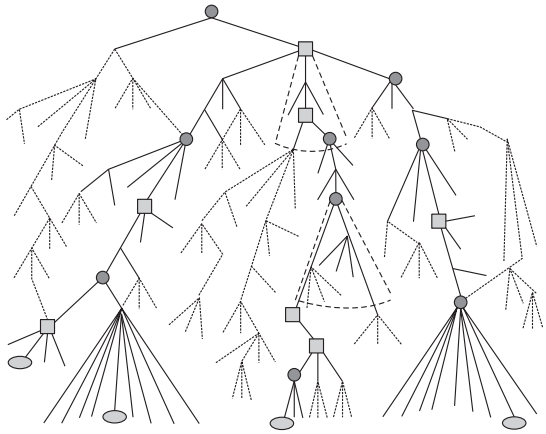


Рис. 2 Применение гидов при обходе дерева поведения

Основная идея представлена на рис. 2, где изображено дерево поведения модели, построенное на основании множества базовых протоколов, описывающих модель (вершины — состояния модели, переходы — базовые протоколы).

Пользователь, зная, какие именно элементы дерева поведения модели (сигналы, изменения атрибутов и состояний и т. п.) должны быть включены в тестовый сценарий, чтобы соответствующее требование было покрыто, может указать определенные состояния и переходы, т. е. задать промежуточные точки (отмечены кружками), через которые осуществляется трассовая генерация. Таким образом существенно ограничивается пространство поиска от одного события (промежуточной цели) в тестовом сценарии до другого. При необходимости путь от начального события до конечного может быть уточнен дополнительными событиями (отмечены квадратами на рис. 2), конкретизирующими путь на дереве поведений. Разработанные методы позволяют создавать гиды по UCM-диаграммам. С помощью UCM2BP UCM-диаграммы транслируются в набор базовых протоколов, а имена соответствующих UCM-элементам протоколов используются для построения гидов.

Чтобы получить желаемые трассы, в инструменте VRS используются созданные гиды. Гиды задают направление, в котором VRS обходит дерево поведения системы. VRS позволяет использовать глубины в гидах, т. е. задавать участки трассы, на которых переход от одного базового протокола к другому возможен с использованием нескольких путей, содержащих между искомыми базовыми протоколами несколько промежуточных протоколов, число которых ограничивается глубиной.

Использование глубин возможно в нескольких ситуациях:

– для сокрытия альтернативного выбора;

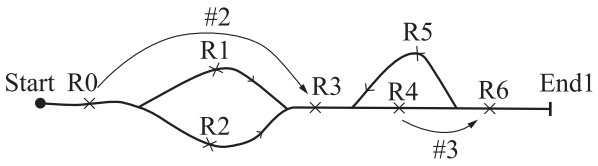


Рис. 3 Пример использования глубины в гидах

– для сокрытия цикла;

– для сокрытия параллелизма.

Пример использования гидов приведен на рис. 3. Если существует необходимость сгенерировать сценарий, который направлен на покрытие ветви, содержащей элемент R6, то без анализа метаданных неизвестно, какой элемент будет использован: R1 или R2, а также будет ли задействован цикл R5. В таком случае, чтобы учесть все варианты путей, которые могут привести к применению R6, необходимо построить 4 гида:

(1) R0, R1, R3, R4, R6;

(2) R0, R1, R3, R4, R5, R4, R6;

(3) R0, R2, R3, R4, R6;

(4) R0, R2, R3, R4, R5, R4, R6.

Однако если указать глубины в гиде, то можно построить лишь один гид: R0, R3#2, R6#3 (# — разделитель имени элемента и величины глубины). Данный гид указывает направление на дереве поведения и интерпретируется следующим образом: применить R0, затем использовать не более одного протокола, прежде чем применить R3, а затем применить R6, использовав при этом не более трех протоколов, выбор которых осуществляет система верификации.

Таким образом, использование критерия ветвей и гидов с глубиной позволяет получить набор сценариев, где каждой ветви соответствует один гид, по которому будет сгенерирована одна трасса. Такой подход решает проблему «взрыва состояний», а также существенно сокращает время генерации сценариев.

6.3 Автоматическая генерация на основе критерия ветвей

1. Генерация базовых протоколов по UCM-проекту.
2. Сопоставление базовых протоколов и элементов UCM-диаграммы.
3. Выделение ветвей в структуре UCM.
4. Генерация гида для каждой ветви.
5. Проведение оптимизации — удаление избыточных диаграмм.

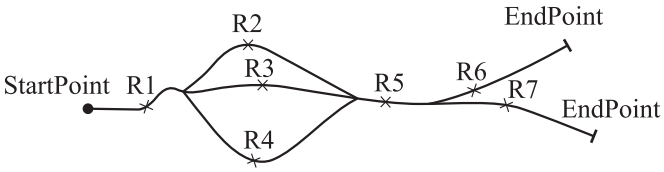


Рис. 4 Пример UCM-диаграммы

Использование программных объектов базовых протоколов при генерации с помощью UCM2BP позволяет сохранить структуру UCM. Программные объекты базовых протоколов содержат ссылки на предыдущие и последующие протоколы, сгенерированные в соответствии со структурой UCM-диаграммы. Также они содержат ссылки на элементы UCM. Все это позволяет однозначно интерпретировать структуру UCM-проекта при генерации MSC-диаграмм в терминах модели базовых протоколов.

UCM-диаграмма, приведенная на рис. 4, состоит из следующих ветвей: R1, R2, R3, R4, R5, R6, R7.

Для каждой ветви генерируется гид, который содержит в себе путь от стартового состояния системы до данной ветви. Для примера на рис. 4 будут сгенерированы следующие гиды:

- R1;
- R1, R2;
- R1, R3;
- R1, R4;
- R1, R5(#2);
- R1, R5(#2), R6;
- R1, R5(#2), R7.

Отметим, что получившийся набор гидов является избыточным, так как главной задачей при использовании критерия ветвей является покрытие каждой ветви не менее одного раза. Так как некоторые ветви покрываются в нескольких гидах, то следует провести оптимизацию и удалить гиды с повторяющимися ветвями. Таким образом, после оптимизации, состоящей из удаления избыточных гидов, будет получен следующий набор:

- R1, R2;
- R1, R3;
- R1, R4;
- R1, R5(#2), R6;
- R1, R5(#2), R7.

Данного набора гидов достаточно для генерации тестовых трасс, удовлетворяющих структурному критерию покрытия по ветвям. Следует отметить, что использование верифицированной модели базовых протоколов для генерации трасс дает уверенность в корректности результата.

Несложно заметить, что некоторые из полученных гидов не ведут процесс генерации трасс до конечных состояний системы. В то же время получение «длинного» гида (до конечного состояния) дает возможность формирования теста, приводящего систему в заранее известное состояние, определяющее окончание работы системы. Подобная генерация также может быть автоматизирована. Для рассматриваемого примера набор «длинных» гидов будет следующим:

- R1, R2, R5, R6;
- R1, R2, R5, R7;
- R1, R3, R5, R6;
- R1, R3, R5, R7;
- R1, R4, R5, R6;
- R1, R4, R5, R7;
- R1, R5(#2), R6;
- R1, R5(#2), R7.

Количество гидов увеличилось. Это объясняется тем, что на уровне генерации гиды без анализа метаданных не дают возможности определить, в какое из альтернативных конечных состояний придет система. Поэтому для сохранения покрытия по критерию ветвей «короткие» гиды продлеваются до всех возможных конечных состояний. Данная особенность приводит к ситуации, когда количество гидов становится избыточным, причем не все из них могут быть покрыты соответствующими трассами. Это происходит из-за несоответствия потока управления, описанного на диаграмме, метаданным, указанным в элементах responsibility. В результате избыточные гиды, по которым не сгенерируется трасса, должны быть исключены из тестового набора.

В результате автоматической генерации на основе анализа модели потока управления получается набор тестов, которые направлены на обнаружение ошибок и удовлетворяют структурному критерию покрытия ветвей.

7 Результаты применения подхода

Результаты применения описанного подхода в четырех проектах среднего размера по разработке модулей телекоммуникационных приложений приведены в таблице. В качестве документации на проект были использованы UCM-диаграммы, а выходом являлся готовый тестовый набор для прогона на реальном коде приложения.

Результаты применения подхода

Проект	Количество базовых протоколов	Количествово ветвей	Количествово гидов	Количество сгенерированных символьных трасс
1	358	149	437	372
2	163	240	139	131
3	191	111	87	72
4	214	200	118	104

Все результаты по проектам были получены с помощью автоматического режима для получения набора трасс, покрывающих модель по критерию ветвей с использованием «длинных» гидов, что позволило ускорить разработку тестовых сценариев в 250 раз по сравнению с традиционным ручным подходом.

В то же время результаты, полученные в рамках применения описанного подхода, свидетельствуют о более чем 70%-ном сокращении временны́х затрат по сравнению с подходом, основанном на простом обходе дерева поведения системы без использования гидов.

Литература

1. IEEE Standard Glossary of Software Engineering Terminology: IEEE Standard 610.12-1990. — New York, N.Y., USA: IEEE, 1990.
2. ITU-T. Specification and description language (SDL). Protocol Z.100. Telecommunication standardization sector of ITU, 2002. 206 p. https://www.itu.int/ITU-T/2005-2008/com17/languages/Z100.pdf.
3. Unified Modeling Language (UML) Resource Page. http://www.uml.org.
4. *Никифоров И. В., Петров А. В., Юсупов Ю. В., Котляров В. П.* Применение различных методик формализации для построения верификационных моделей систем по UCM-спецификациям // Научно-технические ведомости СПбГПУ, 2011. № 3(126). С. 180–184.
5. *Кулямин В. В.* Методы верификации программного обеспечения. — М.: ИСП РАН, 2008. 117 с. http://www.ict.edu.ru/ft/005645/62322e1-st09.pdf.
6. Model-based testing of reactive systems / Eds. M. Broy, B. Jonsson, J.-P. Katoen, M. Leucker, A. Pretschner. — Lecture notes in computer science ser. — Springer, 2005. Vol. 3472. 659 p.
7. *Utting M., Legeard B.* Practical model-based testing: A tools approach. — San Francisco, CA, USA: Morgan-Kaufmann Publs., 2007. 456 p.
8. *Ambert F., Bouquet F., Chemin S., Guenaud S., Legeard B., Peureux F., Vacelet N., Utting M.* BZ-TT: A tool-set for test generation from Z and B using constraint logic programming // Formal Approaches to Testing of Software, FATES 2002 (Workshop of CONCUR'02) Proceedings. P. 105–119.
9. *Abrial J.-R.* The B-book: Assigning programs to meanings. — Cambridge: Cambridge University Press, 1996. 813 p.

10. *Farchi E., Hartman A., Pinter S. S.* Using a model-based test generator to test for standard conformance // IBM Syst. J., 2002. Vol. 41. No. 1. P. 89–110.
11. GOTCHA-TCBeans: Model driven engineering technology. IBM Haifa Research Lab. www.haifa.ibm.com/projects/verification/gtcb/index.html.
12. Rational TAU: Standards-based, model-driven development solution for complex systems. IBM Software. http://www-03.ibm.com/software/products/en/ratitau.
13. *Brinksma E.* A theory for the derivation of tests // 8th Conference (International) on Protocol Specification, Testing and Verification Proceedings. — North Holland, 1988. P. 63–74.
14. *Tretmans J.* A formal approach to conformance testing. — Enschede, The Netherlands: University of Twente, 1992. PhD Thesis. 282 p.
15. *Tretmans J., Belinfante A.* Automatic testing with formal methods // 7th European Conference on Software Testing, Analysis and Review Proceedings. — Barcelona, Spain, 1999. P. 8–10.
16. *Fernandez J.-C., Jard C., Jéron T., Viho C.* Using on-the-fly verification techniques for the generation of test suites // Computer aided verification / Eds. R. Alur, T. A. Henzinger. — Lecture notes in computer science ser. — Springer, 1996. Vol. 1102. P. 348–359.
17. UPPAAL: Design verification for embedded systems. — Uppsala, Sweden: UP4ALL International AB, 2012. http://www.uppaal.com/.
18. *Кулямин В. В., Петренко А. К., Косачев А. С., Бурдонов И. Б.* Подход UniTesK к разработке тестов // Программирование, 2003. № 29(6). С. 25–43.
19. UniTESK: Индустриальная технология надежного тестирования. — М.: ИСП РАН, 2014. http://www.unitesk.ru.
20. Model-based testing with SpecExplorer. Microsoft Research. http://research.microsoft.com/ specexplorer.
21. Spec Explorer. MSDN Library. http://msdn.microsoft.com/en-us/library/ee620411.aspx.
22. *Ammann P., Black P. E.* Abstracting formal specifications to generate software tests via model checking // 18th Digital Avionics Systems Conference Proceedings. IEEE, 1999. Vol. 2. P. 10.A.6-1–10.A.6-10.
23. *Gargantini A., Heitmeyer C.* Using model checking to generate tests from requirements specifications // ACM SIGSOFT Software Engineering Notes, 1999. Vol. 24. No. 6. P. 146–162.
24. *Visser W., Pasareanu C. S., Khurshid S.* Test input generation with Java PathFinder // ACM SIGSOFT Software Engineering Notes, 2004. Vol. 29. No. 4. P. 97–107.
25. *Myers G. J.* The art of software testing. — John Wiley & Sons, 1979. 192 p.
26. *Engel C., Hähnle R.* Generating unit tests from formal proofs // Tests and proofs / Eds. Yu. Gurevich, B. Meyer. — Lecture notes in computer science ser. — Springer-Verlag, 2007. Vol. 4454. P. 169–188.
27. *Gotlieb A., Botella B., Rueher M.* Automatic test data generation using constraint solving techniques // ACM SIGSOFT Software Engineering Notes, 1998. Vol. 23. No. 2. P. 53–62.
28. *Boyapati C., Khurshid S., Marinov D.* Korat: Automated testing based on Java predicates // Symposium (International) on Software Testing and Analysis Proceedings, 2002. P. 123–133.

29. *Sen K., Agha G.* CUTE and jCUTE: Concolic unit testing and explicit path model-checking tools // Conference (International) on Computer Aided Verification Proceedings, 2006. P. 419–423.
30. *Бурдонов И. Б., Косачев А. С., Пономаренко В. Н., Шнитман В. З.* Обзор подходов к верификации распределенных систем. — М.: ИСП РАН, 2006. Препринт № 16. 61 с.
31. *Utting M., Pretschner A., Legeard B.* A taxonomy of model-based testing. Waikato, New Zealand: Department of Computer Science, The University of Waikato, 2006. Technical Report. 17 p.
32. *Дробинцев П. Д.* Интегрированная технология обеспечения качества программных продуктов с помощью верификации тестирования. Дисс. . . . канд. техн. наук. — СПб.: СПбГПУ, 2006.
33. Message Sequence Chart (MSC).: ITU-T Recommendation Z.120. — ITU, 2004. http://www.itu.int/ITU-T/2005-2008/com17/languages/Z120.pdf.
34. *Колчин А. В.* Разработка инструментальных средств для проверки формальных моделей асинхронных систем. Дисс. . . . канд. физ.-мат. наук. — Киев: Институт кибернетики им. В. М. Глушкова НАН Украины, 2009. 140 с.
35. *Beizer B.* Software testing techniques. — 2nd ed. — International Thomson Computer Press, 1990. 580 p.
36. *Buhr R. J. A., Casselman R. S.* Use Case Maps for object-oriented systems. — Prentice Hall, 1995. 302 p.
37. Use Case Map Navigator. http://jucmnav.softwareengineering.ca.

Поступила в редакцию 28.01.15

CONTROL FLOW BASED TEST SUITE GENERATION

N. Voinov, P. Drobintsev, I. Nikiforov, V. Kotlyarov, and I. Selin

Saint-Petersburg State Polytechnical University, 9 Polytechnicheskaya Str., St. Petersburg 195251, Russian Federation

Abstract: The article is devoted to description of an approach to test suite generation in accordance with standard structured coverage criteria based on the control flow model. The approach is based on automatic test generation with usage of symbolic verification. The main advantage of the approach is reducing the number of generated tests obtained due to analysis of control flow data and reducing the state space for the verification system. The article contains the main ideas of the approach, the formal model of control flow, and the tools for model analysis. The results of piloting the approach in a set of projects devoted to software development are also presented.

Keywords: testing automation; formal model; coverage criterion

DOI: 10.14357/08696527150104

References

1. IEEE Standard. 1990. IEEE Standard Glossary of Software Engineering Terminology. ANSI/IEEE Standard 610.12-1990. New York, N.Y.: IEEE.
2. ITU-T. 2002. Specification and description language (SDL). Protocol Z.100. Telecommunication standardization sector of UTI. 206 p. Available at: https://www.itu.int/ITU-T/2005-2008/com17/languages/Z100.pdf (accessed March 26, 2015).
3. Unified Modeling Language (UML) Resource Page. Available at: http://www.uml.org/ (accessed March 26, 2015).
4. Nikiforov, I. V., A. V. Petrov, Yu. V. Yusupov, and V. P. Kotlyarov. 2011. Primenenie razlichnykh metodik formalizatsii dlya postroeniya verifikatsionnykh modeley sistem po UCM-spetsifikatsiyam [Application of miscellaneous formalization methods for creating verification system models based on UCM-specifications]. *Nauchno-Tehnicheskie Vedomosti SPbSPU* [St. Petersburg State Polytechnical University J.] 3(126):180–184.
5. Kulyamin, V. V. 2008. *Metody verifikatsii programmnogo obespecheniya* [Methods of software verification]. 117 p. Available at: http://www.ict.edu.ru/ft/005645/62322e1-st09.pdf (accessed March 25, 2015).
6. Broy, M., B. Jonsson, J.-P. Katoen, M. Leucke, and A. Pretschner. 2005. Model-based testing of reactive systems. Lecture notes in combustion science ser. Springer. Vol. 3472. 659 p.
7. Utting, M., and B. Legeard. 2007. *Practical model-based testing: A tools approach*. Morgan-Kaufmann. 456 p.
8. Ambert, F., F. Bouquet, S. Chemin, S. Guenaud, B. Legeard, F. Peureux, N. Vacelet, and M. Utting. 2002. BZ-TT: A tool-set for test generation from Z and B using constraint logic programming. *Formal Approaches to Testing of Software, FATES 2002 (Workshop of CONCUR'02) Proceedings.* 105–119.
9. Abrial, J.-R. 1996. *The B-Book: Assigning programs to meanings*. Cambridge University Press. 813 p.
10. Farchi, E., A. Hartman, and S. S. Pinter. 2002. Using a model-based test generator to test for standard conformance. *IBM Syst. J.* 41(1):89–110.
11. GOTCHA-TCBeans: Model driven engineering technology. IBM Haifa Research Lab. Available at: www.haifa.ibm.com/projects/verification/gtcb/index.html (accessed March 26, 2015).
12. Rational TAU: Standards-based, model-driven development solution for complex systems. Available at: http://www-03.ibm.com/software/products/en/ratitau (accessed March 26, 2015).
13. Brinksma, E. 1988. A theory for the derivation of tests. *8th Conference (International) on Protocol Specification, Testing and Verification Proceedings*. North Holland. 63–74.
14. Tretmans, J. 1992. A formal approach to conformance testing. Enschede, The Netherlands: University of Twente. PhD Thesis. 282 p.
15. Tretmans, J., and A. Belinfante. 1999. Automatic testing with formal methods. *7th European Conference on Software Testing, Analysis and Review Proceedings.* Barcelona, Spain. 8–10.

16. Fernandez, J.-C., C. Jard, T. Jéron, and C. Viho. 1996. Using on-the-fly verification techniques for the generation of test suites. *Computer aided verification*. Eds. R. Alur and T. A. Henzinger. Lecture notes in computer science ser. Springer. 1102:348–359.
17. UPPAAL. 2012. Design verification for embedded systems. Uppsala, Sweden: UP4ALL International AB. Available at: http://www.uppaal.com/ (accessed March 26, 2015).
18. Kulyamin, V. V., A. K. Petrenko, A. S. Kosachev, and I. B. Burdonov. 2003. Podkhod UniTesK k razrabotke testov. [UniTesk approach to software tests development]. *Programmirovanie* [Software Development] 29(6):25–43.
19. UniTESK: Industrial technology of reliable testing. ISP RAS. Available at: http://www.unitesk.com/ (accessed March 26, 2015).
20. Model-based testing with Spec Explorer. Microsoft Research. Available at: http://research.microsoft.com/specexplorer (accessed March 26, 2015).
21. Spec Explorer. MSDN Library. Available at: http://msdn.microsoft.com/en-us/library/ee620411.aspx (accessed March 26, 2015).
22. Ammann, P., and P. E. Black. 1999. Abstracting formal specifications to generate software tests via model checking. *18th Digital Avionics Systems Conference Proceedings*. IEEE. 2:10A.6-1–10.A.6-10.
23. Gargantini, A., and C. Heitmeyer. 1999. Using model checking to generate tests from requirements specifications. *ACM SIGSOFT Software Engineering Notes* 24(6):146–162.
24. Visser, W., C. S. Pasareanu, and S. Khurshid. 2004. Test input generation with Java PathFinder. *ACM SIGSOFT Software Engineering Notes* 29(4):97–107.
25. Myers, G. J. 1979. *The art of software testing*. John Wiley & Sons. 192 p.
26. Engel, C., and R. Hahnle. 2007. Generating unit tests from formal proofs. *Tests and proofs*. Eds. Yu. Gurevich and B. Meyer. Lecture notes in computer science ser. Springer-Verlag. 4454:169–188.
27. Gotlieb, A., B. Botella, and M. Rueher. 1998. Automatic test data generation using constraint solving techniques. *ACM SIGSOFT Software Engineering Notes* 23(2):53–62.
28. Boyapati, C., S. Khurshid, and D. Marinov. 2002. Korat: Automated testing based on Java predicates. *Symposium (International) on Software Testing and Analysis Proceedings*. 123–133.
29. Sen, K., and G. Agha. 2006. CUTE and jCUTE: Concolic unit testing and explicit path model-checking tools. *Conference (International) on Computer Aided Verification Proceedings*. 419–423.
30. Burdonov, I. B., A. S. Kosachev, V. N. Ponomarenko, and V. Z. Shnitman. 2006. Obzor podkhodov k verifikatsii raspredelennykh sistem [Review of approaches to distributed software systems verification]. ISP RAS. Preprint No. 16. 61 p.
31. Utting M., A. Pretschner, and B. Legeard. 2006. A taxonomy of model-based testing. Waikato, New Zealand: Department of Computer Science, The University of Waikato. Technical Report. 17 p.
32. Drobintsev, P. D. 2006. Integrirovannaya tekhnologiya obespecheniya kachestva programmnykh produktov s pomoshch'yu verifikatsii testirovaniya [Integrated technology for support of software systems quality based on verification and testing]. St. Petersburg: St. Petersburg State Polytechnical University. PhD Thesis.

33. ITU. 2004. Message Sequence Chart (MSC). Recommendation Z.120. http://www.itu.int/ITU-T/2005-2008/com17/languages/Z120.pdf (accessed March 26, 2015).
34. Kolchin, A. V. 2009. Razrabotka instrumentalnykh sredstv dlya proverki formalnykh modeley asinkhronnykh sistem [Development of software tools for checking of formal models of asynchronius systems]. Kiev: V. M. Glushko Institute of Cybernetics NAS. PhD Thesis. 140 p.
35. Beizer, B. 1990. *Software testing techniques*. 2nd ed. International Thomson Computer Press. 580 p.
36. Buhr, R. J. A., and R. S. Casselman. 1995. *Use Case Maps for object-oriented systems*. Prentice Hall. 302 p.
37. Use Case Map Navigator. Available at: http://jucmnav.softwareengineering.ca (accessed March 26, 2015).

Received January 28, 2015

Contributors

Voinov Nikita V. (b. 1984) — Candidate of Science (PhD) in technology, associate professor, Saint-Petersburg State Polytechnical University, 9 Polytechnicheskaya Str., St. Petersburg 195251, Russian Federation; voinov@ics2.ecd.spbstu.ru

Drobintsev Pavel D. (b. 1980) — Candidate of Science (PhD) in technology, associate professor, Saint-Petersburg State Polytechnical University, 9 Polytechnicheskaya Str., St. Petersburg 195251, Russian Federation; drob@ics2.ecd.spbstu.ru

Nikiforov Igor V. (b. 1988) — Candidate of Science (PhD) in technology, assistant, Saint-Petersburg State Polytechnical University, 9 Polytechnicheskaya Str., St. Petersburg 195251, Russian Federation; igor.nikiforovv@gmail.com

Kotlyarov Vsevolod P. (b. 1944) — Candidate of Science (PhD) in technology, professor, Saint-Petersburg State Polytechnical University, 9 Polytechnicheskaya Str., St. Petersburg 195251, Russian Federation; vpk@spbstu.ru

Selin Ivan A. (b. 1988) — assistant, Saint-Petersburg State Polytechnical University, 9 Polytechnicheskaya Str., St. Petersburg 195251, Russian Federation; ivanselin93@gmail.com

АНАЛИЗ ПОКРЫТИЯ UCM-МОДЕЛИ ТЕСТОВЫМИ СЦЕНАРИЯМИ*

***Н. В. Воинов*[1], *П. Д. Дробинцев*[2], *В. П. Котляров*[3], *И. В. Никифоров*[4]**

Аннотация: Рассмотрены подходы к анализу покрытия UCM (Use Case Maps) моделей тестовыми сценариями, сгенерированными по интегральным критериям покрытия. Дан обзор существующих критериев автоматической генерации тестовых сценариев по высокоуровневым UCM-спецификациям. Предложены два подхода анализа покрытия UCM-модели: автоматический, который предоставляет информацию о покрытых и непокрытых элементах, ветвях и путях в сводной форме, и визуальный, который позволяет пользователю наглядно удостовериться в покрытии UCM-модели. Описанные подходы и концепции реализованы в инструменте анализа, который значительно сокращает время формирования покрывающего множества тестов. Дано направление будущих работ по анализу покрытия сигналов UCM-модели.

Ключевые слова: критерии генерации; тестовые сценарии; UCM; спецификации; анализ

DOI: 10.14357/08696527150105

1 Введение

В последнее время изменяется традиционная методология разработки программного обеспечения (ПО), в которую включаются этапы детального анализа исходных требований и высокоуровневого дизайна на ранних стадиях разработки. Такое направление развития связано с повышением требований к качеству ПО при условии удовлетворения исходным спецификациям. Для сокращения времени проверки корректности требований применяют различные технологии автоматизации верификации и тестирования, которые основаны на формальном представлении исходных спецификаций на языках, использующихся как входные для доказательных инструментальных средств.

*Статья рекомендована к публикации в журнале Программным комитетом конференции «Tools & Methods of Program Analysis» («Инструменты и методы анализа программ», TMPA-2014), г. Кострома, 14–15 ноября 2014 г.

[1]Санкт-Петербургский государственный политехнический университет, Voinov@ics2.ecd.spbstu.ru

[2]Санкт-Петербургский государственный политехнический университет, drob@ics2.ecd.spbstu.ru

[3]Санкт-Петербургский государственный политехнический университет, vpk@ics2.ecd.spbstu.ru

[4]Санкт-Петербургский государственный политехнический университет, igor.nikiforovv@gmail.com

Одним из наиболее перспективных языков описания требований является язык UCM [1]. Его отличительные особенности — простота графической нотации и ориентированность на создание поведенческих сценариев систем. Язык может быть использован для описания требований как сложных программных продуктов, так и их модулей. При этом язык UCM используется на ранней стадии дизайна перед использованием таких языков, как SDL (System Description Language), MSC (Message Sequence Charts), UML (Unified Modeling Language). На сегодняшний день одной из наиболее мощных технологий автоматизации верификации и тестирования, которые используют язык UCM как исходный язык описания спецификаций на систему, является технология VRS/TAT (Verification Requirements Specification/Test Automation Toolset) [2].

Отличительной особенностью технологии VRS/TAT является возможность автоматизировать верификацию формальной модели и по проверенной модели автоматически сгенерировать множество тестовых сценариев. Генерация тестовых сценариев осуществляется по определенному критерию покрытия формальной модели. С этим связана проблема оценки и анализа покрытия исходной формальной модели тестовыми сценариями.

Целью настоящей работы является разработка в рамках технологии VRS/TAT автоматических подходов анализа покрытия формальной UCM-модели тестовыми сценариями для сокращения времени формирования конечного множества покрывающих тестов.

2 Анализ технологий повышения качества программного продукта на основе UCM-спецификаций

В настоящее время существуют технологии проектирования формальных систем, использующих стандартизованный язык UCM [1]. К таким технологиям относятся технологии SPEC-VALUe [3], TR_TROOP [4], USHLTD [5], UCM CPN (Colored Petri Nets) [6] и VRS/TAT [2].

Сравнение технологий проектирования UCM-спецификаций проведено по следующим критериям:

(1) соответствие исходного языка спецификаций, т. е. языка спецификаций, используемого в технологии для перехода от текстового неформального описания требований к высокоуровневому дизайну системы, задачам тестирования;

(2) возможность использования целевого языка формальной модели, т. е. формального языка, для применения методов доказательства корректности и средств генерации исполнимого кода и тестовых сценариев;

(3) применение в сравниваемых технологиях итеративного процесса разработки. Следует отметить, что процесс разработки может быть линейным, т. е. содержать строгую последовательность действий, без внесения изменений в исходную модель до момента тестирования сгенерированного по модели

кода. А может быть итеративным, характеризуемым возможностью корректировки на этапе создания формальной модели, когда анализ, выявление ошибок и исправление исходных спецификаций приводят к повторению процесса доказательства корректности и перегенерации модели;

(4) уровень автоматизации работ при проектировании. Технология может быть ручной, полуавтоматической либо автоматизированной и избавленной от ручной работы по анализу и исправлению найденных ошибок;

(5) возможность верификации, характеризуемая наличием поддержки верификации моделей в рассматриваемой технологии;

(6) возможность генерации исполнимого кода: поддерживает ли технология генерацию исполнимого кода системы на основе формального описания модели;

(7) наличие в технологии опции генерации тестовых сценариев, выражающейся в возможности поддержки технологией генерации тестовых сценариев, используемых для тестирования как модели, так и конечной системы;

(8) возможность автоматического исполнения тестовых сценариев;

(9) степень поддержки технологией автоматического анализа результатов верификации и тестовой генерации, т. е. имеет ли технология автоматизированные средства анализа результатов тестирования и верификации и обратного отображения результатов анализа на исходную модель;

(10) наличие в технологии автоматизированного механизма отладки процесса генерации тестовых сценариев.

Сведения о рассмотренных в разделе технологиях по предложенным критериям представлены в таблице.

Сравнение по первому критерию показало, что на ранних этапах разработки важным является использование высокоуровневой нотации для описания модели спецификаций. Все рассмотренные технологии используют UCM как исходный формальный язык спецификаций.

Анализ по второму критерию, проводимый с целью выяснения, какой целевой язык используется в технологии, определяет возможности дальнейшей работы со спецификациями. Язык LOTOS не поддерживает возможность генерации кода системы, но позволяет проводить верификацию исходных спецификаций. Язык SDL позволяет проводить как верификацию, так и генерацию исполнимого кода, но его использование затрудняется сложностью понимания сгенерированной модели системы, а также сложностью отслеживания и отображения сценариев поведения системы на исходные модели. Язык базовых протоколов [7], который представляет собой тройку Хоара, состоящую из предусловия, действия, выполняемого системой, и постусловия, в которое переходит система, позволяет проводить как верификацию модели, так и генерацию тестовых сценариев в формате MSC [8], которые могут быть использованы не только для тестирования, но

Анализ технологий работы с UCM-спецификациями

Технология	Критерии сравнения и их реализация в технологиях									
	Исходный язык	Целевой язык	Итеративный процесс	Автоматизированный подход	Верификация	Генерация исполнимого кода	Генерация тестовых сценариев	Исполнение тестовых сценариев	Анализ результатов	Механизм отладки
	1	2	3	4	5	6	7	8	9	10
CPEC-VALUe	UCM	LOTOS	+	∓	+	−	+	+	−	∓
RT-TROOP	UCM	MSC/SDL	+	∓	+	+	−	−	−	∓
UCM/SDL HLTD	UCM	SDL	+	∓	+	+	+	−	∓	∓
VRS/TAT	UCM	Базовые протоколы	+	∓	+	+	+	+	∓	∓
UCM CPN	UCM	CPN	+	±	+	−	−	−	∓	−

Условные обозначения: «−» — не поддерживается; «∓» — поддерживается частично; «+» — полностью поддерживается.

и для генерации исполнимого кода [9]. Существующие инструментальные реализации раскрашенных сетей Петри (CPN) позволяют проводить анализ системы, но не позволяют генерировать исполнимый код.

Сравнение технологий по третьему критерию показало, что все процессы являются итеративными. Все технологии позволяют после анализа модели выявить недочеты, внести в нее изменения и за несколько итераций добиться необходимой степени корректности.

Анализ технологий по четвертому критерию показывает, что все подходы в технологиях являются полуавтоматическими, так как требуют либо ручного анализа результатов верификации (технологии SPEC-VALUe, UCM CPN и UCM/SDL HLTD), либо полной автоматизации преобразования исходного языка в целевой (технологии VRS/TAT).

Данные сравнительной таблицы по пятому критерию позволяют сделать вывод, что все технологии поддерживают возможность верификации моделей.

Из данных таблицы по шестому критерию следует, что не все перечисленные технологии позволяют генерировать исполнимый код системы. Использование SDL и системы Telelogic позволяет осуществлять генерацию исполнимого кода системы, использование языка базовых протоколов совместно с преобразователем в MSC также позволяет перейти к SDL-модели. Однако технологии SPEC-VALUe и UCM CPN не поддерживают такой возможности, так как они направлены на верификацию и генерацию тестовых сценариев, а не на исполнимый код системы.

Седьмому критерию удовлетворяют только три технологии — SPEC-VALUe, VSR/TAT и UCM/SDL HLTD. Они поддерживают возможность генерации тестовых сценариев. Остальные рассматриваемые технологии направлены только на создание верифицируемых моделей систем.

Еще более сложная ситуация возникает при сравнении технологий по восьмому критерию. Этому критерию отвечают только две технологии из рассмотренных — SPEC-VALUe и VRS/TAT. На их основе можно производить автоматическое исполнение тестов. Остальные технологии существенно проигрывают по этому критерию.

Сравнение технологий по девятому критерию показало, что только три из них: UCM/SDL HLTD, UCM CPN и VRS/TAT — позволяют осуществить анализ результатов генерации, но при этом данный анализ не полностью автоматизирован и требует ручной работы. Остальные технологии (UCM/SDL HLTD и UCM CPN) не поддерживают концепцию анализа результатов на уровне исходной модели вообще либо не являются даже частично автоматизированными.

Анализ принятых в технологиях подходов по десятому критерию указывает на то, что все они, кроме UCM CPN, поддерживают механизм отладки процесса генерации, но во всех случаях он делается вручную и не имеет под собой инструментальной поддержки.

Проведенный сравнительный анализ пяти технологий работы с UCM-спецификациями четко указывает на то, что технология VRS/TAT является наиболее мощной из всех рассмотренных, так как позволяет не только проводить верификацию модели, но и автоматизировать генерацию и исполнение тестовых сценариев.

3 Критерии генерации тестовых сценариев

Выделим основные критерии генерации тестовых сценариев, которые используются в технологии VRS/TAT.

3.1 Критерий покрытия UCM-элементов

Согласно данному критерию покрытие должно происходить по всем элементам UCM-модели [1]. Регистрируется наличие каждого элемента UCM-модели в тестовом сценарии хотя бы один раз. Этот критерий позволяет проверить все состояния программного продукта, но он является наиболее слабым критерием, так как не гарантирует проверки требований.

3.2 Критерий покрытия ветвей UCM-модели

Этот критерий позволят сгенерировать множество тестовых сценариев, которое в совокупности покрывает все ветви UCM-модели. Под ветвью UCM-модели понимается линейный участок поведения, на границах которого присутствуют

элементы начала или конца сценария (StartPoint, EndPoint), элементы альтернативного поведения (OrFork, OrJoin, FailurePoint), элементы многопоточного поведения (AndFork, AndJoin) и между границами присутствуют элементы, выражающие действия системы (Responsibility).

Генерация тестовых сценариев, удовлетворяющих этому критерию, может быть осуществлена автоматически по исходной UCM-модели без дополнительной информации от пользователя.

3.3 Критерий покрытия путей UCM-модели

Это критерий прохождения всех возможных вариантов поведения модели. Критерий путей в общем случае не может быть выполнен из-за наличия бесконечного числа вариантов поведения системы [10]. В технологии VRS/TAT этот критерий используется с ограничением по числу итераций цикла, ограничениями на интерливинг, а также выбором только интересующих поведений пользователя. Таким образом, для выполнения данного критерия от пользователя требуется формирование дополнительной информации о путях, которые должны присутствовать в тестовых сценариях.

3.4 Критерий покрытия по GDL-формулам

Это пользовательский критерий покрытия, задаваемый на языке GDL (Guides Definition Language). GDL — язык пользовательских функций, созданный для направленной генерации тестовых сценариев по дереву поведения системы [11].

Рассмотрим язык GDL подробнее. Он построен на основе определения пользователем формул четырех различных типов: «M», «R», «S» и «T».

Формулы «M» позволяют определить UCM-подмодель путем указания множества элементов UCM-диаграммы. Таким образом, формула M описывает те элементы, которые будут использованы при обходе UCM-диаграммы.

Формулы альтернативного выбора («S») позволяют задать подмножество элементов указанием одного элемента.

Формулы пользовательских последовательностей («R») определяют последовательности UCM-элементов, которые должны быть представлены в генерируемом сценарии.

Формулы «T» определяют совокупность правил, описанных в формулах типа «M», «S» и «R», которые будет использованы при генерации конкретного тестового набора.

GDL-формулы позволяют в краткой форме описать набор интересующих пользователя поведений системы и произвести генерацию тестовых сценариев, удовлетворяющих данным поведениям.

Процесс генерации тестовых сценариев осуществляется автоматически, независимо от выбранного критерия генерации, но при этом остается необходимость провести анализ получившегося тестового набора для выявления непокрытых частей UCM-модели.

4 Анализ покрытия UCM-модели тестовыми сценариями

В работе предлагается два способа проведения анализа покрытия UCM-модели тестовыми сценариями. Первый способ заключается в автоматическом обходе тестовых сценариев, сборе информации о покрытых UCM-элементах, UCM-ветвях и UCM-путях исходной формальной модели и представлении непокрытых частей модели в сводной форме. Второй подход заключается в визуальном отображении тестовых сценариев на UCM-модели для того, чтобы пользователь мог наглядно удостовериться в покрытии UCM-модели.

4.1 Автоматический анализ

Для проведения автоматического анализа покрытия UCM-модели был разработан алгоритм (рис. 1), входом которого является множество тестовых сценариев и исходная формальная UCM-модель.

1. Первым этапом алгоритма является обход UCM-модели и формирование общего множества UCM-элементов и UCM-ветвей. Назовем эти множества U и B соответственно.

Рис. 1 Алгоритм формирования сводной таблицы о непокрытых элементах и ветвях UCM-модели

2. Вторым этапом алгоритма является обход всех тестовых сценариев и формирование множества UCM-элементов и UCM-ветвей, содержащихся в тестовых сценариях. Назовем эти множества $T_{\text{эл}}$ и $T_{\text{вт}}$ соответственно.

   Эти два множества характеризуют покрытие UCM-модели по критерию элементов и ветвей.

   Формально эти множества можно записать как

$$T_{\text{эл}} = \sum_{i=0}^{N} \sum_{j=0}^{M} (E_{ij} \in U); \quad T_{\text{вт}} = \sum_{i=1}^{N} \sum_{j=0}^{M} (B_{ij} \in B),$$

   где N — число тестовых сценариев; M — количество UCM-элементов тестового сценария; i и j — порядковые номера тестового сценария и UCM-элемента соответственно; U — множество всех UCM-элементов проекта; B — множество всех UCM-ветвей проекта, сформированных на этапе 1.

3. Третий шаг алгоритма заключается в поиске разницы между множеством всех UCM-элементов проекта (U) и множеством покрытых UCM-элементов ($T_{\text{эл}}$) тестовых сценариев, а также между множеством всех UCM-ветвей (B) и множеством всех покрытых тестовыми сценариями UCM-ветвей ($T_{\text{вт}}$). Такая разница множеств и показывает непокрытые элементы проекта. Назовем эти два множества $S_{\text{эл}}$ и $S_{\text{вт}}$ соответственно.

   Формально эти разницы можно записать следующим образом:

$$\text{S}_{\text{эл}} = U - T_{\text{эл}}; \quad \text{S}_{\text{вт}} = B - T_{\text{вт}}.$$

4. Последний, четвертый, этап алгоритма заключается в формировании сводной таблицы по покрытым и непокрытым элементам и ветвям модели.

Представленный формальный алгоритм вычисления непокрытых элементов и ветвей позволяет автоматизировать процесс анализа и формирования сводной информации о покрытии. Предложенный алгоритм справедлив для анализа покрытия по критериям UCM-элементов и UCM-ветвей.

Рассмотрим правила анализа покрытия по критерию путей и критерию GDL-формул. Как уже говорилось, эти два критерия в общем случае недостижимы, поэтому от пользователя требуется дополнительная информация о задании интересующих его поведений. Данные поведения задаются в виде последовательности ключевых UCM-элементов, наличие которых в тестовом сценарии обеспечивает покрытие данной последовательности. Такие последовательности называются гидами [12].

Правило. Если все заданные пользователем последовательности обеспечены тестовыми сценариями, то считается, что выбранный критерий генерации удовлетворен, иначе — не удовлетворен.

Формально это правило можно записать в следующем виде:

$$\forall\, G_i \in G\ \exists\, T_j \in T \Rightarrow K\,,$$

где G — множество гидов; G_i — i-й гид; T — множество тестовых сценариев; T_j — j-й тестовый сценарий; K — выбранный критерий генерации.

Автоматизация анализа по рассматриваемым критериям возможна за счет использования алгоритма поиска вхождения последовательности ключевых UCM-элементов гида в последовательность UCM-элементов трассы.

Отношение вхождения одной последовательности событий в другую было дано в работе [13] и задается выражением:

$$U_m \subset U_n \Leftrightarrow (m \le n) \wedge (\forall\, i:\ (1 \le i \le m) \\ \exists\, j_i:\ ((j_0 = 0) \wedge (j_i > j_{i-1})) \wedge (U_m^i = U_n^{j_i}))\,,$$

где U_m — события критериальной цепочки; U_n — события трассы.

Технология VRS/TAT поддерживает четыре критерия генерации тестовых сценариев и позволяет автоматически проверить их выполнение на конечном тестовом наборе.

4.2 Визуальный анализ

Несмотря на наличие автоматизированного подхода вычисления непокрытых элементов и ветвей UCM-модели, необходимо предоставить пользователю возможность визуального контроля покрытия непосредственно на UCM-диаграммах. Каждому тестовому сценарию можно задать уникальный цвет, которым он будет подсвечен на диаграммах исходной модели. Таким образом можно подсветить на UCM-диаграмме весь тестовый набор для оценки его покрытия.

На рис. 2 представлена UCM-диаграмма с подсвеченным сценарием (он изображен пунктиром), который проходит по двум уровням проекта, организованным через структурный элемент Stub [1].

Возможность визуального изображения сценариев на разных уровнях абстракций системы делает тестовый набор наглядным, что актуально для проектов любой сложности, а также способствует ручному анализу и пониманию UCM-модели.

5 Проверка покрытия требований тестовыми сценариями

В разд. 3 и 4 рассмотрены критерии генерации тестовых сценариев и способы автоматического и визуального анализа покрытия. Но тестирование в первую очередь направлено на проверку удовлетворения разрабатываемой системы исходным спецификациям. Поэтому важной особенностью разработки тестов является возможность создания тестовых сценариев, обеспечивающих стопроцентное

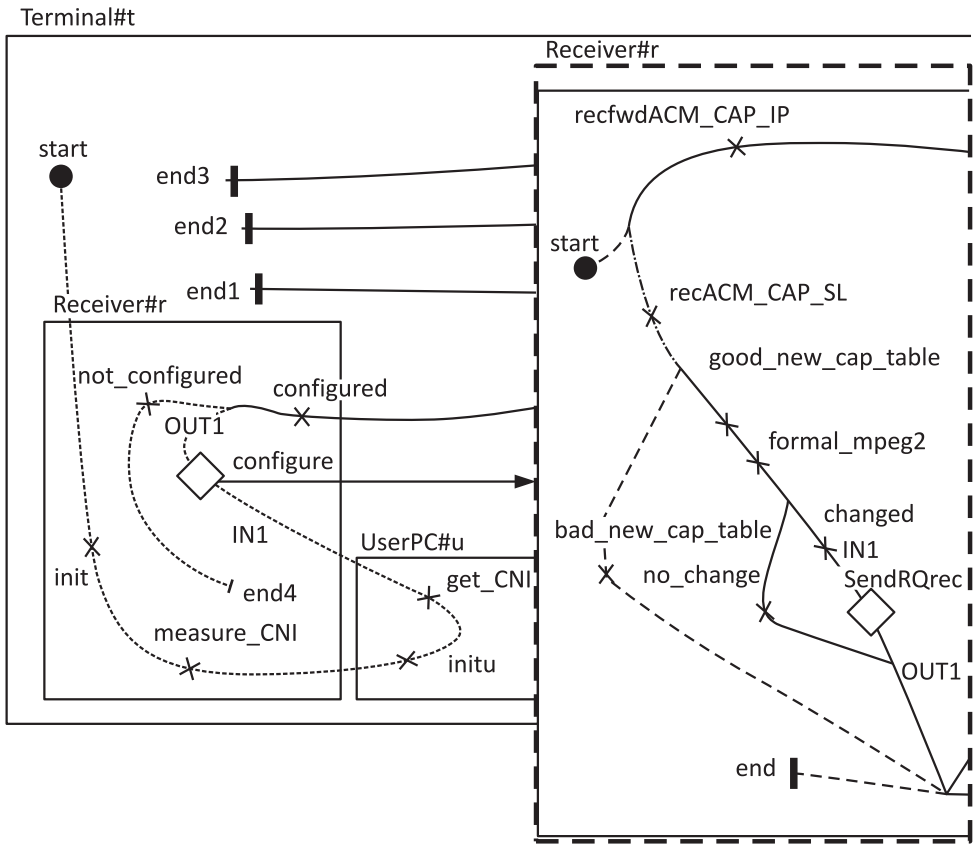


Рис. 2 Визуализация тестового сценария на UCM-диаграмме

покрытие требований. Для этого используется матрица отслеживания [14], представленная на рис. 3.

Матрица отслеживания содержит в себе как минимум следующую информацию:

- идентификатор требования (столбец 1);
- неформализованное описание требования (столбец 2);
- сценарий проверки требования (столбец 3);
- идентификатор сценария проверки требования (столбец 4);
- критериальную цепочку [15] событий (столбец 5).

По неформальному описанию требований выделяются сценарии проверки, согласуемые с заказчиком. Сценарии проверки представляют собой последова-

1	2	3	4	5
Идентификатор требования	Исходный текст требования	Сценарий проверки требования	Идентификатор сценария	Критериальные цепочки
TRS_18081-2210 [7]	Когда установлена опция «Не беспокоить», пользовательский статус отправляется на сервер с сообщением «Не беспокоить»	scen#064: 1. Установить опцию «Do not Disturb» в меню KPTT_MyStatus и нажать LSK “Select” или “Center Select” 2. Проверить, что KPTT_Confirmation_Notice отобразилось в меню	scen#064	: TRS_18081_2209 KPTT_MyStatus06 KPTT_Confirmation_N
TRS_18081-2209 [6]	Если установлена опция «Не беспокоить», то у пользователя не должно быть возможности организовать PTT	scen#128: 1. Проверить состояние в меню DnD 2. Перейти на KPTT_Contacts в меню, выбрать контакт 3. Нажать на кнопку PTT	scen#128	: TRS_18081_2212 KPTT_Contacts39

Рис. 3 Пример матрицы отслеживания

тельности действий пользователя и откликов системы на входные воздействия, которые необходимо осуществить и наблюдать для доказательства корректности требования.

Затем по сценариям проверки осуществляется разработка критериальных цепочек, которые являются последовательностями основных событий, наблюдаемых в поведении системы при проверке некоторого конкретного требования. Критериальные цепочки не только формализуют требование и сценарий его проверки в последовательности событий, но и являются критерием его проверки.

После того как в матрице отслеживания появились критериальные цепочки в виде последовательности UCM-элементов, можно переходить к процессу генерации тестовых сценариев по частичному критерию путей. Такой переход становится возможным, так как критериальные цепочки задают множество поведений, которые пользователь должен проверить в системе.

Таким образом, можно наблюдать связь между частичным критерием путей и матрицей отслеживания, что в свою очередь обеспечивает стопроцентную проверку требований и позволяет автоматически провести генерацию покрывающего тестового набора.

6 Инструментальная реализация анализа покрытия

Для решения задач анализа покрытия реализован набор программных средств, интегрированных в инструмент автоматического анализа UCM GA (Guides Analyzer). Инструмент внедрен в автоматическую цепочку тестирования VRS/TAT с целью получения набора тестовых сценариев, покрывающих функциональные требования на проектируемую систему.

Анализатор UCM GA реализует подходы проверки покрытия поведенческой UCM-модели тестовыми сценариями. Он позволяет не только автоматически, но и визуально оценить покрытие поведенческой модели на уровне UCM-диаграмм. Использование анализатора UCM GA позволяет сократить время формирования покрывающего множества тестовых сценариев в 3 раза [16].

7 Направление будущих работ

При тестировании телекоммуникационных систем основополагающим способом обмена информацией являются транзакции и сложные сигналы. На сегодняшний день в анализаторе покрытия UCM GA отсутствует возможность проверить наличие всех сложных сигналов разрабатываемой системы и комбинаций их параметров в тестовом наборе, поэтому в направлении будущих работ ставится задача создать метод анализа покрытия сложных сигналов UCM-модели, реализовать его в инструменте автоматического анализа и визуализировать точки обмена сообщениями на уровне UCM-модели.

Литература

1. User requirements notation (URN): ITU-T Recommendation Z.151, 10/2012. http://www.itu.int/rec/T-REC-Z.151-201210-I/en.
2. *Веселов А. О., Иванов А. С., Тютин Б. В., Котляров В. П.* Автоматизация тестирования телекоммуникационных приложений // Научно-технические ведомости СПбГПУ, 2009. № 3. С. 208–212.
3. *Amyot D.* Specification and validation of telecommunication system with use Case Maps and LOTOS. — Ottawa, Canada: University of Ottawa, 2001. PhD Thesis. 128 p.
4. *Bordeleau F.* A systematic and traceable progression from scenario models to communicating hierarchical state machines. — Ottawa, Canada: Carleton University, 2000. PhD Thesis. 389 p.
5. *Sales I.* A bridging methodology for Internet protocols standards development. — Ottawa, Canada: School of Information Technology and Engineering, 2001. PhD Thesis. 119 p.
6. *Визовитин Н. В., Непомнящий В. А.* Алгоритмы трансляции UCM-спецификаций в раскрашенные сети Петри. — Новосибирск: Институт систем информатики им. А. П. Ершова, 2012. Препринт № 168. 56 с.
7. *Летичевский А. А.* Верификация и тестирование интерактивных систем, специфицированных базовыми протоколами. — Киев: Ин-т кибернетики им. В. М. Глушкова НАН Украины, 2005. Дисс. . . . канд. физ.-мат. наук. 138 с.
8. Message Sequence Chart (MSC). ITU Recommendation Z.120, 11/1999. http://www.itu.int/ITU-T/studygroups/com10/languages/Z.120_1199.pdf.
9. *Letichevsky A., Kapitonova J., Letichevsky, Jr., A., Volkov V., Baranov S., Weigert T.* Basic protocols, message sequence charts, and the verification of requirements specifications // Comput. Netw., 2005. Vol. 49. No. 5. P. 661–675.
10. *Beizer B.* Software testing techniques. — 2nd ed. — New York, NY, USA: Van Nostrand Reinhold Co., 1990. 550 p.

11. *Дробинцев П. Д., Никифоров И. В., Котляров В. П.* Методика проектирования тестов сложных программных комплексов на основе структурированных UCM-моделей // Научно-технические ведомости СПбГПУ, 2013. № 3(174). С. 99–105.
12. *Колчин А. В.* Метод направления поиска и генерации тестовых сценариев при верификации формальных моделей асинхронных систем // Проблемы программирования, 2008. Вып. 4. С. 5–12.
13. *Баранов С. Н., Котляров В. П.* Формальная модель требований, используемая в процессе генерации кода приложения и кода тестов // Моделирование и анализ информационных систем, 2011. Т. 18. № 4. С. 118–130.
14. *Drobintchev P., Kotlyarov V., Nikiforov I.* Technology aspects of state explosion problem resolving for industrial software design // 7th Spring/Summer Young Researchers' Colloquium on Software Engineering Proceedings. — Kazan: National Research Technical University, 2013. P. 46–51.
15. *Drobintsev P. D., Nikiforov I. V., Kotlyarov V. P.* Semantics adjustment of UCM real time constructions for implementation in translator of UCM to basic protocols // Humanities and Science University J., 2013. № 5. С. 207–215.
16. *Никифоров И. В.* Методы автоматизации построения поведенческой модели программного продукта на основе UCM-спецификаций. — СПб.: СПбГПУ, 2013. Дисс. . . . канд. техн. наук. 205 с.

Поступила в редакцию 30.01.15

ANALYSIS OF UCM-MODEL COVERAGE BY TEST SCENARIOS

N. V. Voinov, P. D. Drobintsev, I. V. Nikiforov, and V. P. Kotlyarov

Saint-Petersburg State Polytechnical University, 9 Polytechnicheskaya Str., St. Petersburg 195251, Russian Federation

Abstract: The article observes approaches to analysis of UCM-models coverage by test scenarios generated based on integral coverage criteria. Existing criteria for automatic generation of test scenarios from high-level UCM-specifications are reviewed. Two approaches to analysis of UCM-model coverage are proposed: the automatic one which provides information about covered and uncovered elements, branches, and paths in one view, and the visual one which allows the user to explicitly make sure that a UCM-model is covered by test scenarios. The described approaches are implemented in the analysis tool which significantly reduces the time needed to create a test set which covers an UCM-model. Future plans on coverage analysis improvement are also mentioned.

Keywords: test generation criteria; test scenarios; UCM; specifications; analysis

DOI: 10.14357/08696527150105

References

1. User requirements notation (URN). ITU-T Recommendation Z.151, 10/2012. Available at: http://www.itu.int/rec/T-REC-Z.151-201210-I/en (accessed April 2, 2015).

2. Veselov, A. O., A. S. Ivanov, B. V. Tyutin, and V. P. Kotlyarov. 2009. Avtomatizatsiya testirovaniya telekommunikatsionnykh prilozheniy [Test automation of telecommunication applications]. *Nauchno-Tehnicheskie Vedomosti SPbSPU* [St. Petersburg State Polytechnical University J.]. 80(3):208–212.
3. Amyot, D. 2001. Specification and validation of telecommunication system with Use Case Maps and LOTOS. Ottawa, Canada: University of Ottawa. PhD Thesis. 128 p.
4. Bordeleau, F. 1999. A systematic and traceable progression from scenario models to communicating hierarchical state machines. Ottawa, Canada: Carleton University. PhD Thesis. 389 p.
5. Sales, I. 2001. A bridging methodology for Internet protocols standards development. Ottawa, Canada: School of Information Technology and Engineering. PhD Thesis. 119 p.
6. Visovitin, N. V., and V. A. Nepomnyashchiy. 2012. *Algoritmy translyatsii UCM-spetsifikatsiy v raskrashennye seti Petri* [Algorithms of UCM-specifications translation into marked Petri nets]. Novosibirsk: A. P. Ershov Institute of Informatics Systems. 56 p.
7. Letichevskiy, A. A. Verifikatsiya i testirovanie interaktivnykh system, spetsifitsirovannykh bazovymi protokolami [Verification and testing of interactive systems specified with basic protocols]. Kiev. DSc Thesis. 138 p.
8. Message Sequence Chart (MSC). ITU Recommendation Z.120, 11/99. Available at: http://www.itu.int/ITU-T/studygroups/com10/languages/Z.120_1199.pdf (accessed April 1, 2015).
9. Letichevsky, A., J. Kapitonova, A. Letichevsky, Jr., V. Volkov, S. Baranov, and T. Weigert. 2005. Basic protocols, message sequence charts, and the verification of requirements specifications. *Comput. Netw.* 49(5):661–675.
10. Beizer, B. 1990. *Software testing techniques*. 2nd ed. New York, NY, USA: Van Nostrand Reinhold Co. 550 p.
11. Drobintsev, P. D., I. V. Nikiforov, and V. P. Kotlyarov. 2013. Metodika proektirovaniya testov slozhnykh programmnykh kompleksov na osnove strukturirovannykh UCM-modeley [Methods of creating tests for complex software based on structured UCM-models]. *Nauchno-Tehnicheskie Vedomosti SPbSPU* [St. Petersburg State Polytechnical University J.] 174(3):99–105.
12. Kolchin, A. V. 2008. Metod napravleniya poiska i generatsii testovykh stsenariev pri verifikatsii formal'nykh modeley asinkhronnykh sistem [Method of guided search and test scenarios generation for verification of formal models of asynchronous systems]. *Problemy Programmirovaniya* [Issues of Programming] 4:5–12.
13. Baranov, S. N., and V. P. Kotlyarov. 2011. Formalnaya model' trebovaniy, ispol'zuemaya v protsesse generatsii koda prilozheniya i koda testov [Formal model of requirements used during application and tests code generation]. *Modelirovanie i Analiz Informatsionnykh System* [Modeling and Analysis of Informational Systems] 18(4):118–130.
14. Drobintchev, P., V. Kotlyarov, and I. Nikiforov. 2013. Technology aspects of state explosion problem resolving for industrial software design. *7th Spring/Summer Young Researchers' Colloquium on Software Engineering Proceedings*. Kazan. 46–51.
15. Drobintsev, P. D., I. V. Nikiforov, and V. P. Kotlyarov. 2013. Semantics adjustment of UCM real time constructions for implementation in translator of UCM to basic protocols. *Humanities Science University J.* 5:207–215.

16. Nikiforov, I. V. 2013. Metody avtomatizatsii postroeniya povedencheskoy modeli programmnogo produkta na osnove UCM-spetsifikatsiy [Automation methods of software product behavioral model creation based on UCM-specifications]. St. Petersburg: SPbSPU. DSc Thesis. 205 p.

Received January 30, 2015

Contributors

Voinov Nikita V. (b. 1984) — Candidate of Science (PhD) in technology, associate professor, Saint-Petersburg State Polytechnical University, 9 Polytechnicheskaya Str., St. Petersburg 195251, Russian Federation; voinov@ics2.ecd.spbstu.ru

Drobintsev Pavel D. (b. 1980) — Candidate of Science (PhD) in technology, associate professor, Saint-Petersburg State Polytechnical University, 9 Polytechnicheskaya Str., St. Petersburg 195251, Russian Federation; drob@ics2.ecd.spbstu.ru

Nikiforov Igor V. (b. 1988) — Candidate of Science (PhD) in technology, assistant, Saint-Petersburg State Polytechnical University, 9 Polytechnicheskaya Str., St. Petersburg 195251, Russian Federation; igor.nikiforovv@gmail.com

Kotlyarov Vsevolod P. (b. 1944) — Candidate of Science (PhD) in technology, professor, Saint-Petersburg State Polytechnical University, 9 Polytechnicheskaya Str., St. Petersburg 195251, Russian Federation; vpk@spbstu.ru

ОБОБЩЕННЫЙ ТАБЛИЧНЫЙ LL-АНАЛИЗ*

С. В. Григорьев[1], *А. К. Рагозина*[2]

Аннотация: Синтаксический анализ является важным шагом анализа кода. Для работы с неоднозначными грамматиками используются обобщенные алгоритмы синтаксического анализа: Generalized LR (GLR) и Generalized LL (GLL). Для работы со встроенными языками — поддержки их в IDE (Integrated Development Environment), анализа в целях реинжиниринга или поиска уязвимостей (SQL-инъекций) — используется абстрактный синтаксический анализ, основанный на классическом табличном анализе. Ранее был предложен алгоритм обобщенного нисходяшего анализа без использования предиктивных таблиц анализа. В данной статье описан подход к созданию табличного GLL-анализатора на основе предложенного алгоритма, который в дальнейшем будет использоваться для получения абстрактного анализатора. Также в статье описан алгоритм обобщенного нисходящего анализа, модификации, которым он подвергся, и результаты сравнения с алгоритмом обобщенного восходящего анализа, который был реализован ранее.

Ключевые слова: синтаксический анализ; GLL; обобщенный анализ; RNGLR; абстрактный анализ; встроенные языки

DOI: 10.14357/08696527150106

1 Введение

Существует множество средств автоматической и автоматизированной обработки исходного кода программ. В качестве примера можно привести компиляторы, верификаторы, средства анализа исходного кода. Многие из них предполагают перевод кода из текстового представления в специальное структурное представление — абстрактное синтаксическое дерево, предназначенное для дальнейшей обработки. Для такого преобразования исходного кода в абстрактное синтаксическое дерево обычно используется синтаксический анализ, для реализации которого существует большое количество алгоритмов, различающихся классом обрабатываемых языков. Для работы с неоднозначными и недетерминированными грамматиками существуют алгоритмы обобщенного синтаксического анализа: GLR и GLL. Данные алгоритмы анализируют все возможные варианты разбора и строят все деревья. Алгоритм обобщенного нисходящего анализа (GLL) отличается своей простотой и прямолинейностью из-за тесной связи с грамма-

*Статья рекомендована к публикации в журнале Программным комитетом конференции «Tools & Methods of Program Analysis» («Инструменты и методы анализа программ», TMPA-2014), г. Кострома, 14–15 ноября 2014 г.

[1]Санкт-Петербургский государственный университет, Semen.Grigorev@jetbrains.com

[2]Санкт-Петербургский государственный университет, ragozina.anastasiya@gmail.com

тикой, в то время как алгоритм обобщенного восходящего анализа (GLR) имеет более сложную структуру.

Многие языки программирования позволяют конструировать выражения на других языках, называемых встроенными, из строковых литералов и затем передавать их на выполнение соответствующему окружению. Чаще всего такой подход используется для генерации HTML-страниц и SQL-запросов. Трудность при использовании встроенных выражений заключается в том, что ошибки в них могут быть обнаружены только во время выполнения. Это делает систему, созданную с использованием такого подхода, ненадежной и уязвимой. Еще одна проблема, которая появляется при использовании встроенных языков — SQL-инъекции, в результате которых могут быть выполнены действия, не предусмотренные создателем скрипта. Эта ситуация возникает из-за отсутствия фильтрации внешних входных данных, поступающих на сервер. Логика запроса меняется таким образом, что злоумышленник может получить данные из таблиц или даже изменить что-то в базе данных. Для обработки встроенных языков используется статический анализ динамически формируемых выражений (абстрактный синтаксический анализ) [1].

Целью данной работы является получение табличного синтаксического анализатора на основе алгоритма GLL, который станет основой для абстрактного анализатора в дальнейшем. Существует несколько подходов к созданию абстрактных синтаксических анализаторов. Так как в абстрактном анализе процесс разбора сильно зависит от структуры входных данных, которая не является линейным входным потоком, то использование явной генерации кода анализатора затруднено. В статье [1] описан абстрактный анализ на основе классического алгоритма синтаксического анализа LALR. Ранее в рамках проекта YaccConstructor [2] был реализован алгоритм обобщенного восходящего анализа RNGLR (Right Nulled GLR) [3], а позже на его основе был создан абстрактный табличный анализатор [4]. Управляющие таблицы синтаксического анализатора при переходе к абстрактному анализу не меняются: достаточно модифицировать только интерпретатор таблиц. Таким образом, чтобы получить возможность использовать GLL-алгоритм для абстрактного анализа, необходимо изменить процесс построения анализатора по грамматике, так как в оригинальном алгоритме GLL не используются таблицы синтаксического анализа, а код анализатора генерируется явно. В данной работе описан алгоритм обобщенного нисходящего анализа и модификации, которые были внесены в него для получения табличного анализатора. Также обсуждаются особенности реализации табличного GLL-анализатора.

2 Обобщенный синтаксический анализ

Синтаксические анализаторы можно разделить на два класса — нисходящие и восходящие, каждый из которых имеет как преимущества, так и недостатки. Структура нисходящих парсеров полностью соответствует структуре грамматики,

что упрощает процесс их написания и отладки. Существенным недостатком таких синтаксических анализаторов является то, что класс языков, обрабатываемых ими, весьма ограничен. Причина этих ограничений заключается в том, что любая LL(k)-грамматика должна быть однозначной, но далеко не все языки программирования однозначны. Например, леворекурсивные грамматики не принадлежат классу LL(k) ни для какого k. Иногда удается преобразовать неLL-грамматику в эквивалентную ей LL-грамматику с помощью факторизации и устранения левой рекурсии, но проблема существования эквивалентной LL(k)-грамматики для произвольной неLL(k)-грамматики неразрешима [5]. Расширить класс обрабатываемых языков позволяет использование восходящих LR-анализаторов [6]. Такие анализаторы позволяют обрабатывать более широкий класс грамматик, но имеют более сложную структуру. Восходящие анализаторы позволяют работать с леворекурсивными грамматиками, но даже они не могут бороться со скрытой левой рекурсией [7]. К сожалению, усложнение структуры парсера часто приводит к снижению скорости его работы и усложняет процесс диагностики ошибок.

Существует еще один класс синтаксических анализаторов — обобщенные анализаторы, которые используются для работы с неоднозначными грамматиками. К этому классу относятся алгоритмы обобщенного восходящего (GLR) и нисходящего (GLL) анализа [8]. Алгоритм обобщенного восходящего анализа широко известен и впервые был предложен Томитой [9]. Классический вариант алгоритма не способен обрабатывать любые контекстно-свободные грамматики. В дальнейшем было описано множество модификаций данного алгоритма, однако принцип работы всегда оставался схожим: для конкретной грамматики GLR-алгоритм обрабатывает все возможные варианты разбора входной последовательности, используя обход графа в ширину. GLR-анализаторы используют таблицы синтаксического анализа подобно LR-анализаторам. Отличие таких таблиц от классических состоит в том, что допускается несколько переходов в различные состояния, которые определяются по исходному состоянию парсера и входному терминальному символу. Это обусловлено тем, что в грамматике могут присутствовать неоднозначности и входная последовательность может иметь несколько вариантов разбора. В результате в процессе работы могут возникать конфликты вида сдвиг/свертка (shift/reduce) и свертка/свертка (reduce/reduce). В таких конфликтных ситуациях следует либо выбрать один вариант разбора, что может привести к получению некорректного результата, либо рассмотреть все варианты разбора, что и происходит в процессе работы алгоритма. При обработке каждой конфликтной ситуации необходимо создавать новый стек, верхнее состояние которого соответствует возможному переходу, однако хранение большого числа стеков требует слишком большого объема памяти. Для экономии памяти общие части стеков переиспользуются, а в местах возникновения конфликтов происходит разветвление стека. В случае когда на вершинах разных ветвей находятся одинаковые состояния, ветви могут быть объединены. Таким образом, стек организуется в виде графа, и такая структура данных называется Graph Structured Stack (GSS) [9]. Если для какой-либо вершины стека и входного сим-

вола в таблице парсера не существует ни одного перехода, то соответствующая ветка разбора считается ошибочной и отбрасывается.

Тот же принцип работы лежит и в основе алгоритма обобщенного нисходящего анализа. В процессе работы парсера рассматриваются все возможные варианты, что осуществляется за счет использования дескрипторов. Дескриптор — четверка, описывающая текущее состояние анализатора: индекс во входном потоке, метка функции разбора, текущая вершина стека и фрагмент дерева вывода, которое было построено на момент создания дескриптора. Такой алгоритм в худшем случае работает за кубическое время от длины входной цепочки, а для LL-грамматик [8] — за линейное. Синтаксические анализаторы, построенные с помощью такого алгоритма, позволяют обрабатывать грамматики как со скрытой, так и с обычной левой рекурсией, значительно расширяя класс обрабатываемых нисходящими синтаксическими анализаторами языков.

3 Подходы к генерации синтаксических анализаторов

Для автоматического создания синтаксических анализаторов существует несколько подходов. В рамках первого подхода весь код парсера генерируется по грамматике. Чаще всего такой подход используется при генерации анализаторов, построенных методом рекурсивного спуска. При генерации нисходящих анализаторов для каждого нетерминала генерируются функции, которые последовательно вызываются в процессе разбора. Несмотря на то что нисходящие анализаторы просты для написания и отладки и поэтому чаще всего создаются вручную, существуют инструменты для автоматической генерации таких анализаторов. Например, инструмент ANTLR (Another Tool for Language Recognition) [10] —

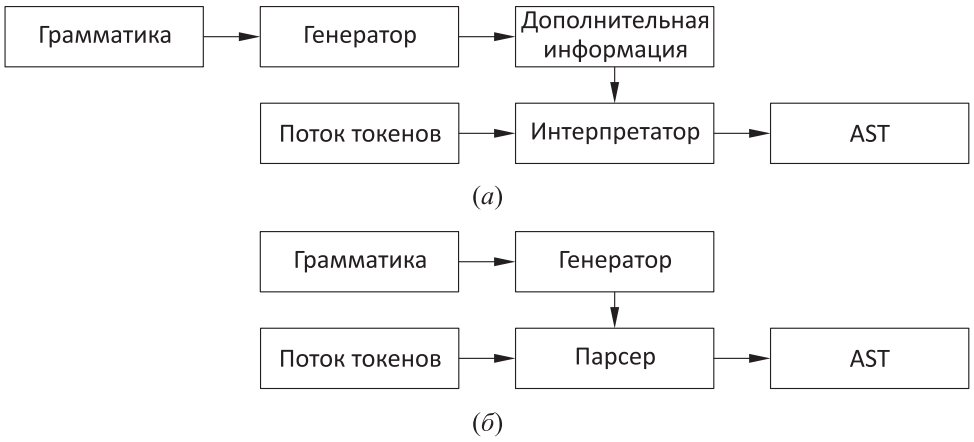


Рис. 1 Структура генератора синтаксических анализаторов: генерирующего весь анализатор (*а*) и только дополнительную информацию (*б*)

генератор парсеров, позволяющий автоматически создавать анализаторы на одном из целевых языков программирования по описанию LL(*)-грамматики на языке, близком к EBNF (Extended Backus–Naur Form). Структура генераторов такого типа изображена на рис. 1, *а*.

Существует еще один подход к генерации синтаксических анализаторов, который используется для получения табличных анализаторов. Отдельно создается интерпретатор, который содержит в себе основную логику алгоритма. Интерпретатор пишется вручную и постоянно переиспользуется. По грамматике каждый раз генерируется дополнительная информация, которая необходима интерпретатору в процессе работы. Структура такого генератора представлена на рис. 1, *б*. Чаще всего в качестве дополнительной информации генерируются таблицы синтаксического анализа, управляющие процессом разбора.

4 Основные принципы обобщенного LL-анализа

Группа ученых во главе с Elizabeth Scott и Adrian Johnstone занимается разработкой алгоритмов синтаксического анализа. Ими было предложено несколько алгоритмов на основе обобщенного восходящего анализа [3, 11, 12] и алгоритм обобщенного нисходящего анализа GLL, о котором и пойдет речь в данной статье.

Синтаксические анализаторы, реализованные с помощью метода рекурсивного спуска, представляют собой набор функций, каждая из которых содержит в себе код для обработки той или иной альтернативы. В алгоритме обобщенного нисходящего анализа для каждого нетерминала генерируется функция, содержащая в себе код для его разбора. Если какой-то нетерминал может быть разобран несколькими способами, то для него генерируется несколько функций, помеченных разными метками. Это метки первого типа, которые используются в алгоритме. Кроме того, бывают метки второго типа — возвратные. Этими метками помечаются фрагменты кода, которые должны быть вызваны после того, как нетерминал был обработан. Для того чтобы запоминать все возможные варианты разбора, используются дескрипторы, содержащие метку функции, которая должна быть вызвана, позицию во входном потоке, стек и дерево разбора. Дескрипторы хранятся в очереди, и выполнение каждый раз возобновляется с точки, которая описана в текущем дескрипторе. Новые дескрипторы добавляются в очередь в процессе работы синтаксического анализатора перед обработкой нетерминалов или после того, как правило было разобрано до конца. Если в процессе выполнения один из вариантов разбора не может быть завершен, то вместо завершения процесса работы парсера с ошибкой из очереди извлекается следующий дескриптор, и процесс продолжается. Разбор завершается, как только очередь становится пустой. Работа парсера контролируется с помощью управляющей функции, которая выполняет следующие действия:

– проверяет, что очередь дескрипторов не пуста;

– извлекает дескриптор;

– присваивает значение глобальным переменным, которые используются в процессе разбора: позиция во входном потоке, стек и дерево;

– вызывает функцию с меткой, изъятой из дескриптора.

На каждом шаге работы алгоритма хранится два указателя: один — в грамматике (слот), а второй — во входном потоке. Слоты имеют следующий вид: $X ::= x_1 \cdots x_i \bullet x_{i+1} \cdots x_q$, где $\bullet$ указывает позицию перед символом $xi + 1$. На каждом этапе разбора имеются слот вида $X ::= a \bullet Xb$ или $x ::= a\bullet$ и позиция во входном потоке i. В процессе разбора рассматриваются следующие ситуации.

1. Если текущий символ в грамматике является терминалом x и совпадает с текущим символом во входном потоке, то указатель в грамматике нужно сдвинуть на одну позицию вправо, $X ::= ax \bullet b$, и увеличить указатель во входном потоке на единицу. Никаких дополнительных действий со стеком при этом не производится. Иначе, если терминал X не совпадает с текущим символом во входном потоке, текущая ветка разбора считается ошибочной, отбрасывается и разбор продолжается с использованием следующего дескриптора.
2. Если текущий символ в грамматике является нетерминалом A, то необходимо в стек записать слот, по которому продолжить разбор после того, как правило для a будет разобрано. Указатель в грамматике перемещается на $A ::= \bullet b$, а указатель во входном потоке остается без изменений.
3. Если указатель в грамматике имеет вид $X ::= a\bullet$ и стек не пуст, то слот $Y ::= bX \bullet c$, который хранится на вершине стека, извлекается и становится текущим.
4. Если текущий слот имеет вид $S ::= c\bullet$ и весь входной поток рассмотрен, то возвращается дерево, построенное в процессе разбора, иначе разбор заканчивается ошибкой.

Процесс работы синтаксического анализатора не детерминирован, и может возникнуть ситуация, когда один и тот же дескриптор будет создаваться снова и снова. Повторное создание дескриптора приведет к тому, что процесс зациклится и никогда не завершится. Для того чтобы избежать создания одинаковых дескрипторов, отдельно поддерживается множество U, содержащее ранее созданные дескрипторы. Каждый раз, прежде чем добавить новый дескриптор в очередь дескрипторов, выполняется проверка того, не был ли он уже добавлен в множество U.

4.1 Организация стека

Как упоминалось ранее, в таблицах синтаксического анализа для алгоритма обобщенного анализа в каждой ячейке может храниться несколько альтернатив для разбора нетерминала. В таких ситуациях каждый раз создаются новые стеки, верхнее состояние которых соответствует каждому возможному переходу.

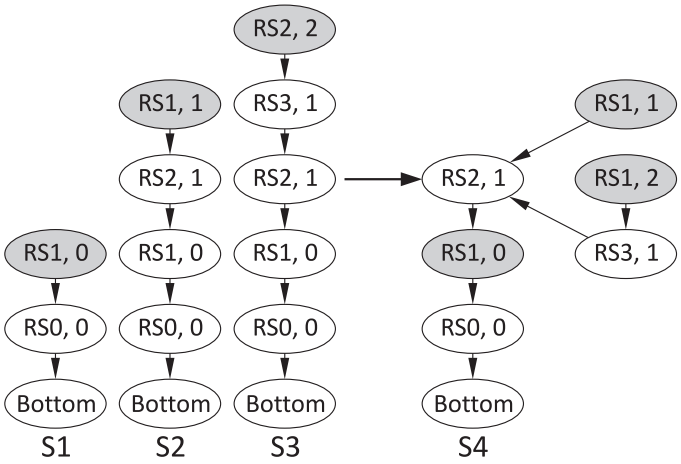


Рис. 2 Переиспользование общих элементов стеков

Проблема такого подхода состоит в том, что для леворекурсивных грамматик число стеков может экспоненциально зависеть от входного потока. Каждый раз при создании нового дескриптора будет создаваться новый стек. Необходимости полностью копировать и каждый раз хранить отдельно каждый стек нет, так как у них есть общие части. Решением проблемы является комбинирование стеков в один с использованием структуры данных GSS (Generic Security Services). С помощью этой структуры можно представить стек в виде графа. Это позволяет вместо хранения всего стека целиком хранить только отдельную вершину графа. Для работы со стеком используется две функции: create( ) для создания новых вершин и pop( ) для изъятия вершин со стека. На вершинах стека хранятся возвратные метки, по которым нужно продолжить разбор после того, как нетерминал будет разобран. На ребрах хранятся фрагменты дерева, которое было построено на момент создания новой вершины стека. Каждая вершина создается лишь однажды и никогда не удаляется в процессе работы алгоритма. Пример переиспользования общих частей стеков представлен на рис. 2: в результате объединения стеков S1, S2 и S3 будет получен стек S4.

4.2 Компактное представление леса разбора

Результатом работы классического синтаксического анализатора для однозначных грамматик является абстрактное синтаксическое дерево, корень которого помечен стартовым нетерминалом, а листья — терминалами. Это дерево используется в дальнейшем для анализа или трансляции. Как упоминалось ранее, для неоднозначных грамматик для одной и той же входной цепочки может существовать несколько деревьев разбора. Для некоторых грамматик число деревьев

может экспоненциально зависеть от размера входа. Для того чтобы уменьшить объем требуемой для хранения деревьев памяти, используется структура Shared Packed Parse Forests (SPPF) [13], которая позволяет хранить деревья более компактно. В SPPF узлы, которые имеют одинаковые поддеревья под ними, переиспользуются, а узлы, которые соответствуют разному выводу одной и той же цепочки из одного и того же нетерминала, комбинируются в упакованный узел.

В алгоритме GLL используется бинаризированная версия SPPF [14], в которой есть промежуточные узлы. Такие промежуточные узлы помечены слотами, т. е. правой частью правила грамматики с указанием позиции в нем. Таким образом, в бинаризированной версии SPPF существуют узлы трех видов: символьные узлы для терминалов или нетерминалов, промежуточные узлы, помеченные слотами, и упакованные узлы, которые позволяют представлять несколько деревьев для одной и той же цепочки. У терминальных узлов нет потомков, а потомками нетерминальных узлов являются упакованные узлы.

4.3 Пример работы алгоритма

Чтобы проиллюстрировать работу оригинального алгоритма и показать, как строятся стек и лес разбора в процессе его работы, обратимся к простому примеру. Рассмотрим неоднозначную грамматику G0 вида

```
0 : S ::= S S
1 : S ::= b
2 : START ::= S
```

В результате работы генератора для такой грамматики будет получен следующий код (более подробно о генерации кода изложено в статье [15]):

```
read the input into I and set I[m] = $
create GSS node u0 = (L0, 0)
index = 0
GSSNode = u0
cN = emptyAST
cR = emptyAST
setR = empty
setP = empty
goto L_START
L0:
 if (IsNotEmpty(setR)) {
 context = setR.Dequeue()
 index = context.Index
 GSSNode = context.Vertex
label = context.Label
 cN = context.AST
```

```
 cR = dummyAST
 } else {
 if (there is an SPPF node (START, 0, m)){
 report success
 } else {report failure}
}

L_START:
 if (test(I[index], S, b)) {
 add(LS, GSSNode, index, emptyAST)}
L_S:
 if (test(I[index], S, b)) {
 add(LS1, GSSNode, index, emptyAST)}
 if (test(I[index], S, SS)) {
 add(LS2, GSSNode, index, emptyAST)}
 goto L0
LS1 :
 cN = getNodeT(b, index)
 index = index + 1
 pop(GSSNode, index, cN)
 goto L0
LS2 :
 GSSNode = create(RS1, GSSNode, index, cN)
 goto LS
RS1 :
 if (test(I[currentIndex], S, S)) {
 GSSNode = create(RS2, GSSNode, index, cN)
 goto LS
 } else { goto L0 }
RS2 :
 pop(GSSNode, index, cN)
 goto L0
```

На вход такому синтаксическому анализатору подается цепочка bbb. Построенный в процессе работы стек изображен на рис. 3, *а*. На вершинах стека хранятся возвратные метки и индекс во входном потоке на момент создания вершины. Эти два параметра позволяют уникально определить вершину. На ребрах стека лежат части дерева, которые были получены до того, как начался разбор данного нетерминала. В данном примере на ребрах лежат деревья только такого вида, как показано на рис. 3, *б*. Деревья, получаемые в процессе выполнения функции pop( ) или create( ), подробное описание которых можно посмотреть в оригинальной статье [14], представлены на рис. 3, *в*. (На рис. 3 на ребрах лежит null для пустых деревьев или нетерминал с указанием соответствующего интервала во входном потоке.)

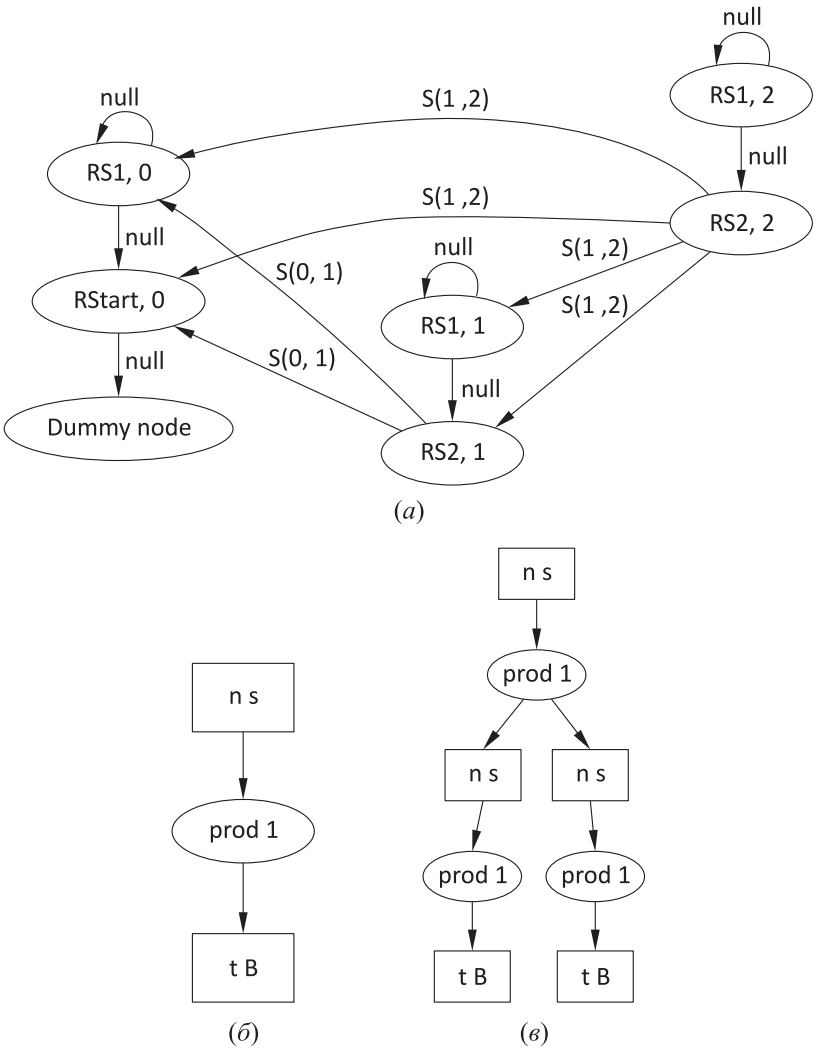


Рис. 3 Процесс вывода цепочки bbb и грамматики G0: (*а*) GSS; (*б*) деревья, хранящиеся на стеке; (*в*) деревья, получаемые в результате операций pop( ) или create( )

Результатом работы анализатора является лес разбора. В данном случае лес разбора состоит из двух деревьев, что проиллюстрировано на рис. 4. На рисунке места неоднозначностей отмечены серым цветом. Нетерминальные ячейки помечены буквой n и именем нетерминала, а под ними находятся узлы с номерами продукций, по которым цепочка была выведена. Терминальные ячейки помечены буквой t и терминалом и не имеют потомков.

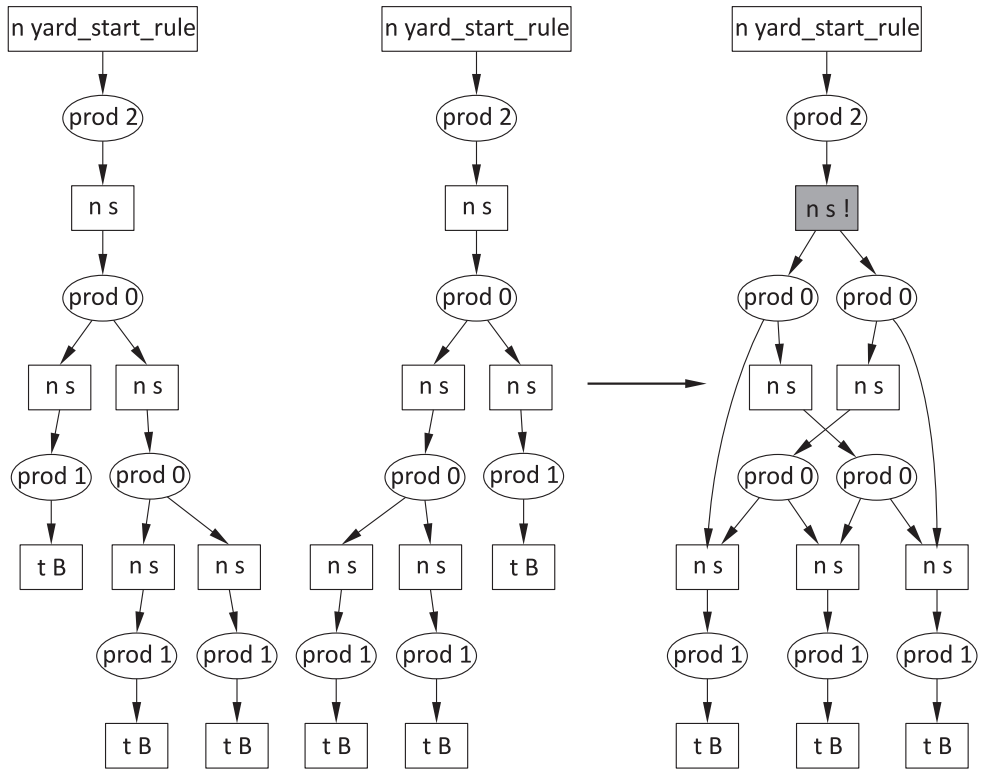


Рис. 4 Сжатие леса разбора в SPPF

5 Абстрактный синтаксический анализ

Абстрактный синтаксический анализ используется для работы со встроенными языками — такими языками, выражения на которых собираются из строк во время исполнения программы и затем передаются соответствующему окружению для выполнения. Выражения на таких языках называются динамически формируемыми выражениями, так как действительное значение, передаваемое на выполнение, будет получено только во время выполнения основной программы. Наиболее распространенные примеры встроенных языков — генерация HTML-страниц, встроенный (в C#, Java, PHP) SQL, динамический SQL.

Так как динамически формируемые выражения — тоже код на некотором языке программирования, то для обеспечения его надежности (отсутствия ошибок, отсутствия мест потенциальных инъекций), так же как и для обычного кода, необходимо выполнение различных анализов, первым шагом многих из которых является синтаксический анализ. Задача осложняется тем, что анализировать каждое возможное значение динамически формируемого выражения нецелесо-

образно и часто невозможно, так как количество таких значений может быть бесконечным. Поэтому предлагаются подходы, которые позволяют проводить анализ некоторого компактного представления множества значений выражения: dataflow уравнения [1], регулярного выражения [1].

Существует несколько подходов к созданию абстрактных синтаксических анализаторов, но чаще всего используется табличный подход. В статье [1] описан пример создания табличного абстрактного анализатора на основе LR-таблиц. В статье [4] описано решение, позволяющее создать абстрактный анализатор на основе обобщенного алгоритма RNGLR. Показано, что использование обобщенного анализа позволяет улучшить абстрактный анализ. Это достигается во многом благодаря использованию структур данных и способов управления ими, свойственных обобщенному анализу. GSS позволяет переиспользовать стеки для общих частей динамически сформированного выражения, а с помощью SPPF можно компактно хранить лес разбора. Это позволяет существенно снизить расход памяти, так как в общем случае при анализе динамически формируемого выражения должно быть получено дерево разбора для каждого возможного значения выражения. Однако на практике часто оказывается, что деревья имеют много общих частей, которые можно переиспользовать.

На основе этого же принципа можно построить абстрактный анализатор с использованием LL-таблиц и алгоритма GLL. Его применение дает следующие преимущества:

(1) высокую скорость работы для леворекурсивных грамматик;

(2) возможность более качественной и простой диагностики и восстановления после ошибок.

Второй пункт является особенно важным, так как качественная и понятная для пользователя диагностика ошибок в абстрактном анализе существенно затруднена сложностью структуры входных данных. По этой причине упрощение механизма диагностики ошибок при сохранении или повышении качества — важная задача.

Для получения перечисленных преимуществ была поставлена задача создания табличного анализатора с использованием алгоритма обобщенного нисходящего синтаксического анализа.

6 Описание модификаций

Поскольку в рамках работы необходимо было реализовать табличный анализатор, процесс работы которого отличается от описанного в статьях [8, 15], в алгоритм были внесены некоторые изменения.

Грамматика перед началом работы синтаксического анализатора представляется в удобной для работы с ней форме. Каждому символу грамматики сопоставляется число; правила хранятся как пара массивов (левые и правые части). В качестве дополнительной информации, которая необходима интерпретатору в процессе работы, генерируются функции для работы с грамматикой,

например, сопоставляющие числам их строковое представление, и таблицы синтаксического анализа. Таблицы, как и в классическом LL-анализе, используются для определения альтернативы, по которой будет осуществляться разбор. В зависимости от символа во входном потоке и текущего нетерминала выбирается альтернатива и в очередь добавляется новый дескриптор. В местах конфликтов создается и записывается дескриптор для каждой из альтернатив. В ячейках таблицы хранятся номера продукций, по которым осуществляется разбор. В дескрипторах вместо метки функции, которую необходимо вызвать, хранится пара чисел: номер правила и позиция в нем. Таким образом, последовательный вызов функций заменяется на обход грамматики с использованием входной цепочки. Вместо нескольких функций, соответствующих нетерминалам, используется пара взаимно рекурсивных функций: управляющая dispatcher( ) и обрабатывающая processing( ). Функция dispatcher( ) выполняет ту же роль, что и раньше: извлекает дескрипторы из очереди, устанавливает значение локальных переменных (позиция в грамматике, позиция во входном потоке, стек и дерево) и вызывает обрабатывающую функцию. Обрабатывающая же функция состоит из последовательности инструкций if-then-else, в телах которых содержится код для обработки всех возможных ситуаций.

1. Если текущий рассматриваемый символ в грамматике $x = A$, где A — терминал, то необходимо перейти к рассмотрению следующего символа в правиле, а указатель во входном потоке увеличить на единицу.
2. Если x — это нетерминальный символ, то в стек записывается текущее правило и запоминается позиция в нем. С использованием этой информации будет осуществляться обработка после того, как нетерминал x будет обработан до конца. А рассматриваемым становится правило, по которому раскрывается x в зависимости от текущего символа во входном потоке. Указатель во входном потоке остается без изменений.
3. Если какое-то правило рассмотрено до конца и текущий стек не пуст, то извлекается дескриптор с вершины стека и продолжается работа с этими данными.

Таким образом, обрабатывающая функция просто выполняет разные действия в зависимости от ситуации. В дескрипторах вместо хранения метки функции, по которой будет осуществляться разбор, теперь хранится пара чисел: номер правила и позиция в нем. В процессе разбора указатель в грамматике сдвигается.

В листинге, приведенном ниже, представлен код получившегося интерпретатора. Вспомогательные функции, такие как pop( ), getNodeP( ), getNodeT, add( ) и create( ), описаны в статье [15].

Инициализация глобальных переменных:

```
read the input into I and set I[m] := $
label = (startRule, 0)
GSSNode = create dummy GSS
```

```
cN = dummyAST
cR = dummyAST
setR = empty
setP = empty
condition = false
stop = false
index = 0
```

Управляющая функция, которая при генерации кода носила название L_Dispatch, выполняет все те же действия, что и ранее:

```
let rec dispatcher () =
  if setR.Count <> 0
  then
    currentContext := setR.Dequeue()
    currentIndex := currentContext.Value.Index
    currentGSSNode := currentContext.Value.Vertex
    currentLabel := currentContext.Value.Label
    currentN := currentContext.Value.Ast
    currentR := dummyAST
    condition := false
  else
    stop := true
```

Обрабатывающая функция, выполняющая различные действия в зависимости от ситуации:

```
and processing () =
 condition := true
 if length rule = 0
 then pop !currentGSSNode !currentIndex !currentN
 elif length rule <> position
 then
  if !currentIndex < inputLength
  then
   if isTerminal curSymbol
   then
    if curSymbol = curToken
    then
     if !currentN = dummyAST
     then currentN := getNodeT !currentIndex
     else currentR := getNodeT !currentIndex
     currentIndex := !currentIndex + 1
     currentLabel := createNewLabel rule (position + 1)
     if !currentR <> dummyAST
     then currentN := getNodeP !currentLabel !currentN !currentR
```

```
     condition := false
   else
    let rules = table[curSymbol][curToken]
    currentGSSNode :=
     create (createNewLabel rule (position + 1))
           !currentGSSNode !currentIndex !currentN
    if Array.length rules <> 0
    then
     for rule in rules do
      let newLabel = createNewLabel rule 0
      addContext newLabel !currentIndex !currentGSSNode dummyAST
  else condition := true
 else
  let curRight = !currentN
  let extension = curRight.extension
  let resTree = findTree key fam
  if key = finalExtension
  then resultAST := Some resTree
  pop !currentGSSNode !currentIndex resTree
       currentN.Value.extension
let control () =
  while not !stop do
     if !condition then dispatcher() else processing()
control()
```

Таким образом, в алгоритме рассмотрены все возможные ситуации, которые могут возникнуть во время разбора. Вместо частного поведения функций, построенных по грамматике, были выделены общие ситуации.

7 Обсуждение

В данной работе описан подход к созданию табличного анализатора на основе алгоритма GLL. В рамках проекта YaccConstructor [2] на языке программирования F# [16] создана рабочая версия анализатора, которая способна работать с левой, правой и скрытыми рекурсиями. К сожалению, его производительность на данном этапе ниже, чем у имеющейся реализации алгоритма RNGLR. В [17] приведено сравнение анализаторов, созданных с использованием разных алгоритмов обобщенного анализа, демонстрирующее, что производительность GLL-анализатора значительно выше, чем производительность RNGLR-анализатора. В дальнейшем планируется оптимизация структур данных для улучшения производительности анализатора. В статье, описывающей технические детали реализации алгоритма GLL, представлены достаточно сложные структуры данных для хранения деревьев и стека, использование которых и планируется в дальнейшем.

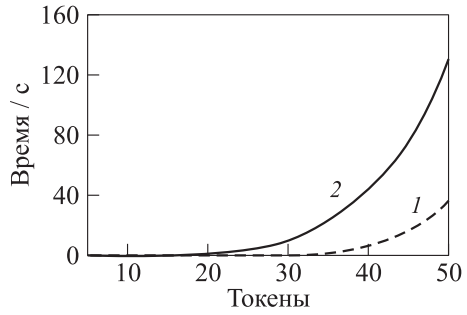


Рис. 5 Сравнение GLL-анализатора (*1*) и RNGLR-анализатора (*2*)

На данном этапе уже были произведены некоторые оптимизации, что позволило значительно улучшить скорость работы самой первой версии анализатора. Для хранения меток, например, используются сжатые числа, что позволяет быстро сравнивать их. Для работы с вершинами стека также используются два сжатых числа — координаты в таблице. Это позволяет быстро осуществлять поиск и обращение к нужной вершине, а сравнение вершин превращается в обычное сравнение чисел. На рис. 5 отображены результаты сравнения производительности RNGLR-анализатора и GLL-анализатора на неоднозначной леворекурсивной грамматике G0, которая ранее рассматривалась в качестве примера.

После этого на базе классического анализатора необходимо получить абстрактный анализатор для работы со встроенными языками. Алгоритм RNGLR имеет сложности с диагностикой ошибок, а в абстрактном анализе процесс работы и структура анализатора усложняются, что делает диагностику ошибок неточной [18]. Диагностика ошибок в классическом LL-анализе гораздо более качественная и простая, чем в LR-анализе. Анализатор на основе GLL наследует преимущества LL-анализаторов, и предполагается, что таких проблем не возникнет в абстрактном GLL-анализаторе. Однако данный вопрос требует отдельных исследований.

Литература

1. *Kyung-Goo D., Hyunha K., Schmidt D. A.* Abstract parsing: Static analysis of dynamically generated string output using LR parsing technology // 16th Symposium (International) on Static Analysis Proceedings. — Berlin: Springer-Verlag, 2009. P. 256–272.
2. YaccConstructor. https://github.com/YaccConstructor/YaccConstructor.
3. *Scott E., Johnstone A.* Right nulled GLR parsers // ACM Trans. Program. Lang. Syst., 2006. Vol. 28. No. 4. P. 577– 618. doi: 10.1145/1146809.1146810.
4. *Grigoriev S. V., Kirilenko I. A.* GLR-based abstract parsing // 9th Central & Eastern European Software Engineering Conference in Russia Proceedings. — New York, NY, USA: ACM, 2013. Article No. 5.
5. *Rosenkrantz D. J., Stearns R. E.* Properties of deterministic top down grammars // 1st Annual ACM Symposium on Theory of Computing Proceedings. — New York, NY, USA: ACM, 1969. P. 165–180.

6. *Aho A. V., Lam M. S., Sethi R., Ullman J. D.* Compilers: Principles, techniques, and tools. — 2nd ed. — Boston: Addison-Wesley Longman Publs. Co., Inc., 2006. 1184 p.
7. *Nederhof M.-J., Sarbo J. J.* Increasing the applicability of LR parsing // 3rd Workshop (International) on Parsing Technologies. — Tilburg, 1993. P. 187–201.
8. *Scott E., Johnstone A.* GLL parsing // Electronic Notes Theoretical Computer Sci., 2010. Vol. 253. Iss. 7. P. 177–189. doi: 10.1016/j.entcs.2010.08.041.
9. *Tomita M.* Efficient parsing for natural language: A fast algorithm for practical systems. — Norwell: Kluwer Academic Publs., 1985. 201 p.
10. ANTLR. http://www.antlr.org.
11. *Scott E., Johnstone A.* Generalized bottom up parsers with reduced stack activity // Computer J., 2005. Vol. 48. No. 5. P. 565–587. doi: 10.1093/comjnl/bxh102.
12. *Scott E., Johnstone A., Economopoulos R.* BRNGLR: A cubic Tomita-style GLR parsing algorithm // Acta Inform., 2007. Vol. 44. No. 6. P. 427– 461. doi: 10.1007/s00236-007-0054-z.
13. *Rekers J. G.* Parser generation for interactive environments. — Amsterdam: University of Amsterdam, 1992. PhD Thesis. 174 p.
14. *Scott E., Johnstone A.* GLL parse-tree generation // Sci. Comput. Program., 2013. Vol. 78. No. 10. P. 1828–1844. doi: 10.1016/j.scico.2012.03.005.
15. *Johnstone A., Scott E.* Modelling GLL parser implementations // 3rd Conference (International) on Software Language Engineering. — Berlin: Springer-Verlag, 2010. P. 42–61.
16. *Syme D., Granicz A., Cisternino A.* Expert F#. — New York, NY, USA: Apress, 2007. 609 p.
17. *Economopoulos R. G.* Generalised LR parsing algorithms. — London, U.K.: Department of Computer Science, Royal Holloway, University of London, 2006. PhD Thesis. 232 p.
18. *Вербицкая Е. А.* Диагностика ошибок при анализе встроенных языков. — СПб.: СПбГУ, 2014. Курсовая работа. 18 с.

Поступила в редакцию 20.01.15

GENERALIZED TABLE-BASED LL-PARSING

S. V. Grigorev and A. K. Ragozina

Saint-Petersburg State University, 7-9 Universitetskaya Nab., St. Petersburg199034, Russian Federation

Abstract: Syntax analysis is an important step of code analysis. The problem is that the grammars have to be in a form which is deterministic, or at least near-deterministic for the chosen parsing technique. Generalized parsing algorithms — Generalized LR and Generalized LL (GLL) — make it possible to remove these restrictions. Abstract analysis makes it possible to parse embedded languages for supporting them in IDE, reengineering tasks, or finding vulnerabilities (SQL-injection). Abstract syntax analysis is based on the classic table-based analysis.

The generalized algorithm of top-down parsing without the use of predictive tables was described earlier in order to extend the class of languages processed by descent analyzers. This paper describes an approach to creation of a table-based GLL-analyzer based on the proposed algorithm, which will be used later for an abstract analyzer. This article describes the algorithm of generalized top-down analysis, its modifications, and the results of comparison with the generalized bottom-up parsing algorithm, which was implemented earlier.

Keywords: generalized parsing; GLL; RNGLR; abstract parsing; string-embedded languages

DOI: 10.14357/08696527150106

References

1. Kyung-Goo, D., K. Hyunha, and D. A. Schmidt. 2009. Abstract parsing: Static analysis of dynamically generated string output using LR-Parsing technology. *16th Symposium (International) on Static Analysis Proceedings*. Berlin: Springer-Verlag. 256–272.
2. YaccConstructor. Available at: http://yaccconstructor.github.io/YaccConstructor/ (accessed February 03, 2015).
3. Scott, E., and A. Johnstone. 2006. Right nulled GLR parsers. *ACM Trans. Program. Lang. Syst.* 28(4):577–618. doi: 10.1145/1146809.1146810.
4. Grigoriev, S. V., and I. A. Kirilenko. 2013. GLR-based abstract parsing. *9th Central & Eastern European Software Engineering Conference in Russia Proceedings*. New York, NY, USA. Article No. 5. 1–9.
5. Rosenkrantz, D. J., and R. E. Stearns. 1969. Properties of deterministic top down grammars. *1st Annual Symposium on Theory of Computing Proceedings*. New York, NY, USA: ACM. 165–180.
6. Aho, A. V., M. S. Lam, R. Sethi, and J. D. Ullman. 2006. *Compilers: Principles, techniques, and tools*. 2nd ed. Boston: Addison-Wesley Longman Publs. Co., Inc. 1184 p.
7. Nederhof, M.-J., and J. J. Sarbo. 1993. Increasing the applicability of LR parsing. *3rd Workshop (International) on Parsing Technologies*. Tilburg. 187–201.
8. Scott, E., and A. Johnstone. 2010. GLL parsing. *Electronic Notes Theoretical Computer Sci.* 253(7):177–189. doi: 10.1016/j.entcs.2010.08.041.
9. Tomita, M. 1985. *Efficient parsing for natural language: A fast algorithm for practical systems*. Norwell: Kluwer Academic Publs. 201 p.
10. ANTLR. Available at: http://www.antlr.org/ (accessed February 03, 2015).
11. Scott, E., and A. Johnstone. 2005. Generalized bottom up parsers with reduced stack activity. *Comput. J.* 48(5):565–587. doi: 10.1093/comjnl/bxh102.
12. Scott, E., A. Johnstone., and R. Economopoulos. 2007. BRNGLR: A cubic Tomita-style GLR parsing algorithm. *Acta Inform.* 44(6):427–461. doi: 10.1007/s00236-007-0054-z.
13. Rekers, J. G. 1992. Parser generation for interactive environments. Amsterdam: Universty of Amsterdam. PhD Thesis. 174 p.

14. Scott, E., and A. Johnstone. 2013. GLL parse-tree generation. *Sci. Comput. Program.* 78(10):1828–1844. doi: 10.1016/j.scico.2012.03.005.
15. Johnstone, A., and E. Scott. 2010. Modelling GLL parser implementations. *3rd Conference (International) on Software Language Engineering*. Berlin: Springer-Verlag. 42–61.
16. Syme, D., A. Granicz, and A. Cisternino. 2007. *Expert F#.* New York, NY, USA: Apress. 609 p.
17. Economopoulos, R. G. 2006. Generalised LR parsing algorithms. London: Department of Computer Science, Royal Holloway, University of London. PhD Thesis. 232 p.
18. Verbitskaya, E. A. 2014. Diagnostika oshibok pri analize vstroennykh yazykov [Error detection in string-embedded languages]. St. Petersburg: Saint-Petersburg State University. Course work. 18 p.

Received January 20, 2015

Contributors

Grigorev Semen V. (b. 1989) — PhD student, Saint-Petersburg State University, 7-9 Universitetskaya Nab., St. Petersburg 199034, Russian Federation; Semen.Grigorev@jetbrains.com

Ragozina Anastasiya K. (b. 1992) — student, Saint-Petersburg State University, 7-9 Universitetskaya Nab., St. Petersburg 199034, Russian Federation; ragozina.anastasiya@gmail.com

СИНТЕЗ УСТОЙЧИВЫХ ЛИНЕЙНЫХ ФИЛЬТРОВ И ЭКСТРАПОЛЯТОРОВ ПУГАЧЁВА ДЛЯ СТОХАСТИЧЕСКИХ СИСТЕМ С МУЛЬТИПЛИКАТИВНЫМИ ШИРОКОПОЛОСНЫМИ ШУМАМИ

И. Н. Синицын[1]***, Э. Р. Корепанов***[2]

Аннотация: Статья посвящена теории аналитического синтеза непрерывных и дискретных равномерно асимптотически устойчивых условно-оптимальных линейных фильтров и экстраполяторов Пугачёва (ЛФП и ЛЭП) для линейных дифференциальных стохастических систем (СтС) с линейными мультипликативными гауссовскими широкополосными шумами, описывающих состояние и наблюдения. Предполагается, что наблюдение входит как в уравнение состояния, так и в уравнение наблюдения. Доказаны теоремы, лежащие в основе алгоритмов синтеза непрерывных устойчивых ЛФП и ЛЭП. Достаточные условия равномерной асимптотической устойчивости сформулированы в виде требований положительной определенности и равномерной стохастической ограниченности некоторых матриц, отражающих свойства наблюдаемости и управляемости. Изложена теория аналитического синтеза непрерывных равномерно асимптотически устойчивых ЛФП и ЛЭП при автокоррелированном широкополосном шуме в наблюдениях. Рассмотрены алгоритмы синтеза дискретных равномерно асимптотически устойчивых ЛФП и ЛЭП как для дискретных, так и для непрерывных СтС с линейными мультипликативными шумами. Приведен иллюстративный пример. Сформулированы некоторые обобщения.

Ключевые слова: мультипликативный автокоррелированный широкополосный шум; стохастическая система (СтС); точность и устойчивость ЛФП и ЛЭП; уравнение Риккати; условно-оптимальный линейный фильтр и экстраполятор Пугачёва (ЛФП и ЛЭП)

DOI: 10.14357/08696527150107

1 Введение

В настоящее время теория ЛФП и ЛЭП для линейных непрерывных и дискретных СтС с аддитивными гауссовскими и негауссовскими широкополосными шумами подробно разработана и реализована в универсальных и специальных программных средствах (см., например, [1–3]).

[1]Институт проблем информатики Федерального исследовательского центра «Информатика и управление» Российской академии наук, sinitsin@dol.ru

[2]Институт проблем информатики Федерального исследовательского центра «Информатика и управление» Российской академии наук, ekorepanov@ipiran.ru

Для СтС с мультипликативными (параметрическими) белыми шумами в уравнениях состояния и наблюдения в [4] для ЛФП и ЛЭП получены соответствующие математические уравнения Риккати для оценки точности фильтрации и экстраполяции.

Рассматривается развитие [4] на случай широкополосных шумов для ЛФП и ЛЭП. На основе известных свойств матричных уравнений Риккати получены достаточные условия равномерной асимптотической устойчивости непрерывных и дискретных ЛФП и ЛЭП и алгоритмы их синтеза. Особое внимание уделено случаю автокоррелированных шумов в наблюдениях. Приводится пример.

2 Синтез устойчивых линейных фильтров Пугачёва

Рассмотрим наблюдаемую непрерывную СтС с мультипликативными и аддитивными негауссовскими широкополосными шумами, описываемую следующими стохастическими дифференциальными уравнениями Ито:

$$\dot{X}_t = aY_t + a_1X_t + a_0 + \left(c_{10} + \sum_{r=1}^{n_y} c_{1r}Y_r + \sum_{r=1}^{n_x} c_{1,n_y+r}X_r\right)U_t,\ \ X_{t_0} = X_0; \quad (1)$$

$$\dot{Y}_t = bY_t + b_1X_t + b_0 + \left(c_{20} + \sum_{r=1}^{n_y} c_{2r}Y_r + \sum_{r=1}^{n_x} c_{2,n_y+r}X_r\right)U_t,\ \ Y_{t_0} = Y_0, \quad (2)$$

где X_t и Y_t — векторы состояния и наблюдения ($\dim X_t = n_x$, $\dim Y_t = n_y$); a_0, b_0, a, a_1, b, b_1 и c_{ij} ($i = 1, 2$, $j = 1, \ldots, n_x$) — векторно-матричные функции t, не зависящие от $X_t = [X_1 \cdots X_{n_x}]^{\mathrm{T}}$ и $Y_t = \left[Y_1 \cdots Y_{n_y}\right]^{\mathrm{T}}$; V_t — векторный гауссовский широкополосный шум ($\dim U_t = n_U$), обладающий конечными ковариационными моментами $K^U(t, t')$.

Тогда, следуя [1, 2], заменив стохастический процесс (СтП) U_t эквивалентным (по критерию совпадения ковариационных моментов) белым шумом $V_э$ с интенсивностью $\nu_э$:

$$U_t = V_э\,,\ \ \nu_э = \nu_э(t) = \int\limits_{-\infty}^{\infty} K^U(t, t')\,dt'\,, \quad (3)$$

придем к следующим результатам:

$$\dot{\hat{X}}_t = \left(aY_t + a_1\hat{X}_t + a_0\right) + \beta_t\left[\dot{Y}_t - \left(bY_t + b_1\hat{X}_t + b_0\right)\right]\,; \quad (4)$$

$$\beta_t = \left(R_t b_1^{\mathrm{T}} + \sigma_{12}\right)\sigma_{22}^{-1}\,; \quad (5)$$

$$\dot{R}_t = a_1R_t + R_ta_1^{\mathrm{T}} - \left(R_tb_1^{\mathrm{T}} + \sigma_{12}\right)\sigma_{22}^{-1}\left(b_1R_t + \sigma_{21}\right) + \sigma_{11}\,; \quad (6)$$

$$\sigma_{11} = \left(c_{10} + \sum_{r=1}^{n_y+n_x} c_{1r} m_r\right) \nu_э \left(c_{10}^{\mathrm{T}} + \sum_{r=1}^{n_y+n_x} c_{1r}^{\mathrm{T}} m_r\right) + \sum_{r,s=1}^{n_y+n_x} c_{1r} \nu_э c_{1s}^{\mathrm{T}} k_{rs}\,; \quad (7)$$

$$\sigma_{12} = \left(c_{10} + \sum_{r=1}^{n_y+n_x} c_{1r} m_r\right) \nu_э \left(c_{20}^{\mathrm{T}} + \sum_{r=1}^{n_y+n_x} c_{2r}^{\mathrm{T}} m_r\right) + \sum_{r,s=1}^{n_y+n_x} c_{1r} \nu_э c_{2s}^{\mathrm{T}} k_{rs}\,; \quad (8)$$

$$\sigma_{22} = \left(c_{20} + \sum_{r=1}^{n_y+n_x} c_{2r} m_r\right) \nu_э \left(c_{20}^{\mathrm{T}} + \sum_{r=1}^{n_y+n_x} c_{2r}^{\mathrm{T}} m_r\right) + \sum_{r=1}^{n_y+n_x} c_{2r} \nu_э c_{2s}^{\mathrm{T}} k_{rs}\,; \quad (9)$$

$$\sigma_{21} = \left(c_{20}^{\mathrm{T}} + \sum_{r=1}^{n_y+n_x} c_{2r}^{\mathrm{T}} m_r\right) \nu_э \left(c_{10}^{\mathrm{T}} + \sum_{r=1}^{n_y+n_x} c_{1r}^{\mathrm{T}} m_r\right) + \sum_{r,s=1}^{n_y+n_x} c_{2r} \nu_э c_{1s}^{\mathrm{T}} k_{rs}\,, \quad (10)$$

где $\hat{X}_t$ — оценка X_t; R_t — ковариационная матрица ошибки фильтрации $\tilde{X}_\tau = \hat{X}_\tau - X_t$.

При этом вероятностные моменты первого и второго порядка $m_t = [m_r]$, $K_t = [k_{rs}]$ составного вектора $\left[\hat{X}_1 \cdots \hat{X}_{n_x} Y_1 \cdots Y_{n_y}\right]^{\mathrm{T}}$ $(r, s = 1, \ldots, n_y + n_x)$ определяются следующими уравнениями:

$$\dot{m}_t = a m_t + a_0\,; \quad (11)$$

$$\dot{K}_t = a K_t + K_t a^{\mathrm{T}} + c_0 \nu_э c_0^{\mathrm{T}} + \sum_{r=1}^{n_y+n_x} \left(c_0 \nu_э c_r^{\mathrm{T}} + c_r \nu_э c_0^{\mathrm{T}}\right) m_r + \sum_{r,s=1}^{n_y+n_x} c_r \nu_э c_s^{\mathrm{T}} (m_r m_s + k_{rs})\,, \quad (12)$$

где

$$a = \begin{bmatrix} b & b_1 \\ a & a_1 \end{bmatrix}\ ;\ \ a_0 = \begin{bmatrix} b_0 \\ a_0 \end{bmatrix}\ ;\ \ c_r = \begin{bmatrix} c_{2r} \\ a_{1r} \end{bmatrix}\ ;\ \ r = 0, 1, \ldots, n_y + n_x\,.$$

Далее, применяя известное матричное неравенство [2, 5–7] к уравнению Риккати (6), получим $\forall t \geq t_0$

$$0 \leq R_t(R_0, t_0) \leq u(t, t_0) R_0 u^{\mathrm{T}}(t, t_0) + \int_{t_0}^{t} u(t, \tau) \bar{\sigma}_{11} u(t, \tau)^{\mathrm{T}}\, d\tau = u(t, t_0) R_0 u(t, t_0)^{\mathrm{T}} + \mathcal{W}(t_0, t)\,.$$

Здесь введены следующие обозначения: $R_0 = R_{t_0}$; $u(t, t_0)$ — фундаментальная матрица уравнения

$$\dot{Z} = \bar{a}_1 Z\,, \quad \bar{a}_1 = a_1 - \sigma_{12}\sigma_{22}^{-1} b_1\,; \tag{13}$$

$$\mathcal{W}(t-t',t) = \int\limits_{t-t'}^{t} u(t,\tau)\bar{\sigma}_{11}(\tau)u(t,\tau)^{\mathrm{T}}\,d\tau\,; \tag{14}$$

$$\bar{\sigma}_{11} = \sigma_{11} - \sigma_{12}\sigma_{22}^{-1}\sigma_{21}\,.$$

Наконец, обобщая понятия равномерной наблюдаемости и управляемости [2, 5–7], введем понятие равномерной ограниченности

$$0 < \alpha_1 I_n \le \mathcal{W}(t-t',t) \le \alpha_2 I_n \;\; \forall\, t \ge t_0 + t'\,, \tag{15}$$

где α_1, α_2 и t' — постоянные; $n = n_y + n_x$; I_n — единичная матрица.

Таким образом, в основе аналитического синтеза равномерно асимптотически устойчивого ЛФП лежит следующее утверждение.

Теорема 2.1. *Пусть процессы X_t и Y_t в дифференциальной СтС* (1)–(3) *обладают конечными ковариационными моментами, а матрица σ_{22}* (9) *не вырождена* ($\det\sigma_{22} \ne 0$). *Тогда ЛФП* (4) *определяется уравнениями* (5)–(12), *а его точность — матричным уравнением Риккати* (6). *Для обеспечения равномерной асимптотической устойчивости ЛФП* (4) *при $Y_t = 0$, $\dot{Y}_t = 0$ (т. е. устойчивости однородного уравнения* (13)) *достаточно положительной определенности матрицы* (14) *и ее равномерной ограниченности* (15).

Замечание 2.1. Фильтр, определяемый уравнением (4), оптимален в классе всех линейных фильтров, причем ЛФП является оптимальным и в более широком классе всех линейных фильтров. В частном случае линейной СтС (1) и (2) без мультипликативных шумов уравнения (4) и (6) совпадают с уравнениями теории оптимальной линейной фильтрации [1, 3]. Для винеровского процесса $W_0(t)$ при отсутствии мультипликативных шумов ЛФП оказывается оптимальным и в классе всех возможных фильтров.

Замечание 2.2. Особенностью ЛФП является то обстоятельство, что $m_r = m_{rt}$, $K_r = K_{rt}$ и $R = R_t$ могут быть вычислены заранее во время синтеза ЛФП, так как не требуют знания результатов наблюдений.

Случай независимых аддитивных и мультипликативных гауссовских белых шумов $V_{э1}$ и $V_{э2}$ соответственно в уравнениях состояния и наблюдения получается на основе уравнений (1) и (2), если принять

$$V_э = \left[V_{э1}^{\mathrm{T}} V_{э2}^{\mathrm{T}}\right]^{\mathrm{T}}; \quad \nu_э = \begin{bmatrix} \nu_{э1} & 0 \\ 0 & \nu_{э2} \end{bmatrix}; \tag{16}$$

$$\left(c_{i0}+\sum_{r=1}^{n_y}c_{ir}Y_r+\sum_{s=1}^{n_x}c_{i,n_y+r}X_r\right)V_{\text{э}}=$$

$$=\left(c'_{i0}+\sum_{r=1}^{n_y}c'_{ir}Y_r+\sum_{s=1}^{n_x}c'_{i,n_y+r}X_r\right)V_{\text{э}i}\,; \quad (17)$$

$$\sigma_{ii}=\left(c'_{i0}+\sum_{r=1}^{n_y+n_x}c'_{ir}m_r\right)\nu_{\text{э}i}\left(c'^{\mathrm{T}}_{i0}+\sum_{r=1}^{n_y+n_x}c'^{\mathrm{T}}_{ir}m_r\right),\ \ \sigma_{ij}=\sigma_{ji}=0\ \ (i\neq j)\,;$$

$$\bar{\sigma}_{11}=\sigma_{11}\,; \quad (18)$$

$$\beta_t=R_tb_1^{\mathrm{T}}\sigma_{22}^{-1}\,. \quad (19)$$

Тогда будет иметь место

Теорема 2.2. *Пусть процессы X_t и Y_t в системе* (1)–(3) *при условиях* (16), (17) *обладают конечными ковариационными моментами, а матрица σ_{22} не вырождена* ($\det\sigma_{22}\neq 0$). *Тогда линейный условно-оптимальный фильтр определяется уравнением* (5) *при условии* (19), *а его точность — матричным уравнением* (9). *Для обеспечения равномерной асимптотической устойчивости ЛФП* (4) *достаточно положительной определенности матрицы* (14) *при условиях* (18) *и* (15).

Замечание 2.3. Для синтеза стационарных ЛФП согласно алгоритмам теорем 2.1 и 2.2 используется стационарное уравнение Риккати, получаемое из (6) при $R_t=R^*$, а также $m_t=m^*$, $K_t=K^*$.

3 Синтез устойчивых линейных экстраполяторов Пугачёва

Обобщая [1, 3] на случай широкополосных шумов U_{1t} и U_{2t} вида (3) и с учетом теоремы 2.2, придем к следующему утверждению.

Теорема 3.1. *Пусть процессы X_t, Y_t в уравнениях*

$$\dot{X}_t=a_1X_t+a_0+\left(c_{10}+\sum_{r=1}^{n_x}c_{1,n_y+r}X_r\right)U_{1t}\,,\ \ X_{t_0}=X_0\,;$$

$$\dot{Y}_t=bY_t+b_1X_t+b_0+\left(c_{20}+\sum_{r=1}^{n_y}c_{2r}Y_r+\sum_{r=1}^{n_x}c_{2,n_y+r}X_r\right)U_{2t}\,,\ \ Y_{t_0}=Y_0\,,$$

где U_{it} $(i=1,2)$ — независимые широкополосные гауссовские СтП, для которых интенсивности $\nu_{\text{э}i}=\nu_{\text{э}i}(t)$ удовлетворяют условиям (3) *и имеют конечные ковариационные моменты. Тогда уравнения ЛЭП имеют вид*:

$$\dot{\hat{X}}_t = \left[a_1(t+\Delta)\hat{X}_t + a_0(t+\Delta)\right] + \\ + \beta_t\left[\dot{Y}_t - (bY_t + b_1\varepsilon_t^{-1}\hat{X}_t + b_0 - b_1\varepsilon_t^{-1}h_t)\right]; \quad (20)$$

$$\varepsilon_t = u(t+\Delta, t)\,, \quad (21)$$

где $u(t,\tau)$ — *фундаментальная матрица уравнения*

$$\dot{u}_t = a_1(t)u_t\,; \\ h_t = h(t) = \int\limits_t^{t+\Delta} u(t+\Delta,\tau)a_0(\tau)\,d\tau\,. \quad (22)$$

Необходимые для вычисления σ_{22} *по формулам* (9) *и* β_t *по формулам* (6) *вероятностные моменты можно найти из уравнений* (11), (12) *для составного вектора* $\left[Y_t^{\mathrm{T}} X_t^{\mathrm{T}} \hat{X}_t^{\mathrm{T}}\right]^{\mathrm{T}}$. *Роль матриц* c_{1r}, c_{2r} *играют матрицы* $[0\,c_{1r}]$ *и* $[c_{2r}\,0]$, *а матрица* ν *диагональна. При этом точность экстраполяции определяется путем интегрирования следующего уравнения*:

$$\dot{R}_t^{\text{э}} = A_\Delta R_t^{\text{э}} + R_t^{\text{э}} A_\Delta^{\mathrm{T}} + R_t^{\text{э}} B_\Delta R_t^{\text{э}} + C_\Delta\,.$$

Здесь введены следующие обозначения:

$$A_\Delta = a_1(t+\Delta)\,; \quad B_\Delta = b_1^{\mathrm{T}}\sigma_{22,2}\sigma_{22,2}^{\mathrm{T}}b_1\,;$$

$$C_\Delta = \left[c_{10}(t+\Delta) + \sum_{r=n_y+1}^{n_y+n_x} c_{12}(t+\Delta)m_r(t+\Delta)\right] \times \\ \times\, \nu_{\text{э}2}(t+\Delta)\left[c_{10}(t+\Delta)^{\mathrm{T}} + \sum_{r=n_y+1}^{n_y+1} c_{12}(t+\Delta)^{\mathrm{T}} m_r(t+\Delta)\right] + \\ + \sum_{r=n_y+1}^{n_y+n_x} c_{1r}(t+\Delta)\nu_{\text{э}2}(t+\Delta)c_{1s}(t+\Delta)^{\mathrm{T}}k_{rs}\,;$$

$$\sigma_{22,2} = \left(c_{20} + \sum_{r=1}^{n_y+n_x} c_{2r}m_r\right)\nu_{\text{э}2}\left(c_{20}^{\mathrm{T}} + \sum_{r=1}^{n_y+n_x} c_{2r}^{\mathrm{T}}m_r\right) + \sum_{r=1}^{n_y+n_x} c_{2r}\nu_{\text{э}2}c_{2s}^{\mathrm{T}}r_{rs}, \quad (23)$$

причем для обеспечения равномерной асимптотической устойчивости достаточно выполнения условий:

$$0 < \alpha_1 I_n \le \mathcal{W}_\Delta\left(t-t', t\right) \le \alpha_2 I_n\,.$$

Здесь

$$\mathcal{W}_\Delta(t-t',t)=\int\limits_{t-t'}^{t} u_\Delta(t,\tau)\mathcal{C}(\tau)u_\Delta(t,\tau)^{\mathrm{T}}\,d\tau\,,$$

где $u_\Delta(t,\tau)$ — *фундаментальная матрица для уравнения*

$$\dot{Z}=A_\Delta Z\,. \tag{24}$$

Замечание 3.1. Из (20) видно, что ЛЭП можно представить в виде последовательного ЛФП, усилителя с коэффициентом усиления ε_t (21) и сумматора, вводящего неслучайное слагаемое h_t (22). Найденный ЛЭП оптимален в классе всех линейных экстраполяторов [1–3].

Замечание 3.2. Сравнивая (12) и (24), заключаем, что для устойчивости ЛЭП достаточно устойчивости ЛФП.

4 Синтез устойчивых линейных фильтров Пугачёва при автокоррелированном шуме в наблюдениях

Следуя [1, 3] с учетом (3), рассмотрим непрерывную дифференциальную СтС с мультипликативными широкополосными шумами при автокоррелированном шуме в наблюдениях, основываясь на следующих уравнениях:

$$\dot{X}_t=a_{11}Y_t+a_{12}X_t+a_{13}N_t+a_{10}+\\ +\left(c_{10}+\sum_{r=1}^{n_y}c_{1r}Y_r+\sum_{r=1}^{n_x}c_{1,n_y+r}X_r+\sum_{r=1}^{n_N}c_{1,n_y+n_x+r}N_r\right)U_t\,; \tag{25}$$

$$\dot{Y}_t=b_{11}Y_t+b_{12}X_t+b_{13}N_t+b_{10}\,; \tag{26}$$

$$\dot{N}_t=a_{21}Y_t+a_{22}X_t+a_{23}N_t+a_{23}+\\ +\left(c_{20}+\sum_{r=1}^{n_y}c_{2r}Y_r+\sum_{r=1}^{n_x}c_{2,n_y+r}X_r+\sum_{r=1}^{n_N}c_{2,n_y+n_x+r}N_r\right)U_t\,; \tag{27}$$

$$\dot{\hat{X}}_t=\alpha_t\left[Y_t^{(1)T}\cdots Y_t^{(s)T}Y_t^{\mathrm{T}}\hat{X}_t^{\mathrm{T}}\right]^{\mathrm{T}}+\beta_t\dot{Y}_t^{(s)}+\gamma_t\,. \tag{28}$$

Здесь U_t — гауссовский широкополосный шум, определенный в разд. 2. Используя результаты [1, 3], определим производные

$$Y_t^{(k)}=b_{k1}Y_t+b_{k2}X_t+b_{k3}N_t+b_{k0}\ \ (k=1,\,\ldots\,,\,s)\,; \tag{29}$$

$$\dot{Y}_t^{(k)} = b_{s+1,1}Y_t + b_{s+1,2}X_t + b_{s+1,3}N_t + b_{s+1,0} + \\ + \left(c_{30} + \sum_{r=1}^{n_y} c_{3r}Y_r + \sum_{r=1}^{n_x} c_{3,n_y+r}X_r + \sum_{r=1}^{n_N} c_{3,n_y+n_x+r}N_r \right) V_э\,, \quad (30)$$

где $V_э$ — эквивалентный гауссовский белый шум, а функции времени b_{kl} определяются рекуррентной формулой

$$b_{k+1,l} = b_{k1}b_{1l} + b_{k2}a_{1l} + b_{k3}a_{2l} + \dot{b}_{kl} \ \ (l = 0,1,2,3; k = 1,\ \ldots\ , s)\,, \quad (31)$$

а

$$c_{sr} = b_{s2}c_{1r} + b_{s3}c_{2r} \ \ (r = 0,1,\ \ldots\ , n_y + n_x + n_N)\,;$$
$$Z_t = \left[Y_t^{(1)\mathrm{T}} \cdots Y_t^{(s)\mathrm{T}} \right]^{\mathrm{T}} = \bar{b}_1 Y_t + \bar{b}_2 X_t + \bar{b}_3 N_t + \bar{b}_0\,;$$
$$\bar{b}_l = \left[b_{1l}^{\mathrm{T}} \cdots b_{sl}^{\mathrm{T}} \right]^{\mathrm{T}} \ \ (l = 0,1,2,3)\,. \quad (32)$$

Из (27) и (30), учитывая несмещенность оценки $\hat{X}_t$ и равенства $K_{xy} = K_{\hat{x}y}$, $K_{x\hat{x}} = K_{\hat{x}}$, найдем

$$\beta_t = \left[R_t b_{s+1,2}^{\mathrm{T}} + (K_{xn} - K_{\hat{x}n}) b_{s_1,3}^{\mathrm{T}} + \zeta_{13} \right] \zeta_{33}^{-1}. \quad (33)$$

Здесь $R_t = K_x - K_{\hat{x}}$ — ковариационная матрица ошибки фильтрации с соответствующими индексами. Рассмотрим ковариационные и взаимные ковариационные матрицы следующих случайных векторов:

$$\zeta_{ij} = c_{i0}\nu_э c_{j0}^{\mathrm{T}} + \sum_{r=1}^{n_y+n_x+n_N} (c_{ir}\nu_э c_{j0}^{\mathrm{T}} + c_{i0}\nu_э c_{jr}^{\mathrm{T}}) m_r + \\ + \sum_{r=1}^{n_y+n_x+n_N} c_{ir}\nu_э c_{jq}^{\mathrm{T}} (m_r m_q + k_{rq}) \ \ (i,j = 1,2,3)\,,$$

где m_r и k_{rq} $(r,q = 1,\ \ldots\ , n_y+n_x+n_N)$ — математические ожидания компонент случайного вектора $\left[Y_t^{\mathrm{T}} X_t^{\mathrm{T}} N_t^{\mathrm{T}} \right]^{\mathrm{T}}$ и элементы его ковариационной матрицы.

Матрица $\alpha_t = [\alpha_1\alpha_2\alpha_3]$ в (4) определяется из равенств $K_{zx} = K_{\hat{x}z}$, $K_{xy} = K_{\hat{x}y}$, $K_{x\hat{x}} = K_{\hat{x}}$ с учетом

$$\kappa_{11} = \begin{bmatrix} K_z & K_{zy} & K_{z\hat{x}} \\ K_{yz} & K_y & K_{y\hat{x}} \\ K_{\hat{x}z} & K_{\hat{x}y} & K_{\hat{x}} \end{bmatrix}$$

из системы уравнений

$$\left.\begin{aligned}
&\alpha_1 K_z + \alpha_2 K_{yz} + \alpha_3 K_{\hat{x}z} = \\
&\quad = (a_{11} - \beta_t b_{s+1,1}) K_{yz} + (a_{12} - \beta_t b_{s+1,2}) K_{\hat{x}z} + (a_{13} - \beta_t b_{s+1,3}) K_{nz}\,; \\
&\alpha_1 K_{zy} + \alpha_2 K_y + \alpha_3 K_{\hat{x}y} = \\
&\quad = (a_{11} - \beta_t b_{s+1,1}) K_y + (a_{12} - \beta_t b_{s+1,2}) K_{\hat{x}y} + (a_{13} - \beta_t b_{s+1,3}) K_{ny}\,; \\
&\alpha_1 K_{z\hat{x}} + \alpha_2 K_{y\hat{x}} + \alpha_3 K_{\hat{x}} = \\
&\quad = (a_{11} - \beta_t b_{s+1,1}) K_{y\hat{x}} + (a_{12} - \beta_t b_{s+1,2}) K_{\hat{x}} + (a_{13} - \beta_t b_{s+1,3}) K_{n\hat{x}}\,.
\end{aligned}\right\} \tag{34}$$

Наконец, учитывая соотношения

$$\left.\begin{aligned}
K_{zy} &= \bar{b}_1 K_y + \bar{b}_2 K_{\hat{x}y} + \bar{b}_3 K_{ny}\,, \\
K_{z\hat{x}} &= \bar{b}_1 K_{y\hat{x}} + \bar{b}_2 K_{\hat{x}} + \bar{b}_3 K_{n\hat{x}}\,;
\end{aligned}\right\} \tag{35}$$

$$K_z = K_{zy}\bar{b}_1^{\mathrm{T}} + K_{z\hat{x}}\bar{b}_2^{\mathrm{T}} + K_{zn}\bar{b}_3^{\mathrm{T}}\,;\;\; K_{yz} = K_y\bar{b}_1^{\mathrm{T}} + K_{y\hat{x}}\bar{b}_2^{\mathrm{T}} + K_{yn}\bar{b}_3^{\mathrm{T}}\,;$$

$$K_{\hat{x}z} = K_{\hat{x}y}\bar{b}_1^{\mathrm{T}} + K_{\hat{x}}\bar{b}_2^{\mathrm{T}} + K_{\hat{x}n}\bar{b}_3^{\mathrm{T}}\,;\;\; K_{nz} = K_{ny}\bar{b}_1^{\mathrm{T}} + K_{n\hat{x}}\bar{b}_2^{\mathrm{T}} + K_n\bar{b}_3^{\mathrm{T}}\,;$$

$$K_{zx} = K_{z\hat{x}}\,;\;\; K_{yx} = K_{y\hat{x}}\,;\;\; K_{x\hat{x}} = K_{\hat{x}}\,;$$

$$\alpha_1(\bar{b}_3\bar{K}_n\bar{b}_3^{\mathrm{T}} - \bar{b}_2 R_t \bar{b}_2^{\mathrm{T}}) = (a_{13} - \beta_t b_{s+1,3})\left[\bar{K}_n\bar{b}_3^{\mathrm{T}} + (K_{nx} - K_{n\hat{x}})\bar{b}_2^{\mathrm{T}}\right]\,; \tag{36}$$

$$\bar{K}_n = K_n - [K_{yn}\, K_{\hat{x}n}]\,\bar{\kappa}_{11}^{-1}\left[K_{yn}^{\mathrm{T}}\, K_{\hat{x}n}^{\mathrm{T}}\right]^{\mathrm{T}}\,; \tag{37}$$

$$\bar{\kappa}_{11} = \begin{bmatrix} K_y & K_{y\hat{x}} \\ K_{\hat{x}y} & K_{\hat{x}} \end{bmatrix}\,; \tag{38}$$

$$\begin{aligned}
\left[\alpha_2 + \alpha_1\bar{b}_1 \;\; \alpha_3 + \alpha_1\bar{b}_2\right] = {} & [(a_{11} - \beta_t b_{s+1,1})a_{12} - b_{s+1,2}] + \\
& + (a_{13} - \alpha_1\bar{b}_3 - \beta_t b_{s+1,3})\,[K_{ny}\, K_{n\hat{x}}]\,\bar{\kappa}_{11}^{-1}\,;
\end{aligned} \tag{39}$$

$$\gamma_t = a_{10} - \beta_t b_{s+1,0} + (a_{13} - \beta_t b_{s+1,3} - \alpha_1\bar{b}_3)\bar{m}_n\,, \tag{40}$$

где

$$\bar{m}_n = m_n - [K_{yn}\, K_{\hat{x}n}]\,\bar{\kappa}_{11}^{-1}\left[m_y^{\mathrm{T}}\, m_x^{\mathrm{T}}\right]\,; \tag{41}$$

$$K_t = \begin{bmatrix} K_y & K_{yx} & K_{yn} \\ K_{xy} & K_x & K_{xn} \\ K_{ny} & K_{nx} & K_n \end{bmatrix}\,;\;\; K_{\hat{x}n} = K_{n\hat{x}}\,; \tag{42}$$

$$\dot{m}_t = Am_t + A_0\,; \tag{43}$$

$$\dot{K}_t = AK_t + K_tA^{\mathrm{T}} + c_0\nu_э c_0^{\mathrm{T}} + \sum_{r=1}^{n_y+n_x+n_N} (c_0\nu_э c_r^{\mathrm{T}} + c_r\nu_э c_0^{\mathrm{T}})m_r + \\ + \sum_{r,q=1}^{n_y+n_x+n_N} c_r\nu_э c_q^{\mathrm{T}}(m_rm_q + k_{rq}), \quad (44)$$

а

$$A = \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{bmatrix}; \quad A_0 = \begin{bmatrix} b_{10} \\ a_{10} \\ a_{20} \end{bmatrix}; \quad c_r = \begin{bmatrix} 0 \\ c_{1r} \\ c_{2r} \end{bmatrix}$$

$$r = 0, 1, \ldots, n_y + n_x + n_N,$$

получим искомые уравнения ЛФП. При этом для нахождения уравнения для ковариационной матрицы ошибки R_t получаем уравнение

$$\dot{R}_t = a_{12}R_t + R_ta_{12}^{\mathrm{T}} + a_{13}(K_{nx} - K_{n\hat{x}}) + (K_{xn} - K_{\hat{x}n})a_{13}^{\mathrm{T}} - \beta_t\zeta_{33}\beta_t^{\mathrm{T}} + \zeta_{11}. \quad (45)$$

Наконец, для нахождения уравнения для $K_{\hat{x}n}$ следует составить уравнение для ковариационной матрицы составного вектора $\left[Y_t^{\mathrm{T}} X_t^{\mathrm{T}} N_t^{\mathrm{T}} \hat{X}_t^{\mathrm{T}}\right]^{\mathrm{T}}$:

$$\dot{K}_{\hat{x}n} = \alpha_3 K_{\hat{x}n} + K_{\hat{x}n}a_{23}^{\mathrm{T}} + (K_x - R_t)a_{22}^{\mathrm{T}} + K_{xy}a_{21}^{\mathrm{T}} + \\ + (\alpha_2 + \alpha_1\bar{b}_1 + \beta_t b_{s+1,1})K_{yn} + (\alpha_1\bar{b}_2 + \beta_t b_{s+1,2})K_{xn} + \\ + (\alpha_1\bar{b}_3 + \beta_t b_{s+1,3})K_n + \beta_t\sigma_{22}. \quad (46)$$

При этом учтено, что $K_{\hat{x}} = K_x - R_t$; $K_{\hat{x}y} = K_{xy}$.

Таким образом, путем обобщения [1, 3] на случай негауссовского белого шума (3) и автокоррелированного шума в наблюдениях приходим к утверждению, определяющему алгоритм синтеза ЛФП.

Уравнения (45) и (46) для составной матрицы $\bar{R} = [R_t^э R_t']$ $(R_t' = K_{\hat{x}n})$ имеют следующий вид:

$$\dot{\bar{R}}_t = \bar{A}\bar{R}_t + \bar{R}_t\bar{A}^{\mathrm{T}} + \bar{R}_t\bar{B}\bar{R}_t + \bar{C}, \quad (47)$$

где $\bar{A}$, $\bar{B}$ и $\bar{C}$ — соответствующие матрицы. Для равномерной асимптотической устойчивости ЛФП достаточно выполнения условий теоремы 2.1 для матрицы $\bar{A}$. В общем случае эти условия очень громоздки и в статье не приводятся. Следует использовать численную проверку.

Таким образом, имеем следующий алгоритм синтеза ЛФП.

Теорема 4.1. *Пусть уравнения* (25)–(27), (3) *допускают существование ковариационных моментов. Тогда уравнение ЛФП имеет вид* (28). *При этом, интегрируя уравнения* (45) *и* (46) *при соответствующих начальных условиях с* β_t

и α_t*, определяемых* (33), (36)–(39)*, найдем* α_t *и* β_t*, после чего* γ_t *определяется согласно* (40) *с учетом* (42)–(44)*. Для обеспечения равномерной асимптотической устойчивости ЛФП достаточно выполнения условий теоремы* 2.1 *для матрицы* $\bar{A}$ *в* (47)*.*

Замечание 4.1. Для получения приближенных критериев устойчивости ЛФП можно, согласно [1, 3], использовать квазилинейный фильтр Пугачёва, получаемый путем статистической линеаризации уравнений (25)–(27) как билинейной системы совместно с уравнением для формирующего фильтра, связывающего U_t с белым шумом. При этом гауссовость не обязательна.

5 Синтез устойчивых линейных экстраполяторов Пугачёва при автокоррелированном шуме в наблюдениях

Аналогично, обобщая [1, 3] на случай широкополосных шумов, будем основываться на уравнениях вида:

$$\dot{X}_t = a_{12}X_t + a_{10} + \left(c_{10} + \sum_{r=1}^{n_x} c_{1r}X_r\right)U_{1t}\,; \tag{48}$$

$$\dot{Y}_t = b_{11}Y_t + b_{12}X_t + b_{13}N_t + b_{10}\,;$$

$$\dot{N}_t = a_{23}N_t + a_{20} + \left(c_{20} + \sum_{r=1}^{n_N} c_{2r}N_r\right)U_{2t}\,, \tag{49}$$

а класс допустимых экстраполяторов определять с помощью уравнения (28), в котором b_1, b_2, b_3 и b_0 находятся по формулам (31) и (32) при $a_{11} = 0$, $a_{12} = 0$, $a_{21} = 0$ и $a_{22} = 0$. Коэффициенты c_{sr} $(r = 0, 1, \ldots, n_N)$ определяются в этом случае по формуле $c_{3r} = b_{s3}c_{2r}$, а $b_{s2}c_{1r} = 0$ $(r = 0, , \ldots, n_x)$. Здесь приняты следующие обозначения:

$$\kappa_{22} = b_{s2}\sigma_{22,2}b_{s2}^{\mathrm{T}}\,,$$

где $\sigma_{22,2}$ вычисляется согласно (23); $\varepsilon_t = u(t + \Delta, t)$; $u(t, \tau)$ — решение однородного уравнения $du/dt = a_{12}u$ при начальном условии $u(\tau, \tau) = I$;

$$\zeta_{22} = c_{20}\nu_{э2}c_{20}^{\mathrm{T}} + \sum_{r=1}^{n_N}(c_{20}\nu_{э2}c_{2r}^{\mathrm{T}} + c_{20}\nu_{э2}c_{20}^{\mathrm{T}})m_{n_y+n_x+r} +$$

$$+ \sum_{r,q=1}^{n_N} c_{2r}\nu_{э2}c_{2q}^{\mathrm{T}}(m_{n_y+n_x+r}m_{n_y+n_x+q} + k_{n_y+n_x+r,n_y+n_x+q})\,; \tag{50}$$

$$\beta_t = \left\{(\varepsilon_t K_x - K_{x\hat{x}})b_{s+1,2}^{\mathrm{T}} + (\varepsilon_t K_{xn} - K_{\hat{x}n})b_{s+1,3}^{\mathrm{T}}\right\}/(b_{s2}\zeta_{22}b_{s2}^{\mathrm{T}})^{-1}\,;$$

$$K_{\hat{x}z} = \varepsilon_t K_{xz}\,;\quad K_{\hat{x}y} = \varepsilon_t K_{xy}\,;\quad K_{\hat{x}} = \varepsilon_t K_{x\hat{x}}\,.$$

Система уравнений для $\alpha_t = [\alpha_1 \alpha_2 \alpha_3]$ имеет вид:

$$\alpha_1 K_z + \alpha_2 K_{yz} + \alpha_3 K_{\hat{x}z} = \\ = -\beta_t b_{s+1,1} K_{yz} + \left\{ a_{12}(t+\Delta) - \beta_t b_{s+1,2} \varepsilon_t^{-1} \right\} K_{\hat{x}z} - \beta_t b_{s+1,3} K_{nz} ;$$

$$\alpha_1 K_{zy} + \alpha_2 K_y + \alpha_3 K_{\hat{x}y} = \\ = -\beta_t b_{s+1,1} K_y + \left\{ a_{12}(t+\Delta) - \beta_t b_{s+1,2} \varepsilon_t^{-1} \right\} K_{\hat{x}y} - \beta_t b_{s+1,3} K_{ny} ,$$

$$\alpha_1 K_{z\hat{x}} + \alpha_2 K_{y\hat{x}} + \alpha_3 K_{\hat{x}} = \\ = -\beta_t b_{s+1,1} K_{y\hat{x}} + \left\{ a_{12}(t+\Delta) - \beta_t b_{s+1,2} \varepsilon_t^{-1} \right\} K_{\hat{x}} - \beta_t b_{s+1,3} K_{n\hat{x}} .$$

Эта система уравнений может быть получена из (34), если заменить a_{11} и a_{13} нулями, а a_{12} и $\beta_{s+1,2}$ — величинами $a_{12}(t+\Delta)$ и $b_{s+1,2}\varepsilon_t^{-1}$. Поэтому оптимальные α_1, α_2 и α_3 можно определить по формулам (36) и (39) с такой же заменой. При этом уравнение для γ_t имеет вид:

$$\gamma_t = a_{10}(t+\Delta) - (\alpha_1 \bar{b}_3 + \beta_t b_{s+1,3}) \bar{m}_n + [a_{12}(t+\Delta) - \alpha_3] \, h_t ,$$

где $\bar{m}_n$ определяется по формуле (41), причем

$$m_x(t+\Delta) = \varepsilon_t m_x(t) + h(t) = \varepsilon_t m_x + h_t ;$$

$$h_t = h(t) = \int_t^{t+\Delta} u(t+\Delta, \tau) a_{10}(\tau) \, d\tau .$$

Для нахождения α_t, β_t и γ_t по полученным формулам необходимо знать величины m_t, K_t, R_t, $K_{\hat{x}}$ и $K_{\hat{x}n}$. Так как математическое ожидание m_t и ковариационная матрица K_t случайного вектора $\left[Y_t^{\mathrm{T}} X_t^{\mathrm{T}} N_t^{\mathrm{T}}\right]$ определяются уравнениями (43) и (44), в которых

$$A = \begin{bmatrix} \bar{b}_1 & \bar{b}_2 & \bar{b}_3 \\ 0 & a_{12} & 0 \\ 0 & 0 & a_{23} \end{bmatrix} ; \quad A_0 = \begin{bmatrix} \bar{b}_0 \\ a_{10} \\ a_{20} \end{bmatrix} ; \quad c_{30} = \begin{bmatrix} 0 & 0 \\ c_{10} & 0 \\ 0 & c_{20} \end{bmatrix} ;$$

$$c_{3r} = \begin{cases} 0 , & r = 0, 1, \ldots, n_y ; \\ \begin{bmatrix} 0 & 0 \\ c_{1,r-n_y} & 0 \\ 0 & 0 \end{bmatrix} , & r = n_y + 1, \ldots, n_y + n_x ; \\ \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & c_{2,r-n_y-n_x} \end{bmatrix} , & r = n_y + n_x + 1, \ldots, n_y + n_x + n_N ; \end{cases}$$

$$m_{\hat{x}}(t) = m_x(t+\Delta)\,,\;\; R_t = \varepsilon_t K_x \varepsilon_t^{\mathrm{T}} - K_{\hat{x}}\,,$$

получим следующее уравнение для R_t:

$$\dot{R}_t = a_{12}(t+\Delta)R_t + R_t a_{12}(t+\Delta)^{\mathrm{T}} - \beta_t b_{s2}\zeta_{22} b_{s2}\beta_t^{\mathrm{T}} + \zeta_{11}(t+\Delta)\,. \qquad (51)$$

Здесь ζ_{22} определяется по формуле (50), а

$$\zeta_{11} = c_{10}\nu_{э1}c_{10}^{\mathrm{T}} + \sum_{r=1}^{n_x}(c_{10}\nu_{э1}c_{1r}^{\mathrm{T}} + c_{1r}\nu_{э1}c_{10}^{\mathrm{T}})m_{n_y+r} + \\ + \sum_{r,q=1}^{n_x} c_{1r}\nu_{э1}c_{1q}^{\mathrm{T}}(m_{n_y+r}m_{n_y+q} + k_{n_y+r,n_y+q})\,. \qquad (52)$$

В рассматриваемом случае матрица $K_{\hat{x}n}$ определяется уравнением

$$\dot{K}_{\hat{x}n} = \alpha_3 K_{\hat{x}n} + K_{\hat{x}n}a_{23}^{\mathrm{T}} + (\alpha_2 + \alpha_1\bar{b}_1 + \beta_t b_{s+1,1})K_{yn} + \\ + (\alpha_1\bar{b}_2 + \beta_t b_{s+1,2})K_{xn} + (\alpha_1\bar{b}_3 + \beta_t b_{s+1,3})K_n\,. \qquad (53)$$

Найденный ЛЭП может быть представлен в виде последовательного соединения ЛФП, усилителя с коэффициентом усиления ε_t и сумматора, вводящего неслучайное слагаемое h_t.

В силу (51) и (53), аналогично теореме 4.1, для проверки устойчивости составляется уравнение для составной матрицы $\bar{R}_t = [R_t R'_t]$ ($R'_t = K_{\hat{x}n}$):

$$\dot{\bar{R}}_t = \bar{\bar{A}}\bar{R}_t + \bar{R}_t\bar{\bar{A}}^{\mathrm{T}} + \bar{R}_t\bar{\bar{B}}\bar{R}_t + \bar{\bar{C}}\,, \qquad (54)$$

где $\bar{\bar{A}}$, $\bar{\bar{B}}$ и $\bar{\bar{C}}$ — соответствующие матрицы.

Таким образом, приходим к следующему алгоритму синтеза ЛЭП.

Теорема 5.1. *Пусть процессы X_t и Y_t удовлетворяют уравнениям* (48) *и* (49) *и имеют конечные ковариационные моменты. Тогда непрерывный ЛЭП имеет вид* (28), *в котором $\bar{b}_0$, $\bar{b}_1$, $\bar{b}_2$ и $\bar{b}_3$ находятся согласно* (31) *и* (32) *при $a_{11} = a_{12} = a_{21} = a_{22} = 0$ и $c_{3r} = b_{s3}c_{2r}$, $b_{s2}c_{1,r} = 0$. При этом интегрирование* (43), (44), (51) *и* (52) *проводится при соответствующих начальных условиях и с учетом полученных коэффициентов α_t, β_t и γ_t. Для обеспечения равномерной асимптотической устойчивости ЛЭП достаточно выполнения условий теоремы* 2.1 *для матрицы $\bar{\bar{A}}$ в* (54).

6 Синтез устойчивых линейных фильтров и экстраполяторов Пугачёва для дискретных стохастических систем

Рассмотрим сначала дискретную наблюдаемую линейную СтС с мультипликативными дискретными широкополосными белыми шумами следующего вида [1, 3]:

$$X_{k+1} = a_k X_k + a_{0k} + \left(c_{0k} + \sum_{r=1}^{n} c_{rk} X_{kr} \right) U_k^d ; \tag{55}$$

$$Y_k = b_k X_k + b_{0k} + \left(l_{0k} + \sum_{r=1}^{n} l_{rk} X_{kr} \right) U_k^d . \tag{56}$$

Здесь $n = n_x = n_y$; X_{kr} и Y_{kr} — r-е компоненты векторов X_k и Y_k; c_{0k}, l_{0k}, c_{rk} и l_{rk} — постоянные коэффициенты; U_k^d — векторный гауссовский широкополосный шум с ковариационной матрицей ν_k^d и матрицей ковариационных функций ν_{kh}^d, удовлетворяющих условию вида (3):

$$U_k^d = V_k^{\text{э}d} ; \;\; \nu_k^d = \nu_{\text{э}k}^d ; \;\; \nu_{\text{э}k}^d = \sum_h \nu_{kh} . \tag{57}$$

Следуя [1, 3], в качестве класса допустимых ЛФП выберем класс линейных фильтров следующего вида:

$$\hat{X}_{k+1} = a_k \hat{X}_k + a_{0k} + \delta_{2k}(Y_k - b_k \hat{X}_k - b_{0k}) ; \tag{58}$$

$$\delta_{2k} = (a_k R_k b_k^{\mathrm{T}} + H_{k12}) \sigma_{22}^{-1} ; \tag{59}$$

$$\left.\begin{aligned}
H_{k11} &= c_{0k} \nu_{\text{э}k}^d c_{0k}^{\mathrm{T}} + \sum_{r=1}^{n} m_{kr} (c_{0k} \nu_{\text{э}k}^d c_{rk}^{\mathrm{T}} + c_{rk} \nu_{\text{э}k}^d c_{0k}^{\mathrm{T}}) + \\
&\qquad + \sum_{r,s=1}^{n} (m_{kr} m_{ks} + k_{krs}) c_{rk} \nu_{\text{э}k}^d c_{sk}^{\mathrm{T}} , \\
H_{k12} &= c_{0k} \nu_{\text{э}k}^d l_{0k}^{\mathrm{T}} + \sum_{r=1}^{n} m_{kr} (c_{0k} \nu_{\text{э}k}^d l_{rk}^{\mathrm{T}} + c_{rk} \nu_{\text{э}k}^d l_{0k}^{\mathrm{T}}) + \\
&\qquad + \sum_{r,s=1}^{n} (m_{kr} m_{ks} + k_{krs}) c_{rk} \nu_{\text{э}k}^d l_{sk}^{\mathrm{T}} , \\
H_{k22} &= l_{0k} \nu_{\text{э}k}^d l_{0k}^{\mathrm{T}} + \sum_{r=1}^{n} m_{kr} (l_{0k} \nu_{\text{э}k}^d l_{rk}^{\mathrm{T}} + l_{rk} \nu_{\text{э}k}^d l_{0k}^{\mathrm{T}}) + \\
&\qquad + \sum_{r,s=1}^{n} (m_{kr} m_{ks} + k_{krs}) l_{rk} \nu_{\text{э}k}^d l_{sk}^{\mathrm{T}} ,
\end{aligned}\right\} \tag{60}$$

$$\sigma_{22} = b_k R_k b_k^{\mathrm{T}} + H_{k22}\,; \tag{61}$$

$$m_{k+1} = a_k m_k + a_{0k}\,; \tag{62}$$

$$K_{k+1} = a_k K_k a_k^{\mathrm{T}} + H_{k11}\,; \tag{63}$$

$$R_{k+1} = a_k R_k a_k^{\mathrm{T}} - \delta_{2k}(b_k R_k a_k^{\mathrm{T}} + H_{k12}^{\mathrm{T}}) + H_{k11}\,. \tag{64}$$

Замечание 6.1. Уравнения (62) и (63) линейны относительно m_k и K_k, в то время как (64) нелинейно относительно R_k в силу зависимости δ_{2k} от R_k. Согласно теории фильтров Калмана и Пугачёва m_k, K_k, R_k и δ_{2k} могут быть вычислены заранее во время синтеза фильтра, поскольку (58) и (62) не содержат результатов наблюдений. Линейный фильтр (57) в случае, когда $X_1, \ldots, X_{k-1}$ входят в него только посредством $\hat{X}_k$, является среднеквадратическим оптимальным фильтром среди всех возможных линейных фильтров.

Для обеспечения равномерной асимптотической устойчивости ЛФП (5), аналогично условиям теоремы 2.1, достаточно выполнения условий положительной определенности и равномерной ограниченности:

$$\mathcal{W}^d(n-n',n) = \sum_{j=n-n'} u_A^d(n,j)\bar{\sigma}_{11}^d u_A^d(n,j)^{\mathrm{T}}\,; \tag{65}$$

$$\bar{\sigma}_{11}^d = \sigma_{11}^d - \sigma_{12}^d(\sigma_{22}^d)^{-1}\sigma_{21}^d\,; \tag{66}$$

$$0 < \alpha_1 I_n \le \mathcal{W}^d(n-n',n) \le \alpha_2 I_n \ \ (\forall\, n > n_0 + n')\,. \tag{67}$$

Таким образом, в основу синтеза равномерно асимптотически устойчивых ЛФП может быть положено следующее утверждение.

Теорема 6.1. *Пусть процессы X_k и Y_k, определяемые в дискретной СтС* (55) *и* (56)*, имеют конечные ковариационные моменты второго порядка, причем матрица σ_{22}* (61) *не вырождена. Тогда уравнения равномерно асимптотически устойчивого ЛФП имеют вид* (57)–(60) *при условиях* (65) *и* (67).

Аналогично составляются алгоритмы синтеза дискретных ЛФП и ЛЭП для широкополосных шумов, а также широкополосных автокоррелированных шумов в наблюдениях путем замены ν_k^d на $\nu_{\text{э}k}^d$.

Наконец, отметим, что дискретные ЛФП и ЛЭП могут быть синтезированы для непрерывных СтС с гауссовскими и негауссовскими мультипликативными шумами методами дискретизации на основе обобщенной формулой Ито [2, 8, 9].

7 Пример

Рассмотрим случай, когда скалярные уравнения (1) и (2) содержат независимые широкополосные шумы U_{1t} и U_{2t}:

$$\dot{X}_t = aY_t + a_1 X_t + a_0 + (c_{10} + c_{11}X_t + c_{12}Y_t)U_{1t}\,;$$

$$\dot{Y}_t = bY_t + b_1 X_t + b_0 + (c_{20} + c_{21}X_t + c_{22}Y_t)U_{2t}\,.$$

Обратим внимание на то, что c_{1r} и c_{2r} здесь не те, что в уравнениях (1) и (2). Они представляют собой соответственно первые и вторые элементы матриц-строк, на которые умножается вектор $[U_{1t}U_{2t}]^{\mathrm{T}}$ в (1) и (2). Для простоты оставляем для них обозначения c_{1r} и c_{2r}.

Для двух типовых скалярных широкополосных СтП эквивалентные интенсивности белых шумов определяются формулами [1, 3]:

$$k(\tau) = K(t, t+\tau) = D\exp(\alpha|\tau|)\,;\;\; \nu^{\text{э}} = 2D\alpha^{-1}\,;$$
$$k(\tau) = K(t, t+\tau) = D\exp(-\alpha|\tau|)(\cos\omega_0\tau + \gamma\sin\omega_0|\tau|)\,;$$
$$\nu^{\text{э}} = 2D(\gamma\omega_0 + \alpha)\beta^{-2} \quad \left(\beta^2 = \omega_0^2 + \alpha^2\right).$$

Тогда в формулах (5) и (9) имеем:

$$\beta_t = \sigma_{22}^{-1} b_1 R_t\,;$$
$$\sigma_{22}^{\text{э}} = \nu_{\text{э}2}\left(c_{20} + c_{21}m_1 + c_{22}m_2\right)^2 + \nu_{\text{э}2}\left(c_{21}^2 k_{11} + 2c_{22}c_{21}k_{21} + c_{22}^2 k_{22}\right).$$

При этом уравнения теоремы 2.1, определяющие m_1, m_2, k_{11}, k_{12}, k_{21}, k_{22} и R_t, имеют следующий вид:

$$\dot m_1 = a_1 m_1 + a m_2 + a_0\,;\;\; \dot m_2 = b_1 m_1 + b m_2 + b_0\,;$$
$$\dot k_{11} = 2\left(a_1 k_{11} + a k_{12}\right) + \nu_{\text{э}1}\left(c_{11}m_1^2 + c_{12}m_2 + c_{10}\right)^2 + \nu_{\text{э}1}\left(c_{11}^2 k_{11} + 2c_{12}c_{11}k_{12} + c_{12}^2 k_{22}\right);$$
$$\dot k_{12} = (a_1 + b)k_{12} + b_1 k_{11} + a k_{22}\,;$$
$$\dot k_{22} = 2(b_1 k_{12} + b k_{22}) + \nu_{\text{э}2}\left(c_{21}m_1 + c_{22}m_2 + c_{20}\right)^2 + \nu_{\text{э}2}\left(c_{21}^2 k_{11} + 2c_{22}c_{21}k_{12} + c_{22}^2 k_{22}\right);$$
$$\dot R_t = 2a_1 R_t - \sigma_{22}^{-1} b_1^2 R_t^2 + \nu_{\text{э}1}\left(c_{11}m_1 + c_{12}m_2 + c_{10}\right)^2 + \nu_{\text{э}2}\left(c_{11}^2 k_{11} + 2c_{11}c_{12}k_{12} + c_{12}^2 k_{22}\right).$$

При этом $\bar\sigma_{11}^{\text{э}}$, входящая в условия устойчивости ЛФП (теорема 3.1), определяется формулой:

$$\bar\sigma_{11}^{\text{э}} = \nu_{\text{э}1}\left(c_{11}m_1 + c_{12}m_2 + c_{10}\right) + \nu_{\text{э}1}\left(c_{11}^2 k_{11} + 2c_{11}c_{12}k_{12} + c_{12}^2 k_{22}\right).$$

При $a = a_0 = c_{12} = 0$, $a_1 = const$ ЛЭП представляет собой последовательное соединение ЛФП и усилителя с коэффициентом $\varepsilon = \exp(a_1\Delta)$.

Приравнивая правые части (29)–(31) нулю, найдем условия для синтеза стационарных ЛФП и ЛЭП с постоянными $\beta_t = \beta^*$, $R_t = R^*$ и $\varepsilon_t = \varepsilon^*$. Для равномерной асимптотической устойчивости ЛФП достаточно неотрицательности коэффициента

$$A = a_1 - \frac{b_1^2 R^*}{\sigma_{22}^*} < 0\,.$$

8 Заключение

Получено обобщение известных результатов по синтезу непрерывных и дискретных равномерно асимптотически устойчивых ЛФП и ЛЭП для случая наблюдаемых дифференциальных и дискретных СтС с мультипликативными гауссовскими широкополосными шумами. Особое внимание уделено ЛФП и ЛЭП при автокоррелированной помехе в наблюдениях. Приводится иллюстративный пример.

Полученные результаты могут быть использованы и для синтеза дискретных ЛФП и ЛЭП для непрерывных СтС, если воспользоваться численными методами приведения нелинейных стохастических дифференциальных уравнений к разностным.

Практический интерес представляет задача синтеза устойчивых ЛФП и ЛЭП по критериям устойчивости, отличным от критерия равномерной асимптотической устойчивости.

Литература

1. *Пугачев В. С., Синицын И. Н.* Стохастические дифференциальные системы. Анализ и фильтрация. — М.: Наука, 1990. 632 с. (Англ. пер. *Pugachev V. S., Sinitsyn I. N.* Stochastic differential systems. Analysis and filtering. — Chichester, NY, USA: Jonh Wiley, 1987. 549 p.)
2. *Пугачев В. С., Синицын И. Н.* Теория стохастических систем. — М.: Логос, 2000. 1000 с.
3. *Синицын И. Н.* Фильтры Калмана и Пугачева. — 2-е изд. — М.: Логос, 2007. 776 с.
4. *Синицын И. Н., Корепанов Э. Р.* Условно оптимальные фильтры и экстраполяторы для стохастических систем с мультипликативными шумами // Информатика и её применения, 2015. Т. 9. Вып. 1. С. 2–8.
5. *Kalman R.* A new approach to linear filtering and prediction problems // J. Basic Eng. (ASME Trans. D), 1960. Vol. 82. P. 35–45.
6. *Bucy R., Joseph P.* Filtering for stochastic processes with applications to guidance. — New York Inter Science Publs., 1968. 238 p.
7. *Ройтенберг Я. Н.* Автоматическое управление. — 3-е изд., перераб. и доп. — М.: Наука, 1992. 576 с.
8. *Синицын И. Н., Синицын В. И.* Лекции по нормальной и эллипсоидальной аппроксимации в стохастических системах. — М.: ТОРУС ПРЕСС, 2013. 476 с.
9. *Синицын И. Н.* Параметрическое статистическое и аналитическое моделирование распределений в нелинейных стохастических системах на многообразиях // Информатика и её применения, 2013. Т. 7. Вып. 2. С. 4–16.

Поступила в редакцию 22.09.14

SYNTHESIS OF STABLE LINEAR PUGACHEV FILTERS AND EXTRAPOLATORS FOR STOCHASTIC SYSTEMS WITH WIDE BAND MULTIPLICATIVE NOISES

I. N. Sinitsyn and E. R. Korepanov

Institute of Information Problems, Federal Research Center "Computer Science and Control," Russian Academy of Sciences, 44-2 Vavilov Str., Moscow 119333, Russian Federation

Abstract: The article is dedicated to the analytical synthesis of continuous and discrete uniquely asymptotically stable conditionally optimal linear Pugachev filters and extrapolators (LPF and LPE) for stochastic systems (StS) with wide band multiplicative Gaussian noises. It is supposed that observation is part of the state and observation equations. The theorems serving as the basis for the algorithms of synthesis of continuous uniquely asymptotical stable LPF and LPE are proven. Continuous LPF and LPE for StS with wide band Gaussian autocorrelated noises are presented. Discrete LPF and LPE for continuous and discrete StS with wide band multiplicative Gaussian noises are considered. An illustrative example is given. Some generalizations are considered.

Keywords: accuracy; continuous stochastic system; discrete stochastic system; linear Pugachev extrapolator; linear Pugachev filter; multiplicative noises; Riccati equation; unique asymptotical stability; wide band gaussian

DOI: 10.14357/08696527150107

References

1. Pugachev, V. S., and I. N. Sinitsyn. 1987. *Stochastic differential systems. Analysis and filtering*. Chichester, NY: Jonh Wiley. 549 p.
2. Pugachev, V. S., and I. N. Sinitsyn. 2001. *Stochastic systems. Theory and applications*. Singapore: World Scientific. 908 p.
3. Sinitsyn, I. N. 2007. *Fil'try Kalmana i Pugacheva* [Kalman and Pugachev filters]. 2nd ed. Moscow: Logos. 776 p.
4. Sinitsyn, I. N., and E. R. Korepanov. 2015. Uslovno optimal'nye fil'try i ekstrapolyatory dlya stokhasticheskikh sistem s mul'tiplikativnymi shumami [Conditionally optimal filters and extrapolators for stochastic systems with multiplicative noises]. *Informatika i ee Primeneniya — Inform Appl.* 9(1):2–8.
5. Kalman, R. 1960. A new approach to linear filtering and prediction problems. *J. Basic Eng. (ASME Trans. D)* 82:35–45.
6. Bucy, R., and P. Joseph. 1968. Filtering for stochastic processes with applications to guidance. New York Inter Science Publs. 238 p.
7. Roytenberg, Ya. N. 1992. *Avtomaticheskoe upravlenie* [Automatic control]. 3rd ed. Moscow: Nauka. 576 p.

8. Sinitsyn, I. N., and V. I. Sinitsyn. 2013. *Lektsii po normal'noy i ellipsoidal'noy approksimatsii v stokhasticheskikh sistemakh* [Lectures on normal and ellipsoidal approximation of distributions in stochastic systems]. Moscow: TORUS PRESS. 476 p.
9. Sinitsyn, I. N. 2013. Parametricheskoe statisticheskoe i analiticheskoe modelirovanie raspredeleniy v nelineynykh stokhasticheskikh sistemakh na mnogoobraziyakh [Parametric statistical and analytical modeling of distributions in nonlinear stochastic systems on manifolds]. *Informatika i ee Primeneniya — Inform. Appl.* 7(2):4–16.

Received September 22, 2014

Contributors

Sinitsyn Igor N. (b. 1940) — Doctor of Science in technology, professor, Honored scientist of RF, Head of Department, Institute of Information Problems, Federal Research Center "Computer Science and Control," Russian Academy of Sciences, 44-2 Vavilov Str., Moscow 119333, Russian Federation; sinitsin@dol.ru

Korepanov Eduard R. (b. 1966) — Candidate of Science (PhD) in technology, Head of Laboratory, Institute of Information Problems, Federal Research Center "Computer Science and Control," Russian Academy of Sciences, 44-2 Vavilov Str., Moscow 119333, Russian Federation; ekorepanov@ipiran.ru

О СХОДИМОСТИ СЛУЧАЙНЫХ СУММ НЕЗАВИСИМЫХ СЛУЧАЙНЫХ ВЕКТОРОВ К МНОГОМЕРНЫМ ОБОБЩЕННЫМ ДИСПЕРСИОННЫМ ГАММА-РАСПРЕДЕЛЕНИЯМ*

А. Ю. Корчагин[1]

Аннотация: Цель данной работы — описать условия сходимости распределений сумм случайного числа независимых неодинаково распределенных многомерных случайных величин к многомерным дисперсионно-сдвиговым смесям нормальных законов и, в частности, к многомерным обобщенным гиперболическим или многомерным обобщенным дисперсионным гамма-распределениям (GVG — generalized variance-gamma).

Ключевые слова: случайная сумма; многомерная дисперсионно-сдвиговая смесь нормальных законов; многомерное обобщенное гиперболическое распределение; многомерное обобщенное дисперсионное гамма-распределение; обобщенное обратное гауссовское распределение; обобщенное гамма-распределение

DOI: 10.14357/08696527150108

1 Введение. Обозначения и вспомогательные результаты

При решении многих задач статистического анализа размер выборки, доступной исследователю, не является заранее определенным и может рассматриваться как случайный. В таких ситуациях объем выборки сам становится *наблюдением*, т. е. статистикой. На естественность такого подхода, в частности, обратил внимание Б. В. Гнеденко в работе [1], в которой было продемонстрировано, что при замене неслучайного объема выборки случайной величиной асимптотические свойства статистик могут радикально измениться. К примеру, вместо ожидаемого в соответствии с классической теорией нормального закона могут возникать распределения с произвольно тяжелыми хвостами.

Свойства статистик, построенных по выборкам случайного объема, рассматривались в десятках работ. Однако условия сходимости распределений таких статистик к дисперсионно-сдвиговым смесям нормальных законов были найдены лишь недавно [2, 3]. В работе [4] приведены критерии сходимости распределений статистик, построенных по выборкам случайного объема, к многомерным

*Исследование выполнено за счет гранта Российского научного фонда (проект 14-11-00397).

[1]Факультет вычислительной математики и кибернетики Московского государственного университета им. М. В. Ломоносова, sasha.korchagin@gmail.com

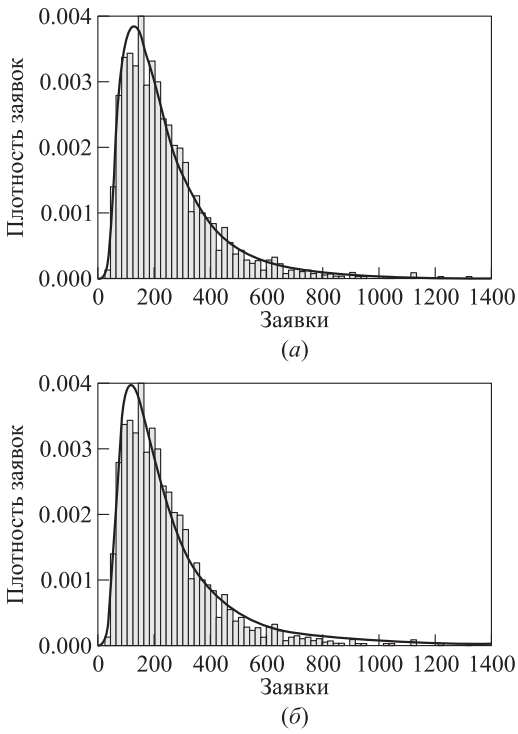
обобщенным гиперболическим законам. Как показано в этой статье, указанная сходимость имеет место тогда и только тогда, когда случайная интенсивность потока информативных событий, в результате которых накапливаются наблюдения, формирующие выборку, имеет асимптотически обобщенное обратное гауссовское распределение.

В некоторых случаях статистические закономерности поведения наблюдаемых величин лучше описываются обобщенными гамма-распределениями с экспоненциально-степенными хвостами. Как показано в работе [5], асимптотическое поведение хвостов смесей нормальных законов в определенном смысле совпадает с аналогичным поведением хвостов смешивающих законов. Следовательно, аналогичная асимптотика должна быть присуща хвостам распределений интенсивностей потоков информативных событий. Действительно, как оказалось, обобщенные гамма-распределения заметно лучше согласуются с эмпирическими распределениями числа событий в книгах заявок в высокочастотных системах электронной торговли на финансовых рынках (P-значения при проверке согласия с помощью критерия хи-квадрат примерно равны $0{,}8$), нежели обобщенные обратные гауссовские распределения, для которых аналогичные P-значения примерно равны $0{,}2$ (см. рисунок). Поэтому поиск критериев сходимости к многомерным дисперсионным гамма-распределениям представляет собой весьма перспективную задачу.

Цель данной работы — уточнить результаты статьи [4] для ситуации, когда рассматриваемые статистики имеют вид случайных сумм. Хотя литература по асимптотическим свойствам случайных сумм случайных величин обширна, условия сходимости распределений сумм случайного числа независимых неодинаково распределенных многомерных случайных величин к многомерным дисперсионно-сдвиговым смесям нормальных законов и, в частности, к многомерным обобщенным гиперболическим или многомерным обобщенным дисперсионным гамма-распределениям еще не описаны. Настоящая работа восполняет этот пробел.

Введем обозначения, которые в дальнейшем будут использоваться без дополнительных комментариев. Пусть $m \in \mathbb{N}$. Векторы $\boldsymbol{x} = \left(x^{(1)}, \ldots, x^{(m)}\right)^\top$ считаются элементами m-мерного евклидова пространства $\mathbb{R}^m$, верхний индекс $\top$ означает транспонирование вектора или матрицы. Скалярное произведение в $\mathbb{R}^m$ будет обозначаться $\langle \cdot, \cdot \rangle$: $\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \boldsymbol{x}^\top \boldsymbol{y} = x^{(1)} y^{(1)} + \cdots + x^{(m)} y^{(m)}$. Как обычно, евклидова норма вектора $\boldsymbol{x}$ обозначается $\|\boldsymbol{x}\| = \langle \boldsymbol{x}, \boldsymbol{x} \rangle^{1/2}$. Если A — вещественная квадратная $(m \times m)$-матрица, то ее определитель обозначается $\det(A)$. Тождественная (единичная) $(m \times m)$-матрица обозначается $\boldsymbol{I}$. Чтобы различать число нуль и нулевой вектор, используются обозначения $0 \in \mathbb{R}$ и $\mathbf{0} = (0, \ldots, 0)^\top \in \mathbb{R}^m$.

Тот факт, что случайный вектор $\boldsymbol{X}$ имеет m-мерное нормальное распределение с вектором средних $\boldsymbol{a}$ и ковариационной матрицей Σ, будем обозначать $\boldsymbol{X} \sim \mathcal{N}_{\boldsymbol{a},\Sigma}$. Это означает, что существуют вектор $\boldsymbol{a} \in \mathbb{R}^m$ и неотрицательно



Гистограммы и графики подогнанных плотностей обобщенного гамма- (*а*) и обобщенного обратного гауссовского (*б*) распределения числа событий в книге заявок (фьючерсы РТС) за пятнадцатисекундные интервалы

определенная симметричная матрица Σ размерности $m \times m$, такие что характеристическая функция $\mathfrak{f}_{\boldsymbol{X}}$ вектора $\boldsymbol{X}$ имеет вид:

$$\mathfrak{f}_{\boldsymbol{X}}(\boldsymbol{t}) = \mathsf{E} \exp\{i\langle \boldsymbol{t}, \boldsymbol{X}\rangle\} = \exp\{i\boldsymbol{a}^{\top}\boldsymbol{t} - \frac{1}{2}\,\boldsymbol{t}^{\top}\Sigma\boldsymbol{t}\}\,, \;\; \boldsymbol{t} \in \mathbb{R}^m\,.$$

Плотность невырожденного нормального распределения $\mathcal{N}_{\boldsymbol{a},\Sigma}$ имеет вид:

$$\phi_{\boldsymbol{a},\Sigma}(\boldsymbol{x}) = \frac{1}{(2\pi)^{m/2}|\Sigma|^{1/2}} \exp\{-\frac{1}{2}\,(\boldsymbol{x}-\boldsymbol{a})^{\top}\,\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{a})\}\,, \;\; x \in \mathbb{R}^m\,,$$

где $|\Sigma|$ — определитель матрицы Σ, а Σ^{-1} — матрица, обратная к Σ. Одномерная стандартная нормальная функция распределения обозначается $\Phi(x)$,

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int\limits_{-\infty}^{x} e^{-y^2/2}\,dy, \;\; x \in \mathbb{R}\,.$$

Будем считать, что все случайные величины и векторы, упоминаемые ниже, заданы на одном и том же вероятностном пространстве $(\Omega, \mathfrak{F}, \mathsf{P})$. Символы $\mathfrak{B}_m$ и $\mathfrak{B}_+$ соответственно обозначают борелевские σ-алгебры подмножеств $\mathbb{R}_m$ и $\mathbb{R}_+ \equiv$ $\equiv [0, \infty)$. Символами $\stackrel{d}{=}$ и $\Longrightarrow$ соответственно будут обозначаться совпадение распределений и сходимость по распределению. Распределение случайного вектора $\boldsymbol{X}$ будет обозначаться $\mathcal{L}(\boldsymbol{X})$.

Семейство $\{\boldsymbol{X}_j\}_{j\in\mathbb{N}}$ $\mathbb{R}^m$-значных случайных векторов называется *слабо относительно компактным*, если каждая последовательность его элементов содержит подпоследовательность, сходящуюся по распределению. Иногда вместо *«слабая относительная компактность»* будет использоваться термин *«слабая компактность»*. В конечномерном случае, рассматриваемом в данной работе, слабая относительная компактность семейства $\{\boldsymbol{X}_j\}_{j\in\mathbb{N}}$ эквивалентна его плотности (tightness) (см., например, [6]):

$$\lim_{R\to\infty} \sup_{n\in\mathbb{N}} \mathsf{P}(\|\boldsymbol{X}_n\| > R) = 0\,.$$

Пусть $\left\{\boldsymbol{S}_{n,k} = \left(S_{n,k}^{(1)}, \ldots, S_{n,k}^{(m)}\right)^\top\right\}$, $n, k \in \mathbb{N}$, — последовательность серий $\mathbb{R}^m$-значных случайных векторов. Для $n, k \in \mathbb{N}$ пусть $\boldsymbol{a}_{n,k} = \left(a_{n,k}^{(1)}, \ldots\right.$ $\left.\ldots, a_{n,k}^{(m)}\right)^\top \in \mathbb{R}^m$ — неслучайные векторы и $b_{n,k} \in \mathbb{R}$ — положительные числа. Назначение векторов $\boldsymbol{a}_{n,k}$ и чисел $b_{n,k}$ — обеспечить слабую относительную компактность семейства случайных векторов $\left\{\boldsymbol{Y}_{n,k} \equiv b_{n,k}^{-1}\left(\boldsymbol{S}_{n,k} - \boldsymbol{a}_{n,k}\right)\right\}_{n,k\in\mathbb{N}}$, когда это требуется.

Рассмотрим семейство $\{N_n\}_{n\in\mathbb{N}}$ неотрицательных случайных величин, таких что при каждом $n, k \in \mathbb{N}$ случайные величины N_n независимы от случайных векторов $\boldsymbol{S}_{n,k}$. Особо заметим, что «построчная» независимость случайных векторов $\{\boldsymbol{S}_{n,k}\}_{k\geqslant 1}$ не требуется. Пусть $\boldsymbol{c}_n = \left(c_n^{(1)}, \ldots, c_n^{(m)}\right)^\top \in \mathbb{R}^m$ — неслучайные векторы и d_n — положительные числа, $n \in \mathbb{N}$. Наша цель — изучить асимптотическое поведение случайных векторов $\boldsymbol{Z}_n \equiv d_n^{-1}\left(\boldsymbol{S}_{n,N_n} - \boldsymbol{c}_n\right)$ при $n \to$ $\to \infty$, уделив особое внимание ситуации, в которой предельные распределения для $\boldsymbol{Z}_n$ имеют вид дисперсионно-сдвиговых смесей нормальных законов.

Характеристическую функцию случайного вектора $\boldsymbol{Y}_{n,k}$ обозначим $\mathfrak{h}_{n,k}(\boldsymbol{t})$, $\boldsymbol{t} \in \mathbb{R}^m$. Пусть $\boldsymbol{Y}$ — $\mathbb{R}^m$-мерный случайный вектор, характеристическая функция которого будет обозначаться $\mathfrak{h}(\boldsymbol{t})$, $\boldsymbol{t} \in \mathbb{R}^m$. Введем случайные величины $U_n =$ $= d_n^{-1} b_{n,N_n}$. Пусть $\boldsymbol{V}_n = \left(V_n^{(1)}, \ldots, V_n^{(m)}\right)^\top$, где $V_n^{(k)} = d_n^{-1}\left(a_{n,N_n}^{(k)} - c_n^{(k)}\right)$ — k-я компонента случайного вектора $d_n^{-1}(\boldsymbol{a}_{n,N_n} - \boldsymbol{c}_n)$. В дальнейшем символ $\boldsymbol{W}_n$ будет обозначать $(m+1)$-мерный случайный вектор $\boldsymbol{W}_n =$ $= \left(U_n, \boldsymbol{V}_n^\top\right)^\top = \left(U_n, V_n^{(1)}, \ldots, V_n^{(m)}\right)^\top$.

Для расстояния полной вариации между распределениями случайных векторов $\boldsymbol{X}$ и $\boldsymbol{Y}$ будем использовать обозначение $\Delta(\boldsymbol{X}, \boldsymbol{Y})$:

$$\Delta(\boldsymbol{X}, \boldsymbol{Y}) = \int\limits_{\mathbb{R}^m} |\mathcal{L}(\boldsymbol{X}) - \mathcal{L}(\boldsymbol{Y})|(d\boldsymbol{x})\,.$$

Понятие многомерной дисперсионно-сдвиговой смеси нормальных законов было введено в работах О. Е. Барндорфф-Нильсена и его коллег [7–9]. Напомним, распределение $\mathbb{R}^m$-значного случайного вектора $\boldsymbol{X}$ является многомерной дисперсионно-сдвиговой смесью нормальных законов, если $\boldsymbol{X} \stackrel{d}{=} \boldsymbol{a} + U\boldsymbol{b} + \sqrt{U}A\boldsymbol{Y}$, где $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^m$; A — вещественная $(m \times m)$-матрица, такая что матрица $\Sigma \equiv AA^\top$ положительно определена; $\boldsymbol{Y}$ — случайный вектор со стандартным m-мерным нормальным распределением $\mathcal{N}_{\boldsymbol{0},\boldsymbol{I}}$; U — неотрицательная случайная величина, независимая от $\boldsymbol{Y}$. Эквивалентно, будем говорить, что вероятностная мера F на $(\mathbb{R}^m, \mathfrak{B}_m)$ является многомерной дисперсионно-сдвиговой смесью нормальных законов, если

$$F(d\boldsymbol{x}) = \int\limits_0^\infty \mathcal{N}_{\boldsymbol{b}+z\boldsymbol{a},\, z\Sigma}(d\boldsymbol{x})G(dz)\,, \tag{1}$$

где смешивающим распределением G является вероятностная мера на $(\mathbb{R}_+, \mathfrak{B}_+)$. Этот факт будем записывать в виде $F = \mathcal{N}_{\boldsymbol{b}+z\boldsymbol{a},\, z\Sigma} \circ G$. Обратим внимание, что в соотношении (1) смешивание происходит одновременно и по параметрам сдвига, и по параметрам масштаба, но так как эти параметры в (1) связаны жесткой зависимостью, при которой параметры положения (*сдвига*) смешиваемых нормальных законов пропорциональны их *дисперсиям*, то фактически смесь (1) является однопараметрической. Именно поэтому смеси вида (1) называются *дисперсионно-сдвиговыми*.

Если в соотношении (1) функция распределения $G(z) = P_{\mathrm{GIG}}(z)$, где $P_{\mathrm{GIG}}(z)$ — функция распределения обобщенного обратного гауссовского закона [10], соответствующая плотности

$$p_{\mathrm{GIG}}(x; \nu, \mu, \lambda) = \frac{\lambda^{\nu/2}}{2\mu^{\nu/2}K_\nu\left(\sqrt{\mu\lambda}\right)}\, x^{\nu-1} \exp\left\{-\frac{1}{2}\left(\frac{\mu}{x} + \lambda x\right)\right\}\,,\quad x > 0\,,$$

то соотношение (1) задает *многомерное обобщенное гиперболическое распределение* [7–9]. Здесь $\nu \in \mathbb{R}$,

$$\mu > 0,\ \lambda \geqslant 0, \text{ если } \nu < 0\,;$$
$$\mu > 0,\ \lambda > 0, \text{ если } \nu = 0\,;$$
$$\mu \geqslant 0,\ \lambda > 0, \text{ если } \nu > 0\,;$$

$K_\nu(z)$ — модифицированная бесселева функция третьего рода порядка ν,

$$K_\nu(z) = \frac{1}{2}\int\limits_0^\infty y^{\nu-1}\exp\left\{-\frac{z}{2}\left(y+\frac{1}{y}\right)\right\}dy\,,\quad z\in\mathbb{C}\,,\ \operatorname{Re} z>0\,.$$

Если в соотношении (1) функция распределения $G(z) = P_{\mathrm{GG}}(z)$, где $P_{\mathrm{GG}}(z)$ — функция обобщенного гамма-распределения [11], соответствующая плотности

$$p_{\mathrm{GG}}(x;\nu,\kappa,\delta) = \begin{cases} \dfrac{|\nu|}{\delta^{k|\nu|}\Gamma(\kappa)}x^{\kappa\nu-1}\exp\left\{-\dfrac{x^\nu}{\delta^{|\nu|}}\right\}, & x\geqslant 0\,;\\ 0\,, & x<0\,, \end{cases}$$

то соотношение (1) задает *многомерное обобщенное дисперсионное гамма-распределение* [3, 12]. Здесь параметры $\nu\in\mathbb{R}$, κ, $\delta\in\mathbb{R}^+$ отвечают соответственно за *степень, форму и масштаб*, $\Gamma(\kappa)$ — эйлерова гамма-функция:

$$\Gamma(\kappa) = \int\limits_0^\infty x^{\kappa-1}e^{-x}\,dx\,.$$

2 Условия сходимости распределений многомерных случайных последовательностей с независимыми случайными индексами

Чтобы получить нетривиальные предельные распределения для $\boldsymbol{Z}_n$ в схеме серий, рассматриваемой в данном разделе, потребуется следующее дополнительное *условие согласованности*: для любого $T\in(0,\infty)$

$$\lim_{n\to\infty}\mathsf{E}\sup_{\|\boldsymbol{t}\|\leqslant T}|\mathfrak{h}_{n,N_n}(\boldsymbol{t})-\mathfrak{h}(\boldsymbol{t})| = 0\,. \tag{2}$$

Несложно убедиться, что условие согласованности (2) вытекает из условия

$$\lim_{n\to\infty}\mathsf{E}\Delta(\boldsymbol{Y}_{n,N_n},\boldsymbol{Y}) = 0\,.$$

Действительно,

$$\mathsf{E}\sup_{\|\boldsymbol{t}\|\leqslant T}|\mathfrak{h}_{n,N_n}(\boldsymbol{t})-\mathfrak{h}(\boldsymbol{t})| = \sum_{k=1}^\infty \mathsf{P}(N_n=k)\sup_{\|\boldsymbol{t}\|\leqslant T}|\mathfrak{h}_{n,k}(\boldsymbol{t})-\mathfrak{h}(\boldsymbol{t})| =$$

$$= \sum_{k=1}^\infty \mathsf{P}(N_n=k)\sup_{\|\boldsymbol{t}\|\leqslant T}\left|\int\limits_{\mathbb{R}^m} e^{i\langle\boldsymbol{t},\boldsymbol{x}\rangle}\mathcal{L}(\boldsymbol{Y}_{n,k})(d\boldsymbol{x}) - \int\limits_{\mathbb{R}^m} e^{i\langle\boldsymbol{t},\boldsymbol{x}\rangle}\mathcal{L}(\boldsymbol{Y})(d\boldsymbol{x})\right| =$$

$$= \sum_{k=1}^{\infty} \mathsf{P}(N_n = k) \sup_{\|\boldsymbol{t}\| \leqslant T} \left| \int\limits_{\mathbb{R}^m} e^{i\langle \boldsymbol{t}, \boldsymbol{x} \rangle} \left[\mathcal{L}(\boldsymbol{Y}_{n,k}) - \mathcal{L}(\boldsymbol{Y}) \right] (d\boldsymbol{x}) \right| \leqslant$$

$$\leqslant \sum_{k=1}^{\infty} \mathsf{P}(N_n = k) \int\limits_{\mathbb{R}^m} \left| \mathcal{L}(\boldsymbol{Y}_{n,k}) - \mathcal{L}(\boldsymbol{Y}) \right| (d\boldsymbol{x}) = \mathsf{E}\Delta(\boldsymbol{Y}_{n,N_n}, \boldsymbol{Y}) \,.$$

Следуя работе [4], для фиксированных случайных векторов $\boldsymbol{Z}$ и $\boldsymbol{Y}$ с характеристическими функциями $f(\boldsymbol{t})$ и $\mathfrak{h}(\boldsymbol{t})$ введем множество $\mathcal{W}(\boldsymbol{Z}|\boldsymbol{Y})$, содержащее все $(m+1)$-мерные случайные векторы $\boldsymbol{W} = (U,\, \boldsymbol{V}^\top)^\top$ с $U \in \mathbb{R}$ и $\boldsymbol{V} \in \mathbb{R}^m$, такие что характеристическая функция $f(\boldsymbol{t})$ может быть представлена в виде:

$$f(\boldsymbol{t}) = \mathsf{E}\mathfrak{h}(U\boldsymbol{t}) e^{i\langle \boldsymbol{t}, \boldsymbol{V} \rangle} \,, \quad \boldsymbol{t} \in \mathbb{R}^m \,, \tag{3}$$

и $\mathsf{P}(U \geqslant 0) = 1$. Какими бы ни были случайные векторы $\boldsymbol{Z}$ и $\boldsymbol{Y}$, множество $\mathcal{W}(\boldsymbol{Z}|\boldsymbol{Y})$ всегда не пусто, поскольку оно очевидно содержит вектор $(0, \boldsymbol{Z}^\top)^\top$. Множество $\mathcal{W}(\boldsymbol{Z}|\boldsymbol{Y})$ может содержать более одного элемента. Несложно видеть, что представление (3) эквивалентно тому, что $\boldsymbol{Z} \stackrel{d}{=} U\boldsymbol{Y} + \boldsymbol{V}$.

Пусть $\Lambda(\boldsymbol{X}_1, \boldsymbol{X}_2)$ — любая вероятностная метрика, метризующая слабую сходимость в пространстве $(m+1)$-мерных случайных векторов. Примером такой метрики является расстояние Леви–Прохорова (см., например, [13, 14]).

Теорема 1 [4]. *Пусть семейство случайных величин $\{U_n\}_{n \in \mathbb{N}}$ слабо относительно компактно. Предположим, что выполнено условие согласованности* (2). *Тогда слабая сходимость нормированных случайных векторов с независимыми случайными индексами*

$$\boldsymbol{Z}_n \Longrightarrow \boldsymbol{Z} \quad (n \to \infty) \tag{4}$$

к некоторому случайному вектору $\boldsymbol{Z}$ имеет место при некоторых $\boldsymbol{c}_n \in \mathbb{R}^m$ тогда и только тогда, когда существует слабо относительно компактная последовательность случайных векторов $\boldsymbol{W}_n^* \equiv (U_n^*, (\boldsymbol{V}_n^*)^\top)^\top \in \mathcal{W}(\boldsymbol{Z}|\boldsymbol{Y})$, $n \in \mathbb{N}$, такая что

$$\lim_{n \to \infty} \Lambda(\boldsymbol{W}_n^*, \boldsymbol{W}_n) = 0 \,.$$

3 Общая теорема о сходимости распределений сумм случайного числа независимых неодинаково распределенных многомерных случайных величин

Рассмотрим последовательность серий $\{\boldsymbol{X}_{n,j}\}_{j \geqslant 1}$, $n \in \mathbb{N}$, независимых, но не обязательно одинаково распределенных в каждой серии случайных величин. Для $n, k \in \mathbb{N}$ положим

$$\boldsymbol{S}_{n,k} = \boldsymbol{X}_{n,1} + \cdots + \boldsymbol{X}_{n,k} \,.$$

Как будет показано в данном разделе, если $\boldsymbol{S}_{n,k}$ — сумма независимых случайных величин, то условие слабой относительной компактности последовательности $\{U_n\}_{n\in\mathbb{N}}$, использованное выше для описания общего случая, может быть заменено, вообще говоря, менее ограничительным условием слабой относительной компактности семейства $\{\boldsymbol{Y}_{n,k}\}_{n,k\in\mathbb{N}}$. Действительно, пусть, к примеру, нормы случайных векторов $\boldsymbol{S}_{n,k}$ имеют моменты некоторого порядка $\delta > 0$. Тогда если положить $b_{n,k} = (\mathsf{E}\|\boldsymbol{S}_{n,k} - \boldsymbol{a}_{n,k}\|^{\delta})^{1/\delta}$, то по неравенству Маркова

$$\lim_{R\to\infty} \sup_{n,k\in\mathbb{N}} \mathsf{P}\left(\|\boldsymbol{Y}_{n,k}\| > R\right) \leqslant \lim_{R\to\infty} \frac{1}{R^{\,\delta}} = 0,$$

т. е. семейство $\{\boldsymbol{Y}_{n,k}\}_{n,k\in\mathbb{N}}$ слабо относительно компактно.

Теорема 2. *Пусть случайные векторы $\boldsymbol{S}_{n,k}$ имеют вид* (5). *Предположим, что семейство случайных векторов $\{\boldsymbol{Y}_{n,k}\}_{n,k\in\mathbb{N}}$ слабо относительно компактно, причем выполнено условие согласованности* (2). *Тогда сходимость* (4) *нормированных многомерных случайных сумм $\boldsymbol{Z}_n$ к некоторому случайному вектору Z имеет место с некоторыми $\boldsymbol{c}_n \in \mathbb{R}^m$ в том и только в том случае, когда существует слабо относительно компактная последовательность случайных векторов $\boldsymbol{W}_n^* \equiv (U_n^*,\, (\boldsymbol{V}_n^*)^{\top})^{\top} \in \mathcal{W}(\boldsymbol{Z}|\boldsymbol{Y})$, $n \in \mathbb{N}$, такая что*

$$\lim_{n\to\infty} \Lambda(\boldsymbol{W}_n^*,\, \boldsymbol{W}_n) = 0\,. \tag{5}$$

Доказательство. Достаточно показать, что в рассматриваемой ситуации условие (4) влечет слабую относительную компактность семейства $\{U_n\}_{n\in\mathbb{N}}$ и сослаться на приведенную выше теорему 1. В дальнейшем симметризацию случайного вектора $\boldsymbol{X}$ будем обозначать $\boldsymbol{X}^{(s)}$, $\boldsymbol{X}^{(s)} = \boldsymbol{X} - \boldsymbol{X}'$, где $\boldsymbol{X}'$ — случайный вектор, независимый от $\boldsymbol{X}$ и такой, что $\boldsymbol{X}' \stackrel{d}{=} \boldsymbol{X}$. Для $q \in (0,1)$ пусть $\ell_n(q)$ — точная нижняя грань q-квантилей случайной величины N_n, $n \in \mathbb{N}$. Предположим, что при каждом n случайные элементы $N_n, X_{n,1}^{(s)}, X_{n,2}^{(s)}, \ldots$ независимы в совокупности и введем случайные векторы

$$\boldsymbol{Q}_n = \frac{1}{d_n} \sum_{j=1}^{N_n} \boldsymbol{X}_{n,j}^{(s)}\,,\ \ n \in \mathbb{N}\,.$$

Используя элементарно доказываемое неравенство симметризации

$$\mathsf{P}\left(\boldsymbol{X}^{(s)} \geqslant R\right) \leqslant 2\,\mathsf{P}\left(\|\boldsymbol{X} - \boldsymbol{a}\| \geqslant \frac{R}{2}\right),$$

справедливое для любого случайного вектора $\boldsymbol{X}$ и любых $\boldsymbol{a} \in \mathbb{R}^m$, $R > 0$ (см., например, [15]), для любых $R > 0$ и $n \in \mathbb{N}$ получаем

$$\mathsf{P}(\|\boldsymbol{Q}_n\| \geqslant R) = \sum_{k=1}^{\infty} \mathsf{P}(N_n = k)\mathsf{P}\left(\left\|\frac{1}{d_n}\sum_{j=1}^{k}\boldsymbol{X}_{n,j}^{(s)}\right\| \geqslant R\right) \leqslant$$

$$\leqslant 2\sum_{k=1}^{\infty} \mathsf{P}(N_n = k)\mathsf{P}\left(\left\|\frac{1}{d_n}\left(\sum_{j=1}^{k}\boldsymbol{X}_{n,j} - \boldsymbol{c}_n\right)\right\| \geqslant \frac{R}{2}\right) =$$

$$= 2\,\mathsf{P}\left(\left\|\frac{1}{d_n}\left(\sum_{j=1}^{N_n}\boldsymbol{X}_{n,j} - \boldsymbol{c}_n\right)\right\| \geqslant \frac{R}{2}\right).$$

Поэтому

$$\lim_{R\to\infty}\sup_n \mathsf{P}(\|\boldsymbol{Q}_n\| \geqslant R) \leqslant 2\lim_{R\to\infty}\sup_n \mathsf{P}\left(\left\|\frac{1}{d_n}\left(\sum_{j=1}^{N_n}\boldsymbol{X}_{n,j} - \boldsymbol{c}_n\right)\right\| \geqslant \frac{R}{2}\right) = 0$$

в силу (4). Следовательно, последовательность случайных векторов $\{\boldsymbol{Q}_n\}_{n\in\mathbb{N}}$ слабо относительно компактна.

Теперь докажем, что

$$C(q) \equiv \sup_n \frac{b_{n,\ell_n(q)}}{d_n} < \infty \tag{6}$$

для каждого $q \in (0,1)$. С этой целью используем многомерный аналог неравенства Леви

$$\mathsf{P}\left(\max_{1\leqslant m\leqslant k}\left\|\sum_{j=1}^{m}\boldsymbol{X}_j^{(s)}\right\| \geqslant R\right) \leqslant 2\,\mathsf{P}\left(\left\|\sum_{j=1}^{k}\boldsymbol{X}_j^{(s)}\right\| \geqslant R\right),$$

справедливого для любых независимых случайных векторов $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_k$ и любого $R > 0$ (см., например, [16, с. 455]), с помощью которого для произвольного $q \in \in (0,1)$ получим следующую цепочку неравенств:

$$2\,\mathsf{P}(\|\boldsymbol{Q}_n\| \geqslant R) = 2\sum_{k=1}^{\infty} \mathsf{P}(N_n = k)\mathsf{P}\left(\left\|\frac{1}{d_n}\sum_{j=1}^{k}\boldsymbol{X}_{n,j}^{(s)}\right\| \geqslant R\right) \geqslant$$

$$\geqslant 2\sum_{k\geq\ell_n(q)} \mathsf{P}(N_n = k)\mathsf{P}\left(\left\|\frac{1}{d_n}\sum_{j=1}^{k}\boldsymbol{X}_{n,j}^{(s)}\right\| \geqslant R\right) \geqslant$$

$$\geqslant \sum_{k\geqslant \ell_n(q)} \mathsf{P}(N_n = k)\mathsf{P}\left(\left\|\frac{1}{d_n}\sum_{j=1}^{\ell_n(q)} \boldsymbol{X}_{n,j}^{(s)}\right\| \geqslant R\right) =$$

$$= \mathsf{P}(N_n \geqslant \ell_n(q))\,\mathsf{P}\left(\left\|\frac{1}{d_n}\sum_{j=1}^{\ell_n(q)} \boldsymbol{X}_{n,j}^{(s)}\right\| \geqslant R\right) = (1-q)\mathsf{P}\left(\left\|\frac{1}{d_n}\sum_{j=1}^{\ell_n(q)} \boldsymbol{X}_{n,j}^{(s)}\right\| \geqslant R\right).$$

Таким образом, слабая компактность семейства случайных векторов $\{\boldsymbol{Q}_n\}_{n\in\mathbb{N}}$, установленная выше, при каждом $q \in (0,1)$ влечет слабую компактность семейства $\{\boldsymbol{Q}_n^{(q)}\}_{n\in\mathbb{N}}$, где

$$\boldsymbol{Q}_n^{(q)} = \frac{1}{d_n}\sum_{j=1}^{\ell_n(q)} \boldsymbol{X}_{n,j}^{(s)}\,,\ \ n \in \mathbb{N}\,.$$

Предположим, что (6) не имеет места. В таком случае существуют некоторое $q^* \in (0,1)$ и некоторая последовательность $\mathcal{N}$ натуральных чисел, такие что

$$\frac{b_{n,\ell_n(q^*)}}{d_n} \longrightarrow \infty\,,\ \ n \to \infty\,,\ \ n \in \mathcal{N}\,. \tag{7}$$

В соответствии с условиями теоремы семейство случайных векторов $\{\boldsymbol{Y}_{n,k} = (\boldsymbol{S}_{n,k} - \boldsymbol{a}_{n,k})/b_{n,k}\}_{n,k\in\mathbb{N}}$ слабо относительно компактно. Поэтому можно выбрать подпоследовательность $\mathcal{N}_1 \subseteq \mathcal{N}$ так, чтобы

$$\boldsymbol{Y}_{n,\ell_n(q^*)} = \frac{1}{b_{n,\ell_n(q^*)}}\left(\sum_{j=1}^{\ell_n(q^*)} \boldsymbol{X}_{n,j} - \boldsymbol{a}_{n,\ell_n(q^*)}\right) \Longrightarrow \boldsymbol{Y}\,,\ \ n \to \infty,\ n \in \mathcal{N}_1\,, \tag{8}$$

где $\boldsymbol{Y}$ — некоторый случайный вектор. Обозначим r-е компоненты случайных векторов $\boldsymbol{Q}_n^{(q^*)}$, $\boldsymbol{Y}_{n,\ell_n(q^*)}^{(s)}$ и $\boldsymbol{Y}^{(s)}$ через $Q_{n;r}^{(q^*)}$, $Y_{n,\ell_n(q^*);r}^{(s)}$ и $Y_r^{(s)}$ соответственно:

$$\boldsymbol{Q}_n^{(q^*)} = \left(Q_{n;1}^{(q^*)}, \ldots, Q_{n;m}^{(q^*)}\right)^\top;$$

$$\boldsymbol{Y}_{n,\ell_n(q^*)}^{(s)} = \left(Y_{n,\ell_n(q^*);1}^{(s)}, \ldots, Y_{n,\ell_n(q^*);m}^{(s)}\right)^\top;$$

$$\boldsymbol{Y}^{(s)} = \left(Y_1^{(s)}, \ldots, Y_m^{(s)}\right)^\top.$$

Тогда из (7) и (8) вытекает, что для любого $R \in \mathbb{R}$ и любого $r \in \{1, \ldots, m\}$

$$\mathsf{P}(Q_{n;r}^{(q^*)} \leqslant R) = \mathsf{P}\left(Y_{n,\ell_n(q^*);r}^{(s)} \leqslant \frac{d_n R}{b_{n,\ell_n(q^*)}}\right) \longrightarrow \mathsf{P}(Y_r^{(s)} \leqslant 0) \geqslant \frac{1}{2}\,,$$

$$n \to \infty\,,\ \ n \in \mathcal{N}_1\,,$$

что противоречит слабой относительной компактности семейства $\{\boldsymbol{Q}_n^{(q^*)}\}$, установленной выше. Таким образом, условие (6) выполнено для любого $q \in (0,1)$.

Несложно убедиться, что $N_n \stackrel{d}{=} \ell_n(\upsilon)$, где υ — случайная величина с равномерным распределением на $(0,1)$. Поэтому с учетом (6) для любых $R \geqslant 0$ и $n \in \mathbb{N}$ имеем

$$\mathsf{P}(U_n \geqslant R) = \mathsf{P}\left(\frac{b_{n,\ell_n(\upsilon)}}{d_n} \geqslant R\right) = \int\limits_0^1 \mathbb{I}\left(\frac{b_{n,\ell_n(q)}}{d_n} \geqslant R\right) dq \leqslant$$

$$\leqslant \int\limits_0^1 \mathbb{I}\left(C(q) \geqslant R\right) dq = \mathsf{P}\left(C\left(\upsilon\right) \geqslant R\right),$$

так что

$$\lim_{R\to\infty} \sup_n \mathsf{P}\left(U_n \geqslant R\right) = \lim_{R\to\infty} \mathsf{P}\left(C(\upsilon) \geqslant R\right) = 0,$$

т. е. последовательность $\{U_n\}_{n\in\mathbb{N}}$ слабо компактна. Тем самым теорема доказана.

4 Сходимость распределений сумм случайного числа независимых неодинаково распределенных многомерных случайных величин к дисперсионно-сдвиговым смесям, в частности к многомерным обобщенным дисперсионным гамма-распределениям

Изучим вопрос о том, при каких условиях предельными распределениями для сумм случайного числа независимых неодинаково распределенных многомерных случайных величин могут быть дисперсионно-сдвиговые смеси нормальных законов и, в частности, многомерные GVG-распределения.

Напомним, что характеристическая функция m-мерного нормального распределения с нулевым вектором математических ожиданий и ковариационной матрицей Σ имеет вид:

$$\mathfrak{f}_{\mathbf{0},\Sigma}(\boldsymbol{t}) = \exp\left\{-\frac{1}{2}\boldsymbol{t}^\top \Sigma \boldsymbol{t}\right\}, \quad \boldsymbol{t} \in \mathbb{R}^m.$$

Предположим, что суммы $\boldsymbol{S}_{n,k}$ неслучайного числа случайных векторов асимптотически нормальны в том смысле, что существует положительно определенная симметричная $(m \times m)$-матрица Σ такая, что для любого $T \in (0,\infty)$

$$\lim_{n\to\infty} \mathsf{E} \sup_{\|\boldsymbol{t}\|\leqslant T} \left|\mathfrak{h}_{n,N_n}(\boldsymbol{t}) - \exp\left\{-\frac{1}{2}\,\boldsymbol{t}^\top \Sigma \boldsymbol{t}\right\}\right| = 0, \tag{9}$$

где $\mathfrak{h}_{n,k}(\boldsymbol{t})$ — характеристическая функция нормированных и центрированных сумм $\boldsymbol{Y}_{n,k} = b_{n,k}^{-1}(\boldsymbol{S}_{n,k} - \boldsymbol{a}_{n,k})$.

Предположим, что центрирующие векторы $\boldsymbol{a}_{n,k}$ и $\boldsymbol{c}_n$ в определенном смысле пропорциональны нормирующим константам $b_{n,k}$ и d_n. А именно: предположим,

что существуют векторы $\boldsymbol{a}_n \in \mathbb{R}^m$ и $\boldsymbol{b}_n \in \mathbb{R}^m$ такие, что для всех $n, k \in \mathbb{N}$ справедливы соотношения:

$$\boldsymbol{a}_{n,k} = d_n^{-1} b_{n,k}^2 \boldsymbol{a}_n\,; \quad \boldsymbol{c}_n = d_n \boldsymbol{b}_n\,, \tag{10}$$

причем существуют пределы:

$$\boldsymbol{a} = \lim_{n\to\infty} \boldsymbol{a}_n\,; \quad \boldsymbol{b} = \lim_{n\to\infty} \boldsymbol{b}_n\,. \tag{11}$$

Теорема 3. *Предположим, что семейство случайных векторов* $\{\boldsymbol{Y}_{n,k}\}_{n,k\in\mathbb{N}}$ *слабо относительно компактно, центрирующие векторы имеют вид* (10) *и удовлетворяют условию* (11). *Предположим, что суммы* $\boldsymbol{S}_{n,k}$ *неслучайного числа случайных векторов асимптотически нормальны в смысле* (9). *Тогда распределения центрированных и нормированных случайных сумм сходятся к распределению некоторого случайного вектора* $\boldsymbol{Z}$:

$$\boldsymbol{Z}_n \Longrightarrow \boldsymbol{Z} \ \ (n \to \infty)$$

тогда и только тогда, когда существует функция распределения G, *такая что* $G(0) = 0$, *распределение* F *случайного вектора* $\boldsymbol{Z}$ *имеет вид*:

$$F = \mathcal{N}_{\boldsymbol{b}+z\boldsymbol{a},\, z\Sigma} \circ G$$

и

$$\mathsf{P}(U_n^2 < x) \Longrightarrow G(x) \ \ (n \to \infty)\,.$$

Доказательство. Выведем теорему 3 в качестве следствия теоремы 2. Во-первых, заметим, что условие (9) — это не что иное, как условие согласованности (2) с $\mathfrak{h}(\boldsymbol{t}) \equiv \exp\{-(1/2)\boldsymbol{t}^\top \Sigma \boldsymbol{t}\}$.

Во-вторых, заметим, что очевидным образом каждая одномерная проекция многомерной дисперсионно-сдвиговой смеси нормальных законов является одномерной дисперсионно-сдвиговой смесью нормальных законов. Недавно в работе [2] было показано, что одномерные дисперсионно-сдвиговые смеси нормальных законов идентифицируемы, т. е. если $a \in \mathbb{R}$, $\sigma > 0$, $\mathsf{P}(Y < x) \equiv \Phi(x)$, а U_1 и U_2 — две неотрицательные случайные величины, то тождество

$$\mathsf{E}\Phi\left(\frac{x - aU_1^2}{\sigma U_1}\right) \equiv \mathsf{E}\Phi\left(\frac{x - aU_2^2}{\sigma U_2}\right)$$

влечет, что $U_1 \stackrel{d}{=} U_2$. Следовательно, множество $\mathcal{W}(\boldsymbol{Z}|\boldsymbol{Y})$ содержит не более одного вектора вида $\boldsymbol{W} = \left(\sigma U,\, (U^2\boldsymbol{a} + \boldsymbol{b})^\top\right)^\top$. Это означает, что в рассматриваемом случае условие (5) сводится к (10). Теорема доказана.

В качестве элементарного следствия теоремы 3 получается утверждение, описывающее условия сходимости распределений случайных сумм к дисперсионным гамма-распределениям или обобщенным гиперболическим распределениям.

Теорема 4. *Предположим, что семейство случайных векторов* $\{\boldsymbol{Y}_{n,k}\}_{n,k\in\mathbb{N}}$ *слабо относительно компактно, центрирующие векторы имеют вид* (10) *и удовлетворяют условию* (11). *Предположим, что суммы* $\boldsymbol{S}_{n,k}$ *неслучайного числа случайных векторов асимптотически нормальны в смысле* (9).

(*a*) *Распределения центрированных и нормированных случайных сумм сходятся к многомерному обобщенному дисперсионному гамма-распределению* $F(d\boldsymbol{x};\boldsymbol{a},\boldsymbol{b},\Sigma,\nu,\kappa,\delta)$ *при* $n\to\infty$ *тогда и только тогда, когда*

$$\mathsf{P}(U_n^2<x)\Longrightarrow P_{\mathrm{GG}}(x;\nu,\kappa,\delta)\ \ (n\to\infty)\,.$$

(*b*) *Распределения центрированных и нормированных случайных сумм сходятся к многомерному обобщенному гиперболическому распределению* $F(d\boldsymbol{x};\boldsymbol{a},\boldsymbol{b},\Sigma,\nu,\mu,\lambda)$ *при* $n\to\infty$ *тогда и только тогда, когда*

$$\mathsf{P}(U_n^2<x)\Longrightarrow P_{\mathrm{GIG}}(x;\nu,\mu,\lambda)\ \ (n\to\infty)\,.$$

Литература

1. *Гнеденко Б. В.* Об оценке неизвестных параметров распределения при случайном числе независимых наблюдений // Тр. Тбилисского математического института, 1989. Т. 92. С. 146–150.
2. *Королев В. Ю.* Обобщенные гиперболические законы как предельные для распределений случайных сумм // Теория вероятностей и ее применения, 2013. Т. 58. Вып. 1. С. 117–132.
3. *Закс Л. М., Королев В. Ю.* Обобщенные дисперсионные гамма-распределения как предельные для случайных сумм // Информатика и её применения, 2013. Т. 7. Вып. 1. С. 105–115.
4. *Korolev V. Yu., Zeifman A. I.* On convergence of the distributions of statistics constructed from samples with random sizes to normal variance-mean mixtures // J. Stat. Planning Inference, 2015 (in press). arXiv:1410.1518v1 [math.PR].
5. *Антонов С. Н., Кокшаров С. Н.* Об асимптотическом поведении хвостов масштабных смесей нормальных распределений // Статистические методы оценивания и проверки гипотез. — Пермь: ПГНИУ, 2006. С. 90–105.
6. *Лоэв М.* Теория вероятностей / Пер. с англ. — М.: Мир, 1962. 720 с. (*Loève M.* Probability theory. — Princeton, NJ, USA: D Van Nostrand, 1955. 702 p.)
7. *Barndorff-Nielsen O. E.* Exponentially decreasing distributions for the logarithm of particle size // Proc. Roy. Soc. Lond. A, 1977. Vol. 353. P. 401–419.
8. *Barndorff-Nielsen O. E.* Hyperbolic distributions and distributions of hyperbolae // Scand. J. Stat., 1978. Vol. 5. P. 151–157.
9. *Barndorff-Nielsen O. E., Kent J., Sørensen M.* Normal variance-mean mixtures and z-distributions // Int. Stat. Rev., 1982. Vol. 50. No. 2. P. 145–159.
10. *Jørgensen B.* Statistical properties of the generalized inverse Gaussian distribution. Lecture notes in statistics ser. — Berlin: Springer, 1982. Vol. 9. 188 p.
11. *Stacy E. W.* A generalization of the gamma distribution // Ann. Math. Stat., 1962. Vol. 33. P. 1187–1192.
12. *Королев В. Ю., Корчагин А. Ю., Зейфман А. И.* О сходимости распределений статистик, построенных по выборкам случайного объема, к многомерным обобщенным дисперсионным гамма-распределениям // Докл. Акад. наук, 2015 (в печати). Т. 462. Вып. 4.

13. Billingsley, P. Convergence of probability measures. — New York, NY, USA: Wiley, 1968. 353 p.
14. *Zolotarev V. M.* Modern theory of summation of random variables. — Utrecht: VSP, 1997. 412 p.
15. *Круглов В. М.* Сходимость числовых характеристик сумм независимых случайных величин со значениями в гильбертовом пространстве // Теория вероятностей и ее применения, 1973. Т. 18. Вып. 4. С. 734–752.
16. *Гихман И. И., Скороход А. В.* Теория случайных процессов. Т. 1. — М.: Наука, 1971. 665 с.

Поступила в редакцию 02.03.15

ON CONVERGENCE OF RANDOM SUMS OF INDEPENDENT RANDOM VECTORS TO MULTIVARIATE GENERALIZED VARIANCE-GAMMA DISTRIBUTIONS

A. Yu. Korchagin

Faculty of Computational Mathematics and Cybernetics, M. V. Lomonosov Moscow State University, 1-52 Leninskiye Gory, GSP-1, Moscow 119991, Russian Federation

Abstract: The purpose of this work is to describe the conditions for convergence of the distributions for sums of a random number of independent not necessarily identically distributed multivariate random variables to multivariate normal variance-mean mixtures, in particular, to multivariate generalized variance-gamma distributions.

Keywords: random sum; multivariate normal variance-mean mixture; multivariate generalized hyperbolic distribution; multivariate generalized variance-gamma distribution; generalized inverse Gaussian distribution; generalized gamma distribution

DOI: 10.14357/08696527150108

Acknowledgments

This work was financially supported by the Russian Science Fundation (grant No. 14-11-00397).

References

1. Gnedenko, B. V. 1989. Ob otsenke neizvestnykh parametrov raspredeleniya pri sluchaynom chisle nezavisimykh nablyudeniy [On estimate of unknown distribution parameters in case of random number of independent values]. *Tbilisi Mathematical Institute* 92:146–150.
2. Korolev, V. Yu. 2013. Obobshchennye giperbolicheskie raspredeleniya kak predel'nye dlya sluchaynykh summ [Generalized hyperbolic distributions as limiting for random sums] *Theory Probab. Appl.* 58(1):117–132.

3. Zaks, L. M., and V. Yu. Korolev. 2013. Obobshchennye dispersionnye gamma-raspredeleniya kak predel'nye dlya sluchaynykh summ [Generalized variance gamma distributions as limiting for random sums]. *Informatika i ee Primeneniya — Inform. Appl.* 7(1):105–115.
4. Korolev, V. Yu., and A. I. Zeifman. 2015 (in press). On convergence of the distributions of statistics constructed from samples with random sizes to normal variance-mean mixtures. *J. Stat. Planning Inference*. Available at: arXiv:1410.1518v1 [math.PR].
5. Antonov, S. N., and S. N. Koksharov. 2006. Ob asimptoticheskom povedenii khvostov masshtabnykh smesey normalnykh raspredeleniy [On asimptotic behavior of tails of normal variance mixtures]. *Statisticheskiye metody otsenivaniya i proverki gipotez* [Statistical methods for estimation and hypothesis validation]. Perm: Perm University Press. 90–105.
6. Loève, M. 1955. *Probability theory*. Princeton, NJ: D Van Nostrand. 702 p.
7. Barndorff-Nielsen, O. E. 1977. Exponentially decreasing distributions for the logarithm of particle size. *Proc. Roy. Soc. Lond. A* 353:401–419.
8. Barndorff-Nielsen, O. E. 1978. Hyperbolic distributions and distributions of hyperbolae. *Scand. J. Stat.* 5:151–157.
9. Barndorff-Nielsen, O. E., J. Kent, and M. Sørensen. 1982. Normal variance-mean mixtures and z-distributions. *Int. Stat. Rev.* 50(2):145–159.
10. Jørgensen, B. 1982. *Statistical properties of the generalized inverse Gaussian distribution*. Lecture notes in statistics ser. Berlin: Springer. Vol. 9. 188 p.
11. Stacy, E. W. 1962. A generalization of the gamma distribution. *Ann. Math. Stat.* (33):1187–1192.
12. Korolev, V. Yu., A. Yu. Korchagin, and A. I. Zeifman. 2015 (in press). O skhodimosti raspredeleniy statistik, postroennykh po vyborkam sluchaynogo ob'yema, k mnogomernym obobshchennym dispersionnym gamma-raspredeleniyam [On convergence of the distributions of statistics constructed from samples with random sizes to multivariate generalized variance-gamma distributions]. *Dokl. Akad. nauk* [Acad. Sci. Rep.] 462(4).
13. Billingsley, P. 1968. *Convergence of probability measures.* New York, NY: Wiley. 353 p.
14. Zolotarev, V. M. 1997. *Modern theory of summation of random variables.* Utrecht: VSP. 412 p.
15. Kruglov, V. M. 1973. Skhodimost' chislovykh kharakteristik summ nezavisimykh sluchaynykh velichin so znacheniyami v gil'bertovom prostranstve [Convergence of numerical characteristics of sums of independent random variables with values in Hilbert space]. *Theory Probab. Appl.* 18(3):734–752.
16. Gikhman, I. I., and A. V. Skorokhod. 1971. *Teoriya sluchaynykh protsessov* [Theory of random processes]. Moscow: Nauka. 665 p.

Received March 2, 2015

Contributor

Korchagin Alexander Yu. (b. 1989) — PhD student, Faculty of Computational Mathematics and Cybernetics, M. V. Lomonosov Moscow State University, 1-52 Leninskiye Gory, GSP-1, Moscow 119991, Russian Federation; sasha.korchagin@gmail.com

ПРОГНОЗИРОВАНИЕ ОБЪЕМОВ ЖЕЛЕЗНОДОРОЖНЫХ ГРУЗОПЕРЕВОЗОК ПО ПАРАМ ВЕТОК*

***Р. К. Газизуллина*[1], *М. М. Стенина*[2], *В. В. Стрижов*[3]**

Аннотация: Работа посвящена исследованию алгоритма непараметрического прогнозирования объемов железнодорожных грузоперевозок. Решается задача прогнозирования числа вагонов с различными грузами, следующих по различным маршрутам. Задана топология железнодорожной сети — для всех возможных пар железнодорожных веток дана информация о всех блоках вагонов, совершивших переезд с одной ветки на другую, включая число вагонов в блоке, вид груза и дату прохождения маршрута. Для построения прогноза используется алгоритм, основанный на свертке эмпирической плотности распределения значений временно́го ряда с функцией потерь. Ранее прогноз выполнялся для каждого железнодорожного узла в отдельности. Предлагается повысить качество прогноза за счет прогнозирования по парам веток вместо прогнозирования отправления всех вагонов с данного узла. Алгоритм проиллюстрирован посуточными данными за полтора года о перевозках 38 типов грузов.

Ключевые слова: прогнозирование; непараметрический метод; загруженность железнодорожного узла; функция потерь; эмпирическое распределение; свертка

DOI: 10.14357/08696527150109

1 Введение

Повышение эффективности транспортировки грузов по железнодорожным путям требует решения задачи прогнозирования объемов и времени перевозок грузов с ветки на ветку и загруженности железнодорожных узлов на основании данных о времени прибытия и отправления вагонов. Решение этой задачи необходимо для повышения качества управления в сложных инфокоммуникативных средах [1]. Управленческие решения принимаются на основании работы систем сбора статистической информации высокой точности и доступности [2, 3]. Данная работа расширяет и уточняет метод решения задачи прогнозирования, изложенный в статье [4]. Используемый в ней алгоритм основан на алгоритме квантильной

*Работа выполнена при финансовой поддержке РФФИ (проект 14-07-31046).

[1]Московский физико-технический институт, rimma.gazizullina@gmail.com

[2]Московский физико-технический институт, mmedvednikova@gmail.com

[3]Вычислительный центр им. А. А. Дородницына РАН, strijov@ccas.ru

регрессии [5, 6], модифицированном сверткой гистограммы с функцией потерь, что позволяет учитывать стоимость ошибок в реальной задаче.

Для получения прогноза по временно́му ряду предлагается построить гистограмму. Прогнозируемое значение (последующий элемент временного ряда) ищется как значение, соответствующее оптимальному значению свертки гистограммы и функции потерь, которая задана исходя из экспертных предположений об экономических потерях при недостаточном или избыточном наличии составов на железнодорожном узле. В работе [4] показано, что данное решение находит более точные прогнозы в сравнении с алгоритмом ARMA (autoregressive–moving-average) [7, 8], также используемым для решения подобных задач. Кроме того, данное решение позволяет в некоторых случаях решить задачу, когда алгоритм ARMA не может работать.

Согласно работе [9], в настоящее время насчитывается свыше 100 классов моделей прогнозирования, в связи с чем возникает задача выбора моделей, которые давали бы адекватные прогнозы для изучаемых процессов или систем. Одними из основных методов решения подобных задач являются методы непараметрической регрессии и модификации, такие как ядерное сглаживание, сглаживание сплайнами, авторегрессия, скользящее среднее и др., описанные в [10–13]. Они заключаются в комбинации взвешенных значений элементов временно́го ряда для получения прогноза. Также для решения подобных задач применяют нейронные сети [14, 15].

Данные, описывающие объемы грузоперевозок, имеют следующий формат. Для каждой станции для каждого дня запись содержит число вагонов с грузами различного типа с указанием ветки прибытия и отправления. Для построения прогнозов используются временны́е ряды двух типов:

(1) число отправленных с ветки вагонов с грузом конкретного типа;

(2) число отправленных на заданную ветку вагонов с грузом конкретного типа.

Основной идеей и новизной работы является использование для прогноза временны́х рядов грузоперевозок по парам веток, в отличие от ранее предложенного метода, использовавшего грузоперевозки по одному узлу. Это позволяет точнее отслеживать особенности грузоперевозок между различными парами веток и, как следствие, точнее предсказывать будущие перевозки. В данной работе предполагается, что благодаря рассмотрению более детализированных данных по перевозкам удастся выявить тренды и периодичность. С другой стороны, решение этой задачи может сделать весь прогноз в целом более точным.

2 Постановка задачи гистограммного прогнозирования

Задан набор временны́х рядов, в котором временной ряд $\mathbf{x} = \{x_i\}_{i=1}^{T}$, и горизонт отсрочки прогноза h (число отсчетов от конца временно́го ряда до точки прогноза включительно). Требуется спрогнозировать следующую точку x_{T+1} временно́го ряда $\mathbf{x}$ так, чтобы выполнялось условие оптимальности функции

потерь (2) и свертки гистограммы (1), построенной по значениям временно́го ряда.

Гистограмма $\mathcal{H}$ — набор пар

$$\mathcal{H} = \{(y_k, g_k)\}_{k=1}^{K}, \tag{1}$$

где K — число интервалов $\left[y_k^{\min}, y_k^{\max}\right]$ со средним значением y_k, на которые разбита ось значений ряда $\mathbf{x}$; g_k — высота столбца гистограммы на интервале $\overline{y}_k$, равная взвешенной сумме количества точек ряда, попавших в этот интервал.

Предполагается, что ряд $\mathbf{x} = \{x_i\}_{i=1}^{T}$ стационарен, т. е. совместное распределение вероятностей T наблюдений $x_1, x_2, x_3, \ldots, x_T$ совпадает с распределением $x_{1+\tau}, x_{2+\tau}, x_{3+\tau} \ldots, x_{T+\tau}$ при любых T и τ.

Введем функцию потерь $L(\hat{y}, y)$ — штраф за несоответствие прогнозируемого значения $\hat{y}$ историческому значению y. Например:

(1) $L(z, x) = (z - x)^2$;

(2) $L(z, x) = |z - x|$;

(3) $L(z, x) = \begin{cases} 0, & \text{если} |z - x| < a\,; \\ |z - x| - a, & \text{если} |z - a| \geq a\,, \end{cases}$

где $a > 0$ — экспертно заданный параметр.

В данной работе используется функция

$$L(z, x) = |z - x|\,. \tag{2}$$

Задача нахождения прогнозируемого значения $\hat{y}$ временно́го ряда с использованием функции L имеет вид:

$$\hat{y} = \operatorname*{argmin}_{z \in \mathbb{N}} \sum_{k=1}^{K} L(z, y_k)\,,$$

где связанная переменная z принадлежит множеству натуральных чисел, так как требуется спрогнозировать число вагонов.

Пусть построен прогноз значений ряда $\{\hat{y_1}, \hat{y_2}, \ldots, \hat{y_k}\}$ в некоторых точках, при этом реальные значения ряда $\mathbf{x}$ в этих же точках равны $\{y_1, y_2, \ldots y_k\}$. Эти значения не использовались при построении прогноза и нужны для получения ретроспективной оценки качества прогностической модели. Вычислим среднюю ошибку ретроспективного прогноза:

$$\text{MeanError} = \frac{1}{k} \sum_{i=1}^{k} L(\hat{y_i}, y_i) \to \min_{\hat{y_1}, \ldots, \hat{y_k}}\,. \tag{3}$$

Чем меньше средняя ошибка, тем точнее прогноз. Таким образом, требуется решить задачу минимизации средней ошибки прогнозирования, т. е. найти значения $\{\hat{y_1}, \hat{y_2}, \ldots, \hat{y_k}\}$ такие, что значение MeanError минимально.

3 Алгоритм прогнозирования

Первым шагом работы алгоритма является разбиение исходных данных для получения временны́х рядов. Для каждого дня τ для всех возможных комбинаций типа груза G, веток отправления и прибытия L и A считается число вагонов с грузом G, отправленных в день τ с ветки L на ветку A. При фиксированных значениях G, L и A получаем зависимость числа вагонов от дня τ, т. е. временно́й ряд $\mathbf{x}$. Если при фиксированных значениях G и L просуммировать временны́е ряды для разных веток прибытия A, то получим временной ряд, соответствующий суммарному количеству отправленных грузов определенного типа с ветки L.

Вторым шагом работы является применение базового алгоритма к полученным на первом этапе временны́м рядам для построения прогнозов. Базовый алгоритм основан на описанном в [4] алгоритме непараметрического прогнозирования временно́го ряда. Для построения гистограммы (1) для каждой точки i временно́го ряда $\mathbf{x}$ определим ее вес как произведение $w_i = w_i^F w_i^H$. Сомножитель w_i^F задает показательную весовую функцию

$$w_i^F = v^{(-i+T+h)/F} \in (0\,,1]\,, \tag{4}$$

убывающую к началу временно́го ряда и равную 1 в точке прогноза. Сомножитель w_i^H задан как

$$w_i^H = \begin{cases} K(i_H, PH)\,, & \text{если } H > 0\,; \\ 1\,, & \text{если } H = 0\,, \end{cases} \tag{5}$$

где индекс H вычисляется в результате решения оптимизационной задачи

$$i_H = \min_{n=0,\ldots,\mathrm{floor}((T+h)/P)} |T + H - nP - i|\ . \tag{6}$$

Эта формула задает вес i-й точки, соответствующей годовой сезонности. Ядро задается выражением:

$$K(x,z) = \begin{cases} \left(1 - \left(\dfrac{x}{z}\right)^2\right)^2\,, & \text{если } |x| < z\,; \\ 0 & \text{иначе.} \end{cases}$$

Взвешенные точки $x_i w_i$ используются для построения (1) гистограммы $\mathcal{H}$.

Настраиваемые параметры: $v \in [0,1]$ в выражении (4) — параметр показательного взвешивания точек ряда, параметр «забывания»; $H \in [0,0,5]$ в выражениях (5) и (6) — параметр ядра весовой функции для годовой сезонности, половина ширины «шапки» годовой сезонности.

Ненастраиваемые параметры: P в выражениях (5) и (6) — длина годового сезонного периода (обычно $P = 365$); $w^{\min}$ в выражении (7) — минимальный допустимый вес; F в выражении (4) — нормировочная константа «забывания».

Предлагается выбрать F следующим образом: $F = (T + H)\varepsilon \log_{10}(0,1)$, где $\varepsilon = 10^{-3}$.

Выберем границы гистограммы, число столбцов и разбиение на столбцы следующим образом:

(1) пусть n — число точек x_i, для которых $w_i > w^{\min}$;

(2) выберем число столбцов (обоснование см. в [16]) $K = \lceil 3\sqrt[3]{n} \rceil$; если $K < 5$, то $K = 5$, если $K > 100$, то $K = 100$;

(3) границы $y_1 = \min\limits_{i:w_i > w^{\min}}(x_i)$, $y_k = \max\limits_{i:w_i > w^{\min}}(x_i)$;

(4) столбцы выбираются равной ширины.

Для каждого $k = 1, \ldots, K$ высота столбца гистограммы g_k равна

$$g_k = \sum_{i=1}^{T} w_i \left[x_i \in y_k\right] \left[w_i > w^{\min}\right], \tag{7}$$

где выражение $[\cdot]$ равно 1, если в скобках стоит истинное логическое выражение, и 0 в противном случае.

Прогнозируемое значение ряда x_{T+h} находится как значение $\hat{y} \in \in \{y_1, \ldots, y_K\}$, соответствующее оптимальному значению свертки распределения $\{g_k\}_{k=1}^{K}$ и функции потерь L:

$$\hat{y} = \operatorname*{argmin}_{z \in \{y_1, \ldots, y_K\}} \sum_{k=1}^{K} g_k L(z, y_k)\,.$$

4 Эксперимент

Цель эксперимента. Сделать прогноз для заданных комбинаций «пара веток – грузы», проверить, что разбиение данных по веткам повышает качество прогноза базового алгоритма.

В эксперименте использованы данные о посуточной загруженности железнодорожных узлов с 1 января 2007 г. по 22 апреля 2008 г. В табл. 1 приведен пример записи.

Коды станций представляют собой шестизначные числа. Станции, в коде которых две первые цифры совпадают, входят в одну железнодорожную ветку. Станций отправления — 1566, станций назначения — 1902, веток — 99. Код

Таблица 1 Вид записи базы данных железнодорожных перевозок

Дата погрузки	Станция отправления	Станция назначения	Количество вагонов	Код груза	Род вагона	Суммарный вес груза	Признак маршрутной отправки
2007-01-01	020108	932902	1	1	216	56	9

груза — натуральное число от 1 до 43; также имеются перевозки, где код груза не указан. Род вагона — натуральное число, в имеющихся данных 75 различных типов вагонов.

Для экспериментов были выбраны данные об отправлении поездов с 83-й ветки на все другие. Весь временной ряд разбивался так, что последние 100 дней образовывали контрольную выборку, а первые 300 — обучающую.

Было проведено два эксперимента, в которых обрабатывались одни и те же данные и выполнялись одни и те же алгоритмы, но в которых преследовались различные цели: в обоих исследовались данные о количестве грузов, отправленных с 83-й ветки на другие. Данные в обоих случаях были разбиты таким образом: временны́е ряды строились для числа вагонов с товарами, отправленных с 83-й ветки на другие конкретные ветки. Для них был запущен алгоритм и найдены значения прогноза и средней ошибки. В первом эксперименте исследовались данные об отправленных грузах на некоторые конкретные ветки для всех видов грузов (при этом графики, отображающие некоторые зависимости, представлены

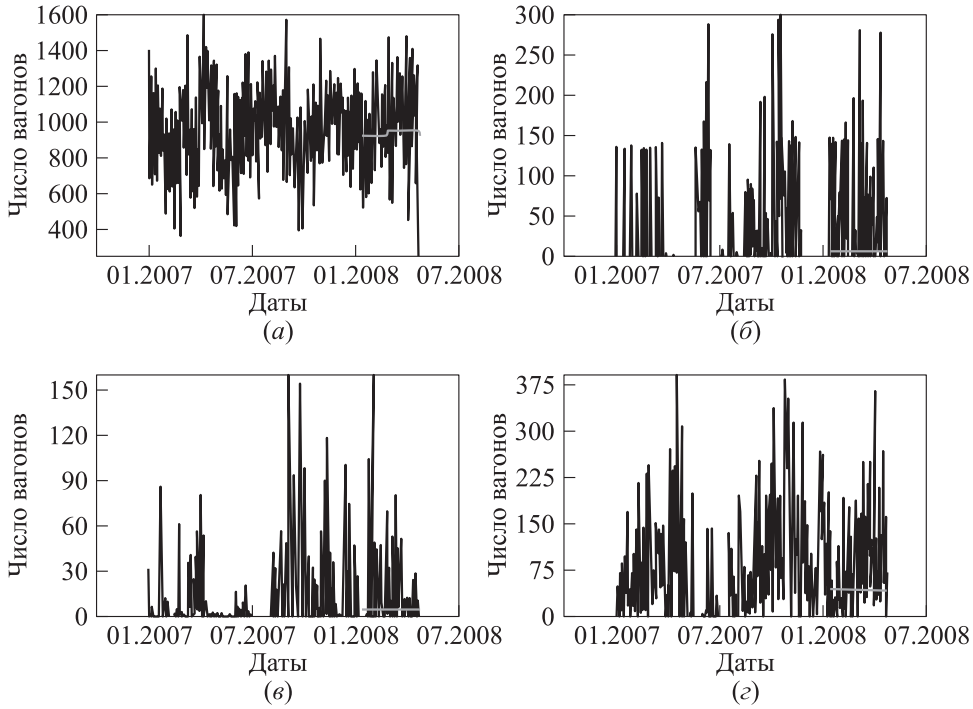


Рис. 1 Графики временны́х рядов числа отправленных вагонов с нефтью с 83-й ветки с прогнозом, выполненным для промежутка в 16 мес.: (*а*) на все ветки; (*б*) на ветку 96; (*в*) на ветку 97; (*г*) на ветку 98

только для некоторых грузов), а также о суммарном количестве различных отправленных грузов на все ветки, — задачей было визуализировать зависимость прогноза и ошибок для разных рядов. Второй эксперимент был проведен для всех видов грузов, отправленных на все ветки, — целью было для каждого вида груза вычислить среднее и максимальное значения усредненной ошибки по всем веткам и сравнить их со средней ошибкой для суммарного количества отправленных грузов этого типа.

Перейдем к описанию полученных результатов. Начнем с первого эксперимента. На рис. 1 изображены графики временны́х рядов — с датой по оси X и числом вагонов по оси Y. Графики приведены как для полного числа отправленных вагонов, так и для вагонов, отправленных на некоторые конкретные ветки. Ветки выбирались таким образом, чтобы отразить различные варианты характеристик временных рядов.

Эксперименты также были проведены для всех грузов, отправленных с 83-й ветки. Для каждого вычисления (т. е. для каждой отдельной ветки прибытия груза, а также для суммарного количества отправленных грузов) была найдена средняя относительная ошибка (3). На рис. 2 представлены результаты этих

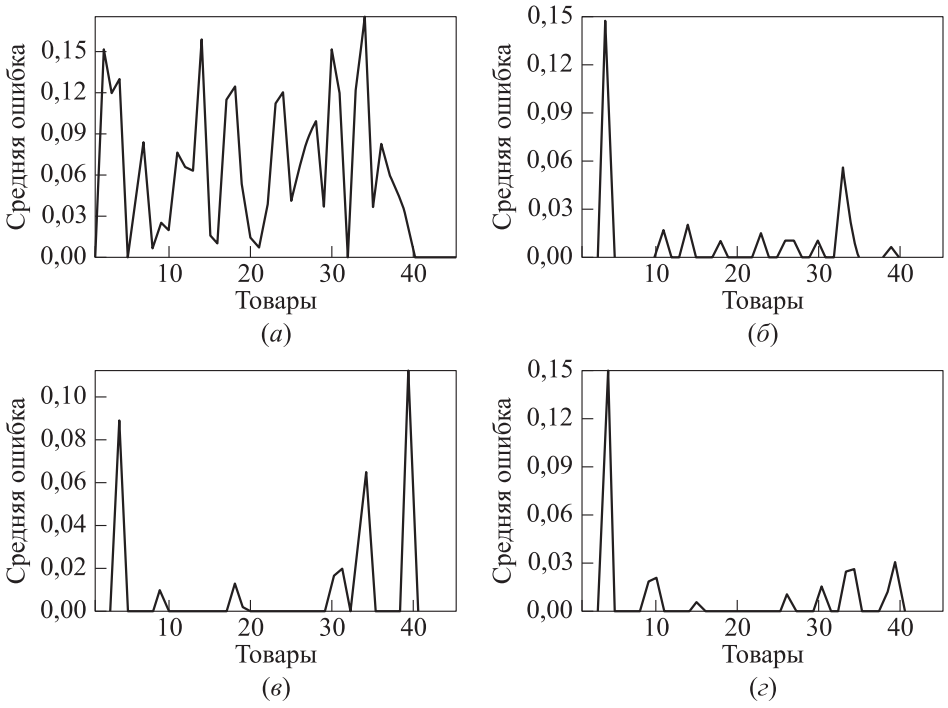


Рис. 2 Средняя ошибка прогноза для всех типов грузов: (*а*) все ветки; (*б*) ветка 96; (*в*) ветка 97; (*г*) ветка 98

вычислений для полного количества отправленных грузов (рис. 2, *а*) и для трех различных веток (рис. 2, *б* – 2, *г*). На рис. 2 отображена зависимость средней относительной ошибки от вида груза.

Для сравнения результатов прогнозирования с базовым алгоритмом (без разбивки по веткам) для некоторых типов грузов были проделаны следующие вычисления: найдена средняя ошибка при прогнозе числа вагонов с нефтью, отправленных с 83-й ветки на каждую другую, а также средняя ошибка при

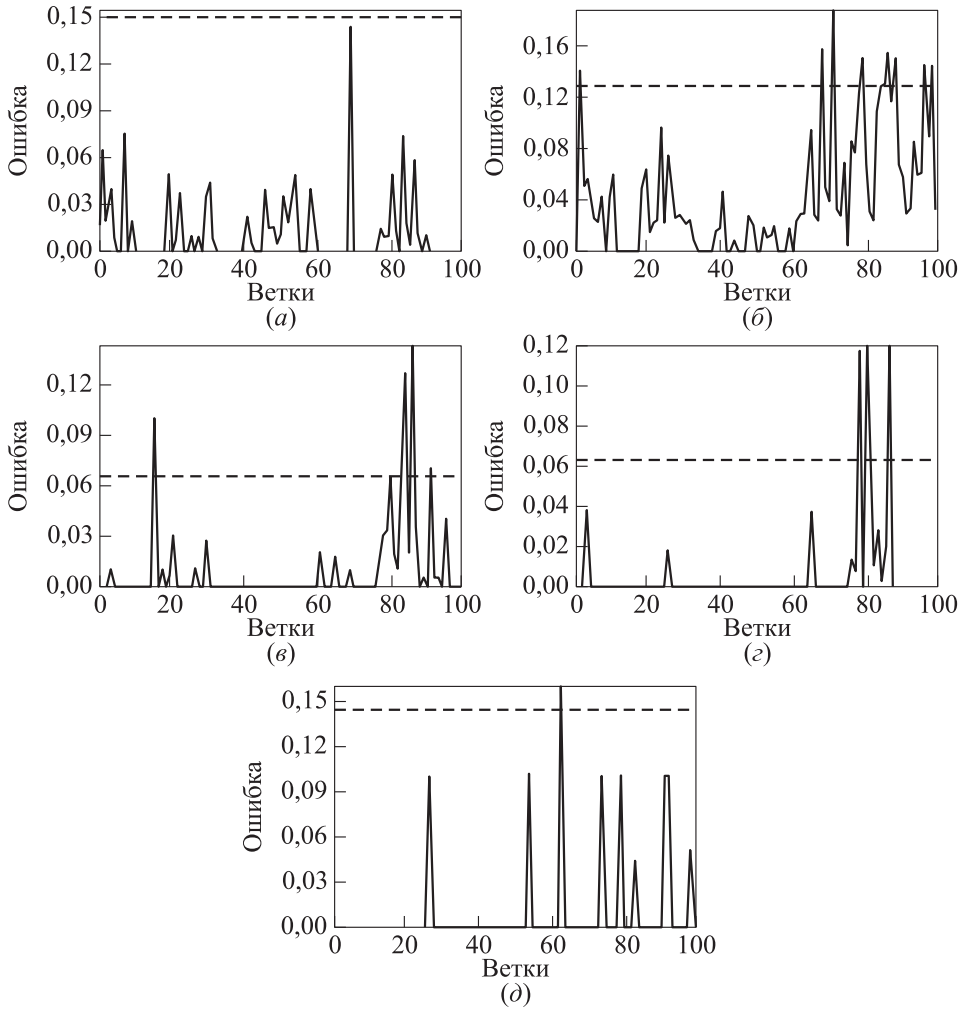


Рис. 3 Сравнение средней ошибки с ошибкой базового алгоритма: (*а*) груз 2; (*б*) груз 4; (*в*) груз 12; (*г*) груз 13; (*д*) груз 15

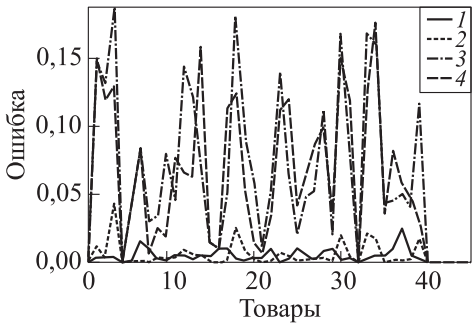


Рис. 4 Минимальная (*1*), средняя (*2*) и максимальная (*3*) ошибки по всем веткам в сравнении с ошибкой базового алгоритма (*4*)

прогнозе суммарного числа вагонов с нефтью, отправленных с 83-й ветки. Результаты этих вычислений представлены в виде графиков зависимости ошибки от ветки. В качестве ориентира горизонтальной штриховой линией отображен уровень ошибки для суммарного количества грузов. Результаты представлены на рис. 3.

Далее рассмотрим результаты второго эксперимента. Для каждого вида груза вычислено среднее, минимальное (ненулевое) и максимальное значение усредненной ошибки по всем веткам прибытия, построены графики зависимости этих значений от вида груза (рис. 4). Для некоторых грузов эти данные выписаны в табл. 2.

Таблица 2 Минимальная, средняя и максимальная ошибка по всем веткам в сравнении с ошибкой базового алгоритма

Тип груза	Минимальная ошибка	Средняя ошибка	Максимальная ошибка	Ошибка базового алгоритма
1	0,001	0,003	0,010	0,004
2	0,005	0,010	0,150	0,150
3	0,003	0,010	0,130	0,120
4	0,010	0,040	0,190	0,130
5	0,001	0,002	0,006	0,009
6	0,001	0,009	0,040	0,030
7	0,020	0,050	0,080	0,080
8	0,010	0,013	0,030	0,010

5 Оценка результатов

На рис. 2 показано, что средняя относительная ошибка прогноза для разных веток почти всегда меньше, чем ошибка прогноза для суммарного числа вагонов. Это означает, что прогноз количества отправленных грузов на определенную ветку в среднем точнее, чем прогноз суммарного числа вагонов. Однако есть и грузы, для которых средняя ошибка прогноза для некоторой конкретной ветки превосходит ошибку прогноза суммарного числа отправленных вагонов.

Это же подтверждают графики на рис. 3, которые дают более подробную информацию о средней ошибке прогнозирования для некоторых грузов. В целом,

средняя относительная ошибка для веток меньше, чем уровень ошибки для суммарного количества. Тем не менее есть и значения ошибок, бóльшие уровня ошибки для суммарного количества. С другой стороны, видно, что веток, для которых средняя ошибка прогноза превосходит ошибку для прогноза суммарного количества грузов, немного (таких нет для груза 2, и 5–10 для остальных рассмотренных типов грузов).

Результаты второго эксперимента — на рис. 4. Таблица 2 также подтверждает тот факт, что средняя ошибка для прогноза количества грузов, отправленных на конкретные ветки, меньше, чем такая же ошибка для прогноза суммарного количества отправленных грузов. Однако средняя ошибка в некоторых случаях меньше минимальной — минимум брался по всем ненулевым значениям, иначе он был бы равен нулю для всех грузов, поскольку для каждого груза есть ветка, на которую такой груз с 83-й ветки не прибывает в течение рассматриваемого периода. Рисунок 4 наглядно демонстрирует, что, во-первых, усредненная по всем веткам подсчитанная ошибка существенно меньше, чем ошибка для суммарного количества грузов. Во-вторых, максимальная ошибка по веткам если и превосходит ошибку для суммы, то ненамного — более того, можно заметить, что они зачастую совпадают или очень близки.

6 Заключение

Поставлен вычислительный эксперимент, который подтвердил предположение о том, что прогноз для отдельных веток точнее, чем прогноз суммарного числа отправленных вагонов. Приведенные результаты подтверждают выдвинутое предположение. Дальнейшие исследования данной задачи могут быть продолжены в следующих направлениях: во-первых, для прогноза количества отправленных грузов можно попытаться в рассматриваемых временны́х рядах выделять тренд и, если возможно, искать периодическую компоненту. Во-вторых, для построения прогноза предлагается применить различные варианты алгоритма. Например, использовать вариацию со скользящим средним.

Литература

1. *Зацаринный А. А., Шабанов А. П.* Методический подход к управлению качеством информации в сложных инфокоммуникационных проектах // Системы и средства информатики, 2011. Т. 21. № 2. С. 3–20.
2. *Синицын И. Н., Корепанов Э. Р., Белоусов В. В., Шоргин В. С., Макаренкова И. В., Конашенкова Т. Д., Агафонов Е. С., Семендяев Н. Н.* Развитие компьютерной поддержки статистических научных исследований систем высокой точности и доступности // Системы и средства информатики, 2011. Т. 21. № 1. С. 3–33.
3. *Синицын И. Н., Шаламов А. С., Сергеев И. В., Синицын В. И., Корепанов Э. Р., Белоусов В. В., Агафонов Е. С., Шоргин В. С.* Методы и средства анализа и моделирования стохастических систем интегрированной логистической поддержки // Системы и средства информатики, 2012. Т. 22. № 2. С. 3–28.

4. *Вальков А. С., Кожанов Е. М., Медведникова М. М., Хусаинов Ф. И.* Непараметрическое прогнозирование загруженности системы железнодорожных узлов по историческим данным // Машинное обучение и анализ данных, 2012. Т. 1. № 4. С. 448–465.
5. *Koenker R., Bassett, Jr., G.* Regression quantiles // Econometrica, 1978. Vol. 46. No. 1. P. 33–50.
6. *Постникова Е.* Квантильная регрессия. — Новосибирск: НГУ, 2006. 34 с.
7. *Cortez P., Rotcha M., Neves J.* Evolving time series forecasting ARMA models // J. Heuristics, 2004. Vol. 10. No. 4. P. 419–429.
8. *Shumway R. H., Stoffer D. S.* Time series analysis and its applications with *R* examples. — Springer, 2006. 590 p.
9. *Тихонов Э. Е.* Прогнозирование в условиях рынка. — Невинномысск: СевКавГТУ, 2006. 221 с.
10. *Хардле В.* Прикладная непараметрическая регрессия / Пер. с англ. А. В. Назина под ред. М. Б. Малютова. — М.: Мир, 1993. 349 с. (*Härdle W.* Applied nonparametric regression. New York – New Rochell – Melbourne – Sydney: Cambridge University Press, 1989. 433 p.)
11. *Шурыгин А. М.* Прикладная стохастика: робастность, оценивание, прогноз. — М.: Финансы и статистика, 2000. 224 с.
12. *Лукашин Ю. П.* Адаптивные методы краткосрочного прогнозирования временны́х рядов. — М.: Финансы и статистика, 2003. 416 с.
13. *Джефри Р.* Непараметрическая эконометрика: вводный курс // Машинное обучение и анализ данных, 2013. Т. 1. № 5. С. 505–518.
14. *Thiesing F. M., Vornberger O.* Forecasting sales using neural networks // Computational intelligence, theory and applications / Ed. B. Reusch. — Lecture notes in computer science ser. — Springer, 1997. Vol. 1226. P. 321–328.
15. *Gheyas I. A., Smith L. S.* Neural network approach to time series forecasting // World Congress on Engineering Proceedings, 2009. Vol. 2. P. 245–253.
16. *Scott D. W.* On optimal and data-based histograms // Biometrika, 1979. Vol. 66. No. 3. P. 605–610.

Поступила в редакцию 20.08.14

LARGE CAPACITY OF RAILWAY CARGO TRANSPORTATION FORECASTING

***R. K. Gazizullina*[1], *M. M. Medvednikova*[1], *and V. V. Strijov*[2]**

[1]Moscow Institute of Physics and Technology, 9 Institutskiy per., Dolgoprudny, Moscow Region 141700, Russian Federation
[2]Dorodnicyn Computing Center, Russian Academy of Sciences, 40 Vavilov Str., Moscow 119333, Russian Federation

Abstract: The article is devoted to research of the algorithm of nonparametric forecasting of railway cargo transportation capacity. The problem considered is

forecasting the number of wagons with various goods, following various routes. The topology of the railway network is given — for all possible pairs of railway lines, information about all blocks of wagons, which have moved from one line to another, including the number of wagons in a block, the type of cargo, and the date of the route, is provided. The algorithm, based on convolution of the empirical density distribution of the values of time series with the loss function is used for prediction. Previously, forecasting was carried out for each railway junction separately. It is proposed to be improved by the quality of forecasting predicting by pairs of lines instead of predicting departure of all wagons from the given junction. The algorithm is illustrated by the daily data on transportation of 38 types of cargo collected during a year and a half.

Keywords: forecasting; nonparametric method; railroad station occupancy; loss function; empirical distribution; compression

DOI: 10.14357/08696527150109

Acknowledgments

The research was financially supported by the Russian Foundation for Basic Research (project 14-07-31046).

References

1. Zatsarinnyi, A. A., and A. P. Shabanov. 2011. Metodicheskiy podkhod k upravleniyu kachestvom informatsii v slozhnykh infokommunikatsionnykh proektakh [Methodological approach to quality management of information in complex infocommunication projects]. *Sistemy i Sredstva Informatiki — Systems and Means of Informatics* 21(2):3–20.
2. Sinitsyn, I. N., E. R. Korepanov, V. V. Belousov, V. S. Shorgin, I. V. Makarenkova, T. D. Konashenkova, E. S. Agafonov, and N. N. Semendyaev. 2011. Razvitie komp'yuternoy podderzhki statisticheskikh nauchnykh issledovaniy sistem vysokoy tochnosti i dostupnosti [Development of computer-aided statistical scientific support for high precision and with high availability systems]. *Sistemy i Sredstva Informatiki — Systems and Means of Informatics* 21(1):3–33.
3. Sinitsyn, I. N., A. S. Shalamov, I. V. Sergeev, V. I. Sinitsyn, E. R. Korepanov, V. V. Belousov, E. S. Agafonov, and V. S. Shorgin. 2012. Metody i sredstva analiza i modelirovaniya stokhasticheskikh sistem integrirovannoy logisticheskoy podderzhki [Analysis and modeling methods and tools for integrated logistic support of stochastic systems]. *Sistemy i Sredstva Informatiki — Systems and Means of Informatics* 22(2):3–28.
4. Val'kov, A. S., E. M. Kozhanov, M. M. Medvednikova, and F. I. Khusainov. 2012. Neparametricheskoe prognozirovanie zagruzhennosti sistemy zheleznodorozhnykh uzlov po istoricheskim dannym [Nonparametric forecasting of railroad stations occupancy according to historical data]. *Mashinnoe Obuchenie i Analiz Dannykh* [J. Machine Learning Data Analysis] 4(1): 448–465.
5. Koenker, R., and G. Bassett 1978. Regression quantiles. *Econometrica* 46(1):33–50.

6. Postnikova, E. 2006. *Kvantil'naya regressiya* [Quantile regression]. Novosibirsk: NGU [Novosibirsk: NSU]. 34 p.
7. Cortez, P., M. Rotcha, and J. Neves. 2004. Evolving time series forecasting ARMA models. *J. Heuristics* 10(4):419–429.
8. Shumway, R. H., and D. S. Stoffer. 2006. Time series analysis and its applications with *R* examples. Springer. 590 p.
9. Tikhonov, E. 2006. *Prognozirovanie v usloviyakh rynka* [Forecasting in market conditions]. Nevinnomyssk. 221 p.
10. Härdle, W. 1989. Applied nonparametric regression. New York – New Rochell – Melbourne – Sydney: Cambridge University Press, 1989. 433 p.
11. Shurygin, A. M. 2000. *Prikladnaya stokhastika: Robastnost', otsenivanie, prognoz* [Applied stochastic: Robustness, estimation, forecast]. Moscow: Finansy i Statistika. 224 p.
12. Lukashin, Y. P. 2003. Adaptivnye metody kratkosrochnogo prognozirovaniya vremennykh ryadov [Adaptive methods of short-term time series forecasting]. Moscow: Finansy i Statistika. 416 p.
13. Dzheffri, R. 2008. Neparametricheskaya ekonometrika: Vvodnyy kurs [Nonparametric econometrics: An introductory course]. *Kvantil'* 7:56.
14. Thiesing, F. M., and O. Vornberger. 1997. Forecating sales using neural networks. *Computational intelligence, theory and applications*. Ed. B. Reusch. Lecture notes in computer science ser. 1226:321–328.
15. Gheyas, I. A., and L. S. Smith. 2009. Neural network approach to time series forecasting. *World Congress on Engineering Proceedings*. 2:245–253.
16. Scott, D. W. 1979. On optimal and data-based histograms. *Biometrika* 66(3):605–610.

Received August 20, 2014

Contributors

Gazizullina Rimma K. (b. 1993) — student, Moscow Institute of Physics and Technology, 9 Institutskiy per., Dolgoprudny, Moscow Region 141700, Russian Federation; rimma.gazizullina@gmail.com

Stenina Mariya M. (b. 1991) — student, Moscow Institute of Physics and Technology, 9 Institutskiy per., Dolgoprudny, Moscow Region 141700, Russian Federation; mmedvednikova@gmail.com

Strijov Vadim V. (b. 1967) — Candidate of science (PhD) in physics and mathematics, associate professor, scientist, Dorodnicyn Computing Center, Russian Academy of Sciences, 40 Vavilov Str., Moscow 119333, Russian Federation; strijov@gmail.com

НЕКОТОРЫЕ ПОДХОДЫ К ФОРМИРОВАНИЮ НОРМАТИВНО-ТЕХНИЧЕСКОЙ БАЗЫ ЕДИНОГО ИНФОРМАЦИОННОГО ПРОСТРАНСТВА РОССИИ В ЧАСТИ ИНФОРМАЦИОННЫХ РЕСУРСОВ

А. А. Зацаринный[1], *Э. В. Киселев*[2]

Аннотация: В статье развиваются методические подходы к формированию нормативно-технической базы для создания единого информационного пространства Российской Федерации (ЕИП РФ) в части вопросов систематизации и взаимодействия информационных ресурсов (ИР). В составе федерального уровня ЕИП РФ как мегасистемы предлагается создание двух централизованных компонентов с собственными ИР — центра управления (ЦУ) и центра взаимодействия с мировым информационным пространством (ЦВ МИП). Предложена обобщенная модель формирования и взаимодействия защищенных ИР (ЗИР) в составе единого пространства. Изложен общий подход к формированию, взаимодействию и использованию ЗИР на стороне участника мегасистемы. Предложена обобщенная модель участника, включающая общий контур ЗИР, в который входят три самостоятельных контура соответственно для открытых, конфиденциальных и закрытых ИР. Определены некоторые вопросы проектирования профилей сред открытых систем участников в составе единого пространства. Предложена модель процесса создания профиля среды открытой системы для единой информационной системы (ЕИС) участника на базе российского руководства Р 50.1.041-2002.

Ключевые слова: защищенные информационные ресурсы; центр управления ЕИП РФ; центр взаимодействия с мировым информационным пространством; шлюзы безопасного взаимодействия ЕИП РФ; общероссийская система электронного взаимодействия; контуры защищенных информационных ресурсов участника; банк данных участника; архив информации участника; единая информационная (информационно-телекоммуникационная) система участника в защищенном исполнении; проектирование профилей сред для открытых систем участников

DOI: 10.14357/08696527150110

[1]Институт проблем информатики Федерального исследовательского центра «Информатика и управление» Российской академии наук, azatsarinny@ipiran.ru

[2]Институт проблем информатики Федерального исследовательского центра «Информатика и управление» Российской академии наук, ekiselev @ipiran.ru

1 Информационные ресурсы единого информационного пространства Российской Федерации и собственные централизованные компоненты

Информационные ресурсы ЕИП РФ территориально распределены по многочисленным участникам, входящим в это пространство [1]. Каждый участник в рамках ЕИП РФ при взаимодействии передает информацию из своих ИР другим участникам и соответственно получает от них необходимую ему информацию. Естественно, необходимы некоторые компоненты, которые должны координировать взаимодействие между участниками, осуществлять контроль и управлять функционированием ЕИП РФ в целом.

Для координации, контроля и управления функционированием ЕИП РФ предлагается создание в составе федерального уровня данной мегасистемы двух собственных централизованных компонентов:

– ЦУ ЕИП РФ, включающего определенный набор собственных ЗИР этого центра;

– ЦВ МИП. Соответственно ЦВ МИП должен иметь определенный состав ЗИР ЦВ МИП, которые через шлюзы безопасного взаимодействия (ШБВ) должны обмениваться доступными ИР с мировым информационным пространством.

Центр управдения ЕИП РФ и ЦВ МИП предназначены непосредственно для использования в интересах ЕИП РФ. Кроме того, эти компоненты могут эффективно использоваться и в других случаях, например [2]:

– при решении проблемы перехода на систему государственного стратегического планирования и коренного изменения ситуации в области использования интеллектуального потенциала страны;

– при централизации управления научно-техническим развитием России;

– при создании и вводе в действие комплексной системы мониторинга национальной безопасности России.

Ниже кратко рассматриваются ЗИР ЦУ в интересах ЕИП РФ: конфигурационная база данных (КБД) и электронная библиотека документации (ЭБД).

Конфигурационная база данных из состава ЗИР ЦУ содержит сведения обо всех (или первоначально о наиболее значимых) составных частях инфраструктуры ЕИП РФ и взаимосвязях между ними. В КБД по линии участников ЕИП РФ могут быть внесены: серверы и наиболее важные рабочие места; системные сервисы (серверы баз данных, HTTP-серверы, Microsoft Exchange и т. п.); наиболее важные базы, каталоги и другие хранилища данных; автоматизированные системы, их подсистемы и автоматизированные рабочие места, роли пользователей. В качестве нормативно-справочной информации в КБД вносят сведения о структурных подразделениях, процессах, операциях.

На основе КБД можно автоматически формировать различные документы, в том числе формуляры и общие описания систем участников ЕИП РФ. При наличии в КБД текстовых описаний потенциально возникает возможность автоматического формирования инструкций по эксплуатации комплексов технических средств, технологических инструкций и программной документации.

Второй составной частью является *ЭБД из состава ЗИР ЦУ* для коллективного доступа и использования всеми участниками ЕИП РФ.

Электронная библиотека документации должна обеспечивать всем пользователям документации следующее:

- наглядность структуры документации, возможность понять со стороны участников ЕИП РФ, какой документ нужен пользователю;
- возможность быстро найти любой нужный документ;
- уверенность в том, что найденный документ содержит актуальную информацию.

В рамках ЦУ может быть рассмотрен вопрос создания электронной библиотеки, интегрированной с КБД.

2 Предложения по формированию и взаимодействию информационных ресурсов в составе единого информационного пространства Российской Федерации

Основу ЕИП РФ составляет совокупность ИР многочисленных участников, входящих в это пространство. В [1] определено, что ЕИП РФ оперирует исключительно ЗИР. Соответственно, все предложения по формированию ИР и взаимодействию ИР в составе ЕИП РФ ориентированы только на ЗИР.

Предлагается обобщенная модель формирования и взаимодействия ЗИР в рамках ЕИП РФ (рис. 1). Данная модель охватывает федеральный и региональный уровни ЕИП РФ, на которых формируются и функционируют ЗИР многочисленных участников из числа государственных и негосударственных структур России.

Укрупненно *федеральный уровень ЕИП РФ* включает следующие государственные ЗИР участников:

- ресурсы высших органов государственной власти (ЗИР ВОГВ) РФ. Сюда входят ЗИР: Президента Российской Федерации; Совета Федерации и Государственной Думы Федерального Собрания Российской Федерации; Правительства Российской Федерации; высших судебных органов Российской Федерации; федеральных органов исполнительной власти Российской Федерации;
- ресурсы иных структур федерального уровня (СФУ), включая органы государственной власти (ЗИР СФУ ОГВ);

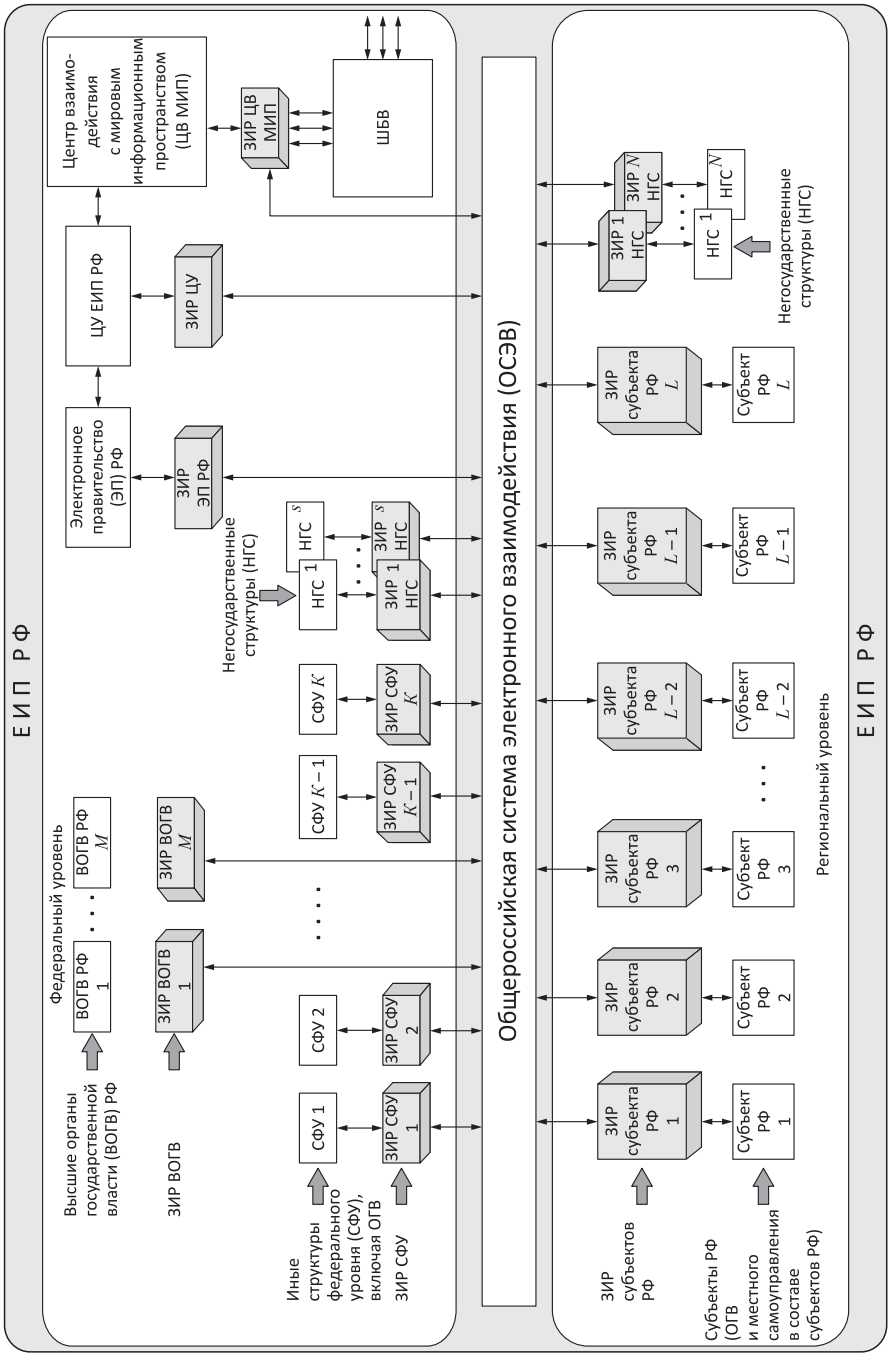


Рис. 1 Обобщенная модель формирования и взаимодействия ЗИР в ЕИП РФ

– ресурсы электронного правительства Российской Федерации (ЗИР ЭП РФ);
– ресурсы ЦУ ЕИП РФ (ЗИР ЦУ ЕИП РФ);
– ресурсы ЦВ МИП (ЗИР ЦВ МИП).

Центр взаимодействия с мировым информационным пространством выполняет функции посредника между ЕИП РФ и мировым информационным пространством. Защищенные ИР из состава ЕИП РФ, подлежащие передаче в мировое информационное пространство, аккумулируются в ЗИР ЦВ МИП, где они проходят необходимые проверки, коррекцию и фильтрацию, после чего через ШБВ выдаются в мировое информационное пространство. В обратном направлении ИР, поступающие через ШБВ из мирового информационного пространства, аккумулируются в ЦВ МИП, проходят необходимые проверки, обработку, коррекцию и фильтрацию, преобразуются в ЗИР ЦВ МИП и далее выдаются адресно в соответствующие ЗИР из состава ЕИП РФ.

Региональный уровень ЕИП РФ укрупненно включает ЗИР участников в рамках каждого субъекта РФ, в том числе:

– органов государственной власти (ОГВ) соответствующего субъекта РФ. Соответственно они имеют ЗИР ОГВ этого субъекта РФ;
– органов местного самоуправления (ОМСУ) данного субъекта РФ. Соответственно они имеют ЗИР ОМСУ субъекта РФ, которые предлагается приравнять к государственным ЗИР данного субъекта РФ.

Учитывая большое число субъектов РФ (более 80), целесообразно и экономически крайне выгодно консолидировать те и другие ЗИР в рамках ЕИП данного субъекта РФ. Именно такой вариант показан на рис. 1, где консолидированные ЗИР обозначены как ЗИР субъекта РФ.

Негосударственные структуры (НГС) в качестве участников ЕИП РФ присутствуют на федеральном и региональном уровнях. Соответственно ЗИР НГС подключаются к тому или другому уровню — в зависимости от их тяготения (например, территориального тяготения).

Всю совокупность многочисленных информационных взаимодействий в составе ЕИП РФ предлагается реализовать посредством вновь создаваемой *общероссийской системы электронного взаимодействия (ОСЭВ)*.

Общероссийская система электронного взаимодействия, будучи важнейшим коммуникационным компонентом глобального масштаба, фактически поддерживает функционирование всего ЕИП РФ. К ОСЭВ подключаются на правах ее абонентов ЗИР всех участников ЕИП РФ. Общероссийская система электронного взаимодействия должна базироваться на мощном «связном фундаменте» в виде совокупности телекоммуникационных сетей, систем и сети Интернет.

Прототипом ОСЭВ может служить система межведомственного электронного взаимодействия (СМЭВ). Эта система была создана ОАО «Ростелеком» для федерального уровня инфраструктуры электронного правительства (ИЭП) РФ в рамках ФЦП «Электронная Россия» [3]. Были разработаны материалы

технического проекта по ИЭП РФ (7 книг), основным соисполнителем которых был ИПИ РАН. В частности, там рекомендовалось расширить СМЭВ до регионального уровня. Таким образом, для поэтапного создания ОСЭВ имеется определенный задел в виде СМЭВ. Главная роль в создании ОСЭВ, безусловно, должна принадлежать государству.

3 Общий подход к формированию, взаимодействию и использованию защищенных информационных ресурсов на стороне участника единого информационного пространства Российской Федерации

Общий подход исходит из многочисленности участников и предполагает на стороне участника правильное структурирование ЗИР, которые будут использоваться при взаимодействии с другими участниками в рамках ЕИП РФ. При этом особое внимание следует уделить необходимой степени защищенности самих ЗИР участника. Предлагается обобщенная модель формирования и взаимодействия ЗИР на стороне участника ЕИП РФ (рис. 2).

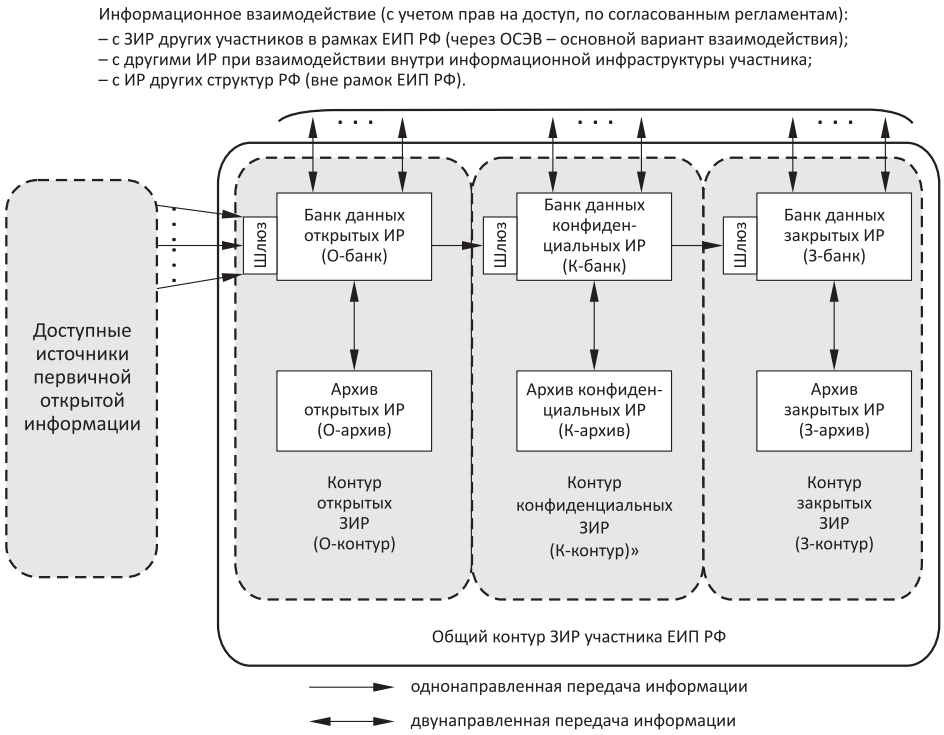


Рис. 2 Обобщенная модель формирования и взаимодействия ЗИР участника ЕИП РФ

На стороне участника ЕИП РФ организуется *общий контур ЗИР*. В пределах этого общего контура сосредоточены все ЗИР, которые необходимы при взаимодействии с другими участниками ЕИП РФ.

Общий контур максимальной конфигурации включает три самостоятельных контура:

(1) *контур открытых ЗИР* (*О-контур*). В нем собирается, обрабатывается, хранится и выдается для использования открытая информация;

(2) *контур конфиденциальных ЗИР* (*К-контур*). В нем собирается, обрабатывается, хранится и выдается для использования конфиденциальная информация;

(3) *контур закрытых ЗИР* (*З-контур*). В нем собирается, обрабатывается, хранится и выдается для использования закрытая информация.

При необходимости может использоваться общий контур сокращенного состава, включающий:

– два самостоятельных контура: О-контур и К-контур; К-контур и З-контур; О-контур и З-контур;

– один самостоятельный контур: только О-контур; только К-контур; только З-контур.

Каждый контур имеет в своем составе:

– банк данных (О-банк, К-банк, З-банк). Он представляет собой упорядоченную совокупность данных, свод необходимой информации для информационного взаимодействия и коллективного использования многими потребителями. Банк данных создается с применением различных систем хранения и передачи информации (например, единой информационной или информационно-телекоммуникационной системы участника);

– архив информации (О-архив, К-архив, З-архив);

– однонаправленные шлюзы, обеспечивающие однонаправленную передачу информации (из источников первичной открытой информации в О-контур; из О-контура в К-контур; из К-контура в З-контур).

Информационное взаимодействие общих контуров ЗИР участников ЕИП РФ осуществляется через указанные банки данных (с учетом прав на доступ, по согласованным регламентам):

– ЗИР конкретного участника с ЗИР других участников в рамках ЕИП РФ (через ОСЭВ — основное направление взаимодействия);

– ЗИР конкретного участника с другими ИР при взаимодействии внутри информационной инфраструктуры этого участника;

– ЗИР конкретного участника с ИР других структур РФ (вне рамок ЕИП РФ).

Фактически общий контур ЗИР участника ЕИП РФ следует создавать на основе ЕИС или единой информационно-телекоммуникационной системы (ЕИТКС) в защищенном исполнении, которая должна соответствовать общим требованиям ЕИП РФ.

Существенным для ЕИС и ЕИТКС является использование типизированного интерфейса (возможно нескольких интерфейсов) информационного взаимодействия, а также типизированных регламентов информационного взаимодействия в рамках ЕИП РФ.

4 Некоторые вопросы проектирования профилей сред для открытых систем участников в составе единого информационного пространства Российской Федерации

В [1] было определено, что потенциальному участнику необходимо срочно отказываться от «лоскутной» информатизации. Количество автономных информационных систем у него должно кардинально сокращаться, информатизация должна перейти в новое качество с таким расчетом, чтобы была создана ЕИС или ЕИТКС.

На пути создания ЕИС (ЕИТКС) потенциальному участнику, безусловно, необходимо выработать единые технические подходы (политику), включая обязательную стандартизацию. Это должно позволить осуществлять первостепенную проработку вопросов создания ЕИС (ЕИТКС).

Единая техническая политика участника позволяет сформировать единый взгляд на систему, ее архитектуру и разработать общий язык для ее определения и описания, а также определяет возможность сопряжения и эффективного взаимодействия различных подсистем / компонентов в рамках единой системы.

С практической точки зрения единая техническая политика должна представлять собой профиль стандартов, обязательных для всех без исключения подразделений участника ЕИП РФ, а также всех исполнителей работ для участника. Единая техническая политика должна предотвращать «волюнтаризм» в выборе принимаемых решений и сводить на нет попытки «несанкционированной рационализации», периодически предпринимаемые техническими специалистами на местах, которые пытаются вернуться к пресловутой «лоскутной» информатизации.

Современная информационная инфраструктура, включающая информационные, вычислительные и телекоммуникационные ресурсы, представляет собой гетерогенную программно-аппаратную среду, использующую разнородные аппаратные и программные компоненты. При этом возникает проблема совместимости в этой среде, т. е. реализации трех основных качеств: переносимости приложений между различными платформами, взаимодействия систем и их масштабируемости.

Среда, обладающая сочетанием этих трех качеств, определяется в международных стандартах как среда открытой системы (СОС). Среда открытой системы

описывает функциональные особенности информационных систем, необходимые для обеспечения переносимости и функциональной совместимости приложений в гетерогенных (разнородных) сетях, объединяющих аппаратные и программные платформы различных типов.

Открытость среды достигается использованием согласованного набора стандартов (профиля) на все программно-аппаратные компоненты. Обеспечение открытости — достаточно сложная научно-методическая задача, поскольку число стандартов в области информационных технологий достаточно велико и постоянно растет с появлением новых технологий. К тому же каждый участник ЕИП РФ имеет свою специфику.

Одной из важнейших задач в решении проблемы обеспечения открытости информационных систем является создание методики построения профилей как неотъемлемой части информационной инфраструктуры любого уровня.

Профиль СОС представляет собой специально подобранный набор базовых стандартов и устанавливаемых в них вариантов требований (опций), которыми для конкретного класса или области приложений определяют на языке функциональных требований, протоколов, интерфейсов, форматов данных и др. функционирование приложений при их взаимодействии между собой. За последние годы практика проектирования профилей активно развивалась, сложилось достаточно четкое представление о процессах жизненного цикла профилей и их содержания, следствием этого факта стала разработка и принятие рекомендаций на мировом уровне стандартизации, а затем и в России.

Преимущества от использования профилей СОС следующие:

- независимость от поставщика;
- возможность поддержки любого технологического образца;
- обеспечение взаимосвязи между деловой (основной) деятельностью конкретной организации (в нашем случае участника ЕИП РФ) и применяемыми информационными технологиями;
- обеспечение интеграции и функциональной совместимости;
- обеспечение реализации структуры инвестиций, дающей максимальную отдачу от вложений в информационные технологии;
- обеспечение наглядной и полной информации о результатах деятельности конкретной организации.

Наиболее подходящими являются рекомендации по стандартизации, разработанные с учетом требований стандарта IEEE Std 1003.23-98 «Стандарт института инженеров по электротехнике и электронике. Руководство по проектированию профилей среды открытой системы организации-пользователя». Отечественные рекомендации по стандартизации Р 50.1.041-2002 «Информационные технологии. Руководство по проектированию профилей среды открытой системы (СОС) организации-пользователя» [4], разработанные на основе указанного стандарта, введены в действие 1 января 2004 г.

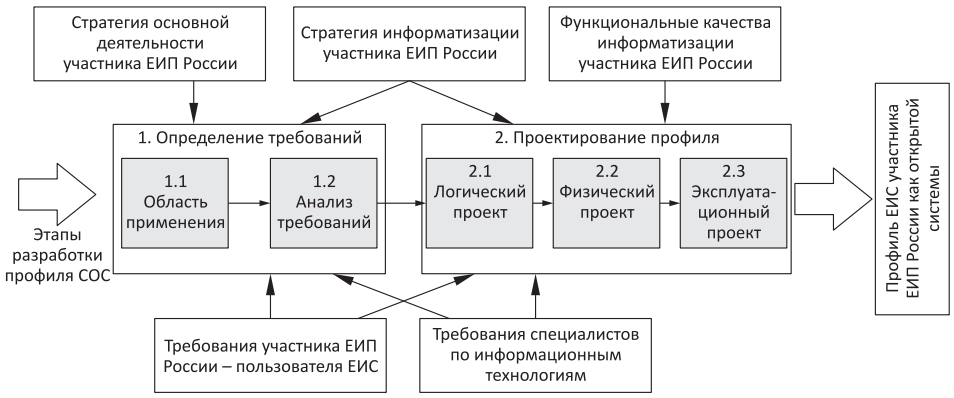


Рис. 3 Модель процесса создания профиля СОС — ЕИС участника ЕИП РФ на базе руководства Р 50.1.041-2002

Рекомендации руководства Р 50.1.041-2002 предназначены оказывать содействие пользователям, проектировщикам, заказчикам и специалистам по применению и эксплуатации сложных информационных систем при создании профилей СОС организации-пользователя, которые устанавливают требования организаций к обработке информации и телекоммуникациям. В зависимости от целого ряда факторов, включая функции, процессы и задачи, выполняемые организацией, может быть принят различный подход к проектированию профилей СОС. Эти рекомендации не содержат всех необходимых сведений для создания профиля СОС, они устанавливают методологию выбора стандартов и процедуры проектирования профилей СОС организации-пользователя и при практическом применении должны быть дополнены другими технологиями, имеющими отношение к созданию профилей СОС с учетом специфики конкретной организации.

Общая модель разработки профиля СОС организации-пользователя является расширением этапов установления требований и проектирования профиля в соответствии с взаимосвязью между стратегией деловой (основной) сферы деятельности организации и требованиями к функциональным возможностям ЕИС (ЕИТКС) и проектированию профиля СОС.

Модель процесса создания профиля СОС — ЕИС участника ЕИП РФ на базе руководства Р 50.1.041-2002 показана на рис. 3.

Согласно рекомендациям руководства Р 50.1.041-2002 модель процесса создания профиля СОС включает две последовательные стадии:

(1) «Определение требований», в которую входят этапы «Область действия» и «Анализ требований»;

(2) «Проектирование профиля», включающую три последующих этапа «Логический проект», «Физический проект» и «Эксплуатационный проект».

Стадия «Определение требований» базируется на использовании:

- стратегии сферы основной деятельности участника ЕИП РФ;
- стратегии информатизации участника ЕИП РФ;
- требований участника ЕИП РФ — пользователя ЕИС (ЕИТКС);
- требований специалистов по информационным технологиям.

Стадия «Проектирование профиля» базируется на использовании:

- стратегии информатизации участника ЕИП РФ;
- функциональных качеств информатизации участника ЕИП РФ;
- требований участника ЕИП РФ — пользователя ЕИС (ЕИТКС);
- требований специалистов по информационным технологиям.

Конечным результатом выполнения обеих стадий, выполненных по рекомендациям руководства Р 50.1.041-2002, является профиль ЕИС (ЕИТКС) участника как открытой системы. Такая система после реализации согласно профилю способна органично включаться в ЕИП РФ, взаимодействовать с другими участниками и обмениваться ИР.

5 Заключение

В статье обоснованы некоторые предложения по совершенствованию нормативно-технического обеспечения ЕИП РФ:

1. Для координации, контроля и управления функционированием ЕИП РФ предложено создание целевым назначением в составе федерального уровня данной мегасистемы двух собственных централизованных компонентов:
   (а) ЦУ ЕИП РФ, имеющего определенный набор собственных ЗИР;
   (б) ЦВ МИП, включающего определенный состав ЗИР.
2. Разработана обобщенная модель формирования и взаимодействия ЗИР в рамках ЕИП РФ.
3. Обоснован общий подход к формированию, взаимодействию и использованию ЗИР на стороне участника ЕИП РФ. Предложена обобщенная модель, включающая общий контур ЗИР, в который входят три самостоятельных контура: открытых ЗИР (О-контур); конфиденциальных ЗИР (К-контур); закрытых ЗИР (З-контур).
4. Определены некоторые вопросы проектирования профилей сред для открытых систем участников в составе ЕИП РФ. Предложена для использования модель процесса создания профиля СОС — ЕИС (ЕИТКС) участника ЕИП РФ на базе российского руководства Р 50.1.041-2002.

Литература

1. *Зацаринный А. А., Киселев Э. В.* Некоторые подходы к формированию нормативно-технической базы для создания единого информационного пространства России // Системы и средства информатики, 2014. Т. 24. № 4. С. 206–220.
2. *Колин К. К.* Вызовы XXI века и стратегические приоритеты развития России // Стратегические приоритеты, 2014. № 2. С. 25–42.
3. Электронная Россия (2002–2010 годы): Федеральная целевая программа. Утверждена Постановлением Правительства Российской Федерации от 28 января 2002 г. № 65.
4. Информационные технологии. Руководство Р 50.1.041-2002 по проектированию профилей среды открытой системы (СОС) организации-пользователя. Утверждено Постановлением Госстандарта России от 14 ноября 2002 г. № 415-ст. Дата введения — 1 января 2004 г.

Поступила в редакцию 03.02.15

SOME APPROACHES TO FORMING THE REGULATORY AND TECHNICAL BASE FOR THE UNIFIED INFORMATION SPACE OF RUSSIA IN THE FIELD OF INFORMATION RESOURCES

A. A. Zatsarinny and E. V. Kiselev

Institute of Information Problems, Federal Research Center "Computer Science and Control," Russian Academy of Sciences, 44-2 Vavilov Str., Moscow 119333, Russian Federation

Abstract: The methodical approaches to forming the regulatory and technical base for the unified information space of the Russian Federation (UIS RF) in the field of systematization and interaction of information resources are developed. It is proposed to create two centralized components of the UIS RF federal level as a mega system with their own information resources: the control center and the world information space interaction center. The authors suggest a generalized model of forming and interaction for secured information resources included in the unified information space. The general approach to creation, interaction, and usage of secured information resources at the site of a mega system participant is described as well. There is a participant-generalized model which includes a secured information resources general circuit. The circuit includes three independent circuits for open, confidential, and enclosed information resources. Some issues of design of profiles of open system environments for participants of the unified information space are determined. Finally, a model of the process of creation of an open system environment profile for a participant of the unified information system is suggested on the basis of the Russian guidance R 50.1.041-2002.

Keywords: secured information resources; UIS RF control center; world information space interaction center; UIS RF secured interaction gateways;

All-Russian System of Electronic Interaction; participant's secured information resources circuits; participant's databank; participant's information archive; participant's unified information (information and telecommunication) secured system; participant's open system environment profile design

DOI: 10.14357/08696527150110

References

1. Zatsarinny, A. A., and E. V. Kiselev. 2014. Nekotorye podkhody k formirovaniyu normativno-tekhnicheskoy bazy dlya sozdaniya edinogo informatsionnogo prostranstva Rossii [Some approaches to forming regulatory and technical base for cration of unified information space of Russia]. *Sistemy i Sredstva Informatiki — Systems and Means of Informatics* 24(4):206–220.
2. Kolin, K. K. 2014. Vyzovy XXI veka i strategicheskie prioritety razvitiya Rossii [Challenges and strategic priorities of Russia development]. *Strategicheskie Prioritety* [Strategic Priorities] 2:25–42.
3. Elektronnaya Rossiya (2002–2010): Federal'naya tselevaya programma [Electronic Russia (2002–2010). Federal program]. Utverzhdena Postanovleniem Pravitel'stva Rossiyskoy Federatsii ot 28 yanvarya 2002 g. No. 65 [Approved by resolution #65 of Russian Government on 28.01.2002].
4. Informatsionnye tekhnologii. Rukovodstvo R 50.1.041-2002 po proektirovaniyu profiley sredy otkrytoy sistemy (SOS) organizatsii-pol'zovatelya. [Information technologies. Guidance R 50.1.041-2002. Design of organization-user open system environment profiles]. Utverzhdeno Postanovleniem Gosstandarta Rossii ot 14 noyabrya 2002 g. No. 415-st. Data vvedeniya — 1 yanvarya 2004 g. [Approved by GosStandard of Russia resolution #415-st on 14.01.2002. Valid since 1.01.2004].

Received February 03, 2015

Contributors

Zatsarinny Alexander A. (b. 1951) — Doctor of Science in technology, professor, Deputy Director, Federal Research Center "Computer Science and Control," Russian Academy of Sciences, 44-2 Vavilov Str., Moscow 119333, Russian Federation; azatsarinny@ipiran.ru

Kiselev Eduard V. (b. 1938) — senior scientist, Institute of Information Problems, Federal Research Center "Computer Science and Control," Russian Academy of Sciences, 44-2 Vavilov Str., Moscow 119333, Russian Federation; ekiselev @ipiran.ru

ТЕХНОЛОГИЯ ПРЕДОТВРАЩЕНИЯ ДУБЛИРОВАНИЯ БИБЛИОГРАФИЧЕСКИХ ОПИСАНИЙ В БАЗЕ ДАННЫХ НАУЧНЫХ ПУБЛИКАЦИЙ БИАС ИПИ РАН

***М. Ю. Заикин*[1], *В. С. Долгополов*[2], *О. Л. Обухова*[3], *И. В. Соловьев*[4]**

Аннотация: Представлена технология предотвращения дублирования библиографических описаний (БО), разработанная для научной базы данных (БД) библиографической информационно-аналитической системы (БИАС) ИПИ РАН (с февраля 2015 г. — в составе ФИЦ ИУ РАН). Приведен анализ причин, порождающих дублирование записей. Разработанная технология опирается на применение программных модулей *определения схожести*, использующих методы нечеткого поиска по алгоритму Оливера, и средств визуализации полученных результатов, которые встроены в систему на уровне формирования контента БД. Введено понятие *индекса схожести*, используемое в алгоритмах поиска дублирующих БО. Представлена формальная модель данных, заложенная в основу построения БД, базирующаяся на принципах фасетной навигации, разработанной авторами. Исследование формальной модели позволило разработать алгоритмы, на которых строилась технология предотвращения дублирования БО. Применение разработанных программных средств дало возможность обнаружить и удалить дублирующие БО в БД научных публикаций БИАС ИПИ РАН.

Ключевые слова: программные модули определения схожести; индекс схожести; метод нечеткого поиска по алгоритму Оливера; фасетная навигация

DOI: 10.14357/08696527150111

1 Введение

Основной целью создания БИАС ИПИ РАН [1] (с февраля 2015 г. ИПИ РАН — в составе Федерального исследовательского центра «Информатика и управление» РАН) была разработка программного комплекса для сбора, хранения и анализа библиографической информации о научных публикациях сотрудников. Для заполнения БД БИАС используются реферативные базы на-

[1]Институт проблем информатики Федерального исследовательского центра «Информатика и управление» Российской академии наук, maxim@amsd.com

[2]Институт проблем информатики Федерального исследовательского центра «Информатика и управление» Российской академии наук, slava@amsd.com

[3]Институт проблем информатики Федерального исследовательского центра «Информатика и управление» Российской академии наук, obuhova@amsd.com

[4]Институт проблем информатики Федерального исследовательского центра «Информатика и управление» Российской академии наук, soloviev@amsd.com

учного цитирования Web of Science (WoS), Scopus, Российского индекса научного цитирования (РИНЦ) на платформе eLibrary, тематические интернет-ресурсы, находящиеся в открытом доступе, а также документально подтвержденные сведения от авторов. Каждая научная статья в БИАС представлена в виде БО [2].

В задачи БИАС входит предоставление пользователям поисково-справочных сервисов. Реализован поиск публикаций по совокупности параметров в произвольных сочетаниях. Также в системе поддерживаются механизмы связывания описаний цитирующей и цитируемой публикаций, что позволяет создавать карты связи публикаций, определяя тем самым неформальные научные коллективы, работающие над определенной тематикой. К удобным возможностям относится объединение посредством взаимных ссылок описания статьи и ее переводной версии в иностранном издании. Примеры БО русскоязычных статей и их переводов, объединенных соответствующими ссылками вида «*публикация – переводная версия*», представлены на рис. 1.

Одна из важных целей создания БИАС ИПИ РАН — анализ публикационной активности авторов. Библиографический анализ цитирования позволяет

Издание: английское название / русское название	Theory of Probability & Its Applications		Теория вероятностей и её применения
Адрес журнала в Интернете	http://epubs.siam.org/loi/tprbau		http://www.mathnet.ru/php/archive.phtml?jrnid=tvp&wshow=contents&option_lang=rus
Адрес статьи в БИАС	http://bias.ipiran.ru/database/article/view/id/193.html	⇔	http://bias.ipiran.ru/database/article/view/id/198.html

(*а*)

Издание: английское название / русское название	Automation and remote control		Автоматика и телемеханика
Адрес журнала в Интернете	http://link.springer.com/journal/10513		http://www.mathnet.ru/php/archive.phtml?jrnid=at&wshow=contents&option_lang=rus
Адрес статьи в БИАС	http://bias.ipiran.ru/database/article/view/id/2602.html	⇔	http://bias.ipiran.ru/database/article/view/id/1868.html

(*б*)

Рис. 1 Примеры объединения БО в переводных изданиях

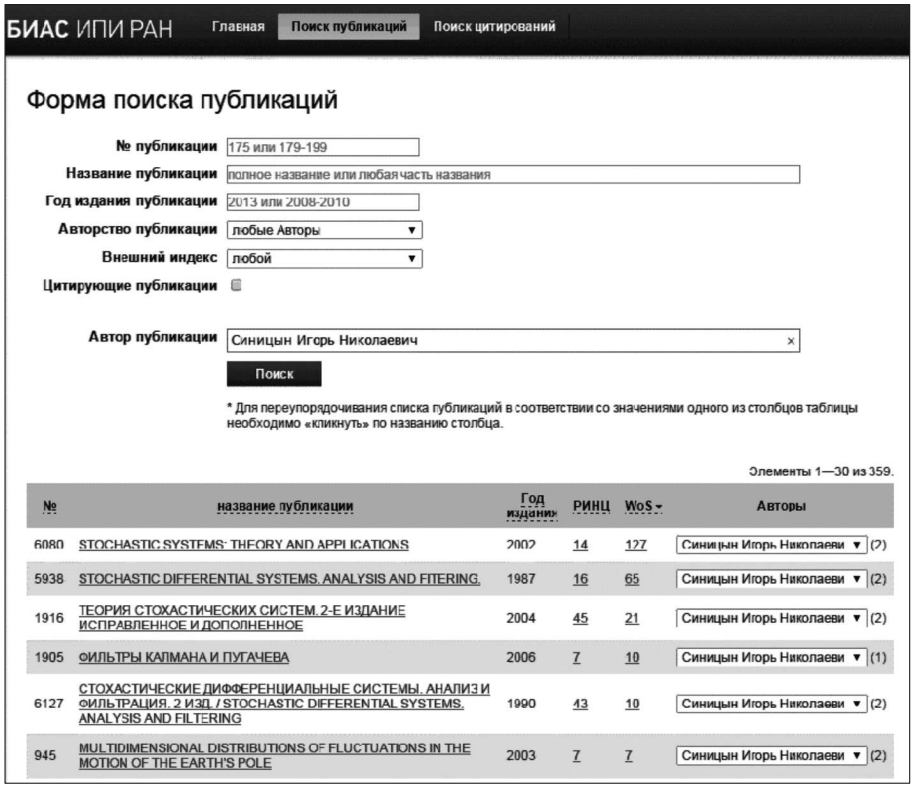


Рис. 2 Фрагмент перечня публикаций автора с указанием числа цитирований каждой статьи по версиям WoS и РИНЦ

определить уровень востребованности и значимости научной работы. Вычисляемые библиографические показатели служат количественной характеристикой продуктивности ученого. Для выполнения этой задачи в БД БИАС ведутся списки цитирований публикаций авторов института. Учитываются цитирования, проиндексированные в WoS и РИНЦ на платформе eLibrary. На рис. 2 представлен пример фрагмента перечня публикаций автора с указанием количества цитирований каждой статьи по версии WoS и РИНЦ. Количество цитирований является ссылкой на описания цитирующих статей.

К целям создания БИАС относится подготовка статистических отчетов о публикационной деятельности ИПИ РАН для различных систем учета. Требования отчетности Президиума РАН, Федерального агентства научных организаций, Министерства образования и науки различаются сочетанием параметров, отображающих публикационную активность института. К примеру, в одних случаях

необходимым показателем является количество публикаций сотрудников ИПИ РАН за 2008–2013 гг., которые цитируются в научных работах, в других — количество цитирующих статей, опубликованных в 2008–2013 гг. Для автоматизации подготовки отчетов используются соответствующие программные модули БИАС ИПИ РАН.

2 Исследование причин множественности библиографических описаний публикаций

В БД БИАС каждая публикация представлена БО, состоящим из (основных) полей: «Название публикации», «Издание», «Год» и «Авторы». На первоначальном этапе заполнения БД разработчики исходили из того, что БО взаимно однозначно определяет публикацию. Но вскоре была обнаружена проблема, связанная с тем, что БО одной и той же публикации в разных ресурсах могут отличаться значениями в одном или нескольких полях. Как следствие, занесение библиографической информации из различных источников — без применения специальных средств контроля — приводило к тому, что *разные* БО в БД определяли одну и ту же публикацию. Для корректного формирования трудов сотрудников ИПИ РАН необходимо было решить проблему, связанную с дублированием представления публикаций в форме БО в БД научных публикаций.

Причинами появления *разных* БО одной и той же статьи являются следующие обстоятельства:

(1) базы индексации научных статей трансформируют формулы, используемые в названиях публикаций, в свои внутренние стандарты, которые, в общем случае, отличаются от первоначального представления;

(2) при цитировании публикаций в переводных изданиях авторы используют разные версии переводов;

(3) к весьма распространенным ситуациям относятся неточности и искажения в описаниях цитируемых публикаций.

Рассмотрим несколько примеров порождения дублирующих (т. е. определяющих одну и ту же статью) БО, занесенных в БД научных публикаций из разных источников.

2.1 Использование формул в названиях публикаций

В случае если авторы выносят в *название* публикации формулы с использованием верхних или нижних индексов, разные БД научного цитирования индексируют эти статьи в своем внутреннем стандарте.

К примеру, статья, опубликованная в журнале «Automation and Remote Control», 2011, том 72, № 1, авторы: *Печинкин А. В., Шоргин С. Я. и др.*, под названием: «$\mathrm{Geo}_m/G/1/n$ system with LIFO discipline without interrupts and

constrained total amount of customers», проиндексирована в различных системах следующим образом:

- в WoS под названием «Geo$(m)/G/1/n$ system with LIFO discipline without interrupts and constrained total amount of customers»;
- в eLibrary под названием «GEOM$/G/1/N$ system with LIFO discipline without interrupts and constrained total amount of customers».

Формально в данном примере представлены разные БО, что приводит к дублированию представления одной и той же статьи в БД.

2.2 Использование переводных версий научных журналов

Особую сложность представляют публикации, БО которых формируется из пристатейных списков используемой литературы. Для достижения полноты наполнения БД БИАС используются списки цитирований, являющиеся обязательной частью любой статьи, доступные в таких ресурсах, как Web of Science, Springer Link [3], eLibrary. Но извлечение библиографической информации из списков цитируемых источников может привести к множественности представления одной публикации в БД БИАС. Использование в ссылках разных версий переводных изданий приводит к образованию двух и более БО одной и той же статьи. Учитывая, что переводные версии российских научных журналов выпол-

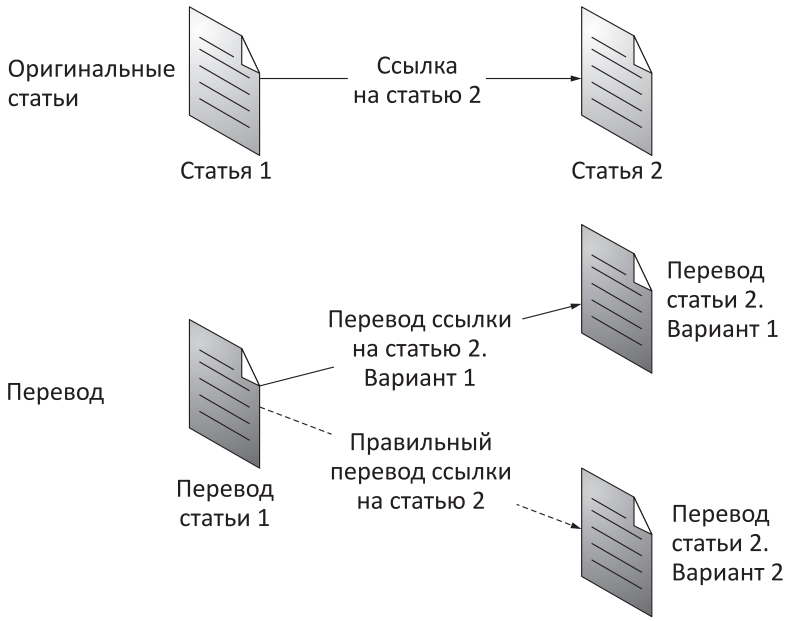


Рис. 3 Пример использования в ссылках разных версий переводных изданий

няют разные переводчики, возникновение таких коллизий — распространенный случай.

Проиллюстрируем это на примере, приведенном на рис. 3.

2.3 Неточности и искажения в цитированиях публикаций

Для получения информации о БО публикаций сотрудников ИПИ РАН из БД РИНЦ (на платформе eLibrary) использовался программный интерфейс для доступа к данным. В eLibrary поддерживается возможность включения в перечень научных статей автора публикаций из списков цитирования. Но формальной обработки на достоверность данное включение не проходит. Весьма распространенной является ситуация ошибочного цитирования: искажаются названия статей, допускаются ошибки в указании выходных данных издания. Это служит источником появления некорректных вариантов публикаций.

К примеру, статья Синицына И. Н., опубликованная в журнале «Doklady Physics», 2002, том 47, выпуск 7, оказалась в БД в трех вариантах:

- «Stochastic Model of Motion of the Pole of Deformable Earth»;
- «Stochastic Model of the Pole's Motion for Deformable Earth»;
- «A Stochastic Model of the Motion of the Deformable-Earth Pole».

3 Технология обнаружения дублирующих библиографических описаний

Выявление причин дублирования БО создало предпосылки для решения следующих задач:

(1) выявление и удаление из БД дублирующих БО;

(2) разработка технологии предотвращения появления в БД дублирующих БО.

Под термином *технология* подразумевается применение комплекса программных модулей, решающих поставленную задачу в соединении со средствами визуализации полученных результатов. Поскольку основная причина появления дублирующих БО связана с искажениями в названиях публикаций, то на первом этапе объектом исследования стало поле «Название публикации» (как части БО).

3.1 Применение методов нечеткого поиска для обнаружения искажений в названиях публикаций

Для анализа искажений в поле названия публикации использовался поиск по сходству (fuzzy string search) по алгоритму Оливера, описанному в работе [4]. В рамках данной статьи не рассматриваются причины предпочтения метода нахождения наибольшей общей подпоследовательности (LCS — longest common subsequence) другим распространенным методам, к примеру, реализующим алгоритм Левенштейна [5]. Вопросы преимущества выбранного метода станут темой следующей статьи.

Применение алгоритма Оливера позволяет находить в двух строках совпадающие фрагменты и вычислять в процентах отношение их длины к суммарной длине сравниваемых строк. Алгоритм Оливера реализован в виде стандартной PHP-функции `similar_text`. Приведем реализацию этого алгоритма на C#:

```
float similar_text (String first, String second)
 {
  int sim = similar(first.ToLower(), second.ToLower());
  return (float)(sim * 200) / (first.Length + second.Length);
 }
   int similar(String first, String second)
 {
  int p, q, l, sum;
  int pos1 = 0, pos1 = 0, max = 0;
  char[] arr1 = first.ToCharArray();
  char[] arr2 = second.ToCharArray();
  int firstLength = arr1.Length;
  int secondLength = arr2.Length;
  for (p = 0; p < firstLength; p++)
  {
   for (q = 0; q < secondLength; q++)
   {
    for (l = 0; (p + l < firstLength) && (q + l < secondLength)
     && (arr1[p + l] == arr2[q + l]); l++) ;
    if (l > max)
    {
      max = l;
      pos1 = p;
      pos2 = q;
    }
   }
  }
  sum = max;
  if (sum > 0)
  {
   if (pos1 > 0 && pos2 > 0)
   {
    sum += similar(first.Substring(0, pos1 >
     firstLength ? firstLength : pos1), second.Substring(0,
pos2 > secondLength ? secondLength : pos2));
   }
   if ((pos1 + max < firstLength) && (pos2 + max < secondLength))
   {
    sum += similar(first.Substring(pos1 + max),
     second.Substring(pos2 + max));
   }
  }
  return sum;
 }
```

Определение. Введем обозначение *индекса схожести* двух строк `Ind_Sim`, являющегося результатом выполнения функции `similar_text`. Диапазон значений: $0\% \leq$ `Ind_Sim` $\leq 100\%$.

Проиллюстрируем работу функции `similar_text` на примере описанного выше случая, когда для исходной публикации «A Stochastic Model of the Motion of the Deformable-Earth Pole» в БД были занесены искаженные варианты названия публикации.

Протокол 1.

```
String first = a stochastic model of the motion
 of the deformable-earth pole
String second = stochastic model of motion
 of the pole of deformable earth

  similar(61,58)
  f=(a stochastic model of the motion
   of the deformable-earth pole)
  s=(stochastic model of motion of the pole of deformable earth)
  max=20, pos1=2, pos2=0
  similar(39,38)
  f=(the motion of the deformable-earth pole)
  s=(motion of the pole of deformable earth)
  max=14, pos1=4, pos2=0
  similar(21,24)
  f=(deformable-earth pole)
  s=(pole of deformable earth)
  max=10, pos1=0, pos2=8
  similar(11,6)
  f=(-earth pole)
  s=( earth)
  max=5, pos1=1, pos2=1
  similar(1,1)
  f=(-)
  s=( )
  max=0, pos1=0, pos2=0
```

`Ind_Sim` $= 82{,}35\%$.

Протокол 2.

```
String first = a stochastic model of the motion
 of the deformable-earth pole
String second = stochastic model of the pole's motion
 for deformable earth
```

```
similar(61,58)
f=(a stochastic model of the motion of the deformable-earth pole)
s=(stochastic model of the pole's motion for deformable earth)
max=24, pos1=2, pos2=0
similar(35,34)
f=(motion of the deformable-earth pole)
s=(pole's motion for deformable earth)
max=11, pos1=13, pos2=17
similar(13,17)
f=(motion of the)
s=(pole's motion for)
max=7, pos1=0, pos2=7
similar(6,3)
f=(of the)
s=(for)
max=1, pos1=0, pos2=1
similar(5,1)
f=(f the)
s=(r)
max=0, pos1=0, pos2=0
similar(11,6)
f=(-earth pole)
s=( earth)
max=5, pos1=1, pos2=1
similar(1,1)
f=(-)
s=( )
max=0, pos1=0, pos2=0
```

$\texttt{Ind_Sim} = 80{,}67\%$.

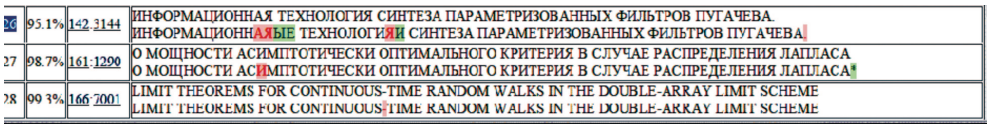
Заметим, что значение *индекса схожести* исходной статьи и ее модификаций в рассматриваемом примере не меньше $80{,}67\%$.

3.2 Этапы проверки базы данных

Выбранная методика и разработанные программные средства позволили провести ревизию БД для выявления дублирующих БО. Протоколы работы программ *определения схожести*, вычисляющих `Ind_Sim` названий статей, предоставлялись модераторам системы, которые принимали решение о достоверности той или иной информации. Для удобства работы использовались средства визуализации результатов работы функции `similar_text`:

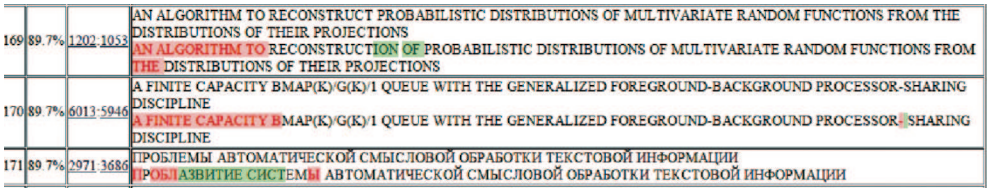
- для второй строки красным цветом подсвечиваются удаленные слова или символы;
- зеленым цветом — добавленные слова или символы;
- для обеих строк указываются ссылки на соответствующее БО в БД БИАС.

Фрагменты протоколов работы программ представлены на рис. 4.



26	95.1%	142:3144	ИНФОРМАЦИОННАЯ ТЕХНОЛОГИЯ СИНТЕЗА ПАРАМЕТРИЗОВАННЫХ ФИЛЬТРОВ ПУГАЧЕВА ИНФОРМАЦИОННАЯЫЕ ТЕХНОЛОГИЯИ СИНТЕЗА ПАРАМЕТРИЗОВАННЫХ ФИЛЬТРОВ ПУГАЧЕВА
27	98.7%	161:1290	О МОЩНОСТИ АСИМПТОТИЧЕСКИ ОПТИМАЛЬНОГО КРИТЕРИЯ В СЛУЧАЕ РАСПРЕДЕЛЕНИЯ ЛАПЛАСА О МОЩНОСТИ АСИМПТОТИЧЕСКИ ОПТИМАЛЬНОГО КРИТЕРИЯ В СЛУЧАЕ РАСПРЕДЕЛЕНИЯ ЛАПЛАСА*
28	99.3%	166:7001	LIMIT THEOREMS FOR CONTINUOUS-TIME RANDOM WALKS IN THE DOUBLE-ARRAY LIMIT SCHEME LIMIT THEOREMS FOR CONTINUOUS-TIME RANDOM WALKS IN THE DOUBLE-ARRAY LIMIT SCHEME

(*а*)



169	89.7%	1202:1053	AN ALGORITHM TO RECONSTRUCT PROBABILISTIC DISTRIBUTIONS OF MULTIVARIATE RANDOM FUNCTIONS FROM THE DISTRIBUTIONS OF THEIR PROJECTIONS AN ALGORITHM TO RECONSTRUCTION OF PROBABILISTIC DISTRIBUTIONS OF MULTIVARIATE RANDOM FUNCTIONS FROM THE DISTRIBUTIONS OF THEIR PROJECTIONS
170	89.7%	6013:5946	A FINITE CAPACITY BMAP(K)/G(K)/1 QUEUE WITH THE GENERALIZED FOREGROUND-BACKGROUND PROCESSOR-SHARING DISCIPLINE A FINITE CAPACITY BMAP(K)/G(K)/1 QUEUE WITH THE GENERALIZED FOREGROUND-BACKGROUND PROCESSOR-SHARING DISCIPLINE
171	89.7%	2971:3686	ПРОБЛЕМЫ АВТОМАТИЧЕСКОЙ СМЫСЛОВОЙ ОБРАБОТКИ ТЕКСТОВОЙ ИНФОРМАЦИИ ПРОБЛАЗВИТИЕ СИСТЕМЫ АВТОМАТИЧЕСКОЙ СМЫСЛОВОЙ ОБРАБОТКИ ТЕКСТОВОЙ ИНФОРМАЦИИ

(*б*)

Рис. 4 Фрагменты протоколов работы программ определения схожести для $95\% \leq \leq \mathsf{Ind_Sim} \leq 100\%$ (*а*) и $75\% \leq \mathsf{Ind_Sim} \leq 95\%$ (*б*)

Этап 1. Диапазон $95\% \leq \mathtt{Ind_Sim} \leq 100\%$.

На этом этапе были выявлены случаи, которые можно отнести к ошибкам правописания.

Этап 2. Диапазон $75\% \leq \mathtt{Ind_Sim} \leq 95\%$.

На этом этапе объектом внимания были БО публикаций, названия которых различались на некоторое количество терминов. Надо заметить, что в этот диапазон попадают и все публикации, соотносящиеся с одной тематикой. Как правило, в этом случае авторы дают название публикаций путем комбинации одних и тех же терминов.

Проиллюстрируем это на примере двух статей Шевцовой И. Г., опубликованных в журнале «Доклады Академии наук», 2012, том 443, выпуск 6: «О точности нормальной аппроксимации для сумм независимых случайных величин» и «О точности нормальной аппроксимации для сумм независимых симметричных случайных величин».

Для этих статей $\mathtt{Ind_Sim} = 91{,}92\%$.

Анализ протоколов работы программ определения схожести позволил получить два результата:

(1) из БД было удалено $\sim 6\%$ БО;

(2) было установлено, что значение *индекса схожести* $< 75\%$ может не рассматриваться при поиске дублирования БО. В большинстве случаев значение `Ind_Sim` менее 75% означает замену двух или более терминов в названиях публикаций. Учитывая специфику БД БИАС, в которой названия публикаций состоят в среднем из 9 слов (на данный момент максимальная длина названия составляет 46 слов), замена двух–трех терминов приводит к изменению смыслового значения.

4 Технология предотвращения дублирования библиографических описаний

Алгоритм работы программной реализации технологии представим на языке описания формальной модели данных, которая заложена в структуру БД БИАС ИПИ РАН.

4.1 Формальная модель данных

Концептуальная модель данных подробно описана в [6]. Формальное представление БО научных публикаций базируется на принципах адаптивной фасетной навигации, разработанной авторами [7], в которой используется описание объекта как множество совокупностей: {фасетный признак: список значений} [8].

Фасетный признак определяется компонентом БО, к примеру: F_{title} — название публикации; F_{journal} — название издания; F_{year} — год издания; F_{authors} — автор. В рамках данной статьи опускается множество других фасетных признаков как не актуальных для данного исследования. Поэтому представляющий интерес фрагмент БО можно представить в виде следующей *формулы определения БО*:

$\Phi = \{F_{\text{title}} = \text{значение}$
$+ F_{\text{journal}} = \text{идентификатор}_{\text{journal}}$
$+ F_{\text{volum}} = \text{значение}$
$+ F_{\text{number}} = \text{значение}$
$+ F_{\text{year}} = \text{значение}$
$+ F_{\text{authors}} = \text{ссылка на список объектов типа «автор»}\}.$

4.2 Алгоритм работы программного модуля предотвращения дублирования библиографических описаний

Целью работы программного модуля предотвращения дублирования БО является выявление тех случаев, которые могут привести к коллизиям и требуют вмешательства модераторов системы. Модераторам системы доступны первоисточники научных журналов, по которым можно установить истинное БО публикации.

Используем обозначения:

– Φ^{new} — новое БО;

– $\Phi^{\exists}$ — существующее на текущий момент в базе данных БО.

На рис. 5 представлен алгоритм работы модуля предотвращения дублирования БО. Здесь `Ind_Sim` — индекс схожести $F^{\text{new}}_{\text{title}}$ и $F^{\exists}_{\text{title}}$, а ID — уникальный идентификатор, который присваивается каждому из возможных значений определенного признака на этапе программной реализации.

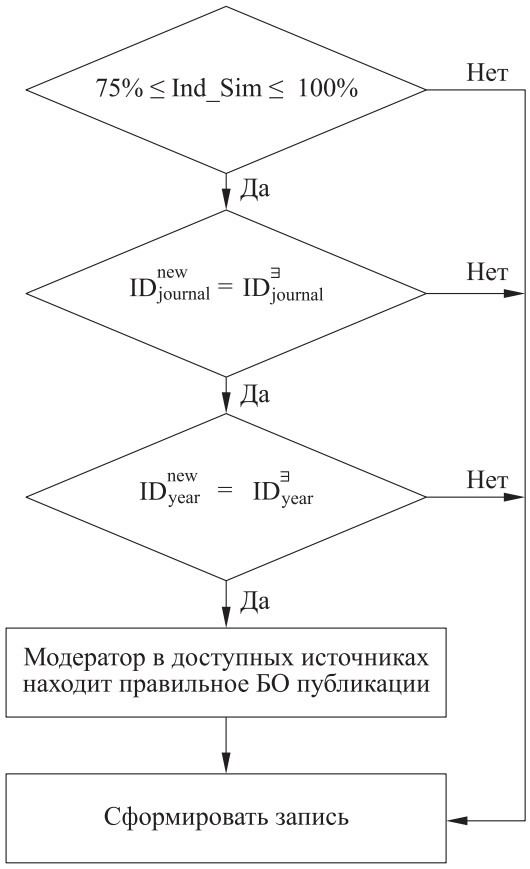


Рис. 5 Алгоритм работы модуля предотвращения дублирования БО

4.3 Программная реализация

Центральной частью разработанной технологии является использование программного модуля определения схожести, обращение к которому происходит при занесении новой библиографической записи посредством вызова функции «*Проверить название публикации на похожие*». Результатом ее работы является список БО публикаций, названия которых удовлетворяют условию $75\% \leq \texttt{Ind_Sim} \leq 100\%$.

Список формируется с использованием элементов визуализации работы программных модулей. При разработке технологии предотвращения дублирования большое внимание было уделено средствам визуализации протоколов работы программного модуля определения схожести. Это связано с тем, что результаты

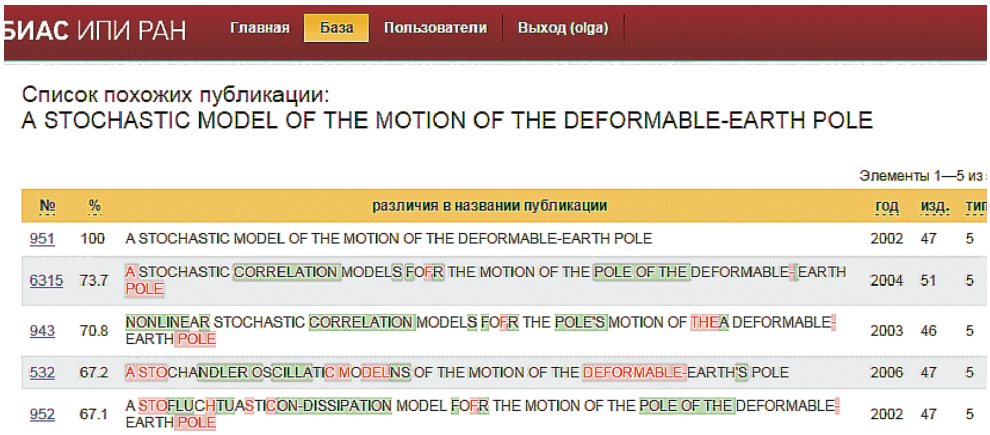


Рис. 6 Результат работы программного модуля определения схожести

работы этих модулей необходимы модераторам системы для принятия решения, какое из следующих возможных действий надо выполнить:

1. Занести новую запись.
2. Проигнорировать новую запись как ошибочную.
3. Ошибочной является существующая запись, ее необходимо заменить новой.

Чтобы определить, какое действие необходимо выполнить, модераторы должны исследовать первоисточники, а именно: каталоги журналов, в которых опубликована анализируемая статья. Поэтому остановимся на применяемых средствах визуализации. Для иллюстрации действующих механизмов приведем несколько примеров, используя описанный выше случай множественности БО публикации И. Н. Синицына.

1. Название новой публикации «A stochastic model of the motion of the deformable-Earth pole». Список БО представлен на рис. 6.

   `Ind_Sim` с существующим БО реальной публикации равен 100%.

   Красным цветом подсвечиваются удаленные символы по отношению к оригиналу, зеленым — новые символы. Левый столбец определяет идентификатор публикации в БД, который служит ссылкой для перехода к описанию данной публикации.

2. Название новой публикации (искаженное) «Stochastic Model of the Pole's Motion for Deformable Earth». Список БО представлен на рис. 7.

   Ind_Sim с существующим БО реальной публикации равен 79,3%.

3. Название новой публикации (искаженное) «Stochastic Model of Motion of the Pole of Deformable Earth». Список БО представлен на рис. 8.

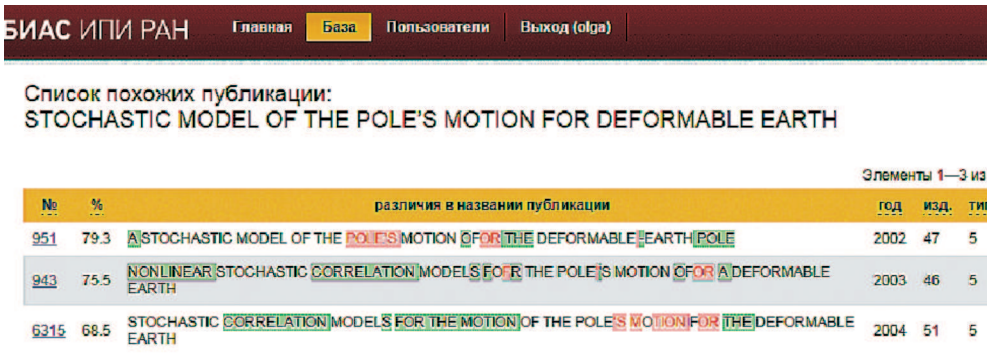


Рис. 7 Результат работы программного модуля определения схожести

Рис. 8 Результат работы программного модуля определения схожести

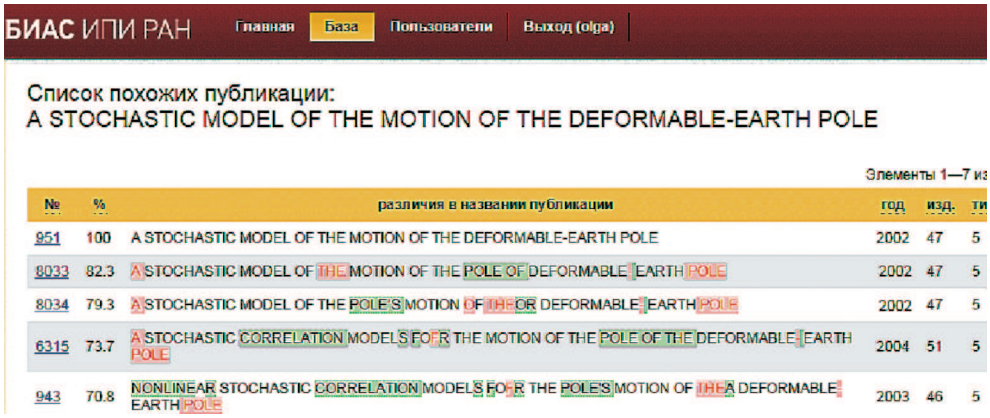


Рис. 9 Результат работы программного модуля определения схожести

`Ind_Sim` с существующим БО реальной публикации равен 82,3%.

Любопытно, что для данного примера более высокий `Ind_Sim` — с публикацией № 6315 «Stochastic correlation models for the motion of the pole of the deformable earth», опубликованной в 2004 г. Но так как с этой публикацией у искомой различаются $F_{\text{journal}} = \text{ID}_{\text{journal}}$ и $F_{\text{year}} = \text{ID}_{\text{year}}$, то искомой является публикация № 951.

Надо заметить, что значение индекса схожести правильной статьи и ее модификаций лежит в пределах $79{,}3\% \leq \texttt{Ind_Sim} \leq 100\%$, что можно увидеть на рис. 9.

5 Заключение

Помимо БД научного цитирования Web of Science, Scopus, РИНЦ существенным источником информации для заполнения БД БИАС являются БО цитируемых публикаций, указанные в пристатейных списках используемой литературы. Эти источники информации представляют особый интерес в связи с тем, что вышеперечисленные системы индексируют только журнальные публикации и, не в полном объеме, труды конференций. Для формирования полного перечня научных трудов сотрудников института, включающего книги, монографии, доклады на конференциях, производится анализ цитируемых источников с целью получения информации обо всех публикациях авторов ИПИ РАН. Но ошибки и специфические особенности оформления БО цитируемых источников создают множество коллизий, порожденных появлением в БД нескольких ошибочных БО одной и той же публикации.

Для достижения достоверности наполнения БД научных публикаций БИАС ИПИ РАН была разработана технология предотвращения дублирования БО, опирающаяся на использование двух компонентов: программных модулей *определения схожести* по алгоритму Оливера и средств визуализации полученных результатов. Сложность алгоритма Оливера составляет $O(N^3)$, где N — длина более длинной из двух строк. Сравнение эталона необходимо выполнить для каждой уникальной статьи. Поэтому в разработанной технологии сложность алгоритма определяется как $K \cdot O(N^3)$, где K — число уникальных статей в БД. При конфигурации сервера:

- процессор Intel Xeon E5420 2,5 ГГц,
- память 8 ГБайт,
- MS SQL 2005

и количестве записей в БД $= 10\,000$ среднее время вычисления индекса схожести при занесении новой записи $= 130$ мс, максимальное время $= 466$ мс.

Визуализация протоколов работы программных систем позволяет модераторам БИАС ИПИ РАН получать полную информацию в форме, удобной как для принятия решений, так и для работ по дальнейшему устранению возникших

коллизий в БД. Предопределенные поля визуализации протоколов позволяют перейти на БО публикации в режиме редактирования. Использование этой технологии гарантирует то, что информация в БД БИАС ИПИ РАН будет полной и достоверной, что соответствует основным требованием к БД.

Литература

1. Библиографическая информационно-аналитическая система ИПИ РАН. http://bias.ipiran.ru.
2. ГОСТ 7.1-2003 Библиографическая запись. Библиографическое описание. http://diss.rsl.ru/datadocs/doc_291wu.pdf.
3. База данных SpringerLink. http://link.springer.com.
4. *Oliver I.* Programming classics: Implementing the World's best algorithms. — Prentice Hall PTR, 1994. 386 p.
5. *Левенштейн В. И.* Двоичные коды с исправлением выпадений, вставок и замещений символов // Докл. Акад. наук СССР, 1965. Т. 163. Вып. 4. С. 845–853.
6. *Заикин М. Ю., Обухова О. Л., Соловьев И. В.* Библиографическая информационно-аналитическая система ИПИ РАН // Системы и средства информатики, 2014. Т. 24. Вып. 1. С. 244–259.
7. *Обухова О. Л., Бирюкова Т. К., Гершкович М. М., Соловьев И. В.* Модель адаптивной фасетной навигации в открытых электронных коллекциях // Системы и средства информатики, 2008. Т. 18. Вып. 1. С. 171–185.
8. *Обухова О. Л., Бирюкова Т. К., Гершкович М. М., Соловьев И. В.* Модель фасетного информационного поиска в коллекции научных материалов // Системы и средства информатики, 2009. Т. 19. Вып. 2. С. 163–174.

Поступила в редакцию 16.12.14

TECHNOLOGY FOR PREVENTION OF DUPLICATION OF BIBLIOGRAPHIC DESCRIPTIONS IN THE SCIENTIFIC DATABASE BIAS IPI RAS

M. Yu. Zaikin, V. S. Dolgopolov, O. L. Obuhova, and I. V. Soloviev

Institute of Information Problems, Federal Research Center "Computer Science and Control," Russian Academy of Sciences, 44-2 Vavilov Str., Moscow 119333, Russian Federation

Abstract: The paper considers the developed technology aimed at avoiding duplication of bibliographic descriptions in the scientific database Bibliographic Information-Analytical System (BIAS) of IPI RAS. The analysis of the reasons of duplications is given. The constituent parts of the developed software are the modules of *definition of similarity* using the methods of fuzzy search based on the Oliver algorithm and the modules of visualization of the results which are

built into the system at the level of formation of the database content. Program modules of visualization allow moderators of BIAS IPI RAS to receive full information about the conflicts. They will be able to decide on further action using additional information. The concept of *similarity index* used in the software modules of *definition of similarity* is introduced. The paper considers the formal data model underlying construction of the database, built on the principles of facet navigation. Application of the developed software made it possible to detect and remove duplicate bibliographic descriptions in the scientific database.

Keywords: similarity index; software modules of definition of similarity; method of fuzzy search on the Oliver algorithm; facet navigation

DOI: 10.14357/08696527150111

References

1. Scientific database Bibliographic Information-Analytical System (BIAS) of IPI RAS. Available at: http://bias.ipiran.ru (accessed April 7, 2015).
2. GOST 7.1-2003. Bibliograficheskaya zapis'. Bibliograficheskoe opisanie [Bibliographic record. Bibliographic description]. Available at: http://diss.rsl.ru/datadocs/doc_291wu.pdf (accessed April 7, 2015).
3. Database SpringerLink. Available at: http://link.springer.com/ (accessed April 7, 2015).
4. Oliver, I. 1994. *Programming classics: Implementing the World's best algorithms*. Prentice Hall PTR. 386 p.
5. Levenshteyn, V. I. 1965. Dvoichnye kody s ispravleniem vypadeniy, vstavok i zameshcheniy simvolov [Binary codes with correction for deletions, insertions, and substitutions of symbols]. *Dokl. Akad. Nauk SSSR* 163(4):845–833.
6. Zaikin, M. Yu., O. L. Obuhova, and I. V. Soloviev. 2014. Bibliograficheskaya informatsionno-analiticheskaya sistema IPI RAN [Bibliographic information-analytical system (BIAS) of IPI RAS]. *Sistemy i Sredstva Informatiki — Systems and Means of Informatics* 24(1):244–258.
7. Chochia, A. P., I. V. Soloviev, O. L. Obuhova, T. K. Biryukova, and M. M. Gershkovich. 2008. Model' adaptivnoy fasetnoy navigatsii v otkrytykh elektronnykh kollektsiyakh [The model for adaptive facet navigation in open digital collections]. *Sistemy i Sredstva Informatiki — Systems and Means of Informatics* 18(1):171–185.
8. Obuhova, O., I. Soloviev, T. Biryukova , M. Gershkovich, and A. Chochia. 2009. Model' fasetnogo informatsionnogo poiska v kollektsii nauchnykh materialov [A model of facet information retrieval within the collection of scientific materials]. *Sistemy i Sredstva Informatiki — Systems and Means of Informatics* 19(2):163–174.

Received December 16, 2014

Contributors

Zaikin Maxim Yu. (b. 1973) — scientist, Institute of Information Problems, Federal Research Center "Computer Science and Control," Russian Academy of Sciences, 44-2 Vavilov Str., Moscow 119333, Russian Federation; maxim@amsd.com

Dolgopolov Vyacheslav S. (b. 1962) — scientist, Institute of Information Problems, Federal Research Center "Computer Science and Control," Russian Academy of Sciences, 44-2 Vavilov Str., Moscow 119333, Russian Federation; slava@amsd.com

Obuhova Olga L. (b. 1952) — scientist, Institute of Information Problems, Federal Research Center "Computer Science and Control," Russian Academy of Sciences, 44-2 Vavilov Str., Moscow 119333, Russian Federation; obuhova@amsd.com

Soloviev Ivan V. (b. 1976) — programmer I, Institute of Information Problems, Federal Research Center "Computer Science and Control," Russian Academy of Sciences, 44-2 Vavilov Str., Moscow 119333, Russian Federation; soloviev@amsd.com

ПОСТРОЕНИЕ СИСТЕМЫ УПРАВЛЕНИЯ ПОТОКАМИ ПАЦИЕНТОВ

Г. Я. Илюшин[1], *В. И. Лиманский*[2]

Аннотация: Рассмотрена организационная модель системы здравоохранения в части оказания населению первичной медицинской помощи и складывающаяся в процессе ее реформирования иерархическая трехуровневая структура лечебно-профилактических учреждений (ЛПУ). Выполнен анализ существующих способов маршрутизации пациентов, включая самозапись пациентов в ЛПУ первого уровня (прикрепляемый контингент), перенаправление пациентов в ЛПУ второго и третьего уровня с помощью выдаваемых врачами талонов и направлений. Представлен анализ преимуществ и недостатков этих способов. Рассмотрены применяемые для маршрутизации пациентов электронные системы управления потоками пациентов и некоторые проблемы их внедрения, в частности внедрение «Единой медицинской информационно-аналитической системы» (ЕМИАС) в г. Москве. Проведен анализ электронных систем управления потоками пациентов (СУПП), выделены их основные компоненты и предложена структура построения такой системы, основанная на принципах единой точки взаимодействия компонентов распределенных систем. Рассмотрены процессы, протекающие в системе при реализации основных прецедентов, и определены потоки данных между компонентами системы, возникающие при исполнении этих процессов. Рассмотрены также подходы к организации электронного документооборота между ЛПУ с использованием электронных направлений.

Ключевые слова: медицинские информационные системы (МИС); системы управления потоками пациентов (СУПП); лечебно-профилактические учреждения (ЛПУ); распределенные информационные системы; диаграммы последовательностей

DOI: 10.14357/08696527150112

1 Введение

Первичная медико-санитарная помощь является основой системы оказания медицинской помощи и включает в себя мероприятия по профилактике, диагностике, лечению заболеваний и состояний, медицинской реабилитации. Существовавшая организационная модель не могла в полной мере обеспечить возможности получения пациентами качественной медицинской помощи. В рамках

[1]Институт проблем информатики Федерального исследовательского центра «Информатика и управление» Российской академии наук, ilushin@ipiran.ru

[2]Институт проблем информатики Федерального исследовательского центра «Информатика и управление» Российской академии наук, vlimansky@ipiran.ru

реформирования системы здравоохранения осуществляется поэтапное формирование трехуровневой системы оказания первичной медико-санитарной помощи в г. Москве [1].

Первый уровень представлен амбулаторно-поликлиническими учреждениями, оказывающими первичную врачебную медико-санитарную помощь и наиболее востребованные виды первичной специализированной помощи. Эти учреждения имеют прикрепленное население и работают по территориально-участковому принципу. В них оказывается медицинская помощь по специальностям терапия, хирургия, неврология, кардиология, оториноларингология, офтальмология, эндокринология, инфекционные болезни, выполняются лабораторные и инструментальные исследования и другие медицинские процедуры.

Второй уровень — амбулаторно-поликлинические учреждения, оказывающие первичную специализированную медико-санитарную помощь. Эти учреждения оснащены оборудованием экспертного класса, имеют в своем составе дневные стационары, амбулаторно-хирургические центры и другие специализированные подразделения. Для данных учреждений выбран термин «амбулаторный центр». За каждым амбулаторным центром функционально закреплены несколько учреждений первого уровня с учетом транспортной и территориальной доступности. Учреждения первого уровня получили статус филиалов амбулаторного центра.

Третий уровень образуют консультативно-диагностические отделения в городских стационарах. Их основные функции — оказание консультативной помощи, принятие согласованных решений по направлению пациентов для оказания стационарной специализированной помощи, обеспечение медицинских услуг и диагностических исследований, отсутствующих в амбулаторных центрах.

Сходные процессы реформирования системы лечебных учреждений для оптимизации процессов оказания амбулаторно-поликлинической помощи населению путем специализации ЛПУ на оказание конкретных видов медицинских услуг происходят и в других регионах РФ. В рассмотренной трехуровневой системе оказания первичной медицинской помощи дорогостоящее медицинское оборудование концентрируется в отдельных медицинских учреждениях, в основном в амбулаторных и диагностических центрах, что способствует более рациональному его использованию и снижению затрат на оказание медицинских услуг при сохранении должного уровня их качества.

В соответствии с законодательством [2] первичная медико-санитарная помощь оказывается гражданам по территориально-участковому принципу в определенных медицинских организациях. В результате пациент оказывается прикрепленным к определенной и при этом только к одной единственной поликлинике, куда он может самостоятельно обратиться для получения медицинских услуг по системе обязательного медицинского страхования (ОМС) в соответствии с перечнем специальностей врачей, открытых пациентам для самозаписи. На специальности и процедуры, закрытые для самозаписи в медицинских организациях первого уровня, в медицинские организации второго и третьего уровней (амбулаторные и диагностические центры) или в другие специализированные медицинские ор-

ганизации, участвующие в процессе оказания первичной медицинской помощи, пациент самостоятельно записаться не может. Направить пациента к специалистам в указанные медицинские организации или, как часто говорят, к узкопрофильным специалистам может только врач поликлиники, к которой пациент прикреплен.

2 Применение талонов и направлений для маршрутизации пациентов

Маршрутизация пациентов между ЛПУ в процессе оказания первичной медицинской помощи осуществляется на базе медицинских информационных систем (МИС) ЛПУ с использованием различных систем и приложений управления потоками пациентов. Системы управления потоками пациентов решают задачи самозаписи пациентов в ЛПУ первого уровня и направления врачами пациентов к узкопрофильным специалистам. Самозапись пациентов в ЛПУ первого уровня, к которым эти пациенты прикреплены, может осуществляться различными способами, в том числе:

- через работников регистратуры поликлиники при непосредственной явке пациента в ЛПУ или по телефонному звонку;
- при посещении районного многофункционального центра (МФЦ) или другой точки предоставления госуслуг;
- с помощью информационных киосков, установленных в ЛПУ;
- с использованием публичных сайтов, осуществляющих самозапись пациентов в ЛПУ данного города или региона (например, портала госуслуг);
- с помощью мобильных приложений, устанавливаемых на мобильных устройствах пациентов (смартфонах, планшетах и других гаджетах).

Направление врачом пациентов к узкопрофильным специалистам данного или другого ЛПУ может осуществляться путем выдачи талонов на прием, где указаны дата и время посещения специалиста. Такой способ направления пациентов на прием к узкопрофильным специалистам не всегда удобен как врачам, так и пациентам по следующим причинам:

(1) врач при выдаче талона вынужден отвлекаться от лечебного процесса и тратить дополнительное время на оформление талона в условиях жесткого лимита времени приема (как правило, 12–15 мин на пациента), он должен запросить в системе расписание приема по нужной специальности, согласовать дату и время приема с пациентом. Кроме того, запись пациента к нужному специалисту на требуемую процедуру или исследование в момент приема пациента может оказаться невозможной ввиду отсутствия свободных талонов. В этом случае врач вынужден пригласить пациента на новый прием либо принять на себя обязательства выписать талон пациенту в его

отсутствие. В последнем случае дату и время приема врач выбирает самостоятельно, без согласования с пациентом, а пациент тем или иным способом должен отслеживать появление для него нового талона на прием;

(2) пациент при выдаче талона врачом не видит всех возможных вариантов времени посещения нужного специалиста и вынужден соглашаться с одним из вариантов, предложенным ему врачом, и, как следствие, не может выбрать наиболее удобное ему время посещения специалиста. Если впоследствии выяснится, что пациент по каким-либо причинам не может посетить врача в указанное в талоне время, пациент будет вынужден этот талон аннулировать и снова идти на прием к врачу за новым талоном, тратя свое время и создавая избыточную нагрузку для врача.

Другой известный способ перенаправления врачом пациентов к узкопрофильным специалистам — путем выдачи электронных направлений — свободен от перечисленных недостатков. В электронном направлении указывается только ЛПУ и профиль специалиста, которого пациент должен посетить, а дату и время приема пациент, получив направление, выберет позже при самостоятельном оформлении талона на прием. Иными словами, электронное направление «открывает» пациенту профили специалистов, ранее недоступные ему для самозаписи. В данном случае врач, выдающий направление, не отвлекается на лишние рутинные процедуры, а возможность выдачи направления не зависит от наличия или отсутствия в тот момент свободных мест на прием у нужного специалиста. Если впоследствии пациент по каким-либо причинам аннулирует полученный талон, то он сможет повторно записаться на прием к нужному узкопрофильному специалисту, используя то же самое электронное направление.

В различных городах и регионах страны реализуются свои СУПП, в Москве и Московской области — это ЕМИАС, в Пермском крае и Волгоградской области — это «Электронная регистратура», в Республике Мордовия и Бурятия — это «Единая электронная регистратура» и т. д. В одних системах реализованы только функции самозаписи пациентов, в других помимо этого — функции перенаправления врачами пациентов путем выдачи талонов, в третьих — выдача (врачами) и запись на прием (пациентами) по электронным направлениям.

Системы управления потоками пациентов являются распределенными многокомпонентными системами. В их состав входят как новые, специально создаваемые для этой системы компоненты — точки доступа пациентов к системе, реализуемые в виде публичных интернет-сайтов и приложений мобильных устройств, так и унаследованные компоненты — МИС ЛПУ, предоставляющие доступ к системе врачам и работникам регистратур ЛПУ. Все компоненты СУПП совместно участвуют в реализации процессов создания, использования и ликвидации основных артефактов (талонов и электронных направлений), находящихся под управлением субъектов двух категорий — врачей и пациентов. В таких системах необходимо тщательно определять потоки данных между компонентами системы и процессы управления этими потоками. Неудачные или плохо проду-

манные решения могут привести к тупиковым ситуациям при развитии системы, потребовать ее перепроектирования.

Показателен в этом смысле процесс внедрения ЕМИАС. На первом этапе ее внедрения (2013 г.) было реализовано два вида точек доступа, реализующих функции самозаписи пациентов на прием к врачам: публичные интернет-сайты и информационные киоски, устанавливаемые в ЛПУ. На следующем этапе (вторая половина 2013 г.) пациентам в качестве нового типа точки доступа было предложено мобильное приложение, реализующее помимо функции самозаписи функцию записи на прием по выдаваемым врачами электронным направлениям. В очередной версии мобильного приложения ЕМИАС (версия 3.0 от 30 апреля 2014 г.) по результатам жалоб пользователей на неправильную работу приложения функция записи по электронному направлению была исключена. Если за полгода не удалось устранить ошибки, то здесь, вероятно, проблема кроется не только в дефектах самого мобильного приложения. Далее рассматривается возможная структура СУПП, организация в ней потоков данных и бизнес-процессов.

3 Особенности реализации системы управления потоками пациентов

Система управления потоками пациентов находится под управлением субъектов двух категорий — это врачи и пациенты. Врачи выполняют свои функции по маршрутизации пациентов (выдача талонов и электронных направлений), используя функционал рабочих мест МИС. Пациенты осуществляют получение талонов на прием в режиме самозаписи или по электронным направлениям в различных точках доступа к СУПП либо непосредственно — с помощью публичных сайтов, мобильных приложений или информационных киосков, либо опосредованно — через работников регистратур ЛПУ и сотрудников центров обслуживания населения. Работники регистратуры при выполнении своих функций используют функционал рабочих мест МИС ЛПУ, а сотрудники центров услуг — специализированные сайты или иные приложения. Таким образом, в состав системы управления потоками пациентов входят следующие компоненты:

- МИС ЛПУ;
- публичные и специализированные сайты;
- мобильные приложения.

Информационные киоски, располагаемые в вестибюлях ЛПУ, не являются самостоятельными объектами, поскольку функционируют в рамках МИС соответствующих ЛПУ, и могут рассматриваться как своеобразные рабочие места пациентов этих МИС.

Пациенты и врачи создают талоны во взаимодействии с МИС тех ЛПУ, где впоследствии будут оказаны заказанные медицинские услуги и где эти вновь созданные талоны сохраняются. В результате талоны одного пациента оказываются, в общем случае, рассредоточены по многим ЛПУ. Между тем пациенту,

обратившемуся к СУПП через любую точку доступа (сайт, мобильное приложение, информационный киоск), должен предоставляться актуальный список всех имеющихся у него на данный момент талонов, с тем чтобы он мог просмотреть эти талоны (куда и когда надо идти на прием), распечатать или аннулировать их. Похожая ситуация складывается и с электронными направлениями. Кроме того, электронные направления, которые создаются в ЛПУ, направляющих пациента, должны быть переданы в ЛПУ предоставления услуг, с тем чтобы МИС этих ЛПУ могли на основании этих направлений выдать пациентам нужные талоны, погасить соответствующие направления после выдачи по ним талонов или возобновить действие направлений при аннулировании полученных по ним талонов.

Сформировать актуальные списки талонов и электронных направлений путем последовательного опроса МИС всех ЛПУ, расположенных в ареале действия системы (в некотором регионе, районе или городе), крайне затратно, долго и практически невозможно. Поэтому в системе необходимо создать еще один компонент — сервер-накопитель, назначением которого будет хранение и поддержание актуального состояния списков всех артефактов, выданных пациентам в этой системе.

Из сказанного следует, что МИС ЛПУ, функционирующие в рамках СУПП, должны взаимодействовать с другими МИС этой системы, с приложениями конечных пользователей, реализующих функционал точек доступа пациентов в систему, и с сервером-накопителем. Приложения конечных пользователей (сайты и мобильные приложения) должны взаимодействовать с сервером-накопителем и всеми МИС, входящими в систему. Характер взаимодействия компонентов такой системы можно определить, если не как «все со всеми», то во всяком случае как «многие ко многим». Учитывая, что МИС ЛПУ могут предлагать для взаимодействия различные программные интерфейсы, поскольку по отношению к создаваемой СУПП являются унаследованными, реализовать взаимодействие компонентов такой системы было бы крайне тяжело, а сама система получилась бы неуправляемой и неустойчивой. Исходя из сказанного, взаимодействие компонентов такой системы целесообразно осуществлять через специальный шлюз в соответствии с принципами интеграции информационных систем через «единый узел» [3]. Функции такого шлюза можно возложить на введенный ранее в систему сервер-накопитель, который после этого может именоваться как шлюз-накопитель СУПП. Протоколы взаимодействия с мобильными приложениями и аналогичные протоколы взаимодействия с публикационными сайтами, как правило, отличаются по программной реализации. Эту функцию также можно возложить на шлюз-накопитель СУПП. Структурная схема такой системы приведена на рис. 1.

Цифры на стрелках на рис. 1 обозначают разные виды взаимодействий между компонентами системы:

Взаимодействие вида 1 осуществляется между приложениями конечных пользователей и МИС ЛПУ в процессе получения пациентами талонов на

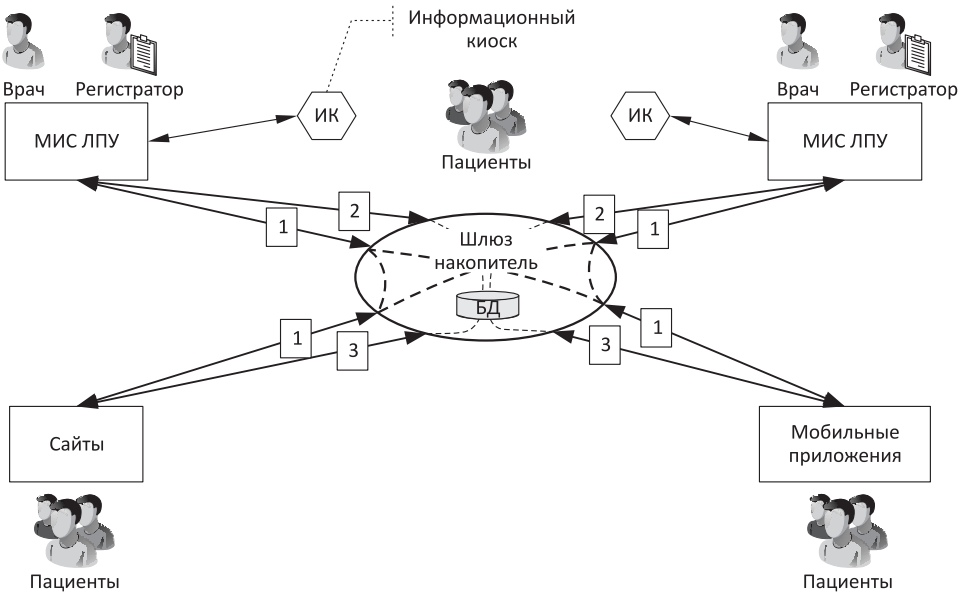


Рис. 1 Структурная схема СУПП

прием в режиме самозаписи или по электронным направлениям. При данном виде взаимодействия шлюз-накопитель выполняет функции коммутации между приложениями и МИС, осуществляет преобразование программных протоколов, оставаясь прозрачным для сторон взаимодействия.

Взаимодействие вида 2 осуществляется между МИС и шлюзом-накопителем при создании и актуализации списков артефактов. При данном виде взаимодействия МИС передают шлюзу-накопителю вновь создаваемые талоны и электронные направления, а также информацию об аннулировании талонов, погашении направлений и возобновлении их действия. Кроме этого шлюз-накопитель осуществляет доставку электронных направлений МИС ЛПУ предоставления услуги, получаемые им от ЛПУ, создающих эти направления.

Взаимодействие вида 3 осуществляется между приложениями конечных пользователей и шлюзом-накопителем при запросе приложениями актуальных списков талонов и электронных направлений. Кроме того, данный вид взаимодействия осуществляется при авторизации приложений в СУПП.

Основными действиями, выполняемыми пациентом в СУПП, или, пользуясь терминологией универсального процесса проектирования (Unified Process) [4], основными прецедентами (Use case), являются запись пациентом на прием к врачу в режиме самозаписи или по электронным направлениям. Эти два прецедента очень похожи, отличие заключается в том, что в случае самозаписи пациент сначала выбирает ЛПУ, затем запрашивает список профилей медицинских спе-

Прецедент: CreateCouponOnAssignment
Краткое описание: Создание талона по направлению
Главные актеры: Пациент
Второстепенные актеры: Нет
Предусловия: 1. Электронное направление выдано пациенту
Основной поток: 1. Пациент запрашивает у СУПП список действующих направлений 2. Пациент запрашивает из ЛПУ список врачей 3. Пациент запрашивает из ЛПУ даты и время приема выбранного врача 4. Пациент дает команду на создание талона, МИС проверяет параметры для создания талона 5. МИС создает талон на прием к врачу и заносит талон и электронное направление в реестр ЛПУ 6. МИС передает талон на хранение шлюзу-накопителю СУПП
Постусловия: 1. Талон занесен в реестр ЛПУ
Альтернативные потоки: CouponCancel

Рис. 2 Спецификация прецедента записи пациента на прием к врачу по электронному направлению

циалистов и далее — список врачей по выбранной специальности. В случае записи на прием по направлению процесс получения талона начинается сразу с запроса списка врачей, поскольку ЛПУ и профиль медицинского специалиста уже заданы в электронном направлении. Спецификация прецедента записи пациента на прием к врачу по электронному направлению приведена на рис. 2.

Как видно из описания основного потока прецедента, на пятом шаге происходит создание талона на прием, который вместе с электронным направлением, на основании которого этот талон был создан, заносится в реестр МИС ЛПУ, а затем вновь созданный талон (шаг 6 основного потока) передается на сохранение в хранилище шлюза-накопителя. В случае, когда к моменту попытки создания талона время приема у заданного врача оказывается занято другим пациентом, который успел получить талон на это же время, выполнение основного потока прерывается и возникает альтернативный поток CouponCancel. В этом случае талон не создается, а соответствующее электронное направление остается в хранилище шлюза-накопителя.

Диаграмма последовательности действий (sequence diagram) СУПП процесса реализации прецедента записи пациента на прием по электронному направлению приведена на рис. 3.

При намерении пациента записаться в ЛПУ по направлению приложение обращается к шлюзу-накопителю с запросом списка выданных пациенту электронных направлений. Шлюз-накопитель выбирает из своих баз данных (БД)

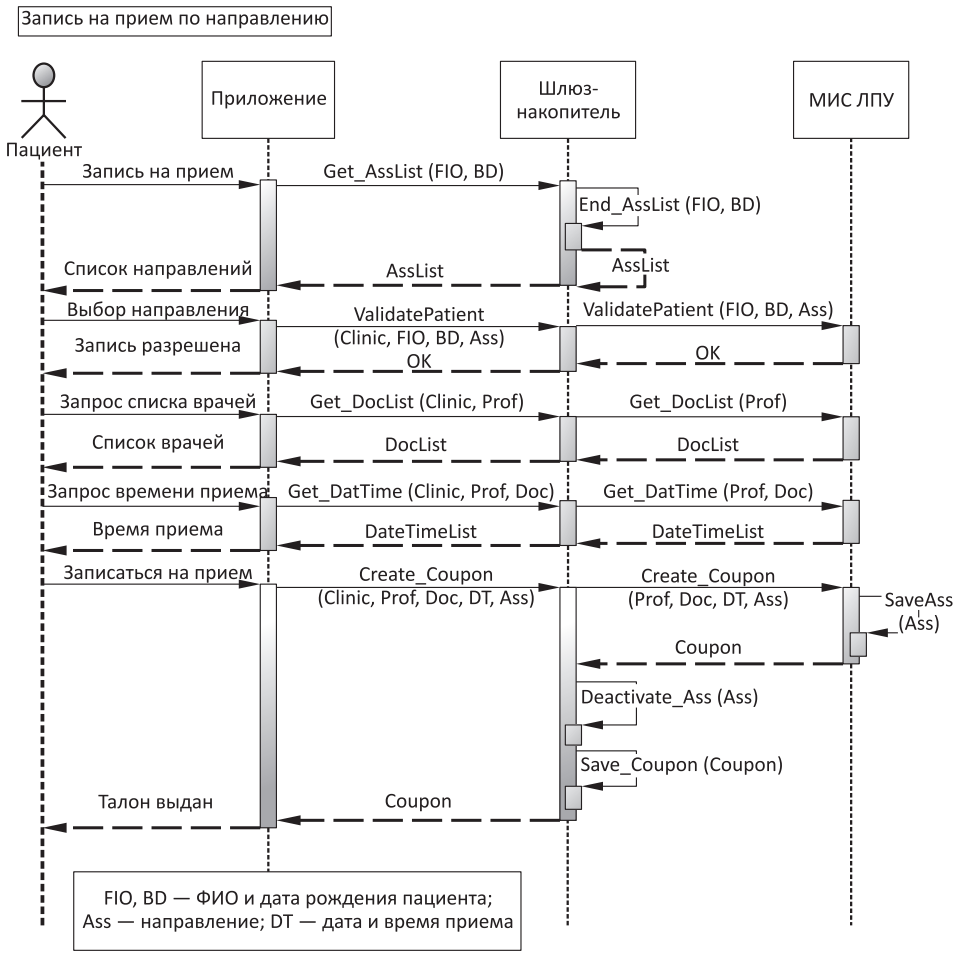


Рис. 3 Диаграмма последовательности действий в процессе записи пациента на прием по электронному направлению

все направления данного пациента, срок действия которых не истек, и возвращает полученный список приложению. После выбора пациентом конкретного направления приложение инициирует запрос на проверку правомерности обращения пациента в данное ЛПУ для получения услуг по направлению. На данной диаграмме процесс записи пациента на прием начинается сразу с запроса списка медицинских специалистов, поскольку профиль необходимой медицинской услуги задан в электронном направлении.

На последнем этапе реализации процесса записи на прием, при создании в МИС ЛПУ талона на прием, вместе с талоном заносится полученное от шлю-

за-накопителя направление. Получив от МИС ЛПУ вновь созданный талон, шлюз-накопитель помечает направление, по которому этот талон был создан, как использованное, а сам талон помещает в свою БД. Таким образом, шлюз-накопитель всегда хранит актуальные списки электронных направлений и талонов на прием, выданных пациентам в любых ЛПУ ареала действия СУПП, и может предоставить их приложениям при запросе из любой точки доступа СУПП.

При перенаправлении пациента к другому специалисту врач должен обеспечить пациенту не только возможность получения талона на прием, но и передать медицинскому специалисту, реализующему требуемую услугу, некую дополнительную информацию, в том числе: диагноз, описание требуемой услуги и все необходимые для успешного выполнения этой услуги сведения. После оказания услуги информация о ее результатах — уточненный диагноз, выданные рекомендации или результаты исследований — должна быть передана направившему пациента врачу. В настоящее время этот вид взаимодействия между врачами направляющей и принимающей стороны выполняется, если судить по доступным в Интернете публикациям и сообщениям, с помощью выписок из электронной болезни пациента на бумажных носителях, а в качестве курьера используется, как правило, сам пациент. Бумажный документооборот используется даже в тех случаях, когда и направляющая, и принимающая стороны оснащены МИС, а маршрутизация пациентов между ЛПУ осуществляется с помощью электронных СУПП.

Между тем нетрудно заметить, что электронные направления, выдаваемые врачами, передаются при получении пациентами талонов на прием в ЛПУ предоставления услуги и становятся доступными медицинским специалистам, оказывающим пациентам услуги. В данном случае целесообразно для передачи всей необходимой сопроводительной медицинской информации от врача, направляющего пациента к специалисту, оказывающему затребованные услуги, ввести в состав направления 2–3 дополнительных поля с соответствующей информацией. В электронном направлении обязательно содержится «обратный адрес», в соответствии с которым, по завершении оказания услуги, инициатору услуги должны быть переданы ее результаты и в автоматическом режиме внесены в амбулаторную карту пациента. Таким образом, электронное направление может быть использовано не только в качестве средства маршрутизации пациентов, но и в качестве контейнера для доставки информации от врача — инициатора услуги (врача, выдавшего пациенту электронное направление) к исполнителю услуги (медработнику, реализующему заказанную услугу в интересах пациента) и обратно.

4 Заключение

Проведен анализ складывающейся в ходе реформы системы здравоохранения РФ трехуровневой системы организации ЛПУ при оказании пациентам первичной медицинской помощи. Рассмотрены способы маршрутизации пациентов

путем самозаписи в прикрепленные ЛПУ и перенаправления врачами пациентов в амбулаторные и диагностические центры путем выдачи им талонов и электронных направлений.

Сформулированы предложения по реализации электронных СУПП в соответствии с принципами «единой точки» взаимодействия компонентов системы, определены и классифицированы потоки данных, возникающие при функционировании системы между ее компонентами. Приведено описание бизнес-процессов, протекающих в СУПП при записи пациента на прием в лечебные учреждения с участием врача и пациента. Основные решения по реализации предлагаемой СУПП апробированы при создании макета системы с использованием в качестве точки доступа мобильного приложения для смартфона и планшета.

Литература

1. *Гриднев О. В., Гаврисюк Е. В., Картавцева Л. Р., Ковалева В. В.* Реформирование первичной медико-санитарной помощи в рамках программы модернизации здравоохранения в г. Москве // Заместитель главного врача: лечебная работа и медицинская экспертиза, 2014. № 1. С. 14–21.
2. Об основах охраны здоровья граждан в Российской Федерации: Федеральный закон № 323-ФЗ от 21.11.2011. http://www.consultant.ru/document/cons_doc_LAW_173308.
3. *Илюшин Г. Я., Лиманский В. И.* Методы интеграции облачных сервисов на примере здравоохранения // Системы и средства информатики, 2014. Т. 24. № 2. С. 205–221.
4. *Арлоу Д., Нейштадт А.* UML 2 и Унифицированный процесс. Практический объектно-ориентированный анализ и проектирование. — 2-е изд. — СПб.: Символ-Плюс, 2007. 624 с.

Поступила в редакцию 23.10.14

CREATION OF PATIENT FLOWS CONTROL SYSTEM

G. Y. Ilushin and V. I. Limansky

Institute of Information Problems, Federal Research Center “Computer Science and Control,” Russian Academy of Sciences, 44-2 Vavilov Str., Moscow 119333, Russian Federation

Abstract: This article deals with the health care system organizational model regarding rendering of primary medical care to the population and the hierarchical three-level structure of medical institutions formed during the process of its reforming. The analysis of the existing ways of patient routing is carried out, including patient booking in medical institutions of the first level (the attached contingent), the redirection of patients to medical institutions of the second and the third levels by means of coupons and electronic assignments issued by doctors. The analysis of advantages and disadvantages of these ways is presented. The

electronic control systems of patient flows used for patient routing are examined as well as some problems of their implementation, in particular, implementation of EMIAS in Moscow. The analysis of electronic control systems of patient flows is given. Their main components are defined. The structure of this system based on the principles of a single point of distributed systems components interaction is suggested. The processes taking place in the system during realization of the main precedents are examined. The data flows between components of the system arising during these processes are defined. The paper also deals with approaches to organization of electronic documents flow between the medical institutions based on application of electronic appointments.

Keywords: medical information systems (MIS); electronic control systems of patient flows; distributed information systems; sequence diagram

DOI: 10.14357/08696527150112

References

1. Gridnev, O. V., E. V. Gavrisyuk, L. R. Kartavtseva, and V. V. Kovaleva. 2014. Reformirovanie pervichnoy mediko-sanitarnoy pomoshchi v ramkakh programmy modernizatsii zdravookhraneniya v g. Moskve [Reforming of primary medical-sanitary help within the program of health care modernization in Moscow]. *Zamestitel' glavnogo vracha: Lechebnaya rabota i meditsinskaya ekspertiza* [Deputy of chief physician: Medical work and medical examination] 1:14–24.
2. Federal'nyy zakon ot 21.11.2011 No. 323-FZ (red. ot 21.07.2014) "Ob osnovakh okhrany zdorov'ya grazhdan v Rossiyskoy Federatsii" [The Federal Law of 21.11.2011 No. 323-FZ (an edition of 21.07.2014) "About basis of citizens health protection in the Russian Federation"].
3. Ilushin, G. Ya., and V. I. Limanskiy. 2014. Metody integratsii oblachnykh servisov na primere zdravookhraneniya [Methods of cloud service integration by the example of health care]. *Sistemy i Sredstva Informatiki — Systems and Means of Informatics* 24(2):205–221.
4. Arlou, D., and A. Neyshtadt. 2007. *UML 2 i unifitsirovannyy protsess. Prakticheskiy ob"ektno-orientirovannyy analiz i proektirovanie* [UML 2 and the unified process. Practical object oriented analysis and design]. 2nd ed. St. Petersburg: Simvol-Plus. 624 p.

Received October 23, 2014

Contributors

Ilushin Gennadiy Y. (b. 1947) — Candidate of Science (PhD) in technology, Head of Laboratory, Institute of Information Problems, Federal Research Center "Computer Science and Control," Russian Academy of Sciences, 44-2 Vavilov Str., Moscow 119333, Russian Federation; ilushin@ipiran.ru

Limansky Valery I. (b. 1952) — Candidate of Science (PhD) in technology, senior scientist, Institute of Information Problems, Federal Research Center "Computer Science and Control," Russian Academy of Sciences, 44-2 Vavilov Str., Moscow 119333, Russian Federation; vlimansky@ipiran.ru

ИНФОРМАЦИОННЫЕ РЕСУРСЫ КОНТРАСТИВНЫХ ЛИНГВИСТИЧЕСКИХ ИССЛЕДОВАНИЙ: ТИПОЛОГИЧЕСКИЕ БАЗЫ ДАННЫХ*

М. Г. Кружков[1]

Аннотация: Рассматриваются информационные ресурсы, использующиеся в контрастивных лингвистических исследованиях, и их основные характеристики. Основными видами таких информационных ресурсов являются типологические базы данных (ТБД) и электронные корпуса текстов. В рамках данной работы основное внимание уделяется ТБД. Среди ТБД выделяются ТБД широкого профиля и специализированные ТБД. Типологические базы данных широкого профиля (или «характеристические» ТБД) представляют собой универсальные хранилища данных о широком спектре характеристик различных естественных языков; они могут использоваться как справочные инструменты, а также при решении задач, связанных с классификацией языков. Специализированные ТБД используются при детальном исследовании отдельных языковых явлений на примере ограниченных наборов языков: они тщательно моделируют структуру этих явлений и включают примеры, иллюстрирующие их функционирование в рассматриваемых языках. В статье также уделяется внимание проблемам, связанным с формированием ТБД и интеграцией данных из нескольких ТБД с различной структурой.

Ключевые слова: контрастивная лингвистика; базы данных; типологические базы данных; электронные корпуса текстов

DOI: 10.14357/08696527150113

1 Введение

Контрастивная лингвистика является одним из направлений современной лингвистики, которое ставит перед собой задачу сравнения языковых явлений, наблюдаемых в различных естественных языках. Контрастивные лингвистические исследования, все более популярные в последние десятилетия, имеют как теоретическое, так и прикладное значение, позволяя выявлять общие особенности языков и специфичные характеристики отдельных языков. С одной стороны, контрастивная лингвистика дает материал для языковой типологии и для выявления языковых универсалий, с другой стороны, она непосредственно связана

*Работа выполнена при поддержке РФФИ (проект 13-06-00403).

[1]Институт проблем информатики Федерального исследовательского центра «Информатика и управление» Российской академии наук, magnit75@yandex.ru

с различными образовательными дисциплинами, такими как преподавание иностранных языков, теория и практика перевода [1], а также используется при создании систем машинного перевода.

В лингвистике традиционный ручной сбор исходных данных, которые необходимы для проведения экспериментальных исследований, часто является трудоемким и дорогостоящим этапом работы. Формирование информационных ресурсов в форме корпусов и баз данных (БД) позволяет существенно экономить время для проведения лингвистических исследований [2–5], что в особенности актуально для задач контрастивной лингвистики, в процессе решения которых сопоставляются тексты на разных языках.

В конце прошлого века сформировалась потребность в развитии систем и средств информатики, позволяющих, с одной стороны, эффективно хранить и обрабатывать большие объемы текстов на разных языках и, с другой стороны, фиксировать результаты экспериментальных и теоретических исследований, получаемые лингвистами в процессе контрастивного анализа. Одним из наиболее эффективных средств информатики, используемых для решения этих задач, являются БД. При этом наиболее востребованы БД следующих двух видов:

(1) компьютерные БД, разработанные в целях проведения типологических исследований (ТБД). Такие БД представляют собой структурированные систематизированные сведения об определенных характеристиках различных естественных языков;

(2) электронные корпуса текстов, активно использующиеся при проведении экспериментальных лингвистических исследований, среди которых в особенности следует выделить параллельные[1] и сопоставимые[2] корпуса. Этот вид информационных ресурсов будет подробно рассмотрен в отдельной работе, а в рамках данной статьи сосредоточим внимание на ТБД.

Типологические БД содержат информацию о свойствах и характеристиках различных языков, например числе грамматических родов, наблюдаемых в языках. Есть языки, в которых род отсутствует, а есть с двумя (мужской, женский), тремя (мужской, женский, средний) и бóльшим числом грамматических родов. Информация, включаемая в ТБД, извлекается исследователями из описаний грамматик соответствующих языков, генерируется в результате взаимодействия с носителями этих языков или предоставляется экспертами, свободно ими владеющими. Также эта информация может генерироваться в результате ручного или автоматического анализа текстов на этих языках (в том числе, возможно, с использованием текстов электронных корпусов). Типологические БД могут отличаться по количеству и составу рассматриваемых языков, а также по природе,

[1]Параллельные (переводные) корпуса представляют собой собрания оригинальных текстов на одном из сравниваемых языков, дополненные переводом на другой язык (или несколькими переводами на один или несколько языков).

[2]Сопоставимые корпуса представляют собой сбалансированные собрания оригинальных текстов на сравниваемых языках.

количеству, составу и структуре включенных в них признаков, характеризующих каждый язык в целом или отдельные языковые явления.

Электронные корпуса и ТБД являются эффективными средствами для проведения контрастивных лингвистических исследований, но это принципиально разные инструменты. Типологические БД включают в себя справочную информацию о свойствах и характеристиках различных языков либо отдельных языковых единиц. Электронные корпуса представляют собой, с одной стороны, (пополняемые) хранилища текстовых и языковых данных, с другой стороны, инструменты, с помощью которых могут быть получены новые лингвистические знания.

Цель данной статьи состоит в том, чтобы рассмотреть средства информатики, используемые для формирования ТБД (разд. 2), а также описать два основных вида ТБД, используемых в контрастивных лингвистических исследованиях:

(1) характеристические ТБД, содержащие данные о широком спектре языковых явлений (разд. 3);

(2) специализированные ТБД, создаваемые для детального исследования отдельных языковых явлений (разд. 4).

Рассматриваются основные свойства ТБД этих двух видов и решаемые с их использованием задачи.

2 Особенности создания типологических баз данных

На сегодняшний день системы управления БД (СУБД) являются наиболее распространенным инструментом, позволяющим структурировать и систематизировать языковые данные, а также обеспечить к ним удобный доступ и возможность поиска. Однако лингвисты часто не имеют достаточных навыков работы с СУБД, в результате им нередко приходится хранить свои данные в электронных таблицах или даже в файлах как текстовые документы.

Dimitriadis и Musgrave сделали попытку устранить этот пробел, представив обзор особенностей построения и использования реляционных[1] БД, предназначенный специально для лингвистов [6]. В нем они освещают такие вопросы, как основные преимущества БД, их модели, СУБД, используемые для управления данными, виды БД и платформ, основы структурированного описания данных, языка запросов SQL (structured query language), особенности хранения данных разных типов и т. д.

При рассмотрении вопроса построения схемы реляционной БД (или *реляционной структуры*) отмечается, что, хотя не существует универсального способа построения схемы, необходимо, чтобы она *отражала логическую организацию*

[1] Реляционная БД представляет собой множество таблиц, связанных друг с другом отношениями (их название проистекает от английского слова relation — отношение).

обрабатываемых данных. Если это условие выполняется, то данные в ТБД будут позволять поддерживать решение лингвистических задач на основе этих данных.

Отмечается, что схема эффективной БД должна быть достаточно *гибкой*, чтобы иметь возможность адаптироваться к возможным будущим потребностям исследователей, и одновременно максимально *простой*, позволяя при этом поддерживать решение типологических задач, которые могут вставать перед исследователями. Создатели ТБД должны иметь четкое представление о том, как именно будет использоваться их БД в лингвистических исследованиях, и еще на этапе разработки схемы стремиться обеспечить реализацию необходимых лингвистам функций. При этом необходимо учитывать различия в теоретических предпосылках, которых могут придерживаться лингвисты как будущие пользователи ТБД, и в задачах, которые они перед собой ставят. Это означает, что лингвисты должны принимать активное участие в процессе проектирования БД.

Иногда возникает потребность в интеграции типологических языковых данных из различных внешних источников. Проблемы, возникающие при интеграции, и пути их решения рассматриваются в работе [7]. Авторы [7] разработали концепцию унифицированной системы ТБД (Typological Database System, TDS), которая включает в себя глобальную онтологию лингвистических понятий и набор локальных онтологий — по одной для каждой БД, интегрированной в систему. В процессе реализации концепции выяснилось, что основная трудность связана не с задачей интеграции языковых данных, хранящихся в различных форматах и в различных программных средах, а со следующими двумя проблемами.

Во-первых, это необходимость описать и поддерживать в TDS, а также визуализировать огромное число описательных параметров. Во-вторых, что оказалось наиболее сложной проблемой, это различия в структурах, терминологии и теоретических подходах, использовавшихся при создании различных ТБД, интегрируемых в TDS. Эти различия обусловлены тем, что разные ТБД разрабатывались независимыми коллективами с разными концептуальными подходами, задачами, научными парадигмами и интересами.

Исходя из опыта интеграции БД, авторы работы [7] дают разработчикам ТБД ряд советов, в частности, они рекомендуют:

- документировать все теоретические предпосылки и процедуры, чтобы иметь возможность легко восстановить принципы, на которых строилась ТБД;
- фиксировать все источники информации, использованные при создании ТБД, что помогает при выверке и дополнении содержимого ТБД;
- использовать стандарты для тех полей, для которых они существуют (в частности, стандарт ISO 639-3 [8] для однозначной идентификации языков);
- избегать использования полей с текстовыми комментариями, по возможности заменяя их четко структурированными полями;
- избегать использования Null-значений в ТБД из-за неоднозначности их интерпретации [9].

3 «Характеристические» типологические базы данных широкого профиля

Ядро схемы ТБД может иметь относительно простую «матричную» структуру, в которой для n заданных языков и m их признаков в идеальном случае необходимо проставить $n \cdot m$ значений этих признаков; иначе говоря, n языков *характеризуются* максимум по m признакам. Типологические БД, в основе которых лежит подобный принцип построения схемы, можно условно назвать «характеристическими».

Основными отличительными особенностями «характеристических» ТБД являются перечни сопоставляемых языков и наборы признаков для описания языков. В некоторых ТБД эти наборы относительно невелики — исследователи ограничивают свою задачу узкими специализированными наборами свойств и небольшой группой языков. Типологические БД обычно создаются небольшими исследовательскими группами для исследования конкретных языковых явлений [10]. Наборы признаков определяются в соответствии с целями исследования, наборы языков — в соответствии с масштабом исследуемого языкового явления либо с целями и задачами исследования. Если данное языковое явление исследуется в глобальном масштабе, то выборка языков обычно делается с учетом максимальной ареальной и генетической представительности, т. е. исследователи стремятся включить в ТБД максимум представителей различных языковых групп и регионов. Однако ТБД также могут охватывать относительно небольшие группы языков, например романские (по родовому принципу) или африканские (по географическому принципу). Наконец, иногда авторы ТБД ограничиваются языками, для которых доступны соответствующие данные (например, в виде опубликованных грамматик), — в этом случае набор языков часто не является генетически или ареально сбалансированным.

Существуют по крайней мере два достаточно амбициозных проекта, в рамках которых на протяжении многих лет разрабатываются ТБД, поддерживающие обширные массивы признаков для больших групп языков.

Типологическая БД World Atlas of Language Structures (WALS) на 27 декабря 2014 г. содержала информацию о 2679 языках, характеризуемых максимум 192 признаками [11]. Иными словами, не для каждого языка обязательно указаны все 192 признака, например для русского языка приведены значения для 156 из них [12].

Вторая обширная коллекция признаков языков содержится в БД «Языки Мира» (ЯМ), включающей информацию о 3821 признаке для 315 языков [13].

Видно, что в подобных масштабных проектах акцент делается либо на широкий спектр языков (при ограниченном наборе признаков — WALS), либо на обширные наборы признаков (при ограниченном наборе языков — ЯМ). Однако даже при таких ограничениях в рамках этих проектов часто не представляется возможным заполнить соответствующими значениями все ячейки признаков ТБД. Так, в WALS в 2009 г. из 360 960 ячеек было заполнено около 58 000 (16%),

т. е. для каждого признака были проставлены значения в среднем для 400 языков из 2560 [11].

Темпы эволюции ТБД иллюстрирует следующий пример. В 2009 г. ТБД WALS включала информацию о 141 признаке для 2560 языков, т. е. за 5 лет число языков увеличилось на 119, а максимальное число признаков — на 51. Для подобного пополнения иногда необходимо провести трудоемкую и дорогостоящую работу, которая может включать в себя, в том числе, проведение лингвистических экспедиций для непосредственного общения с носителями соответствующих языков.

Результаты проекта WALS изначально были опубликованы в виде книги с картами мира, показывающими глобальное распределение языковых признаков [14]. На картах каждый язык обозначен множеством географически распределенных кружков, а для визуализации значений признаков используются разные цвета кружков. Каждый признак изначально мог иметь от 2 до 9 возможных значений, поскольку больше чем девять цветов трудно было четко передать на карте. В издание внесли свой вклад более чем 40 авторов. Их задача состояла в сборе информации из различных изданий (справочников по грамматике, словарей, научных статей). Кроме этих источников использовалась информация, которая была получена благодаря их личным связям с экспертами или носителями соответствующих языков. В книге авторы приводят текстовое описание каждого признака (вместе с его значениями), дают пояснения относительно его теоретической значимости и комментируют особенности его географического распределения.

Вместе с книгой была выпущена также ее электронная версия на компакт-диске. Пользователи компьютеров могли просматривать карты, опубликованные на диске, а также делать собственные запросы к WALS, например: какие корреляции существуют между структурными признаками в различных областях грамматики, а также как географически распределяются те или иные признаки. Поиск корреляций можно осуществлять как с привязкой, так и без привязки к географической информации. При этом совместное исследование признаков и ареальных моделей оказалось продуктивным для обнаружения значимых корреляций, важных для понимания географических особенностей типологии языков [15].

С 2008 г. WALS стала доступна через Интернет [16]. Онлайн-версия WALS продолжает совершенствоваться как в плане пополнения ТБД новыми данными, так и в плане расширения ее функциональности. В ней появляются новые «главы», посвященные новым признакам (как было отмечено, на 27.12.2014 их число достигло 192). Также появилась функция обратной связи, позволяющая пользователям комментировать и обсуждать как сами признаки, так и значения этих признаков, проставленные для конкретных языков.

Были сняты некоторые ограничения, наложенные на структуру WALS в начале проекта. Например, ограничение на число возможных значений признаков (9) приводило к тому, что на многих картах было задействовано низкоинформатив-

ное значение «другой» или «смешанный». В настоящее время максимальное число возможных значений признаков увеличено до 28. При этом интерактивный пользовательский интерфейс позволяет при необходимости агрегировать несколько значений вместе, таким образом пользователь может свести несколько значений какого-либо числового признака к таким номинативным значениям, как «среднее», «ниже среднего» и «выше среднего». Также пользователи могут создавать свои собственные составные признаки путем комбинирования значений двух или более различных признаков.

Важно отметить, что в WALS концепция использования признаков и их значений не унифицирована, а используемые термины не имеют одного и того же значения в разных главах, т. е. WALS не имеет общей онтологии, а лингвистические термины определяются заново в каждой главе ее авторами. Этот недостаток сохраняется и в электронной версии ТБД WALS, поскольку его гораздо труднее устранить, чем другие, уже упомянутые выше.

С начала 1980-х гг. в Институте языкознания РАН ведется работа над ТБД ЯМ [17]. Изначально она разрабатывалась как компьютерный справочный аппарат для энциклопедии «Языки мира», издающейся с 1993 г. по настоящее время (на сегодня опубликовано 20 томов [18]). В дальнейшем идея трансформировалась в проект по разработке БД, создаваемой на основе энциклопедии, позволяющей извлекать сведения лингво-типологического характера без непосредственного обращения к самому энциклопедическому изданию.

Признаки языков в ЯМ классифицированы в рамках единой иерархической структуры (модели). Первому уровню иерархии соответствуют наименования классов языковых явлений, которые соответствуют классам, содержащимся в схемах статей энциклопедии «Языки мира». Внутри классов выделяются возможные аспекты рассмотрения данного класса, на более низких ступенях может располагаться несколько уровней подаспектов, а элементы, расположенные на последнем уровне, называются характеристиками. Внутри аспекта может содержаться до пяти уровней иерархии, последний из которых (самый низкий) представлен характеристиками, а все промежуточные по определению являются подаспектами. Например: класс — «просодические явления», аспект — «ударение», подаспект — «тип ударения», характеристики — «нефонологическое», «фонологическое».

По своей роли структурные элементы делятся на классифицирующие и фактографические. Классифицирующие элементы (классы, аспекты и подаспекты) играют вспомогательную роль, они служат своего рода координатами, задающими место той или иной характеристики в общей модели. Ко второму типу относятся сами характеристики. В модель иерархической структуры включается перечень наиболее распространенных характеристик, но если при описании конкретного языка наблюдаются явления, которые не описываются стандартными характеристиками модели, формулируется новая характеристика, аналогичная по своим параметрам характеристикам, присутствующим в модели. Таким образом, ТБД ЯМ примечательна тем, что в ней, в отличие, например, от ТБД

WALS, реализована единая онтология для описания лингвистических явлений.

Самым очевидным вариантом использования «характеристических» ТБД широкого профиля является поиск справочной информации о языках и о языковых конструкциях. Однако у них есть и другие возможности использования, которые рассматриваются в [17]. Они могут применяться в процессе решения задач, связанных с классификацией языков, например, при выяснении степени близости языков, которая может вычисляться по различным алгоритмам на основе данных «характеристических» ТБД. Подобные задачи могут иметь ареальную, генетическую или даже историческую ориентацию, поскольку степень близости языков может объясняться как генетическими, так и ареальными факторами, а на основе информации о генетических и/или ареальных связях между языками могут быть сделаны выводы о путях миграции человечества при расселении по планете. Результаты генетической классификации языков на основе ТБД WALS и ЯМ сопоставляются в работе [19]. По мнению авторов этой работы, результаты генетического анализа, полученные на основе данных WALS, не совсем точно отражают степень близости языков, в то время как результаты, полученные на основе ЯМ, дают возможность правильно выделить несколько групп языков. Можно предположить, что этот результат обусловлен более обширным пространством признаков в ЯМ и наличием единой иерархической системы классификации этих признаков.

4 Специализированные типологические базы данных

В отличие от чисто «характеристических» ТБД широкого профиля, многие специализированные ТБД могут также включать в себя более или менее широкие наборы исходного языкового материала, т. е. некоторые конструкции или текстовые фрагменты, на основании анализа которых исследователи затем характеризуют и классифицируют соответствующие языки. Такие ТБД могут иметь более сложную структуру, которая разрабатывается в соответствии с целями и теоретическими предпосылками исследователей, работающих над конкретными проектами.

Например, «ТБД личных и указательных местоимений» (ТБД ЛУМ) [20] помимо общей языковой информации включает в себя наборы местоимений для 109 типологически различных языков, включая их парадигмы, а также резюме местоименных систем для каждого из этих языков. При этом каждое местоимение классифицировано в соответствии с заданным набором морфосинтаксических атрибутов.

Целью подобных ТБД, как правило, является выявление *типологических тенденций* и *типологически необычных явлений* (т. е. исключений из этих тенденций). Например, в результате анализа содержания ТБД ЛУМ было установлено, что ее данные соответствуют известным лингвистическим универсалиям морфосинтаксических атрибутов, в частности, совместимы с иерархией падежей,

установленной Blake [21], в которой грамматических падежей больше, чем локальных, а локальных, в свою очередь, больше, чем падежей других типов. Вместе с тем было показано, например, что тайские местоимения отличаются от местоимений, существующих в других языках, настолько существенно, что встает вопрос о том, являются ли они в действительности личными местоимениями или их следует отнести к классу существительных. Таким образом, работа над ТБД ЛУМ поставила перед исследователями такие фундаментальные вопросы, как что же, собственно, является личным местоимением и действительно ли личные местоимения существуют во всех языках.

Еще один пример ТБД с исходным языковым материалом — «ТБД интенсификаторов и возвратных местоимений» (ТБД ИВМ) [22]. Для этой относительно небольшой ТБД была разработана специальная типология возвратных местоимений. Типологическая БД ИВМ содержит 689 примеров для более чем 100 языков, и каждый пример помечен соответствующим типом, характеризующим иллюстрируемый интенсификатор или возвратное местоимение, что позволило авторам создать поисковый интерфейс для поиска нужных примеров. Из-за скромного объема ТБД ИВМ ее нельзя использовать в статистических процедурах, она имеет, главным образом, описательную цель.

Чтобы показать, как ТБД могут использоваться для описания языковых объектов и явлений, рассмотрим ряд ТБД, разработанных морфологической группой университета Суррея (Великобритания) [10]. Во-первых, это ТБД по синкретизму в различных языках (ТБДСин). Синкретизм обычно определяется как «отношение между словами, имеющими одинаковую форму при различных значениях морфосинтаксических характеристик»[1] [23]. Однако в процессе работы над ТБДСин авторам пришлось дать более точное определение рассматриваемому явлению, поскольку вышеприведенное определение подразумевает, что существуют заранее определенные наборы *морфологических и синтаксических характеристик* (признаков) *и их возможных значений* для каждого рассматриваемого языка. Однако различные исследователи могут по-разному выделять наборы таких характеристик. В связи с этим авторы приняли более строгое определение синкретизма, данное на основе распределительного принципа, описанного Comrie в работе [24].

Очевидно, что синкретизм — достаточно сложное для описания явление. В ТБДСин выделяется 10 возможных наборов характеристик для описания синкретизмов: число, падеж, род, определенность, лицо, время, наклонение, залог, вид, наличие/отсутствие отрицания. Значения этих характеристик группируются в комбинации в отдельной таблице, а сам синкретизм описывается как пара таких комбинаций. Типологическая БД по синкретизму в различных языках предусматривает запись только бинарных синкретизмов; если более двух комбинаций имеют общую форму, то такой многоместный синкретизм разбивается

[1]Например, прилагательное *большой* может иметь характеристики {*мужской род, именительный падеж*}, {*женский род, творительный падеж*}, {*женский род, дательный падеж*} и т. д.

на несколько бинарных[1]. В таблице, описывающей параметры синкретизма, содержится ссылка на внешний ресурс, в котором приводятся парадигмы, иллюстрирующие описанный синкретизм для данного языка.

Суррейская группа также разработала ТБД по супплетивизму в различных языках (ТБДСуп), т. е. явлению, при котором две флективные формы одной лексической единицы не имеют явной фонологической связи (как, например, в парах *человек/люди*, *идти/шел*). Супплетивизм — нерегулярное явление, поэтому для него сложно построить регулярную модель. Авторы для его описания используют схему данных, которая во многом повторяет схему ТБДСин. Она также включает в себя 10 таблиц с наборами языковых характеристик (число, падеж и т. д.), значения которых также группируются в комбинации в отдельной таблице. Далее в таблице более высокого уровня основы лексических единиц привязываются к комбинациям, в которых они проявляются, а затем в таблице еще более высокого уровня конкретная лексема объединяется с различными основами, находящимися между собой в отношении супплетивизма. Таким образом, описываемое в ТБДСуп явление представляет собой древовидную логическую структуру, которая, тем не менее, может быть представлена с помощью иерархии таблиц реляционной БД.

5 Заключение

В данной работе описаны ТБД, используемые для поддержки контрастивных лингвистических исследований. Были рассмотрены проблемы, связанные с формированием и интеграцией ТБД, а также подробно описаны два основных вида ТБД, а именно: характеристические ТБД широкого профиля и специализированные ТБД.

Для двух наиболее известных ТБД первого вида проведен сопоставительный анализ (ТБД WALS и ТБД ЯМ). В результате сравнения этих БД отмечается, что ТБД WALS включает в себя значительно больше языков, чем ТБД ЯМ, в то время как ЯМ содержит значительно более широкую коллекцию признаков, чем WALS. Кроме этого, в ТБД ЯМ, в отличие от ТБД WALS, реализована глобальная онтология для описания лингвистических понятий. Отмечается, что ТБД широкого профиля используются главным образом для получения справочной информации о языках, а также в задачах, связанных с классификацией языков.

Для создания специализированных ТБД обычно требуется меньше ресурсов, чем для создания ТБД широкого профиля, поэтому специализированных ТБД существует значительно больше и они более разнообразны, поскольку посвящены исследованию разноплановых языковых явлений. Специализированные ТБД более детально моделируют структуру исследуемых явлений и иллюстрируют

[1]Возможно, для решения этой проблемы было бы удобно использовать отношение «многие-ко-многим».

их примерами. Было показано, что помимо собственно задачи моделирования исследуемых языковых явлений специализированные ТБД могут служить инструментом для выявления кросслингвистических тенденций, а также для обнаружения явлений, выступающих исключениями из этих тенденций.

В продолжение анализа информационных ресурсов, обеспечивающих проведение контрастивных лингвистических исследований, планируется рассмотреть возможности электронных корпусов текстов в рамках отдельной работы. Электронные корпуса не содержат в явном виде информацию о свойствах языков содержащихся в них текстов, однако корпусный инструментарий является одним из эффективных средств получения и проверки такой информации. Кроме этого, сегодня формируются БД нового вида, которые существенно расширяют возможности электронных корпусов и которые предлагается называть надкорпусными. Такие БД в настоящее время используются, в частности, для контрастивного анализа лингвоспецифичных единиц. Описание подобных БД приводится в работе [25].

Литература

1. *Гак В. Г.* О контрастивной лингвистике // Новое в зарубежной лингвистике. Вып. XXV. Контрастивная лингвистика / Сост. В. П. Нерознак, В. Г. Гак. — М.: Прогресс, 1989. С. 5–17.
2. *Добровольский Д. О., Кретов А. А., Шаров С. А.* Корпус параллельных текстов // Научная и техническая информация. Сер. 2. Информационные процессы и системы, 2005. № 6. С. 16–27.
3. *Добровольский Д. О., Кретов А. А., Шаров С. А.* Корпус параллельных текстов: архитектура и возможности использования // Национальный корпус русского языка: 2003–2005. — М.: Индрик, 2005. С. 263–296.
4. *Андреева Е. Г., Касевич В. Б.* Грамматика и лексика (на материале англо-русского корпуса параллельных текстов) // Национальный корпус русского языка: 2003–2005. — М.: Индрик, 2005. С. 297–307.
5. *Добровольский Д. О.* Корпус параллельных текстов в исследовании культурно-специфичной лексики // Национальный корпус русского языка: 2006–2008. Новые результаты и перспективы. — СПб.: Нестор-История, 2009. С. 383–401.
6. *Dimitriadis A., Musgrave S.* Designing linguistic databases: A primer for linguists // The use of databases in cross-linguistic studies. Empirical approaches to language typology. — Berlin – New York: Mouton De Gruyter, 2009. Vol. 41. P. 13–76.
7. *Dimitriadis A., Windhouwer M., Saulwick A., Goedemansand R., Bíró T.* How to integrate databases without starting a typology war: The typological database system // The use of databases in cross-linguistic studies. Empirical approaches to language typology. — Berlin – New York: Mouton De Gruyter, 2009. Vol. 41. P. 155–208.
8. Codes for the representation of names of languages — Part 3: Alpha-3 code for comprehensive coverage of languages. http://www.iso.org/iso/catalogue_detail?csnumber=39534.

9. *Date C. J.* An introduction to database systems. — 8th ed. — Oxford: Blackwell, 2004. 1024 p.
10. *Brown D. P., Tiberius C., Chumakina M., Corbett G. G., Krasovitsky A.* Databases designed for investigating specific phenomena // The use of databases in cross-linguistic studies. Empirical approaches to language typology. — Berlin – New York: Mouton De Gruyter, 2009. Vol. 41. P. 117–154.
11. *Haspelmath M.* The typological database of the World Atlas of Language Structures // The use of databases in cross-linguistic studies. Empirical approaches to language typology. — Berlin – New York: Mouton De Gruyter, 2009. Vol. 41. P. 283–300.
12. WALS: Language Russian. http://wals.info/languoid/lect/wals_code_rus.
13. *Соловьев В. Д.* Типологические базы данных: перспективы использования // Вопросы языкознания, 2010. № 1. С. 94–110.
14. The World Atlas of Language Structures / Eds. M. Haspelmath, M. S. Dryer, D. Gil, B. Comrie. — Oxford: Oxford University Press, 2005. 712 p.
15. *Dryer M.* Large linguistic areas and language sampling // Studies Language, 1989. Vol. 13. Iss. 2. P. 257–292.
16. World Atlas of Language Structures. http://wals.info.
17. *Виноградов В. А., Новиков А. И., Ярославцева Е. И.* База данных «Языки мира» как инструмент лингвистического исследования // Вопросы языкознания, 2003. № 3. С. 3–14.
18. Языки мира: Многотомное энциклопедическое издание. http://www.iling-ran.ru/langworld.
19. *Polyakov V., Solovyev D., Wichnann S., Belyaev O.* Using WALS and Jazyki Mira // Linguistic Typology, 2009. Vol. 13. No. 1. P. 137–167.
20. *Bliss H., Ritter E.* A typological database of personal and demonstrative pronouns // The use of databases in cross-linguistic studies. Empirical approaches to language typology. — Berlin – New York: Mouton De Gruyter, 2009. Vol. 41. P. 77–116.
21. *Blake B. J.* Case. — Cambridge: Cambridge University Press, 1994. 248 p.
22. *Gast V.* A contribution to 'two-dimensional' language description: The Typological Database of Intensifiers and Reflexives // The use of databases in cross-linguistic studies. Empirical approaches to language typology. — Berlin – New York: Mouton De Gruyter, 2009. Vol. 41. P. 216–241.
23. *Matthews P. H.* The Concise Oxford Dictionary of Linguistics. — Oxford: Oxford University Press, 1997. 432 p.
24. *Comrie B.* Form and function in identifying cases // Paradigms. The Economy of inflection / Ed. F. Plank. — Berlin – New York: Mouton De Gruyter, 1991. P. 41–55.
25. *Kruzhkov M. G., Buntman N. V., Loshchilova E. Ju., Sitchinava D. V., Zalisniak A. A., Zatsman I. M.* A database of Russian verbal forms and their French translation equivalents // Компьютерная лингвистика и интеллектуальные технологии, 2014. Вып. 13(20). С. 284–297.

Поступила в редакцию 13.02.15

INFORMATION RESOURCES FOR CONTRASTIVE STUDIES: TYPOLOGICAL DATABASES

M. G. Kruzhkov

Abstract: This article presents information resources used in contrastive linguistic studies and their principle features. There are two main types of such information resources: typological databases and electronic text corpora. The attention is focused on typological databases, which can be subdivided into general-purpose typological databases and specialized typological databases. General-purpose typological databases act as repositories of a wide range of data on a wide variety of languages. They may be utilized as reference resources and may also be helpful while dealing with language classification problems. Specialized typological databases are used for closer investigation of specific language phenomena in restricted sets of languages. They supply detailed models of such phenomena and include examples to illustrate their functioning in the considered languages. The paper also looks into problems related to development of typological databases and to integration of data from heterogeneous typological databases.

Keywords: contrastive linguistic studies; databases; typological databases; electronic text corpora

DOI: 10.14357/08696527150113

Acknowledgments

The work was performed with financial support of the Russian Foundation for Basic Research (grant No. 13-06-00403).

References

1. Gak, V. G. 1989. O kontrastivnoy lingvistike [On contrustive linguistics]. *Novoe v zarubezhnoy lingvistike. Vyp. XXV "Kontrastivnaya lingvistika"* [New developments in foreign linguistics. Vol. XXV "Contrastive linguistics"]. Moscow: Progress. 5–17.
2. Dobrovol'skiy, D. O, A. A. Kretov, and S. A. Sharov. 2005. Korpus parallel'nykh tekstov [Corpus of parallel texts]. *Nauchnaya i Tekhnicheskaya Informatsiya. Ser. 2 "Informatsionnye Protsessy i Sistemy"* [Scientific and Technical Information. Ser. 2 "Informational Processes and Systems"] 6:16–27.
3. Dobrovol'skiy, D. O., A. A. Kretov, and S. A. Sharov. 2005. Korpus parallel'nykh tekstov: Arkhitektura i vozmozhnosti ispol'zovaniya [Corpus of parallel texts: Architecture and applications]. *Natsional'nyy korpus russkogo yazyka: 2003–2005* [Russian National Corpus: 2003–2005]. Moscow: Indrik. 263–296.

4. Andreeva, E. G., and V. B. Kasevich. 2005. Grammatika i leksika (na materiale anglo-russkogo korpusa parallel'nykh tekstov) [Grammar and lexis (based on the English-Russian parallel corpus)]. *Natsional'nyy korpus russkogo yazyka: 2003–2005* [Russian National Corpus: 2003–2005]. Moscow: Indrik. 297–307.
5. Dobrovol'skiy, D. O. 2009. Korpus parallel'nykh tekstov v issledovanii kul'turno-spetsifichnoy leksiki [Using a parallel corpus for investigation of culture-specific lexis]. *Natsional'nyy korpus russkogo yazyka: 2006–2008. Novye rezul'taty i perspektivy* [Russian National Corpus: 2006–2008. New results and prospects]. St. Petersburg: Nestor-Istoriya. 383–401.
6. Dimitriadis, A., and S. Musgrave. 2009. Designing linguistic databases: A primer for linguists. *The use of databases in cross-linguistic studies. Empirical approaches to language typology*. Berlin – New York: Mouton De Gruyter. 41:13–76.
7. Dimitriadis, A., M. Windhouwer, A. Saulwick, R. Goedemansand, and T. Bíró. 2009. How to integrate databases without starting a typology war: The Typological Database System. *The use of databases in cross-linguistic studies. Empirical approaches to language typology*. Berlin – New York: Mouton De Gruyter. 41:155–208.
8. Codes for the representation of names of languages — Part 3: Alpha-3 code for comprehensive coverage of languages. Available at: http://www.iso.org/iso/catalogue_detail?csnumber=39534 (accessed February 27, 2015).
9. Date, C. J. 2004. *An introduction to database systems*. 8th ed. Oxford: Blackwell. 1024 p.
10. Brown, D. P., C. Tiberius, M. Chumakina, G. G. , and A. Krasovitsky. 2009. Databases designed for investigating specific phenomena. *The use of databases in cross-linguistic studies. Empirical approaches to language typology*. Berlin – New York: Mouton De Gruyter. 41:117–154.
11. Haspelmath, M. 2009. The typological database of the World Atlas of Language Structures. *The use of databases in cross-linguistic studies. Empirical approaches to language typology*. Berlin – New York: Mouton De Gruyter. 41:283–300.
12. WALS: Language Russian. Available at: http://wals.info/languoid/lect/wals_code_rus (accessed February 27, 2015).
13. Solov'ev, V. D. 2010. Tipologicheskie bazy dannykh: Perspektivy ispol'zovaniya [Typological databases: Application prospects]. *Zh. Voprosy Yazykoznaniya* [Problems of Linguistics] 1:94–110.
14. Haspelmath, M., M. S. Dryer, D. Gil, and B. Comrie, eds. 2005. *The World Atlas of Language Structures*. Oxford: Oxford University Press. 712 p.
15. Dryer, M. 1989. Large linguistic areas and language sampling. *Studies Language* 13:257–292.
16. World Atlas of Language Structures. Available at: http://wals.info/ (accessed February 27, 2015).
17. Vinogradov, V. A., A. I. Novikov, and E. I. Yaroslavtseva. 2003. Baza dannykh "Yazyki mira" kak instrument lingvisticheskogo issledovaniya [Database "Languages of the World" as a tool for linguistic studies]. *Zh. Voprosy Yazykoznaniya* [Problems of Linguistics] 3:3–14.
18. Yazyki mira. Mnogotomnoe entsiklopedicheskoe izdanie [Languages of the World Multivolume encyclopedia]. Available at: http://www.iling-ran.ru/langworld/ (accessed February 27, 2015).

19. Polyakov, V., D. Solovyev, S. Wichnann, and O. Belyaev. 2009. Using WALS and Jasyki Mira. *Linguistic Typology* 13(1):137–167.
20. Bliss, H., and E. Ritter. 2009. A typological database of personal and demonstrative pronouns. *The use of databases in cross-linguistic studies. Empirical approaches to language typology*. Berlin – New York: Mouton De Gruyter. 41:77–116.
21. Blake, B. J. 1994. *Case*. Cambridge: Cambridge University Press. 248 p.
22. Gast, V. 2009. A contribution to 'two-dimensional' language description: The Typological Database of Intensifiers and Reflexives. *The use of databases in cross-linguistic studies. Empirical approaches to language typology*. Berlin – New York: Mouton De Gruyter. 41:216–241.
23. Matthews, P. H. 1997. *The Concise Oxford Dictionary of Linguistics*. Oxford: Oxford University Press. 432 p.
24. Cormie, B. 1991. Form and function in identifying cases. *Paradigms. The Economy of inflection*. Ed. F. Plank. Berlin – NewYork: Mouton De Gruyter. 41–55.
25. *Kruzhkov, M. G., N. V. Buntman, E. Ju. Loshchilova, D. V. Sitchinava, A. A. Zalisniak, and I. M. Zatsman.* 2014. A database of Russian verbal forms and their French translation equivalents. *Komp'yuternaya Lingvistika i Intellektual'nye Tekhnologii* [Computer Linguistics and Intellectual Technologies] 13(20):284–297.

Received February 13, 2015

Contributor

Kruzhkov Mikhail G. (b. 1975) — leading programmer, Institute of Information Problems, Federal Research Center "Computer Science and Control," Russian Academy of Sciences, 44-2 Vavilov Str., Moscow 119333, Russian Federation; magnit75@yandex.ru

О ВКЛАДЕ АКАДЕМИКА И. А. МИЗИНА В ТЕОРИЮ И ПРАКТИКУ СОЗДАНИЯ ОТЕЧЕСТВЕННЫХ ИНФОРМАЦИОННО-ТЕЛЕКОММУНИКАЦИОННЫХ СИСТЕМ: К 80-ЛЕТИЮ СО ДНЯ РОЖДЕНИЯ

И. А. Соколов[1], *А. А. Зацаринный*[2], *В. Н. Захаров*[3]

Аннотация: Статья посвящена 80-летнему юбилею академика И. А. Мизина, директора ИПИ РАН в период 1989–1999 гг., выдающегося ученого, конструктора и инженера. Дана краткая биографическая справка, характеризующая научно-практическую деятельность И. А. Мизина. Вклад академика И. А. Мизина в теорию и практику создания отечественных информационно-телекоммуникационных сетей рассматривается применительно к трем направлениям его деятельности. Первое — разработка и внедрение системы обмена данными (СОД) в интересах автоматизированных систем управления (АСУ) Вооруженных Сил, первой отечественной сети с пакетной коммутацией. Второе направление — обоснование информационных технологий для создания крупных территориальных систем передачи данных в регионах России. Третье направление — создание и развитие информационных сетей в интересах органов государственной власти с учетом требований по защите информации.

Ключевые слова: главный конструктор; академик; информационно-телекоммуникационные сети; информационные технологии; система обмена данными; пакетная коммутация; методы передачи информации; защита информации; каналы связи; испытания; органы государственной власти

DOI: 10.14357/08696527150114

1 Введение

12 апреля 2015 г. исполняется 80 лет со дня рождения академика И. А. Мизина, директора Института проблем информатики РАН в период 1989–1999 гг. Игорь Александрович ушел из жизни рано, в возрасте всего 64 лет, но внес огромный вклад как в развитие теории отечественных информационно-телекоммуникационных систем (ИТС), так и в практику их разработки и внедрения.

[1]Федеральныйо исследовательский центр «Информатика и управление» Российской академии наук, isokolov@ipiran.ru

[2]Федеральный исследовательский центр «Информатика и управление» Российской академии наук, azatsarinny@ipiran.ru

[3]Федеральный исследовательский центр «Информатика и управление» Российской академии наук, vzakharov@ipiran.ru

И. А. Мизин родился 12 апреля 1935 г. в Москве в семье военнослужащего. В 1952 г. с отличием окончил среднюю школу и поступил в одно из самых престижных военных учебных заведений — Военно-воздушную академию имени Н. Е. Жуковского. После окончания академии в 1959 г. по специальности «Эксплуатация радиотехнических средств ВВС» И. А. Мизин молодым лейтенантом был направлен для прохождения службы в одну из ведущих организаций военно-промышленного комплекса — НИИ автоматической аппаратуры, в котором проработал более 30 лет, последовательно пройдя ступени карьерного роста от инженера до заместителя директора, руководителя крупного научно-производственного коллектива, главного конструктора СОД [1, 2].

В 1966 г. защитил кандидатскую диссертацию на тему «Разработка метода обмена информацией по закрытым телеграфным каналам связи в командной системе управления специального назначения», а в 1972 г. — докторскую на тему «Вопросы исследования и разработки информационных сетей территориальных автоматизированных систем управления». В 1975 г. И. А. Мизину присвоено ученое звание профессора по кафедре «Автоматизированные системы управления». В 1984 г. он был избран членом-корреспондентом АН СССР по специальности «Вычислительная техника».

В течение последних 10 лет жизни (1989–1999 гг.) И. А. Мизин — директор ИПИ РАН, генеральный конструктор работ по модернизации информационных сетей органов государственной власти, генеральный конструктор АСУ Вооруженных Сил РФ. В 1997 г. был избран академиком РАН по специальности «Вычислительная техника и элементная база».

Важно, что на всех этапах жизненного пути И. А. Мизин умело сочетал активную работу инженера-конструктора систем и комплексов передачи данных в интересах различных заказчиков (в том числе в качестве главного конструктора) с глубокими теоретическими исследованиями в этой предметной области. Он автор более 190 научных работ в открытой печати, в том числе 12 моно-

графий, 12 изобретений, более ста отчетов по НИР, а также огромного числа закрытых проектов по созданию систем и комплексов различного назначения. Научные работы И. А. Мизина отличались глубиной и оригинальностью решений, сочетанием серьезного математического аппарата с инженерной интуицией и практической направленностью.

Наиболее известные работы И. А. Мизина:

- Передача информации в сетях с коммутацией сообщений. — М.: Связь, 1972, 1977 (2-е изд., перераб. и доп.). Совместно с Уринсоном Л. С., Храмешиным Г. К.
- Сети коммутации пакетов. — М.: Радио и связь, 1986. Совместно с Богатыревым В. А. и Кулешовым А. П.
- Протоколы информационно-вычислительных сетей: Справочник / Под ред. И. А. Мизина, А. П. Кулешова. — М.: Радио и связь, 1990.
- О концепции создания российской общегосударственной и региональных интегрированных сетей передачи информации // Электросвязь, 1993. № 12.
- Состояние и перспективы развития информационных телекоммуникационных технологий для сферы образования и науки // Образование и информатика: Доклады II Международного конгресса ЮНЕСКО, 1996.
- Некоторые проблемы создания единой информационно-телекоммуникационной системы общенационального масштаба как основы информатизации сферы образования в России // Системы и средства информатики. Вып. 8.
- Информационные технологии в образовании: от компьютерной грамотности к информационной культуре общества. — М.: Наука, 1996. С. 114–124. Совместно с Киселевым Э. В., Соколовым И. А., Шоргиным С. Я.
- Современное состояние проблематики интегрированных информационно-телекоммуникационных систем и сетей // Системы и средства информатики. — М.: Наука, 1999. Вып. 9. С. 11–23.

В настоящей статье научный и практический вклад академика И. А. Мизина рассматривается в рамках трех направлений его научно-практической деятельности, связанных с разработкой и внедрением СОД с пакетной коммутацией в интересах АСУ Вооруженных Сил, обоснованием информационных технологий для создания крупных территориальных систем передачи данных и созданием информационных сетей в интересах органов государственной власти.

2 И. А. Мизин — создатель системы обмена данными в интересах автоматизированной системы управления стратегического назначения

Становление И. А. Мизина и как инженера-конструктора, и как ученого происходило в стенах НИИ автоматической аппаратуры (НИИ АА), ведущего в стране предприятия в области автоматизированных систем управления [3].

В первые годы И. А. Мизин занимался исследованиями и разработкой методов повышения достоверности передачи информации по каналам связи различной физической природы, методов помехоустойчивого кодирования, вероятностных моделей дискретных каналов связи, которые были реализованы в нескольких устройствах повышения достоверности для автоматизированных систем с искажениями. Его целеустремленность и работоспособность, а также достигнутые успехи были отмечены руководством института: в 1963 г. он назначается начальником лаборатории аппаратуры передачи данных.

Судьбоносным для И. А. Мизина стал 1967 год. Постановлением ЦК КПСС и Совета Министров была задана разработка автоматизированной командной системы боевого управления, а НИИ АА определен головной организацией по созданию этой системы. При формировании коллектива исполнителей И. А. Мизин руководством института был назначен Главным конструктором СОД, которая должна была обеспечить беспрецедентно высокие вероятностно-временны́е характеристики информационного обмена данными между объектами, распределенными по территории всей страны.

«*В то уже далекое время, когда в 1967 году в НИИ автоматической аппаратуры была поставлена задача создать телекоммуникационную СОД для проектируемой тогда большой территориально распределенной АСУ, проблема ее проектирования представлялась молодому коллективу разработчиков во главе с его руководителем И. А. Мизиным не такой уж сложной. Однако жизнь быстро опровергла это изначальное представление, поскольку предъявленные к ней оперативно-технические требования оказались весьма жесткими как в части научно-теоретической проработки системных вопросов, так и в плане ее конкретного системотехнического проектирования и внедрения. Практически поиск приемлемых научных и системотехнических решений пришлось начинать почти с нуля*», — вспоминает Г. К. Храмешин, один из ближайших соратников Мизина [4].

Проведенные предпроектные исследования показали, что структура создаваемой СОД должна строиться на новых решениях, отличных от используемых тогда иерархических структур с непосредственными связями управляющих звеньев АСУ с многочисленными объектами ее нижних уровней управления. Такую структуру практически невозможно было реализовать из-за слишком больших объемов технических средств передачи-приема данных на управляющих объектах обслуживаемой АСУ, которые стали бы очень громоздкими и энергоемкими, а СОД в целом оказалась бы системой с низкими показателями надежности функционирования.

В связи с этим И. А. Мизиным совместно с Л. С. Уринсоном и Г. К. Храмешиным был предложен принципиально новый в то время вариант структурного построения СОД на основе центров коммутации пакетов (ЦКП) двух модификаций: главных — в районах размещения объектов высших звеньев АСУ — и территориальных, размещаемых по всей территории СССР [5]. Объекты управления, являющиеся абонентами СОД, подключаются двумя–тремя канала-

ми связи к двум–трем ближайшим ЦКП. Такое построение позволяло обеспечить резкое сокращение необходимого количества каналов связи, высокую надежность связности всех абонентов-объектов, но вместе с тем приводило к необходимости решения комплекса сложных вопросов обеспечения процессов их информационного взаимодействия.

В рамках такой концепции под непосредственным руководством И. А. Мизина был проработан и успешно решен целый ряд новых научно-теоретических и системотехнических задач, обеспечивших выполнение множества требований в интересах АСУ. Основные из этих задач [3, 4]:

- обоснование структуры циркулирующих в СОД сообщений и системы их адресования;
- разработка совокупности (стека) алгоритмов, протоколов и интерфейсов, обеспечивающих передачу через СОД всех необходимых видов информации;
- создание в составе СОД подсистемы контроля функционирования и управления ее работой;
- создание специализированных аппаратно-программных комплексов ЦКП и аппаратно-программных комплексов оконечных средств, устанавливаемых на объектах управления АСУ — абонентах СОД;
- разработка комплекса организационно-технических мероприятий по защите информации, циркулирующей в СОД.

Успешное решение этих задач позволило обеспечить высокие вероятностно-временны́е и надежностные характеристики доведения информации, а также показатели достоверности передачи сообщений по каналам связи, в том числе низкого качества.

Кроме того, коллективом И. А. Мизина был своевременно выполнен комплекс работ по вводу СОД в эксплуатацию, который включал отладку средств и комплексов на объектах эксплуатации, проведение различных видов испытаний (автономных, предварительных, государственных) программно-аппаратных средств, комплексов и опытного участка СОД в целом, а также поддержание функционирования системы в штатном режиме.

Сравнительно немногочисленный коллектив инженеров, который во второй половине 1960-х годов заложил основы новых системотехнических решений для создания СОД и ее технических комплексов, под руководством И. А. Мизина постоянно рос, повышал свой научно-технический потенциал и к моменту постановки системы на опытную эксплуатацию (1980 г.) превратился в мощнейший коллектив высококвалифицированных опытных ученых, конструкторов, инженеров и специалистов.

В процессе создания СОД коллектив, возглавляемый главным конструктором И. А. Мизиным, работал в тесном контакте как с сотрудниками других подразделений института, так и с учеными и специалистами институтов и заводов внешней кооперации. Основные из них: НПО «Красная заря» (г. Ленинград),

НИИССУ (г. Москва), НИИ автоматики (г. Москва), Ереванский НИИ математических машин, Пензенский НИИЭИ, Тамбовский завод «Тамбоваппарат», Рязанский завод «Красное знамя», Горьковский НИИРС, КБ завода «Сигнал» (г. Кишинев), Электромеханический завод (г. Свердловск), монтажные предприятия «Каскад» и др. [3].

В своей работе И. А. Мизин и его коллектив разработчиков постоянно и конструктивно взаимодействовали с Генеральным заказчиком АСУ (маршал А. И. Белов, генералы К. Н. Трофимов, А. П. Зименков, полковник А. П. Жуковский и др.) и с головными военными НИИ, прежде всего с 16 ЦНИИИ связи и 27 ЦНИИ [6, 7].

Создание первой в СССР телекоммуникационной системы с коммутацией пакетов для командной АСУ и ее постановка в режим круглосуточной эксплуатации стало крупным научно-техническим достижением отечественной системотехники. И. А. Мизин, как Главный конструктор, в 1981 г. стал лауреатом Ленинской премии СССР, а ряд его сотрудников удостоены Государственной премии СССР, награждены орденами и медалями.

Новые решения, заложенные в основу построения СОД, оказались эффективными и перспективными, обеспечившими возможность ее эволюционного развития в базовую СОД (БСОД) как основы объединенной СОД.

Именно поэтому БСОД, которая в 1980 г. была поставлена на опытную эксплуатацию, с 1985 г. в режиме боевого дежурства более четверти века успешно выполняла задачи по информационному обмену в интересах сотен территориально удаленных объектов АСУ стратегического назначения [8].

Сегодня можно смело утверждать, что многие системотехнические решения по построению и функционированию СОД, разработанные и успешно внедренные под руководством И. А. Мизина, предвосхитили ряд идей, которые в последующем легли в основу передовых зарубежных разработок в области телекоммуникационных сетей и технологий. Эти результаты остались неизвестными международной научной общественности в силу закрытого характера работ НИИ АА.

3 Информационные технологии для создания крупных территориальных телекоммуникационных сетей

И. А. Мизин был крупным ученым в области информатики и внес большой вклад в развитие информационно-вычислительных и телекоммуникационных систем не только специального, но и общегосударственного назначения.

К наиболее значимым достижениям в этой области следует отнести развитие теоретических основ информационно-телекоммуникационных технологий, разработку информационных технологий построения крупномасштабных информационно-вычислительных и коммуникационных сетей, а также создание и внедрение комплекса базовых технических и программных средств [1, 2, 9].

Естественно, что методологической базой для этих достижений стали теоретические результаты и системотехнические решения по созданию СОД с использованием метода пакетной коммутации, разработанные в период работы в НИИ АА.

Большое значение для развития отечественной информатики имели концептуальные работы И. А. Мизина по состоянию и перспективам развития телекоммуникационных технологий. В своем докладе на эту тему, сделанном на II Международном конгрессе ЮНЕСКО «Образование и информатика» (1996 г.), И. А. Мизин определил основные этапы развития телекоммуникационных технологий следующим образом: телеграфные и телефонные сети (докомпьютерная эпоха); передача данных между отдельными абонентами по выделенным и коммутируемым каналам с использованием модемов; сети передачи данных с коммутацией пакетов; локальные вычислительные сети (ЛВС); цифровые сети интегрального обслуживания (ISDN — Integrated Services for Digital Natwork) — узкополосные, а затем широкополосные; высокоскоростные распределенные сети — Frame Relay, SMDS (Switched Multimegabit Data Service), ATM (Asynchronous Transfer Mode); высокоскоростные локальные сети — Fast Ethernet, FDDI (Fiber Distributed Data Interface), FDDI II; информационные супермагистрали. При этом в качестве наиболее перспективных технологий И. А. Мизин определил распределенные сети на оптоволокне, беспроводные сети мобильных абонентов, Интернет и электронную почту. Эти основополагающие выводы остаются актуальными и сегодня.

Особое внимание И. А. Мизин уделял развитию информатики и информационных технологий с учетом специфических условий России. Он отмечал, что создание современной телекоммуникационной инфраструктуры России является сложной масштабной задачей, для успешного решения которой необходимо комплексное проведение работ по трем направлениям: реализация крупномасштабных общегосударственных проектов; развитие и поддержка региональных телекоммуникационных проектов; поддержка деятельности негосударственных организаций в этой области.

И. А. Мизиным была поставлена задача описания класса крупномасштабных телекоммуникационных систем двойного применения как полносвязных территориально структурированных мультисетевых систем общенационального масштаба, совмещающих функции специальных систем и систем общего пользования [9].

Научные работы И. А. Мизина отличались глубиной и оригинальностью решений, сочетанием серьезного математического аппарата с инженерной интуицией, практической направленностью. Он всегда уделял большое внимание научной обоснованности предлагаемых решений, в частности методам математического моделирования и оптимизации ИТС. По инициативе И. А. Мизина в 1992 г. в ИПИ РАН был создан новый отдел — «Отдел исследований и разработки принципов построения информационно-вычислительных систем», одной из задач которого стало решение научных задач по разработке методов

и средств моделирования реальных телекоммуникационных систем. В частности, проводились фундаментальные теоретические исследования, разрабатывались математические и программные средства и модели, их экспериментальная апробация, анализ и оптимизация характеристик ИТС (процессов доставки информации с разными дисциплинами обслуживания и входными потоками, процессов управления и комплексной обработки информации).

В этой области под руководством и при непосредственном участии И. А. Мизина получен целый ряд результатов [9]. Среди наиболее значимых результатов следует отметить комплекс математических методов, алгоритмов и программ для расчета общесетевых характеристик телекоммуникационных сетей большой размерности, учитывающих реальные технологии, определяемые действующими международными стандартами и рекомендациями. Под руководством И. А. Мизина коллективом ИПИ РАН в рамках сотрудничества с компанией SOTAS (США) были разработаны концептуальная, математическая и программная модели телекоммуникационной сети, использующей популярную в 1990-е гг. технологию передачи Frame Relay.

Особо необходимо отметить усилия И. А. Мизина по практической реализации полученных научных результатов. Так, в начале 1990-х гг. в России начались процессы создания телекоммуникационных сетей, обеспечивающих ограниченный спектр услуг и построенных, как правило, на оборудовании зарубежного производства. И. А. Мизин получил предложение от Государственного комитета по науке и технике (ГКНТ) построить сеть передачи данных в интересах администрации Псковской области. К моменту поступления такого предложения коллективом ИПИ РАН был обоснован замысел создания отечественного коммутатора, реализующего протокол X.25 и его модификации. Этот коммутатор и стал технической основой региональной сети передачи данных в Псковской области, которая под руководством И. А. Мизина в период 1994–1995 гг. была разработана и введена в действие [10].

Рассматривая вклад академика И. А. Мизина в решение стратегических проблем информатизации общества в России, нельзя не сказать о разработанной в ИПИ РАН под его научным руководством концепции создания единой ИТС общенационального масштаба (ЕИТС). Основные положения этой концепции были опубликованы еще в 1996 г., но многие из них остаются актуальными и сегодня.

4 И. А. Мизин — Генеральный конструктор комплекса работ по созданию информационных систем органов государственной власти

На начало 1990-х гг. пришелся переломный момент в становлении и использовании информационных технологий в России в различных сферах: промышленности, науке, образовании, а также в государственном управлении.

В сентябре 1994 г. вышел указ Президента РФ, поставивший актуальнейшую задачу модернизации информационных и телекоммуникационных систем органов государственной власти на базе использования последних достижений в этой отрасли. Этим же указом И. А. Мизин был назначен Генеральным конструктором, а Институт проблем информатики РАН определен головной организацией.

Предстояла не только сложная, но и во многом пионерская работа [9]. Необходимо было с новых позиций изучить процессы управления, вычленить основные функциональные задачи, сформулировать количественные и качественные характеристики, необходимый набор исходной информации и, наконец, создать комплексы аппаратно-программных средств (в том числе сетевого исполнения), позволяющие с требуемыми характеристиками автоматизировать эти процессы. В итоге требовалось создать компьютерные системы поддержки принятия решений и автоматизации процессов государственного управления.

С одной стороны, требовалось обеспечить преемственность в использовании уже имеющегося задела, с другой — внедрить самые современные и перспективные технологии и средства для его реализации. Масштаб и сложность такой задачи многим представлялись откровенно пугающими; нередки были голоса, предрекающие полный крах замысла и скорое прекращение работы.

И здесь в полной мере проявились такие качества личности И. А. Мизина, как многоплановость, широта и целеустремленность. В нем органично сочетались качества ученого, способного воспринять и правильно оценить новое, и инженера, точно знающего, что необходимо для достижения цели.

Именно в процессе этой работы были сформулированы фундаментальные принципы автоматизации процессов государственного управления (применительно к России), разработаны системно-технические решения (вплоть до конкретных аппаратно-программных решений) построения всех видов автоматизированных систем, которые долгое время (вплоть до наших дней) являются категориями первого выбора для многих разработчиков в нашей стране. Кратко об основных из этих решений [9].

В качестве первого из них следует назвать выбор архитектуры объектовых комплексов средств автоматизации (КСА), или, в терминологии, предложенной И. А. Мизиным, комплексов средств информатизации (КСИ). Что предпочесть — централизованную архитектуру, основанную на одном мощном центральном вычислителе с подключенными к нему терминалами, или децентрализованную клиент-серверную архитектуру на базе интеллектуальных автоматизированных рабочих мест (АРМ) и целевых серверов, объединенных ЛВС?

В результате углубленных аналитических исследований и практических экспериментов, включая анализ опыта крупнейших зарубежных компаний, в качестве решения была обоснована клиент-серверная архитектура ЛВС. Отказ от многолетней и широко распространенной в Министерстве обороны и других силовых ведомствах практики использования терминальных сетей на базе универсальных ЭВМ (ЕС ЭВМ, IBM) был обусловлен тем, что произошел коренной перелом в использовании средств вычислительной техники и теле-

коммуникаций. Существо этого перелома заключалось, во-первых, в том, что подавляющему большинству пользователей стали доступны мощные персональные ЭВМ (ПЭВМ), а во-вторых, в том, что эти средства перестали служить только для решения узкоспециализированных задач высококвалифицированным персоналом. Поэтому при создании названных выше систем задача состояла в обеспечении пользователя (специалиста) всем набором программных средств, позволяющих ему заниматься своей профессиональной деятельностью. Краткие сроки создания системы не позволяли начинать все разработки функционального программного обеспечения «с нуля», а требовали применения (адаптации) уже готового программного обеспечения независимых производителей. Это, в свою очередь, означало, что перед создателями системы ставилась не только задача интеграции на одном АРМ системного программного обеспечения различных производителей для его совместной работы, но и задача поддержки последующего расширения перечня устанавливаемых программных средств. Имевшиеся на текущий момент тенденции развития архитектур программных систем однозначно указывали на необходимость выбора архитектуры приложения, в которой прикладные и пользовательские сервисы реализованы на клиентской рабочей станции, а данные централизованно хранятся и обрабатываются на сервере.

Другим важным вопросом был вопрос выбора технологии организации ЛВС.

В отличие от сегодняшней ситуации, когда всецело и безраздельно «господствует» протокол Ethernet, в начале 1990-х гг. на равных применялись и Ethernet (тонкий и толстый), и Token Ring, и FDDI, и ряд других протоколов.

На основе анализа отечественной и зарубежной научно-технической литературы по данной проблематике, проведения математического моделирования и стендового макетирования в качестве наиболее рационального решения была выбрана технология Ethernet, в дальнейшем с использованием коммутаторов ЛВС.

Вместе с тем стало понятно, что эффективность решения поставленных задач во многом будет определяться не только архитектурой и технологией ЛВС, но и организацией хранения информации, методами ее поиска и извлечения.

Жесткие ограничения по производительности процессоров начала 1990-х гг., ограниченные объемы оперативной памяти при необходимости реализации сложнейших алгоритмов обработки данных часто приводили коллективы разработчиков к собственным решениям по способам структуризации данных в рамках сетевой модели, построению уникальных индексных систем, собственных языков манипулирования данными, их интерпретаторов и т. п. Особенно остро эти проблемы стояли при обработке неформализованной текстовой информации на русском языке.

В этих условиях, несмотря на серьезные возражения, были обоснованы стратегические решения по унификации подходов к хранению данных в рамках реляционной модели и использованию языка SQL (Structured Query Language) как унифицированного средства взаимодействия с системами управления базами

данных (СУБД). В результате этой работы при наличии нескольких десятков вариантов в 1995 г. в качестве основных были выбраны три линейки СУБД: СУБД Oracle (один из лидеров рынка на тот момент) для мощных серверов, обрабатывающих большие объемы фактографической и документальной информации с интенсивным потоком транзакций; персональная СУБД Microsoft Access как альтернатива многочисленным линейкам СУБД форматов dBASE и Paradox для создания баз данных для небольших групп пользователей; СУБД Microsoft SQL Server для технологических и учетных баз данных.

Впоследствии на основе этих СУБД было разработано множество приложений, которые, с небольшими изменениями, работают и поныне. И сейчас, спустя почти 20 лет, эти линейки СУБД остаются доминирующими в своих сегментах, а их многочисленные конкуренты (Sybase SQL, Watcom SQL, Informix, Interbase, Paradox, FoxPro) практически исчезли с российского IT-рынка.

Третьим важнейшим результатом стали решения по контролю состояния систем, управлению их функционированием и эксплуатацией. Создаваемые крупномасштабные системы требовали новых, современных решений в этой части. И. А. Мизин фундаментально подошел к решению этих вопросов. Здесь необходимо отметить, что особое внимание к этим вопросам было обусловлено опытом Игоря Александровича, приобретенным в период создания под его руководством СОД. Разработанная в ее рамках система контроля состояния и управления сыграла огромную роль в обеспечении длительного непрерывного и эффективного функционирования СОД.

В 1990-е гг. появился целый ряд новых продуктов нового класса — сетевых платформ управления (к тому времени уже были специфицированы и приняты в качестве международных стандартов протоколы SNMP — Simple Network Management Protocol)). Наиболее известными были продукты фирмы Hewlett Packard (HP) — линейка продуктов HP Open View — и Computer Associates (CA) — программный комплекс UniCenter. ИПИ РАН были приобретены оба продукта и проведены всесторонние аналитические исследования на различных конфигурациях серверного оборудования и средствах телекоммуникаций, операционных системах, СУБД, разнообразном общесистемном и прикладном программном обеспечении. Более того, специалисты института прошли обучение на курсах фирм HP и CA.

В результате было принято решение разрабатывать собственную систему управления с использованием отдельных модулей из линейки HP Open View (в части управления аппаратными средствами и контроля трафика) и модулей из состава продукта UniCenter в части управления программным обеспечением.

Еще одно из важнейших направлений работ связано с решениями по обеспечению сопряжения систем, создаваемых на базе современных технологий, и систем, уже созданных и функционирующих на территории РФ, с учетом совместного использования их потенциала (на современном языке — сохранения инвестиций).

Здесь необходимо отметить два важнейших технических решения.

Первое из них — сопряжение с наиболее массовыми и территориально распространенными сетями того времени: телеграфными сетями АТ-50 и Телекс. Такое решение позволяло повысить вероятность доведения сообщений ограниченного объема (команды, сигналы, распоряжения и т. п.) с учетом увеличения множества возможных путей передачи информации за счет разветвленности телеграфной сети. С учетом таких потенциальных возможностей этих сетей в ИПИ РАН был разработан специальный комплекс сопряжения с сетями абонентского телеграфирования АТ-50 и Телекс для обеспечения обмена телеграфными сообщениями между абонентами объектовых ЛВС (Ethernet) и абонентами телеграфных сетей АТ-50 и Телекс.

Вторая работа — сопряжение с объектами управления, подключенными к базовой СОД Вооруженных Сил РФ, которая к началу 1990-х гг. была основной транспортной сетью для автоматизированных систем управления Генерального штаба, различных видов вооруженных сил и родов войск (АСУ ВС РФ). Сопряжение обеспечивало возможность информационного обмена с объектами военных систем. Для реализации такого сопряжения потребовалась разработка специального комплекса и правил по преобразованию принятых в базовой СОД форматов кодограмм в форматы протокола Ethernet, что позволило обеспечить информационное взаимодействие абонентов сети Ethernet модернизируемой системы органов государственной власти и абонентов КСА автоматизированных систем военного назначения. И такой, весьма непростой, комплекс сопряжения был создан.

В процессе проектирования было рассмотрено несколько вариантов реализации этого сопряжения, и в качестве основного был выбран вариант, при котором комплекс сопряжения модернизируемой системы органов государственной власти подключался по выделенным каналам связи к центрам коммутации базовой СОД в виде виртуального объекта управления, обладающего необходимыми правами и характеристиками.

Естественно, что успешному решению этой задачи способствовало то, что И. А. Мизин как Главный конструктор СОД прекрасно владел существом и особенностями технических решений, реализованных в базовой СОД АСУ ВС РФ.

Совершенно новые задачи коллективу ИПИ РАН под руководством И. А. Мизина пришлось решать в области информационной безопасности. Одновременно с формированием нормативной базы (совместно с ФАПСИ, ФСБ и Министерством обороны) были апробированы различные подходы и решения, в том числе и не нашедшие дальнейшего применения. В конечном итоге вышли на использование решений, которые определяют современное состояние защищенности информации, содержащей государственную тайну, в автоматизированных системах. Разработанные средства позволяют обеспечить защиту информации при работе пользователей в доменной структуре, обмене сообщениями электронной почты, доступе к информации в базах данных и при совместной работе

над документами. Средства криптографической защиты реализуют шифрование и имитозащиту файлов, функции электронного замка, шифрование внутрисетевого трафика и защиту внешних каналов связи. Впервые в изделиях были реализованы криптоалгоритмы семейства «Метель», разработаны специальные считывающие устройства, в качестве носителей ключевой информации применена Российская интеллектуальная карта, реализованы различные типы интерфейсов, включая высокоскоростные оптоволоконные. Разработанные изделия успешно прошли сертификацию в ФСБ и широко применяются в защищенных сетях органов государственной власти РФ.

Оперативность и правильность принятия решений руководством страны во многом зависит от наглядности и качества представления данных, что потребовало широкого внедрения технологий геоинформационных систем (ГИС), позволяющих эффективно взаимоувязать картографическую основу и фактографические базы данных, а также быстро отобразить огромный объем накопленных данных в пригодном для восприятия виде. Эффективность ГИС в процессе оперативного принятия решений была настолько очевидной, что под них создавались специализированные средства отображения информации — «Видеостены», которые позволяли получать приемлемые характеристики по разрешению, яркости и размеру картографического изображения при достаточно скромных характеристиках единичного технического элемента (проектора, видеокуба). В состав интегрированного программно-аппаратного комплекса, реализующего функции представления информации для принятия решения, входил экран коллективного пользования проекционного типа на базе видеокубов, индивидуальные средства отображения и аппаратно-программные средства поддержки.

Под руководством И. А. Мизина применительно к задачам государственного управления была показана важность создания ситуационных центров (СЦ). Ситуационный центр должен представлять собой комплекс, состоящий из АРМ для руководства и групп специалистов, а также из комплексов специального программного обеспечения, которые обеспечивают оперативный анализ ситуации, «проигрывание» различных сценариев. Главная задача СЦ — поддержание информационного хранилища данных в актуальном состоянии на основе сбора информации из различных источников и создание условий для интеллектуальной деятельности специалистов по информационной поддержке принятия управленческих решений на основе современных информационных технологий.

Как видим, И. А. Мизиным были предугаданы и сформулированы основные принципы и технические решения для широчайшего класса автоматизированных систем органов государственной власти РФ.

5 Заключение

Вся научно-практическая деятельность академика И. А. Мизина на протяжении более 40 лет была связана с проблематикой информационных и телекоммуникационных сетей. Под его руководством и при его непосредственном

участии были решены многие проблемы информатизации процессов государственного управления, разработаны концептуальные документы по стратегии взаимосогласованного развития систем и комплексов связи и автоматизированного управления войсками, силами и оружием с применением новых информационных технологий, выработаны организационные и системотехнические принципы координации работ по автоматизации процессов управления во всех звеньях Вооруженных Сил РФ, выполнен крупный комплекс работ по созданию базовых функционально-ориентированных межвидовых унифицированных аппаратно-программных средств вычислительной техники для перспективных телекоммуникационных комплексов нового поколения, разработаны алгоритмы автоматической обработки и селекции разнородных потоков текстовой информации и моделирования критических ситуаций, разнесенных во времени и пространстве, созданы математические модели сложных адаптивных телекоммуникационных сетей.

И. А. Мизин постоянно вел большую научно-организационную работу. Он был председателем секции «Теория передачи и обработки информации» Научного совета по комплексной проблеме «Кибернетика» АН СССР, членом комиссии по вычислительным центрам коллективного пользования и сетям ЭВМ при Президиуме АН СССР, членом Совета руководителей Академсети, членом советской рабочей группы международной организации по стандартизации, председателем Научного совета РАН по информационно-вычислительным сетям, членом Президиума научно-технического совета по программе «Информатизация России», членом Координационного совета по информатизации при Администрации Президента РФ, членом научных советов при Совете Безопасности РФ, Совете обороны РФ, Минэкономики РФ, членом Президиума НТС при Правительстве г. Москвы.

С 1996 г. и до последних дней И. А. Мизин вел активную работу как член бюро Отделения информатики, вычислительной техники и автоматизации РАН.

И. А. Мизин много внимания уделял подготовке научных и инженерных кадров. Он был заведующим базовыми кафедрами в Московском государственном университете радиотехники, электроники и автоматики (МИРЭА) и в Московском техническом университете связи и информатики (МТУСИ), председателем двух специализированных советов по защите докторских диссертаций. Подготовил значительное число кандидатов и докторов технических наук.

И. А. Мизин обладал удивительной способностью создавать в науке и промышленности успешно работающие большие коллективы специалистов, объединять вокруг себя талантливую молодежь.

После прихода в ИПИ РАН И. А. Мизин установил и поддерживал активные контакты с зарубежными коллегами. Он получил международную известность и снискал глубокое уважение многих ученых, профессоров и деловых партнеров в США, Германии, Италии, странах Юго-Восточной Азии.

Научная, научно-организационная и педагогическая деятельность И. А. Мизина была по достоинству оценена нашей Родиной. Он был лауреатом Ленинской

и Государственной премий, награжден орденом Трудового Красного Знамени, орденом «За заслуги перед Отечеством» 4-й степени, многими медалями.

Игорь Александрович Мизин умер в расцвете творческих сил 8 сентября 1999 г. после тяжелой болезни. Он похоронен в Москве на Востряковском кладбище.

Коллеги и ученики И. А. Мизина продолжают начатые им работы, реализуя и развивая его научные идеи в различных организациях Российской академии наук, промышленности и в технических университетах.

Литература

1. *Захаров В. Н., Соколов И. А.* Академик Игорь Александрович Мизин // История науки и техники, 2008. № 7. С. 53–56.
2. *Зацаринный А. А., Захаров В. Н., Синицин И. Н.* ИПИ РАН и создание информационно-телекоммуникационных сетей в интересах органов государственной власти Российской Федерации (к 25-летию ИПИ РАН) // Ведомственные корпоративные сети, 2008. № 2(47). С. 118–128.
3. Автоматизация управления. Наш путь: К 50-летию НИИ автоматической аппаратуры им. академика В. С. Семенихина. — М.: НИИ АА, 2006. 210 с.
4. *Храмешин Г. К.* Мизин — главный конструктор первой в СССР телекоммуникационной системы обмена данными на принципах коммутации пакетов. Основные научные и системотехнические проблемы ее создания // И. А. Мизин — ученый, конструктор, человек. — М.: ИПИ РАН, 2010. С. 129–132.
5. *Щукин Л. Б.* И. А. Мизин: системное мышление, новаторство, целеустремленность // И. А. Мизин — ученый, конструктор, человек. — М.: ИПИ РАН, 2010. С. 136–141.
6. *Зацаринный А. А.* О роли и вкладе института в создание и развитие системы обмена данными АСУ ВС // Институт военной связи. История и современность. 1923–2008 гг. — Мытищи, 2008. С. 230–237.
7. *Зацаринный А. А.* Академик И. А. Мизин: военная наука и практика // И. А. Мизин — ученый, конструктор, человек. — М.: ИПИ РАН, 2010. С. 96–128.
8. *Зацаринный А. А.* Аппаратура систем и комплексов обмена данными // История отечественных средств связи. — М.: Столичная энциклопедия, 2013. С. 501–506.
9. *Соколов И. А.* Вклад академика И. А. Мизина в развитие отечественных информационных технологий и систем // И. А. Мизин — ученый, конструктор, человек. — М.: ИПИ РАН, 2010. С. 85–95.
10. *Гайкович Ю. В.* Региональные телекоммуникации от Мизина // И. А. Мизин — ученый, конструктор, человек. — М.: ИПИ РАН, 2010. С. 193–196.

Поступила в редакцию 30.01.15

ABOUT ACADEMICIAN I. A. MIZIN'S CONTRIBUTION TO THEORY AND PRACTICE OF DOMESTIC INFORMATION-TELECOMMUNICATION SYSTEMS CREATION: TO THE 80th ANNIVERSARY

I. A. Sokolov, A. A. Zatsarinny, and V. N. Zakharov

Federal Research Center "Computer Science and Control," Russian Academy of Sciences, 44-2 Vavilov Str., Moscow 119333, Russian Federation

Abstract: The article is devoted to the 80th anniversary of academician I. A. Mizin, head of IPI RAS during 1989–1999, outstanding scientist, designer and engineer. A brief biography concerning his scientific work is presented. The scientific and practical contribution of I. A. Mizin to the theory and its applications in creation of domestic information-telecommunication systems (ITS) is considered in three directions of his work. The first direction is development and implementation of the data communication system for the purposes of Armed Force ACS (automated control system), which was the first domestic network with packet commutation. The second direction is justification of information technologies for creation of the huge territorial system of data communication in the regions of Russia. The third direction is creation and development of information networks for the purposes of government with requirements of information protection.

Keywords: chief designer; academician; information telecommunication networks; information technologies; data communication system; packet commutation; methods of data communication; information protection; link channels; trial-run; government

DOI: 10.14357/08696527150114

References

1. Zakharov, V. N., and I. A. Sokolov. 2008. Akademik Igor' Aleksandrovich Mizin [Academician Igor Alexandrovich Mizin]. *Istoriya Nauki i Tekhniki* [Science and Technics History] 7:53–56.
2. Zatsarinny, A. A., V. N. Zakharov, and I. N. Sinitsyn. 2008. IPI RAN i sozdanie informatsionno-telekommunikatsionnykh setey v interesakh organov gosudarstvennoy vlasti Rossiyskoy Federatsii (k 25-letiyu IPI RAN) [IPI RAS and creation of information telecommunication networks for Russian Government purposes (to the 25th IPI RAS anniversary)]. *Vedomstvennye Korporativnye Seti* [Departmental Corporate Networks] 2(47):118–128.
3. 2006. *Avtomatizatsiya upravleniya. Nash put'. K 50-letiyu NII avtomaticheskoy apparatury im. akademika V. S. Semenikhina* [Control automation. Our path. To the 50th anniversary of academician V. S. Semenikhin Research Institute of Automated Apparatura]. Moscow: NII AA. 210 p.

4. Khrameshin, G. K. 2010. Mizin — glavnyy konstruktor pervoy v SSSR telekommunikatsionnoy sistemy obmena dannymi na printsipakh kommutatsii paketov. Osnovnye nauchnye i sistemotekhnicheskie problemy ee sozdaniya [Mizin as a chief designer of the first in USSR telecommunication system of data exchange made by principles of packet commutation. Main science and systemotechnic problems of its creation]. *I. A. Mizin — uchenyy, konstruktor, chelovek* [Mizin as a scientist, a designer, a man]. Moscow: IPI RAS. 129–132.
5. Shchukin, L. B. 2010. I. A. Mizin: Sistemnoe myshlenie, novatorstvo, tseleustremlennost' [I.A. Mizin: Systemic thinking, innovation, sense of purpose]. *I. A. Mizin — uchenyy, konstruktor, chelovek* [Mizin as a scientist, a designer, a man]. Moscow: IPI RAS. 136–141.
6. Zatsarinny, A. A. 2008. O roli i vklade instituta v sozdanie i razvitie sistemy obmena dannymi ASU VS [About institute role and contribution to creation and development of data exchange system of Armed Forces ACS]. *Nauchno-istoricheskiy sbornik «Institut voennoy svyazi. Istoriya i sovremennost'. 1923–2008 gg.* [Science-historical digest "Institute of military communication. History and modernity"]. Mytishchi. 230–237.
7. Zatsarinny, A. A. 2010. Akademik I. A. Mizin: Voennaya nauka i praktika [An academician I. A. Mizin: Military science and practice]. *I. A. Mizin — uchenyy, konstruktor, chelovek* [Mizin as a scientist, a designer, a man]. Moscow: IPI RAS. 96–128.
8. Zatsarinny, A. A. 2013. Apparatura sistem i kompleksov obmena dannymi [Apparature of data exchange systems and complexes]. *Istoriya otechestvennykh sredstv svyazi* [The history of domestic communication means]. Moscow: Stolichnaya entsiclopediya. 501–506.
9. Sokolov, I. A. 2010. Vklad akademika I. A. Mizina v razvitie otechestvennykh informatsionnykh tekhnologiy i sistem [Academician I. A. Mizin's contribution to development of domestic information technologies and systems]. *I. A. Mizin — uchenyy, konstruktor, chelovek* [Mizin as a scientist, a designer, a man]. Moscow: IPI RAS. 85–95.
10. Gaykovich, Yu. V. 2010. Regional'nye telekommunikatsii ot Mizina [Regional telecommunications by Mizin]. *I. A. Mizin — uchenyy, konstruktor, chelovek* [Mizin as a scientist, a designer, a man]. Moscow: IPI RAS. 193–196.

Received January 30, 2015

Contributors

Sokolov Igor A. (b. 1954) — Academician of the Russian Academy of Sciences, Doctor of Science in technology, Director, Federal Research Center "Computer Science and Control," Russian Academy of Sciences, 44-2 Vavilov Str., Moscow 119333, Russian Federation; isokolov@ipiran.ru

Zatsarinny Alexander A. (b. 1951) — Doctor of Science in technology, professor, Deputy Director, Federal Research Center "Computer Science and Control," Russian Academy of Sciences, 44-2 Vavilov Str., Moscow 119333, Russian Federation; azatsarinny@ipiran.ru

Zakharov Victor N. (b. 1948) — Doctor of Science in technology, associate professor; Scientific Secretary, Federal Research Center "Computer Science and Control," Russian Academy of Sciences, 44-2 Vavilov Str., Moscow 119333, Russian Federation; vzakharov@ipiran.ru

ОБ АВТОРАХ

Андреев Арк Михайлович (р. 1943) — кандидат технических наук, доцент Московского государственного технического университета им. Н. Э. Баумана

Березкин Дмитрий Валерьевич (р. 1966) — кандидат технических наук, старший научный сотрудник Московского государственного технического университета им. Н. Э. Баумана

Воинов Никита Владимирович (р. 1984) — кандидат технических наук, доцент Санкт-Петербургского государственного политехнического университета

Газизуллина Римма Камилевна (р. 1993) — студентка Московского физико-технического института

Григорьев Семен Вячеславович (р. 1989) — аспирант Санкт-Петербургского государственного университета

Долгополов Вячеслав Сергеевич (р. 1962) — научный сотрудник Института проблем информатики Федерального исследовательского центра «Информатика и управление» Российской академии наук

Дробинцев Павел Дмитриевич (р. 1980) — кандидат технических наук, доцент Санкт-Петербургского государственного политехнического университета

Заикин Максим Юрьевич (р. 1973) — научный сотрудник Института проблем информатики Федерального исследовательского центра «Информатика и управление» Российской академии наук

Захаров Виктор Николаевич (р. 1948) — доктор технических наук, доцент, ученый секретарь Федерального исследовательского центра «Информатика и управление» Российской академии наук

Зацаринный Александр Алексеевич (р. 1951) — доктор технических наук, профессор, заместитель директора Федерального исследовательского центра «Информатика и управление» Российской академии наук

Илюшин Геннадий Яковлевич (р. 1947) — кандидат технических наук, заведующий лабораторией Института проблем информатики Федерального исследовательского центра «Информатика и управление» Российской академии наук

Киселев Эдуард Васильевич (р. 1938) — старший научный сотрудник Института проблем информатики Федерального исследовательского центра «Информатика и управление» Российской академии наук

Козлов Илья Андреевич (р. 1989) — аспирант Московского государственного технического университета им. Н. Э. Баумана

Кореньков Владимир Васильевич (р. 1953) — кандидат физико-математических наук, директор лаборатории Объединенного института ядерных исследований (ОИЯИ); заведующий кафедрой Международного университета «Дубна»

Корепанов Эдуард Рудольфович (р. 1966) — кандидат технических наук, заведующий сектором Института проблем информатики Федерального исследовательского центра «Информатика и управление» Российской академии наук

Корчагин Александр Юрьевич (р. 1989) — аспирант факультета вычислительной математики и кибернетики Московского государственного университета им. М. В. Ломоносова

Котляров Всеволод Павлович (р. 1944) — кандидат технических наук, профессор Санкт-Петербургского государственного политехнического университета

Кружков Михаил Григорьевич (р. 1975) — ведущий программист Института проблем информатики Федерального исследовательского центра «Информатика и управление» Российской академии наук

Лиманский Валерий Иванович (р. 1952) — кандидат технических наук, старший научный сотрудник Института проблем информатики Федерального исследовательского центра «Информатика и управление» Российской академии наук

Лукашевич Наталья Валентиновна (р. 1964) — кандидат физико-математических наук, ведущий научный сотрудник Научно-исследовательского вычислительного центра Московского государственного университета им. М. В. Ломоносова

Нечаевский Андрей Васильевич (р. 1982) — инженер-программист Объединенного института ядерных исследований

Никифоров Игорь Валерьевич (р. 1988) — кандидат технических наук, ассистент Санкт-Петербургского государственного политехнического университета

Обухова Ольга Леонидовна (р. 1952) — научный сотрудник Института проблем информатики Федерального исследовательского центра «Информатика и управление» Российской академии наук

Ососков Геннадий Алексеевич (р. 1931) — доктор физико-математических наук, профессор, главный научный сотрудник Объединенного института ядерных исследований

Пряхина Дарья Игоревна (р. 1992) — инженер-программист Объединенного института ядерных исследований

Рагозина Анастасия Константиновна (р. 1992) — студент магистратуры Санкт-Петербургского государственного университета

Селин Иван Андреевич (р. 1993) — ассистент Санкт-Петербургского государственного политехнического университета

Симаков Константин Васильевич (р. 1980) — кандидат технических наук, старший научный сотрудник Московского государственного технического университета им. Н. Э. Баумана

Синицын Игорь Николаевич (р. 1940) — доктор технических наук, профессор, заслуженный деятель науки РФ, заведующий отделом Института проблем информатики Федерального исследовательского центра «Информатика и управление» Российской академии наук

Соколов Игорь Анатольевич (р. 1954) — академик Российской академии наук, доктор технических наук, директор Федерального исследовательского центра «Информатика и управление» Российской академии наук

Соловьев Иван Владимирович (р. 1976) — программист 1-й категории Института проблем информатики Федерального исследовательского центра «Информатика и управление» Российской академии наук

Стенина Мария Михайловна (р. 1991) — студентка Московского физико-технического института

Стрижов Вадим Викторович (р. 1967) — кандидат физико-математических наук, доцент, научный сотрудник Вычислительного центра им. А. А. Дородницына Российской академии наук

Трофимов Владимир Валентинович (р. 1955) — ведущий программист объединенного института ядерных исследований

Ужинский Александр Владимирович (р. 1983) — кандидат технических наук, ведущий программист объединенного института ядерных исследований

Четверкин Илья Игоревич (р. 1988) — программист Научно-исследовательского вычислительного центра Московского государственного университета им. М. В. Ломоносова

Правила подготовки рукописей статей для публикации в журнале «Системы и средства информатики»

Журнал «Системы и средства информатики» публикует теоретические, обзорные и дискуссионные статьи, посвященные научным исследованиям и разработкам в области информационных технологий.

Журнал издается на русском языке. По специальному решению редколлегии отдельные статьи могут печататься на английском языке.

Тематика журнала охватывает следующие направления:

- информационно-телекоммуникационные системы и средства их построения;
- архитектура и программное обеспечение вычислительных машин, комплексов и сетей;
- методы и средства защиты информации.

1. В журнале печатаются статьи, содержащие результаты, ранее не опубликованные и не предназначенные к одновременной публикации в других изданиях.

   Публикация не должна нарушать закон об авторских правах.

   Направляя рукопись в редакцию, авторы сохраняют все права собственников данной рукописи и при этом передают учредителям и редколлегии неисключительные права на издание статьи на русском языке (или на языке статьи, если он отличен от русского) и на ее распространение в России и за рубежом. Авторы должны предоставить в редакцию письмо в следующей форме:

   Соглашение о передаче права на публикацию:

   «Мы, нижеподписавшиеся, авторы рукописи «. . . », передаем учредителям и редколлегии журнала «Системы и средства информатики» неисключительное право опубликовать данную рукопись статьи на русском языке как в печатной, так и в электронной версиях журнала. Мы подтверждаем, что данная публикация не нарушает авторских прав других лиц или организаций, а также не содержит сведений, запрещенных к опубликованию в открытой печати.

   Подписи авторов: (ф. и. о., дата, адрес)».

   Это соглашение может быть предоставлено в бумажном виде или в виде отсканированной копии (с подписями авторов).

   Редколлегия вправе запросить у авторов экспертное заключение о возможности публикации представленной статьи в открытой печати.

2. К статье прилагаются данные автора (авторов) (см. п. 8). При наличии нескольких авторов указывается фамилия автора, ответственного за переписку с редакцией.

3. Редакция журнала осуществляет экспертизу присланных статей в соответствии с принятой в журнале процедурой рецензирования.

   Возвращение рукописи на доработку не означает ее принятия к печати.

   Доработанный вариант с ответом на замечания рецензента необходимо прислать в редакцию.

4. Решение редколлегии о публикации статьи или ее отклонении сообщается авторам.

   Редколлегия может также направить авторам текст рецензии на их статью. Дискуссия по поводу отклоненных статей не ведется.

5. Редактура статей высылается авторам для просмотра. Замечания к редактуре должны быть присланы авторами в кратчайшие сроки.
6. Рукопись предоставляется в электронном виде в форматах MS WORD (.doc или .docx) или LaTeX (.tex), дополнительно — в формате .pdf, на дискете, лазерном диске или электронной почтой. Предоставление бумажной рукописи необязательно.
7. При подготовке рукописи в MS Word рекомендуется использовать следующие настройки.

   Параметры страницы: формат — A4; ориентация — книжная; поля (см): внутри — 2,5, снаружи — 1,5, сверху и снизу — 2, от края до нижнего колонтитула — 1,3.

   Основной текст: стиль — «Обычный», шрифт — Times New Roman, размер — 14 пунктов, абзацный отступ — 0,5 см, 1,5 интервала, выравнивание — по ширине.

   Рекомендуемый объем рукописи — не свыше 20 страниц указанного формата.

   Сокращения слов, помимо стандартных, не допускаются. Допускается минимальное количество аббревиатур.

   Все страницы рукописи нумеруются.

   Шаблоны примеров оформления представлены в Интернете:

   http://www.ipiran.ru/publications/collected/template.doc
8. Статья должна содержать следующую информацию на ***русском и английском языках***:

   – название статьи;
   – Ф.И.О. авторов, на английском можно только имя и фамилию;
   – место работы, с указанием города и страны и электронного адреса каждого автора;
   – сведения об авторах, в соответствии с форматом, образцы которого представлены на страницах:
     http://www.ipiran.ru/journal/collected/2012_22_02_rus/authors.asp и
     http://www.ipiran.ru/journal/collected/2012_22_02_eng/authors.asp;
   – аннотация (не менее 100 слов на каждом из языков). Аннотация — это краткое резюме работы, которое может публиковаться отдельно. Она является основным источником информации в информационных системах и базах данных. Английская аннотация должна быть оригинальной, может не быть дословным переводом русского текста и должна быть написана хорошим английским языком. В аннотации не должно быть ссылок на литературу и, по возможности, формул;
   – ключевые слова — желательно из принятых в мировой научно-технической литературе тематических тезаурусов. Предложения не могут быть ключевыми словами.
   – источники финансирования работы (ссылкм на гранты, проекты, поддерживающие организации и т. п.

9. Требования к спискам литературы.
   Ссылки на литературу в тексте статьи нумеруются (в квадратных скобках) и располагаются в каждом из списков литературы в порядке первых упоминаний.

Списки литературы представляются в двух вариантах:

(1) **Список литературы к русскоязычной части**. Русские и английские работы — на языке и в алфавите оригинала.

(2) **References**. Русские работы и работы на других языках — в латинской транслитерации с переводом на английский язык; английские работы и работы на других языках — на языке оригинала.

Необходимо для составления списка "References" пользоваться размещенной на сайте http://www.translit.ru/ бесплатной программой транслитерации русского текста в латиницу, при этом в закладке «варианты. . . » следует выбрать опцию BNG.

Список литературы "References" приводится полностью отдельным блоком, повторяя все позиции из списка литературы к русскоязычной части, независимо от того, имеются или нет в нем иностранные источники. Если в списке литературы к русскоязычной части есть ссылки на иностранные публикации, набранные латиницей, они полностью повторяются в списке "References".

Примеры ссылок на различные виды публикаций в списке "References":

Описание статьи из журнала:

Zhang, Z., and D. Zhu. 2008. Experimental research on the localized electrochemical micromachining. *Rus. J. Electrochem.* 44(8):926–930. doi:10.1134/S1023193508080077.

Описание статьи из электронного журнала:

Swaminathan, V., E. Lepkoswka-White, and B. P. Rao. 1999. Browsers or buyers in cyberspace? An investigation of electronic factors influencing electronic exchange. *JCMC* 5(2). Available at: http://www.ascusc.org/jcmc/vol5/issue2/ (accessed April 28, 2011).

Описание материалов конференций:

Usmanov, T. S., A. A. Gusmanov, I. Z. Mullagalin, R. Ju. Muhametshina, A. N. Chervyakova, and A. V. Sveshnikov. 2007. Osobennosti proektirovaniya razrabotki mestorozhdeniy s primeneniem gidrorazryva plasta [Features of the design of field development with the use of hydraulic fracturing]. *Trudy 6-go Mezhdunarodnogo Simpoziuma "Novye resursosberegayushchie tekhnologii nedropol'zovaniya i povysheniya neftegazootdachi"* [6th Symposium (International) "New Energy Saving Subsoil Technologies and the Increasing of the Oil and Gas Impact" Proceedings]. Moscow. 267–272.

Описание книги (монографии, сборники):

Lindorf, L. S., and L. G. Mamikoniants, eds. 1972. *Ekspluatatsiya turbogeneratorov s neposredstvennym okhlazhdeniem* [Operation of turbine generators with direct cooling]. Moscow: Energy Publs. 352 p.

Описание переводной книги (в списке литературы к русскоязычной части необходимо указать: / Пер. с англ. — после названия книги, а в конце ссылки указать оригинал книги в круглых скобках):

1. В русскоязычной части:
   Тимошенко С. П., Янг Д. Х., Уивер У. Колебания в инженерном деле / Пер. с англ. — М.: Машиностроение, 1985. 472 с. (*Timoshenko S. P., Young D. H., Weaver W.* Vibration problems in engineering. — 4th ed. — N.Y.: Wiley, 1974. 521 p.)

2. В англоязычной части:
   Timoshenko, S. P., D. H. Young, and W. Weaver. 1974. *Vibration problems in engineering*. 4th ed. N.Y.: Wiley. 521 p.

Описание неопубликованного документа:

Latypov, A. R., M. M. Khasanov, and V. A. Baikov. 2004. Geology and production (NGT GiD). Certificate on official registration of the computer program No. 2004611198. (In Russian, unpubl.)

Описание интернет-ресурса:

Pravila tsitirovaniya istochnikov [Rules for the citing of sources]. Available at: http://www.scribd.com/doc/1034528/ (accessed February 7, 2011).

Описание диссертации или автореферата диссертации:

Semenov, V. I. 2003. Matematicheskoe modelirovanie plazmy v sisteme kompaktnyy tor [Mathematical modeling of the plasma in the compact torus]. D.Sc. Diss. Moscow. 272 p.

Kozhunova, O. S. 2009. Tekhnologiya razrabotki semanticheskogo slovarya informatsionnogo monitoringa [Technology of development of semantic dictionary of information monitoring system]. PhD Thesis. Moscow: IPI RAN. 23 p.

Описание ГОСТа:

GOST 8.586.5-2005. 2007. Metodika vypolneniya izmereniy. Izmerenie raskhoda i kolichestva zhidkostey i gazov s pomoshch'yu standartnykh suzhayushchikh ustroystv [Method of measurement. Measurement of flow rate and volume of liquids and gases by means of orifice devices]. Moscow: Standardinform Publs. 10 p.

Описание патента:

Bolshakov, M. V., A. V. Kulakov, A. N. Lavrenov, and M. V. Palkin. 2006. Sposob orientirovaniya po krenu letatel'nogo apparata s opticheskoy golovkoy samonavedeniya [The way to orient on the roll of aircraft with optical homing head]. Patent RF No. 2280590.

10. Присланные в редакцию материалы авторам не возвращаются.
11. При отправке файлов по электронной почте просим придерживаться следующих правил:
    - указывать в поле subject (тема) название журнала и фамилию автора;
    - использовать attach (присоединение);
    - в состав электронной версии статьи должны входить: файл, содержащий текст статьи, и файл(ы), содержащий(е) иллюстрации.
12. Журнал «Системы и средства информатики» является некоммерческим изданием. Плата за публикацию не взимается, гонорар авторам не выплачивается.

Адрес редакции журнала «Системы и средства информатики»:

Москва 119333, ул. Вавилова, д. 44, корп. 2, ФИЦ ИУ РАН

Тел.: +7 (499)135-86-92 Факс: +7 (495)930-45-05

e-mail: rust@ipiran.ru (Сейфуль-Мулюков Рустем Бадриевич)

http://www.ipiran.ru/journal/collected

Requirements for manuscripts submitted to Journal "Systems and Means of Informatics"

Journal "Systems and Means of Informatics" publishes theoretical, review, and discussion articles on the research and development in the field of information technology.

The journal is published in Russian. By a special decision of the editorial board, some articles can be published in English.

Topics covered include the following areas:

- information and communication systems and tools of their design;
- architecture and software of computational complexes and networks; and
- methods and tools of information protection.

1. The Journal publishes original articles which have not been published before and are not intended for simultaneous publication in other editions. An article submitted to the Journal must not violate the Copyright law. Sending the manuscript to the Editorial Board, the authors retain all rights of the owners of the manuscript and transfer the nonexclusive rights to publish the article in Russian (or the language of the article, if not Russian) and its distribution in Russia and abroad to the Founders and the Editorial Board. Authors should submit a letter to the Editorial Board in the following form:

   Agreement on the transfer of rights to publish:

   "*We, the undersigned authors of the manuscript ". . . ," pass to the Founder and the Editorial Board of the Journal "Systems and Means of Informatics" the nonexclusive right to publish the manuscript of the article in Russian (or in English) in both print and electronic versions of the Journal. We affirm that this publication does not violate the Copyright of other persons or organizations.*

   Author(s) signature(s): (name(s), address(es), date)."

   This agreement should be submitted in paper form or in the form of a scanned copy (signed by the authors).

   The Editorial Board has the right to request from the authors an official expert conclusion that the submitted article has no classified data prohibited for publication.

2. A submitted article should be attached with **the data on the author(s)** (see item 8). If there are several authors, the contact person should be indicated who is responsible for correspondence with the Editorial Board and other authors about revisions and final approval of the proofs.
3. The Editorial Board of the Journal examines the article according to the established reviewing procedure. If authors receive their article for correction after reviewing, it does not mean that the article is approved to be published. The corrected article should be sent to the Editorial Board for the subsequent review and approval.
4. The decision on the article publication or its rejection is communicated to the authors. The Editorial Board may also send the reviews on the submitted articles to the authors. Any discussion upon the rejected articles is not possible.
5. The edited articles will be sent to the authors for proofread. The comments of the authors to the edited text of the article should be sent to the Editorial Board as soon as possible.

6. The manuscript of the article should be presented electronically in the MS WORD (.doc or .docx) or LaTeX (.tex) formats, and additionally in the .pdf format. All documents may be sent by e-mail or provided on a CD or diskette. A hard copy submission is not necessary.
7. The recommended typesetting instructions for manuscript.

   Pages parameters: format A4, portrait orientation, document margins (cm): left — 2.5, right — 1.5, above — 2.0, below — 2.0, footer 1.3.

   Text: font —Times New Roman, font size — 14, paragraph indent — 0.5, line spacing — 1.5, justified alignment.

   The recommended manuscript size: not more than 20 pages of the specified format.

   Use only standard abbreviations. Avoid abbreviations in the title and abstract. The full term for which an abbreviation stands should precede its first use in the text unless it is a standard unit of measurement.

   All pages of the manuscript should be numbered.

   The templates for the manuscript typesetting are presented on site:

   http://www.ipiran.ru/publication/collected/template.doc
8. Articles should enclose data both in **Russian and English**:

   – title;
   – author's name and surname;
   – affiliation — organization, its address with ZIP code, city, country, and official e-mail address;
   – data on authors according to the format (see site):
     http://www.ipiran.ru/journal/collected/2012_22_02_rus/authors.asp and
     http://www.ipiran.ru/journal/collected/2012_22_02_eng/authors.asp;
   – abstract (not less than 100 words) both in Russian and in English. Abstract is a short summary of the article that can be published separately. The abstract is the main source of information on the article and it could be included in leading information systems and data bases. The abstract in English has to be an original text and should not be an exact translation of the Russian one. Good English is required. In abstracts, avoid references and formulae.
   – Indexing is performed on the basis of keywords. The use of keywords from the internationally accepted thematic Thesauri is recommended.

     Important! Keywords must not be sentences.
   – Acknowledgments.
9. References. Russian references have to be presented both in English translation and in Latin transliteration.

   Please take into account the following examples of Russian references appearance:

   Article in journal:

   Zhang, Z., and D. Zhu. 2008. Experimental research on the localized electrochemical micromachining. *Rus. J. Electrochem.* 44(8):926–930. doi:10.1134/S1023193508080077.

   Journal article in electronic format:

   Swaminathan, V., E. Lepkoswka-White, and B. P. Rao. 1999. Browsers or buyers in cyberspace? An investigation of electronic factors influencing electronic

exchange. *JCMC* 5(2). Available at: http://www.ascusc.org/jcmc/vol5/issue2/ (accessed April 28, 2011).

Conference proceedings:

Usmanov, T. S., A. A. Gusmanov, I. Z. Mullagalin, R. Ju. Muhametshina, A. N. Chervyakova, and A. V. Sveshnikov. 2007. Osobennosti proektirovaniya razrabotki mestorozhdeniy s primeneniem gidrorazryva plasta [Features of the design of field development with the use of hydraulic fracturing]. *Trudy 6-go Mezhdunarodnogo Simpoziuma "Novye resursosberegayushchie tekhnologii nedropol'zovaniya i povysheniya neftegazootdachi"* [6th Symposium (International) "New Energy Saving Subsoil Technologies and the Increasing of the Oil and Gas Impact" Proceedings]. Moscow. 267–272.

Books and other monographs:

Lindorf, L. S., and L. G. Mamikoniants, eds. 1972. *Ekspluatatsiya turbogeneratorov s neposredstvennym okhlazhdeniem* [Operation of turbine generators with direct cooling]. Moscow: Energy Publs. 352 p.

Dissertation and Thesis:

Kozhunova, O. S. 2009. Tekhnologiya razrabotki semanticheskogo slovarya informatsionnogo monitoringa [Technology of development of semantic dictionary of information monitoring system]. PhD Thesis. Moscow: IPI RAN. 23 p.

State standards and patents:

GOST 8.586.5-2005. 2007. Metodika vypolneniya izmereniy. Izmerenie raskhoda i kolichestva zhidkostey i gazov s pomoshch'yu standartnykh suzhayushchikh ustroystv [Method of measurement. Measurement of flow rate and volume of liquids and gases by means of orifice devices]. M.: Standardinform Publs. 10 p.

Bolshakov, M. V., A. V. Kulakov, A. N. Lavrenov, and M. V. Palkin. 2006. Sposob orientirovaniya po krenu letatel'nogo apparata s opticheskoy golovkoy samonavedeniya [The way to orient on the roll of aircraft with optical homing head]. Patent RF No. 2280590.

References in Latin transcription are presented in the original language.

References in the text are numbered according to the order of their first appearance; the number is placed in square brackets. All items from the reference list should be cited.

10. Manuscripts and additional materials are not returned to Authors by the Editorial Board.
11. Submissions of files by e-mail must include:
    - the journal title and author's name in the "Subject" field;
    - an article and additional materials have to be attached using the "attach" function;
    - an electronic version of the article should contain the file with the text and a separate file with figures.
12. "System and Means of Informatics" journal is not a profit publication. There are no charges for the authors as well as there are no royalties.

Editorial Board address:
FRC CSC RAS, 44, block 2, Vavilov Str., Moscow 119333, Russia
Ph.: +7 (499) 135 86 92, Fax: +7 (495) 930 45 05
e-mail: rust@ipiran.ru (to Prof. Rustem Seyful-Mulyukov)
http://www.ipiran.ru/english/journal_systems.asp

SYSTEMS AND MEANS OF INFORMATICS (SISTEMY I SREDSTVA INFORMATIKI)

SCIENTIFIC JOURNAL

Volume 25 No.1 Year 2015

Editor-in-Chief and Chair of Editorial Council
Academician I. A. Sokolov

I N T H I S I S S U E:

GRID AND CLOUD SERVICES SIMULATION AS AN IMPORTANT STEP OF THEIR DEVELOPMENT
V. V. Korenkov, A. V. Nechaevskiy, G. A. Ososkov, D. I. Pryakhina, V. V. Trofimov, and A. V. Uzhinskiy 4

COMBINING CORPUS AND THESAURUS INFORMATION FOR EXTRACTING SENTIMENT WORDS
N. V. Loukachevitch and I. I. Chetviorkin 20

MULTICRITERIA METHOD FOR DETECTING NEAR-DUPLICATES IN A STREAM OF TEXT MESSAGES
A. Andreev, D. Berezkin, I. Kozlov, and K. Simakov 34

CONTROL FLOW BASED TEST SUITE GENERATION
N. Voinov, P. Drobintsev, I. Nikiforov, V. Kotlyarov, and I. Selin 54

ANALYSIS OF UCM-MODEL COVERAGE BY TEST SCENARIOS
N. V. Voinov, P. D. Drobintsev, I. V. Nikiforov, and V. P. Kotlyarov 74

GENERALIZED TABLE-BASED LL-PARSING
S. V. Grigorev and A. K. Ragozina 89

SYNTHESIS OF STABLE LINEAR PUGACHEV FILTERS AND EXTRAPOLATORS FOR STOCHASTIC SYSTEMS WITH WIDE BAND MULTIPLICATIVE NOISES
I. N. Sinitsyn and E. R. Korepanov 108

ON CONVERGENCE OF RANDOM SUMS OF INDEPENDENT RANDOM VECTORS TO MULTIVARIATE GENERALIZED VARIANCE-GAMMA DISTRIBUTIONS
A. Yu. Korchagin 127

LARGE CAPACITY OF RAILWAY CARGO TRANSPORTATION FORECASTING
R. K. Gazizullina, M. M. Medvednikova, and V. V. Strijov 142