

Программная инженерия

Том 8
№ 10
2017
Пр
ИН

Учредитель: Издательство "НОВЫЕ ТЕХНОЛОГИИ"

Издается с сентября 2010 г.

DOI 10.17587/issn.2220-3397

ISSN 2220-3397

Редакционный совет

Садовничий В.А., акад. РАН
(председатель)
Бетелин В.Б., акад. РАН
Васильев В.Н., чл.-корр. РАН
Жижченко А.Б., акад. РАН
Макаров В.Л., акад. РАН
Панченко В.Я., акад. РАН
Стемпковский А.Л., акад. РАН
Ухлинов Л.М., д.т.н.
Федоров И.Б., акад. РАН
Четверушкин Б.Н., акад. РАН

Главный редактор

Васенин В.А., д.ф.-м.н., проф.

Редколлегия

Антонов Б.И.
Афонин С.А., к.ф.-м.н.
Бурдонов И.Б., д.ф.-м.н., проф.
Борзовс Ю., проф. (Латвия)
Гаврилов А.В., к.т.н.
Галатенко А.В., к.ф.-м.н.
Корнеев В.В., д.т.н., проф.
Костюхин К.А., к.ф.-м.н.
Махортов С.Д., д.ф.-м.н., доц.
Манцивода А.В., д.ф.-м.н., доц.
Назирова Р.Р., д.т.н., проф.
Нечаев В.В., д.т.н., проф.
Новиков Б.А., д.ф.-м.н., проф.
Павлов В.Л. (США)
Пальчунов Д.Е., д.ф.-м.н., доц.
Петренко А.К., д.ф.-м.н., проф.
Позднеев Б.М., д.т.н., проф.
Позин Б.А., д.т.н., проф.
Серебряков В.А., д.ф.-м.н., проф.
Сорокин А.В., к.т.н., доц.
Терехов А.Н., д.ф.-м.н., проф.
Филимонов Н.Б., д.т.н., проф.
Шапченко К.А., к.ф.-м.н.
Шундеев А.С., к.ф.-м.н.
Щур Л.Н., д.ф.-м.н., проф.
Язов Ю.К., д.т.н., проф.
Якобсон И., проф. (Швейцария)

Редакция

Лысенко А.В., Чугунова А.В.

Журнал издается при поддержке Отделения математических наук РАН, Отделения нанотехнологий и информационных технологий РАН, МГУ имени М.В. Ломоносова, МГТУ имени Н.Э. Баумана

СОДЕРЖАНИЕ

- Грибова В. В., Клещев А. С., Москаленко Ф. М., Тимченко В. А., Федорищев Л. А., Шалфеева Е. А.** Управляемая графовыми грамматиками разработка оболочек интеллектуальных сервисов на облачной платформе IACPaaS. 435
- Васенин В. А., Занчурин М. А., Козицын А. С., Кривчиков М. А., Шачнев Д. А.** Архитектурно-технологические аспекты разработки и сопровождения больших информационно-аналитических систем в сфере науки и образования. 448
- Димитров В. М.** Повышение производительности языка LaOQL за счет параллельной реализации запросов. 456
- Иванова К. Ф.** Интервальное допустимое решение задачи линейного программирования в системе MATLAB. 463
- Шокуров А. В., Кривчикова К. А.** Оценка абсолютной погрешности при поиске внешних параметров группы видеокамер в приложении к задаче локализации и сопровождения объекта. 470

Журнал зарегистрирован

в Федеральной службе

по надзору в сфере связи,

информационных технологий

и массовых коммуникаций.

Свидетельство о регистрации

ПИ № ФС77-38590 от 24 декабря 2009 г.

Журнал распространяется по подписке, которую можно оформить в любом почтовом отделении (индекс: по каталогу агентства "Роспечать" — 22765, по Объединенному каталогу "Пресса России" — 39795) или непосредственно в редакции.

Тел.: (499) 269-53-97. Факс: (499) 269-55-10.

Http://novtex.ru/prin/rus E-mail: prin@novtex.ru

Журнал включен в систему Российского индекса научного цитирования.

Журнал входит в Перечень научных журналов, в которых по рекомендации ВАК РФ должны быть опубликованы научные результаты диссертаций на соискание ученой степени доктора и кандидата наук.

© Издательство "Новые технологии", "Программная инженерия", 2017

SOFTWARE ENGINEERING

PROGRAMMAYA INGENERIA

Vol. 8

N 10

2017

Published since September 2010

DOI 10.17587/issn.2220-3397

ISSN 2220-3397

Editorial Council:

SADOVNICHY V. A., Dr. Sci. (Phys.-Math.), Acad. RAS (*Head*)
 BETELIN V. B., Dr. Sci. (Phys.-Math.), Acad. RAS
 VASIL'EV V. N., Dr. Sci. (Tech.), Cor.-Mem. RAS
 ZHIZHCHEKNO A. B., Dr. Sci. (Phys.-Math.), Acad. RAS
 MAKAROV V. L., Dr. Sci. (Phys.-Math.), Acad. RAS
 PANCHENKO V. YA., Dr. Sci. (Phys.-Math.), Acad. RAS
 STEMPKOVSKY A. L., Dr. Sci. (Tech.), Acad. RAS
 UKHLINOV L. M., Dr. Sci. (Tech.)
 FEDOROV I. B., Dr. Sci. (Tech.), Acad. RAS
 CHETVERUSHKIN B. N., Dr. Sci. (Phys.-Math.), Acad. RAS

Editor-in-Chief:

VASENIN V. A., Dr. Sci. (Phys.-Math.)

Editorial Board:

ANTONOV B.I.
 AFONIN S.A., Cand. Sci. (Phys.-Math)
 BURDONOV I.B., Dr. Sci. (Phys.-Math)
 BORZOV JURIS, Dr. Sci. (Comp. Sci), Latvia
 GALATENKO A.V., Cand. Sci. (Phys.-Math)
 GAVRILOV A.V., Cand. Sci. (Tech)
 JACOBSON IVAR, Dr. Sci. (Philos., Comp. Sci.), Switzerland
 KORNEEV V.V., Dr. Sci. (Tech)
 KOSTYUKHIN K.A., Cand. Sci. (Phys.-Math)
 MAKHORTOV S.D., Dr. Sci. (Phys.-Math)
 MANCIVODA A.V., Dr. Sci. (Phys.-Math)
 NAZIROV R.R., Dr. Sci. (Tech)
 NECHAEV V.V., Cand. Sci. (Tech)
 NOVIKOV B.A., Dr. Sci. (Phys.-Math)
 PAVLOV V.L., USA
 PAL'CHUNOV D.E., Dr. Sci. (Phys.-Math)
 PETRENKO A.K., Dr. Sci. (Phys.-Math)
 POZDNEEV B.M., Dr. Sci. (Tech)
 POZIN B.A., Dr. Sci. (Tech)
 SEREBR'YAKOV V.A., Dr. Sci. (Phys.-Math)
 SOROKIN A.V., Cand. Sci. (Tech)
 TEREKHOV A.N., Dr. Sci. (Phys.-Math)
 FILIMONOV N.B., Dr. Sci. (Tech)
 SHAPCHENKO K.A., Cand. Sci. (Phys.-Math)
 SHUNDEEV A.S., Cand. Sci. (Phys.-Math)
 SHCHUR L.N., Dr. Sci. (Phys.-Math)
 YAZOV Yu. K., Dr. Sci. (Tech)

Editors: LYSENKO A.V., CHUGUNOVA A.V.

CONTENTS

Gribova V. V., Kleschev A. S., Moskalenko Ph. M., Timchenko V. A., Fedorischev L. A., Shalfeeva E. A. A Graph Grammar Managed Development of Intelligent Service Shells on the IACPaaS Cloud Platform	435
Vasenin V. A., Zanchurin M. A., Kozitsyn A. S., Krivchikov M. A., Shachnev D. A. Architectural and Technological Aspects of the Development and Maintenance of Large Information Analysis Systems in the Area of Science and Education	448
Dimitrov V. M. Laconic Object Query Language Using Features of Object Model	456
Ivanova K. F. Interval Admissible Solution of a Linear Programming Problem in MATLAB System	463
Shokurov A. V., Krivchikova K. A. Absolute Error Estimation when Reconstructing the Extrinsic Parameters of a Camera Group in Application to Object Localization and Tracking	470

Information about the journal is available online at:
<http://novtex.ru/prin/eng> e-mail: prin@novtex.ru

В. В. Грибова, д-р техн. наук, зам. дир. по научной работе, e-mail: gribova@iacp.dvo.ru,
А. С. Клещев, д-р физ.-мат. наук, проф., гл. науч. сотр., e-mail: kleschev@iacp.dvo.ru,
Ф. М. Москаленко, канд. техн. наук, ст. науч. сотр., e-mail: philipmm@iacp.dvo.ru,
В. А. Тимченко, канд. техн. наук, ст. науч. сотр., e-mail: vadim@iacp.dvo.ru,
Л. А. Федорищев, канд. техн. наук, науч. сотр., e-mail: fedorischev@iacp.dvo.ru,
Е. А. Шалфеева, канд. техн. наук, доц., ст. науч. сотр., e-mail: shalf@iacp.dvo.ru,
Федеральное государственное бюджетное учреждение науки Институт автоматизации и процессов управления Дальневосточного отделения Российской академии наук, г. Владивосток

Управляемая графовыми грамматиками разработка оболочек интеллектуальных сервисов на облачной платформе IASaaS

Представлена технология разработки оболочек интеллектуальных сервисов на облачной платформе IASaaS — на основе расширенных порождающих графовых грамматик. Грамматика специфицирует информационную структуру, которая управляет процессом решения задачи. Ее расширение представляет собой орграф, который ставит в соответствие вершинам грамматики программных агентов. Технология направлена на уменьшение трудоемкости и сроков разработки и сопровождения оболочек интеллектуальных сервисов.

Ключевые слова: системы, основанные на знаниях, технология разработки систем, основанных на знаниях, программные оболочки систем, основанных на знаниях, мультиагентные системы, иерархические орграфы, агентно-ориентированное программирование, облачные сервисы

Введение

Разработка интеллектуальных систем, а конкретнее систем, основанных на знаниях (СОЗ), с помощью оболочек, в особенности специализированных (ориентированных на определенные предметные области и/или классы решаемых задач), имеет наименьшую трудоемкость по сравнению с другими способами разработки таких систем [1–10].

В работах [11–13] авторами была описана базовая (универсальная) технология разработки как специализированных оболочек СОЗ, так и прикладных СОЗ (с использованием этих оболочек) на облачной платформе IASaaS [14–16]. Оболочки СОЗ при этом предоставляются как облачные сервисы платформы. Вместе с тем продолжаются поиск и создание специализированных технологий, которые учитывая специфику задачи и/или предметной области, упрощали бы по сравнению с базовой технологией разработку и сопровождение специализированных оболочек СОЗ.

В множестве различных задач, решаемых с использованием СОЗ, может быть выделен широкий класс задач, для которых явным образом можно специфицировать *структуру* результата их решения

в форме связного орграфа, а процесс вычисления самого результата (который также представляет собой связный орграф) организовать в виде его пошагового формирования "сверху-вниз" в автоматическом, интерактивном или смешанном режиме.

Целью настоящей работы является описание технологии разработки специализированных оболочек СОЗ на платформе IASaaS на основе информационных структур, управляющих процессом вычисления результата решения задачи.

1. Общие сведения о технологии

В основу предлагаемой в работе технологии разработки специализированных оболочек СОЗ положено явное описание *информационной структуры, управляющей процессом решения задачи* (либо нескольких таких взаимосвязанных структур). Она представляет собой спецификацию структуры результата решения задачи в форме связного орграфа. Будем называть ее дальше *управляющая структура* или *орграф управляющей структуры* (когда важно ее представление). Вершинам этого орграфа ставят в соответствие программные модули (агентов), которые формируют ре-

зультат решения задачи, возможно, предоставляя для этого необходимый пользовательский интерфейс. Обращения к агентам осуществляет обеспечивающий инструментальную поддержку данной технологии *интерпретатор орграфов управляющих структур*. Он обеспечивает взаимодействие пользователя (решающего свою задачу) с управляющей структурой и является ядром решателей задач оболочек, разрабатываемых по данной технологии.

Орграф управляющей структуры, равно как и орграфы, описывающие абстрактный синтаксис языков представления различных видов знаний и данных, могут рассматриваться как порождающие *графовые грамматики* [17, 18].

Грамматика задает множество размеченных корневых иерархических орграфов с возможными петлями и циклами. Сама грамматика имеет аналогичную форму представления — орграф, обладающий такими же свойствами (орграфовое представление грамматики назовем *орграфом грамматики*). Отличие состоит в том, что орграф грамматики имеет дополнительную разметку, задающую семантику правил порождения [19—21]. Данный подход согласуется с общепринятым в настоящее время двухуровневым подходом к формированию информационных единиц [22—28]. В соответствии с ним различают два типа информационных единиц по уровню абстракции представляемой ими информации: информационные единицы, представляющие метаинформацию (абстрактный уровень), и информационные единицы, представляющие информацию (конкретный уровень). Таким образом, орграф грамматики описывает абстрактный синтаксис языка (метаинформацию) представления знаний (данных), в терминах которого формируется множество орграфов информации (представляющих базы знаний, базы данных и другие виды информации). Орграфы грамматики, в свою очередь, формируются на языке описания графовых грамматик оргграфовой связной двухуровневой модели информационных единиц [19, 20]. В ней же выражена наиболее распространенная в настоящее время объектно-ориентированная двухуровневая модель информационных единиц [25—28].

Расширение графовой грамматики представляет собой отдельную информационную единицу специального вида, которая специфицирует связи вершин орграфа грамматики с программными модулями (агентами). Сам орграф грамматики остается неизменным.

Технология разработки специализированных оболочек на основе расширенных порождающих графовых грамматик включает следующие этапы.

Этап 1. Разработка графовых грамматик (метаинформации) и управляемых ими средств формирования информационных единиц (исходных данных и знаний), необходимых решателю задач для получения результата.

Этап 2. Разработка графовой грамматики (возможно, нескольких), специфицирующей структуру результата решения задачи.

Этап 3. Разработка орграфа информации специального вида, называемого *таблицей соответствий*. Она специфицирует конкретное расширение графовой грамматики, которая описывает структуру результата решения задачи¹.

Этап 4. Разработка декларативной спецификации решателя задач, описывающей его связь с формальными параметрами (*формальный параметр* — орграф грамматики (метаинформации), *фактический параметр* — некоторый порожденный по ней орграф информации, обрабатываемый решателем задач), собственными информационными ресурсами, орграфами грамматик временных информационных единиц и пользовательским интерфейсом.

Этап 5. Разработка агентов решателя задач, обрабатывающих или формирующих фрагменты информационных единиц (возможно, только одной), представляющих результат решения задачи².

Этап 6. Разработка новых шаблонов сообщений, необходимых для взаимодействия агентов³.

Этап 7. Разработка пользовательского интерфейса.

Разработка графовых грамматик и управляемых ими средств формирования информационных единиц осуществляется так же, как в базовой технологии разработки оболочек СОЗ, имеет ту же инструментальную поддержку и выполняется по схеме, описанной в работе [11]. Далее будем описывать только те этапы технологии, которые либо имеют свою специфику по сравнению с аналогичными этапами базовой технологии (в этом случае подробно опишем только их отличительные особенности), либо вовсе отсутствуют в последней (этап 3).

2. Таблица соответствий

Таблица соответствий t_G — это орграф информации, описывающий конкретное расширение некоторого орграфа грамматики G . Она ставит в соответствие вершинам орграфа грамматики (управляющей структуры) G обращения к агентам платформы IASaaS для формирования/модификации фрагментов орграфов информации g_1, \dots, g_n ($n \geq 1$) и/или отображения пользовательского интерфейса, отличного от предоставляемого интерпретатором расширенных графовых грамматик. Орграф, описывающий грамматику (метаинформацию) таблиц соответствий, представлен на рис. 1.

¹ Интерпретатор расширенных графовых грамматик формирует (в автоматическом, интерактивном или смешанном режиме) в соответствии с этой грамматикой орграф информации. Формирование осуществляется "сверху-вниз": от вершин, представляющих сложносоставные понятия, к атомарным, начиная с корневой вершины орграфа информации. Процесс модификации орграфа информации также управляется оргграфом грамматики [21].

² При этом агентов, к которым интерпретатор расширенных графовых грамматик обращается через таблицу соответствий, разрабатывают с учетом специальных требований (они описаны в разд. 4).

³ Этап пропускается, если на этапе разработки агентов не потребовалось использование новых шаблонов сообщений.

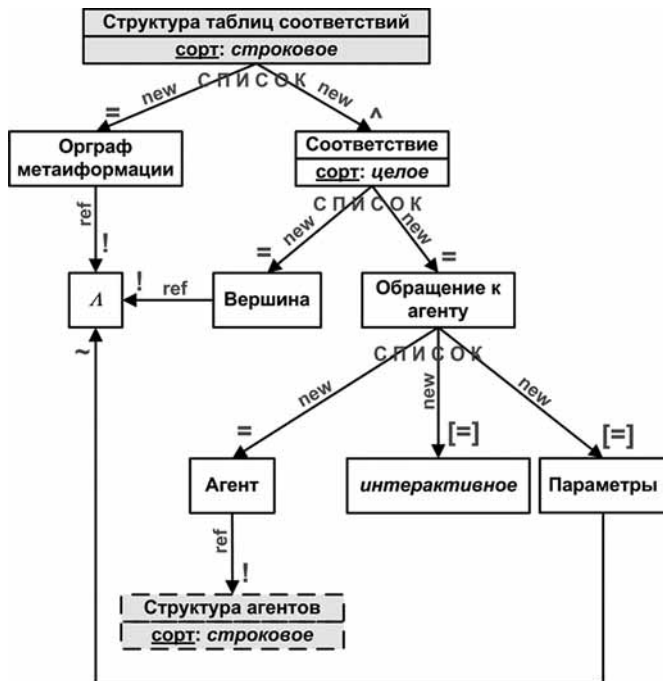


Рис. 1. Орграф грамматики таблиц соответствий

Здесь и далее нотация рисунков, изображающих орграфы грамматик и информации, совпадает с нотацией, используемой в работах [19–21]. Корневые вершины орграфов закрашены серым цветом. Названия сортов и метки терминальных вершин, представляющих константные значения некоторого сорта, даны курсивным начертанием. Спецификаторы дуг имеют следующие символьные обозначения: "копия" — "="; "возможное отсутствие" — "[=]"; "в точности один" — "!"; "непустая последовательность" — "^"; "пропуск" — "~".

Ограничители дуг имеют следующие символьные обозначения: "порождение" — "new"; "ссылка" — "ref". Если из вершины выходит только одна дуга, то тип такого набора исходящих дуг не указывается. Вершины, изображенные прямоугольниками, обведенными штриховыми линиями, принадлежат орграфам, отличным от данного орграфа, т. е. представляющим другую метаинформацию или информацию (так, на рис. 1 это корневая вершина орграфа метаинформации, описывающего структуру декларативных спецификаций агентов платформы IACPaaS). Символом A обозначен специальный тип вершин, семантика которого подробно описана в работе [20]. Здесь же необходимо отметить, что такой тип вершин в орграфах грамматики необходим для спецификации двух особых случаев (правил) порождения орграфов информации. В первом из них необходимо иметь возможность из заданной вершины орграфа информации породить произвольный подграф g' . На рис. 1 входящая в такую вершину дуга помечена спецификатором "~". Во втором случае необходимо иметь возможность из заданной вершины орграфа

информации сделать ссылку на любую вершину произвольного орграфа информации или метаинформации. На рис. 1 входящие в такую вершину дуги помечены спецификатором "!" и ограничителем "ref".

Для каждого обращения к агенту вершина "интерактивное" представляет собой логический признак. Он определяет, как в процессе формирования орграфа информации (результата решения задачи) это обращение должно выполняться: автоматически или по инициативе пользователя. Вершина "Параметры" в t_G представляет собой корневую вершину орграфа метаинформации, специфицирующего передаваемые агенту данные произвольной структуры.

Разработка таблицы соответствий выполняется по схеме, описанной в работе [11]. Ее метаинформацией является орграф грамматики "Структура таблиц соответствий".

3. Разработка решателя задач

Разработка решателя задач оболочки СОЗ выполняется по схеме, описанной в работе [11]. Особенности формирования орграфа информации, представляющего декларативную спецификацию решателя задач для данной технологии, состоят в следующем.

- Агент *Интерфейсный контроллер* решателя задается путем создания ссылки на корневую вершину орграфа информации в Фонде платформы IACPaaS, представляющего декларативную спецификацию агента *"Интерфейсный контроллер расширяемых редакторов и просмотрщиков информационных единиц"*. Данный агент реализует функциональные возможности интерпретатора расширенных графовых грамматик.

- Классы обрабатываемых орграфов информации задаются путем упорядоченного создания ссылок на соответствующие орграфы грамматик, представляющие их метаинформацию. При этом в качестве *первого выходного* формального параметра должен быть указан орграф управляющей структуры.

- Среди собственных информационных ресурсов необходимо указать (путем создания ссылки) орграф информации, представляющий таблицу соответствий, сформированную на этапе 3. В общем случае можно таким образом указать более одной таблицы соответствий. Это связано с тем, что орграф грамматики является иерархическим и может иметь ссылки на фрагменты орграфов других грамматик, которые также могут иметь расширения в виде других таблиц соответствий.

- Необходимо задать название стартовой web-страницы решателя задач на web-сайте платформы IACPaaS.

Интерфейс, формируемый агентом *"Интерфейсный контроллер расширяемых редакторов и просмотрщиков информационных единиц"*, открывает пользователю доступ ко всем функциональным возможностям по редактированию информации в структурном виде, которыми обладает *"Редактор орграфов информации"* платформы IACPaaS.

В процессе формирования орграфа информации g , представляющего результат решения задачи, агент "Интерфейсный контроллер расширяемых редакторов и просмотрщиков информационных единиц" взаимодействует с указанными в таблице соответствий t_G агентами следующим образом. При создании вершины v орграфа g , которая соответствует некоторой вершине w орграфа грамматики G , модификации/просмотру набора дуг, выходящих из вершины v , или модификации/просмотру ее значения, выполняется проверка условия следующего правила. Если в t_G из некоторой ее вершины "Вершина" выходит дуга, входящая в вершину w , то выполняется обращение (путем отправки сообщения) к указанному агенту с соответствующими параметрами, поставленное в соответствие вершине w . "Неявными" параметрами, передаваемыми в сообщениях от агента "Интерфейсный контроллер расширяемых редакторов и просмотрщиков информационных единиц" агентам, на которые сделаны ссылки из t_G , являются:

- при создании вершины v :
 - вершина v_0 орграфа информации g , являющаяся прямым предком вершины v ;
 - вершина w_0 орграфа метаинформации G , являющаяся прямым предком вершины w ;
 - дуга $w_0 \rightarrow w$ орграфа грамматики G ;
- при модификации/просмотру набора дуг, выходящих из вершины v , или модификации/просмотру значения ее поля "значение":
 - вершина v орграфа информации g ;
 - значение логического признака, указывающего на действие "модификация" или "просмотр".

Агент, указанный в t_G в данном соответствии, выполняет порождение или редактирование подграфа с корневой вершиной v и/или отображает пользовательский интерфейс. После завершения работы агента из t_G управление возвращается (также путем отправки сообщения) агенту "Интерфейсный контроллер расширяемых редакторов и просмотрщиков информационных единиц".

4. Разработка агентов

Разработка каждого агента решателя задач оболочки интеллектуального сервиса выполняется по схеме, описанной в работе [12]. Специальное требование к процессу формирования орграфов информации, представляющих декларативные спецификации агентов, непосредственно взаимодействующих с агентом "Интерфейсный контроллер расширяемых редакторов и просмотрщиков информационных единиц", состоит в следующем. В множестве блоков продукций этих агентов в обязательном порядке должны присутствовать:

- 1) блок продукций, выполнение которого инициируется сообщениями, сформированными по шаблону "Шаблон Сформировать подсеть";
- 2) блок продукций, множество шаблонов выходных сообщений которого содержит шаблон "Шаблон Результат формирования подсети", причем работа

этого блока продукций выполняется после работы предыдущего описанного блока (возможно, это один и тот же блок продукций);

3) блок продукций, выполнение которого инициируется сообщениями, сформированными по шаблону "Шаблон Обработать подсеть";

4) блок продукций, множество шаблонов выходных сообщений которого содержит шаблон "Шаблон Результат обработки подсети", причем работа этого блока продукций выполняется после работы предыдущего описанного блока (возможно, это один и тот же блок продукций).

Перечисленные в п. 1–4 шаблоны сообщений присутствуют в Фонде платформы IASPaas. В классах на языке программирования, соответствующих процедурным частям этих шаблонов сообщений, инкапсулированы доступные разработчикам агентов методы обработки (формирования, чтения и модификации) орграфа информации, представляющего содержимое сообщений, созданных по этим шаблонам. В сообщениях в обязательном порядке содержатся "неявные" параметры (см. разд. 3), а также могут содержаться параметры обращения к агенту, описанные в таблице соответствий, и данные, "пришедшие" из элементов пользовательского интерфейса.

5. Разработка пользовательского интерфейса

Разработка пользовательского интерфейса выполняется по схеме, описанной в работе [13]. Специальное требование при разработке интерфейса состоит в том, что при формировании содержимого *стартовой страницы* решателя задач необходимо поместить в текст web-страницы в указанном порядке следующее:

```

{{MaintenanceWork}}
<ui solver = "раздел/папка/название решателя
задач" action = "навигация" />
{{IweCSS}}

```

где:

- {{MaintenanceWork}} — вызов шаблона для отображения системных уведомлений;
- тег <ui/> с двумя параметрами — вызов решателя задач посредством ии-запроса с указанием полного (квалификационного) названия решателя (solver) и параметром обращения к нему (action);
- {{IweCSS}} — вызов шаблона для отображения интерфейса решателя задач с таблицей стилей CSS инструментального средства платформы "Редактор орграфов информации".

6. Пример разработки оболочки для интерактивных систем вывода пустого диалюгкта на основе метода резолюций

В качестве примера применения представленной в работе технологии опишем разработку с ее помощью оболочки интерактивных систем вывода пустого диалюгкта □ на основе метода резолюций [29].

Для простоты изложения рассмотрим метод резолюций применительно к пропозициональной логике. Данный метод позволяет дать ответ на вопрос, является ли анализируемая формула пропозициональной логики F тавтологией, путем построения на ее основе вывода пустого дизъюнкта \square . Для этого необходимо рассмотреть отрицание этой формулы $\neg F$ и показать, что $\neg F$ является противоречием (тождественно-ложной формулой). Перед построением вывода с помощью эквивалентных преобразований (исключение импликации, применение законов де Моргана и дистрибутивности и т. п.) пропозициональная формула $\neg F$ преобразуется в эквивалентную ей конъюнктивную нормальную форму (КНФ) вида $\neg F = D_1 \wedge \dots \wedge D_n$. Каждый дизъюнкт D_i ($i = 1, \dots, n$) представляет собой дизъюнкцию литералов $D_i = L_1 \vee \dots \vee L_k$. Каждый литерал L_j ($j = 1, \dots, k$) имеет вид p или $\neg p$, где p обозначает некоторую пропозициональную переменную. На основе полученного представления $\neg F$ в КНФ формируется множество дизъюнктов Ω , которое изначально имеет вид $\Omega = \{D_1, \dots, D_n\}$. Далее начинается процесс вывода пустого дизъюнкта \square , в котором на каждом шаге выполняются следующие действия.

1. В множестве дизъюнктов Ω ищется *резолювируемая пара*, т. е. пара вида $(D' \vee p, D'' \vee \neg p)$. Поскольку таких пар в Ω может быть много, то, чтобы не решать задачу перебора, выбор пары делается в интерактивном режиме.

2. К резолювируемой паре применяется правило резолюции $(D' \vee p), (D'' \vee \neg p) \Rightarrow (D' \vee D'')$. Результат применения правила — *резолювента* — есть новый дизъюнкт $D_{n+1} = D' \vee D''$.

3. Если полученная резолювента есть пустой дизъюнкт ($D_{n+1} = \square$), то вывод считается законченным, а пропозициональная формула F является тавтологией. В противном случае D_{n+1} добавляется в Ω и выполняется переход к п. 1.

Если на некотором шаге вывода в Ω невозможно добавить ни одной новой резолювенты, то формула F не является тавтологией.

Разрабатываемая оболочка состоит из специализированного редактора "*Редактор базы пропозициональных формул*", управляемого расширенной графовой грамматикой языка пропозициональной логики, и *решателя задач*, агенты которого реализуют алгоритм вывода пустого дизъюнкта, формируя при этом соответствующий граф вывода (который в терминах описываемой технологии является орграфом информации).

Орграф грамматики (метаинформации), описывающий абстрактный синтаксис языка пропозициональной логики, представлен на рис. 2. Орграф информации, описывающий грамматику языка текстового представления (конкретный синтаксис) пропозициональных формул, представлен на рис. 3.

Спецификатор дуги "непустой список" имеет символическое обозначение "*", а ограничитель дуги "порождение или ссылка" — "ref-new". Здесь и далее символ "@" в метках вершин, представленных прямоугольниками, обведенными штриховыми линиями, разделяет метку корневой вершины стороннего орграфа и метку той его вершины (не совпадающей с первой), которая является корневой вершиной повторно используемого подграфа. В квадратных скобках у вершин указаны метки соответствующих им вершин из орграфа грамматики (метаинформации). Орграф информации, представляющий таблицу соответствий для орграфа грамматики языка пропозициональной логики, показан на рис. 4.

Специализированный редактор "*Редактор базы пропозициональных формул*" создается путем подключения к инструментальному средству платформы IASPaas "*Редактор орграфов информации*" расширенной графовой грамматики языка пропозициональной логики в качестве информации, управляющей

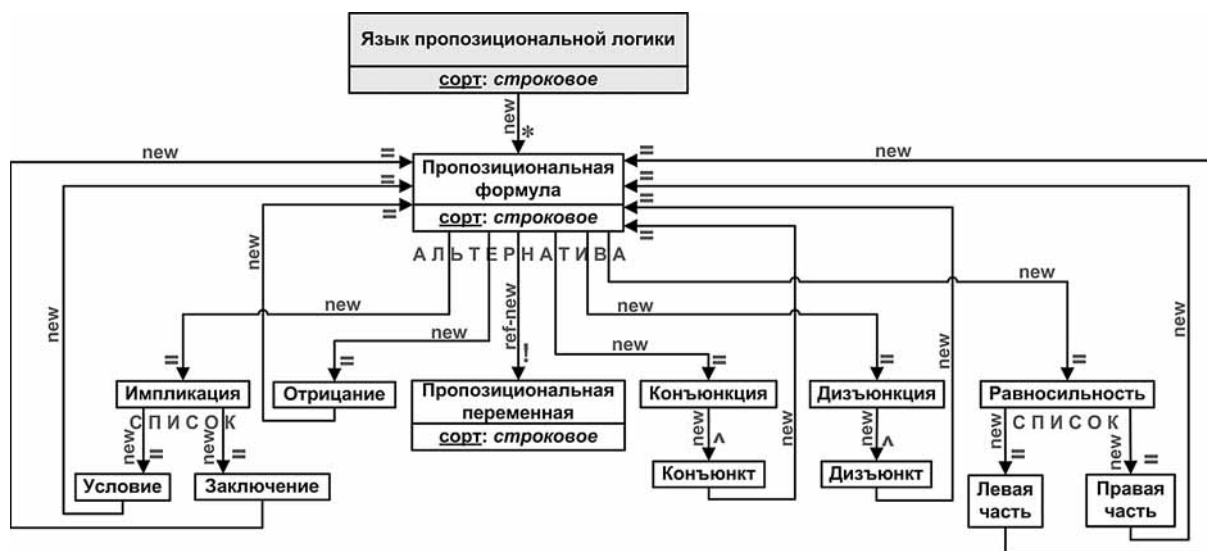


Рис. 2. Орграф грамматики языка пропозициональной логики

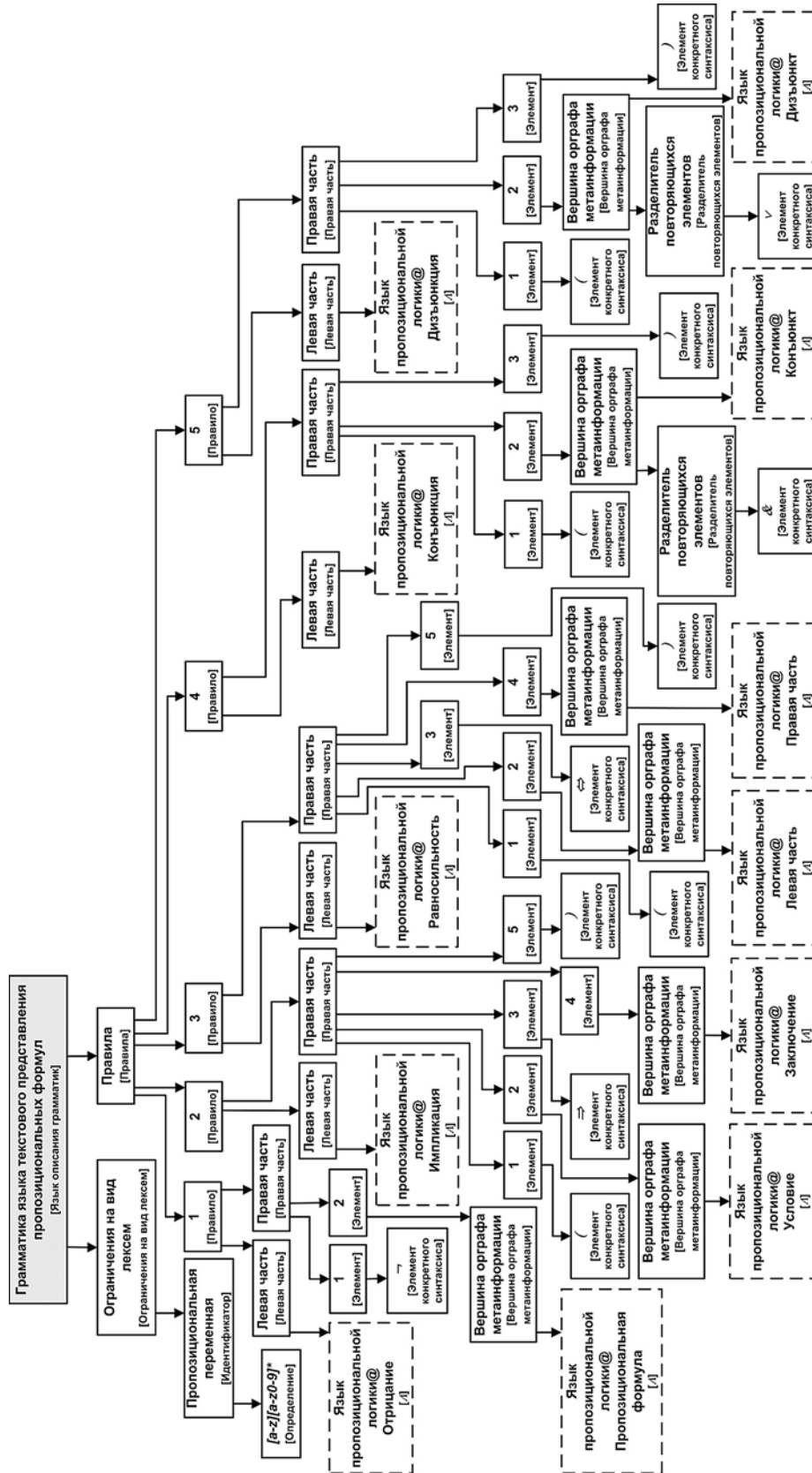


Рис. 3. Орграф информации, описывающий грамматику языка текстового представления пропозициональных формул



Рис. 4. Таблица соответствий для орграфа грамматики языка пропозициональной логики

процессом редактирования и генерацией интерфейса [11, 21].

Орграф грамматики (метаинформации), описывающий абстрактный синтаксис языка представления графов вывода пустого дизъюнкта на основе метода резолюций, показан на рис. 5. Спецификатор дуги "ноль или один" имеет символическое обозначение "[!]".

Орграф информации, представляющий таблицу соответствий для орграфа грамматики, описывающий абстрактный синтаксис языка представления графов вывода пустого дизъюнкта методом резолюций, изображен на рис. 6.

На рис. 7 показан интерфейс инструментального средства платформы "Редактор решателей задач", с помощью которого по правилам, описанным в разд. 3, сформирована декларативная спецификация решателя задач "Решатель задач оболочки интерактивных систем вывода пустого дизъюнкта на основе метода резолюций". Отметим, что в собственных информационных ресурсах указано две таблицы соответствий. Это связано с тем, что орграф грамматики, описывающий абстрактный синтаксис языка представления графов вывода пустого дизъюнкта на основе метода резолюций, имеет ссылку на подграф орграфа грамматики языка пропозициональной логики. С последним связана таблица соответствий, используемая агентом "Интерфейсный контроллер расширяемых редакторов и просмотрщиков информационных единиц" для обращения к агенту "АИСТ" (анализ и синтез текстов [21]), который формирует текстовое представление пропозициональных формул на основе соответствующей грамматики (передаваемой ему в параметрах).

Таблица соответствий для орграфа грамматики языка представления гра-



Рис. 5. Орграф грамматики языка представления графов вывода пустого дизъюнкта на основе

фов вывода пустого дизъюнкта методом резолюций используется агентом "Интерфейсный контроллер расширяемых редакторов и просмотрщиков информационных единиц" для обращения к агентам "Агент Сформировать множество дизъюнктов" и "Агент Применить правило резолюций" (путем отправки сообщений, сформированных по специальным шаблонам, указанным в разд. 4). Обращение к агенту "Агент Сформировать множество дизъюнктов" выполняется при создании вершины, соответствующей вершине "Множество дизъюнктов" из орграфа грамматики. Данный агент применяет



Рис. 6. Таблица соответствий для орграфа грамматики языка представления графов вывода пустого дизъюнкта на основе метода резолюций

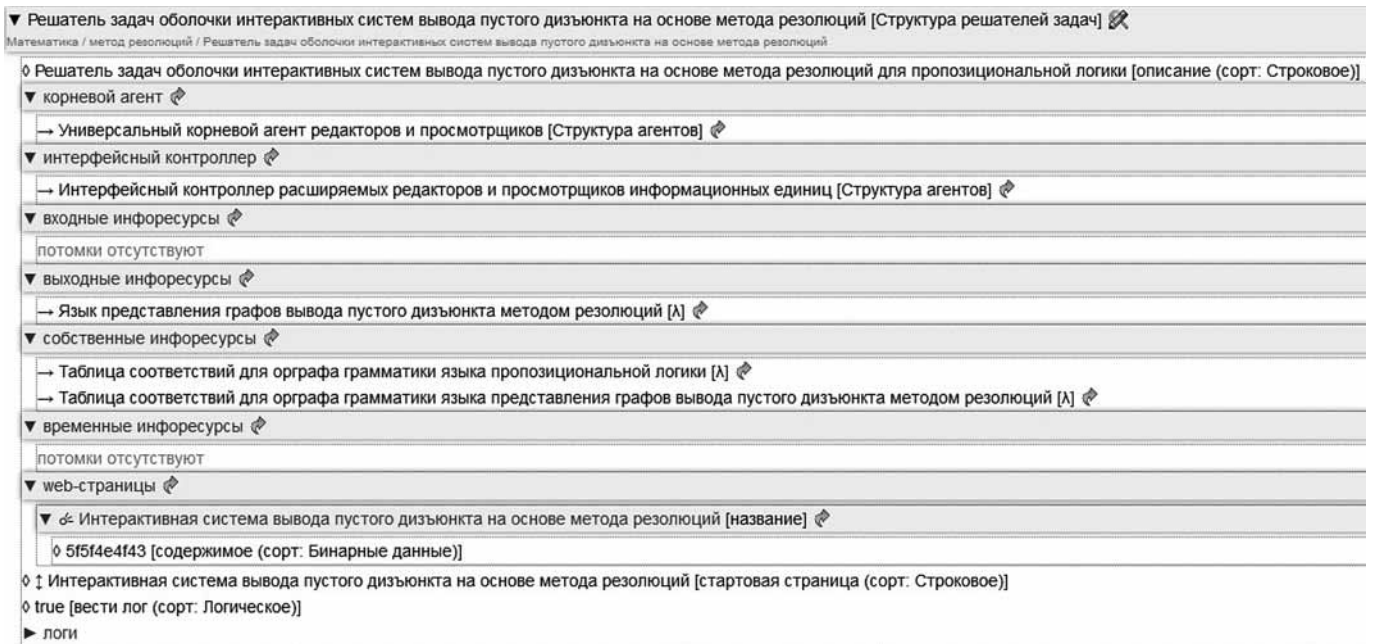


Рис. 7. Интерфейс инструментального средства "Редактор решателей задач", в котором сформирована декларативная спецификация решателя задач

к отрицанию выбранной пользователем пропозициональной формулы набор эквивалентных преобразований с целью получения ее представления в КНФ. Затем формирует начальное множество дизъюнктов, создавая в графе вывода пустого дизъюнкта вершину "Множество дизъюнктов" и делая ее прямыми потомками корневые вершины подграфов, представляющих пропозициональные формулы, являющиеся дизъюнктами в полученной КНФ. Обращение к агенту "Агент Применить правило резолюций" выполняется при создании вершины, соответствующей вершине "Резольвента" из орграфа грамматики. Данный агент применяет правило резолюции к выбранной пользователем на очередном шаге вывода резолювируемой паре пропозициональных формул (дизъюнктов). Если пара не задана, то агент отображает сообщение об этом. В противном случае он создает в графе вывода пустого дизъюнкта вершину "Резольвента". Затем, если резолювируемая пара представляет собой пару только контрарных литералов, то у вершины "Резольвента" потомков не создается, вывод считается завершенным. В противном случае агент делает прямым потомком вершины "Резольвента" и вершины "Множество дизъюнктов" корневую вершину подграфа, представляющего полученную в результате применения правила пропозициональ-

ную формулу (дизъюнкт), расширяя таким образом множество дизъюнктов.

Содержимое стартовой (и единственной) web-страницы "Интерактивная система вывода пустого дизъюнкта на основе метода резолюций" решателя задач "Решатель задач оболочки интерактивных систем вывода пустого дизъюнкта на основе метода резолюций" представлено на рис. 8.

На рис. 9 продемонстрирован пример работы сервиса "Сервис вывода пустого дизъюнкта методом резолюций в пропозициональной логике", созданного с использованием разработанной оболочки. База пропозициональных формул сформирована с помощью специализированного редактора оболочки "Редактор базы пропозициональных формул". Выходным фактическим параметром здесь является орграф информации, представляющий базу графов вывода пустого дизъюнкта методом резолюций.

Все подграфы, корневые вершины которых соответствуют вершине "Пропозициональная формула" в орграфе грамматики языка пропозициональной логики, отображаются в текстовом представлении. Однако при нажатии на кнопку-иконку "☒" у некоторой вершины, соответствующей вершине "Пропозициональная формула", отображение подграфа этой формулы заменяется с текстового представ-

ления на структурное. На рис. 10 показано структурное представление пропозициональной формулы, соответствующей третьему дизъюнкту $\neg a \wedge \neg b$ в множестве дизъюнктов. При нажатии на кнопку-иконку "☒" данная формула вновь будет отображена в текстовом представлении.

```

{{MaintenanceWork}}
<ui solver="Математика / метод резолюций / Решатель задач оболочки
интерактивных систем вывода пустого дизъюнкта на основе метода резолюций"
action="навигация"/>
{{IweCSS}}

```

Рис. 8. Содержимое стартовой web-страницы решателя задач "Решатель задач оболочки интерактивных систем вывода пустого дизъюнкта на основе метода резолюций"

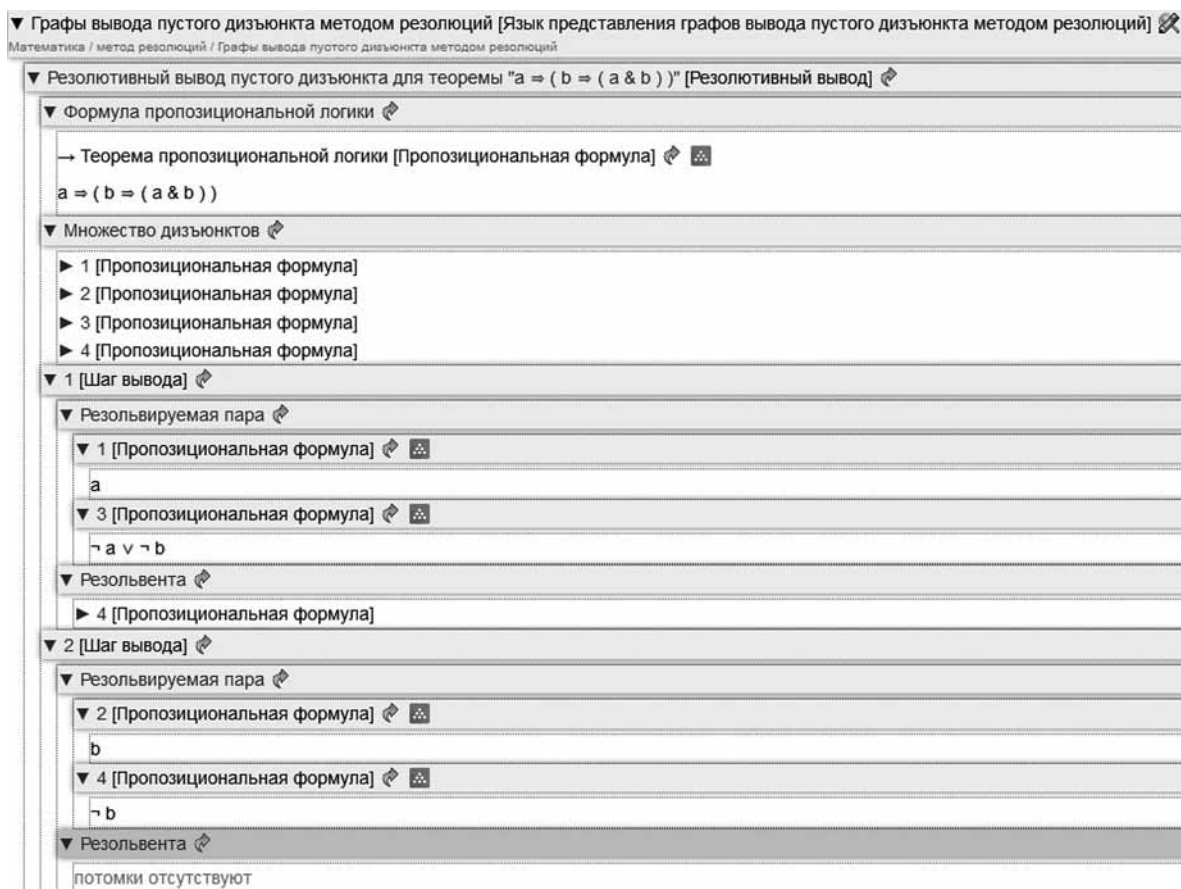
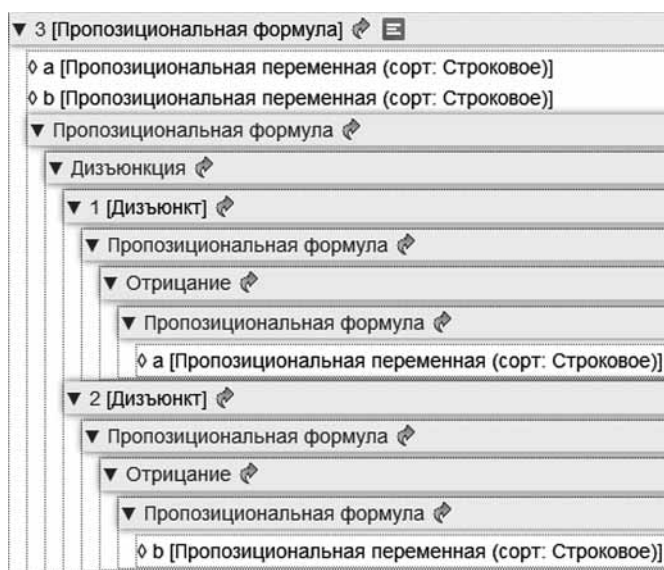


Рис. 9. Интерфейс сервиса "Сервис вывода пустого дизъюнкта методом резолюций в пропозициональной логике"



Заключение

В работе представлена специализированная технология разработки проблемно-ориентированных оболочек СОЗ (предоставляемых как сервисы) на облачной платформе IASaaS. Использование данной технологии ориентировано на класс задач, процесс решения которых можно специфицировать в виде пошагового выполнения отдельных операций. Эти операции инициируются либо пользователем, либо автоматически запускаются на выполнение интерпретатором. Результатом решения задачи является последовательно сформированная "сверху-вниз" информационная единица (возможно, несколько информационных единиц) — оргграф информации.

Принципиальное отличие управляющей структуры, положенной в основу данной технологии, от аналогичной структуры для решателей задач, разрабатываемых по базовой технологии платформы IASaaS, состоит в следующем. Первая описывается на уровне абстракции, соответствующем спецификации задачи, и ставит в соответствие шагам (этапам) ее решения ответственных за эти шаги (этапы)

агентов. При этом не требуется описывать, *как* должны выполняться обращения к этим агентам (эту задачу решает интерпретатор расширенных графовых грамматик). Вторая же требует явно специфицировать схему взаимодействия агентов друг с другом, т. е. для каждого акта взаимодействия зафиксировать его порядок и язык (что соответствует уровню взаимодействия программных компонентов).

Опыт использования представленной в работе технологии позволил значительно расширить изначально заложенные функциональные и интерфейсные возможности интерпретатора, а также область его применения. Принципиально новыми функциями, реализуемыми по данной технологии, являются перечисленные далее.

1. Генерация мультимодальных пользовательских интерфейсов редакторов баз знаний и сложноструктурированных данных. При создании баз знаний и данных для пользователей оказывается удобным представлять их фрагменты разными способами. Например, при формировании базы знаний компьютерного тренажера по офтальмологии описания объектов виртуального мира удобно представлять в структурном виде, сценариев обследования — в форме графа, возможных действий виртуального пациента — в форме таблицы [30]; при формировании базы знаний математических знаний для различных разделов математики и базы доказательств теорем удобно совмещать структурное и текстовое представление [21], при формировании базы знаний системы диагностики неисправностей автономного подводного робота — использовать структурное и табличное представление [31]. Соответствие вершинам орграфа управляющей структуры (в данном случае он описывает структуру базы знаний) агентов, реализующих соответствующие типы интерфейсов, обеспечивает указанную функциональность. В настоящее время реализованы агенты, генерирующие интерфейсы в виде текстов и таблиц.

2. Поддержка контекстно-зависимого редактирования баз знаний и данных, задание контекстных условий при их формировании. В данном случае в качестве управляющей структуры также выступает структура баз знаний или данных. Агенты, присоединенные к вершинам графа метаинформации, обеспечивают вычисление контекстных условий различных типов, формирование фрагментов орграфа по результатам вычислений.

3. Управление сервисами на основе задаче-ориентированного подхода. В этом случае в качестве управляющей структуры выступает спецификация модели задач пользователя — результат декомпозиции решаемой исходной задачи на подзадачи. В настоящее время с использованием данной технологии реализован инструментальный сервис коллективной разработки интеллектуальных сервисов по базовой технологии. Управляющей структурой является модель коллективной разработки — иерархия задач, из которых состоит разработка сервиса. Задачи автоматически генерируются по его проекту (описы-

вается в форме орграфа информации). Присоединение агентов к узлам такой управляющей структуры обеспечивает интерактивное управление процессом разработки (назначение исполнителей, сроков, создание инфраструктуры для каждого типа разработчика в зависимости от задач, синхронизация между подзадачами и т. п.).

Данная технология также упрощает процесс сопровождения СОЗ, в которых может изменяться не только база знаний, но и ее структура, определяемая онтологией (моделью представления знаний), а также функциональность решателя задач. Упрощение происходит за счет явного указания связей между программными модулями (агентами) и элементами управляющей структуры (онтологии). Удаление из управляющей структуры вершины, которой в таблице соответствий поставлено в соответствие обращение к агенту, не нарушает работу интерпретатора расширенных графовых грамматик (последний игнорирует такое соответствие). Разработчик таблицы соответствий в этом случае получает сообщение о ее неполноте. Под изменения модели представления знаний и данных автоматически адаптируются все управляемые ею редакторы баз знаний и данных, ориентированные на экспертов. Расширение функциональности решателя задач оболочки состоит в модификации существующих или добавлении новых соответствий в таблицу соответствий, а также в доработке существующих или разработке новых агентов.

С использованием предложенной технологии и инструментальных средств платформы IASPaas разработана оболочка для исследований в области верификации интуитивных математических доказательств [32]. В созданных с ее использованием сервисах ("Сервис для верификации доказательств в арифметике", "Сервис для верификации доказательств утверждений теории пределов", "Сервис для верификации доказательств утверждений о пределе как функции") заданное интуитивное доказательство верифицируется путем его расширения до полного формального доказательства. При этом выполняется пошаговое построение орграфа этого полного формального доказательства. Также ведется разработка оболочки для создания интеллектуальных систем в практической и образовательной медицине [33].

Работа выполнена при частичной финансовой поддержке РФФИ (проект 16-07-00340, проект 17-07-00299 и проект 15-07-03193).

Список литературы

1. Рыбина Г. В. Интеллектуальные системы: от А до Я. Серия монографий в трех книгах. Книга 3. Проблемно-специализированные интеллектуальные системы. Инструментальные средства построения интеллектуальных систем. М.: Научтехлитиздат, 2015. 180 с.
2. Kumar S., Prasad R. Importance of Expert System Shell in Development of Expert System // International journal of innovative research & development. 2015. Vol. 4, Issue 3. P. 128—133.

3. **Musen M.** The protégé project: a look back and a look forward // Newsletter AI Matters. 2015, Vol. 1, Issue 4. P. 4–12. doi: 10.1145/2757001.2757003.
4. **Rajendra M.** Sonar An Enterprise Intelligent System Development and Solution Framework// International journal of applied science, engineering and technology. 2007. Vol. 4, No. 1. P. 34–39.
5. **Robert J.** Schalkoff Intelligent Systems: Principles, Paradigms and Pragmatics. Sudbury, Mass. Jones and Bartlett Publishers, 2011. 758 p.
6. **Загорюлько Ю. А.** Семантическая технология разработки интеллектуальных систем, ориентированная на экспертов предметной области // Онтология проектирования. 2015. № 1 (15). С. 30–46.
7. **Загорюлько Ю. А., Боровикова О. И.** Программная оболочка для построения многоязычных тезаурусов предметных областей, ориентированная на экспертов // Труды 13-й национальной конференции по искусственному интеллекту с международным участием — КИИ-2012. Белгород: Изд-во БГТУ, 2012, Т. 4. С. 76–83.
8. **Артемьева И. Л.** Специализированные оболочки интеллектуальных систем для сложноструктурированных предметных областей // Одиннадцатая нац. конф. по искусств. интеллекту с междунар. участием, Дубна, 28 сентября — 3 октября 2008: сб. тр. в 3-х томах. М.: ЛЕНАНД, 2008, Т. 1. С. 95–103.
9. **Грибова В. В., Клещев А. С.** Технология разработки интеллектуальных сервисов, ориентированных на декларативные предметные базы знаний. Часть 1. Информационные ресурсы // Информационные технологии. 2013. № 9. С. 7–11.
10. **Грибова В. В., Клещев А. С.** Технология разработки интеллектуальных сервисов, ориентированных на декларативные предметные базы знаний. Часть 2. Решатель задач. Пользовательский интерфейс // Информационные технологии. 2013. № 10. С. 10–14.
11. **Грибова В. В., Клещев А. С., Крылов Д. А., Москаленко Ф. М., Тимченко В. А., Шалфеева Е. А.** Базовая технология разработки интеллектуальных сервисов на облачной платформе IASaaS. Часть 1. Разработка базы знаний и решателя задач // Программная инженерия. 2015. № 12. С. 3–11.
12. **Грибова В. В., Клещев А. С., Крылов Д. А., Москаленко Ф. М., Тимченко В. А., Шалфеева Е. А.** Базовая технология разработки интеллектуальных сервисов на облачной платформе IASaaS. Часть 2. Разработка агентов и шаблонов сообщений // Программная инженерия. 2016. Т. 7, № 1. С. 14–20.
13. **Грибова В. В., Клещев А. С., Крылов Д. А., Москаленко Ф. М., Тимченко В. А., Федорищев Л. А., Шалфеева Е. А.** Базовая технология разработки интеллектуальных сервисов на облачной платформе IASaaS. Часть 3. Разработка интерфейса и пример создания прикладных сервисов // Программная инженерия. 2016. Т. 7, № 3. С. 99–107.
14. **Gribova V. V., Kleshev A. S., Krylov D. A., Moskalenko Ph. M., Timchenko V. A., Shalfeyeva E. A., Goldstein M. L.** A software platform for the development of intelligent multi-agent internet-services // Proceedings of the Distributed Intelligent Systems and Technologies Workshop (DIST'2013). 1–4 July 2013. St. Petersburg, Russia. 2013. P. 29–36.
15. **Gribova V., Kleshev A., Krylov D., Moskalenko P., Timchenko V., Shalfeyeva E.** A Cloud Computing Platform for Lifecycle Support of Intelligent Multi-agent Internet-services // International Conference on Power Electronics and Energy Engineering (PEEE). Hong Kong, PEOPLES R CHINA. 19–20 Apr., 2015. USA. Lancaster: Desteck Publications, Inc., 2015. P. 231–235.
16. **Грибова В. В., Клещев А. С., Крылов Д. А., Москаленко Ф. М., Тимченко В. А., Шалфеева Е. А.** Методы и средства разработки жизнеспособных интеллектуальных сервисов // Вестник дальневосточного отделения Российской академии наук. 2016. № 4. С. 133–141.
17. **Handbook of Graph Grammars and Computing by Graph Transformation**, vol. 1: / Eds. G. Rozenberg. Foundations. World Scientific Publishing Co. Pte. Ltd., 1997. 572 p.
18. **Handbook of Graph Grammars and Computing by Graph Transformation**, vol. 2: Applications, Languages and Tools/ Eds. H. Ehrig, G. Engels, H.-J. Kreowski, G. Rozenberg. World Scientific, 1999. 720 p.
19. **Gribova V. V., Kleshev A. S., Moskalenko F. M., Timchenko V. A.** A Two-Level Model of Information Units with Complex Structure that Correspond to the Questioning Metaphor // Automatic Documentation and Mathematical Linguistics. 2015. Vol. 49, No. 5. P. 172–181.
20. **Gribova V. V., Kleshev A. S., Moskalenko F. M., Timchenko V. A.** A Model for Generation of Directed Graphs of Information by the Directed Graph of Metainformation for a Two-Level Model of Information Units with a Complex Structure // Automatic Documentation and Mathematical Linguistics. 2015. Vol. 49, No. 6. P. 221–231.
21. **Gribova V. V., Kleshev A. S., Moskalenko F. M., Timchenko V. A.** Implementation of a Model of a Metainformation-Controlled Editor of Information Units with a Complex Structure // Automatic Documentation and Mathematical Linguistics. 2016. Vol. 50, No. 1. P. 14–25.
22. **Agrawal A., Vizhanyo A., Kalmar Z., Shi F., Narayanan A., Karsai G.** Reusable Idioms and Patterns in Graph Transformation Languages // Electronic Notes in Theoretical Computer Science. 2005. Vol. 127, No. 1. P. 181–192.
23. **Klempien-Hinrichs R., Kreowski H.-J., Kuske S.** Rule-Based Transformation of Graphs and the Product Type // In Transformation of knowledge, information and data: Theory and Applications / Patrick van Bommel. Information Science Publishing, 2005. P. 29–51.
24. **Atkinson C., Kuhne T.** Model-driven development: A metamodeling foundation // IEEE Software, 2003. Vol. 20, Issue 5. P. 36–41.
25. **Kleppe A., Warmer S., Bast W.** MDA Explained. The Model Driven Architecture: Practice and Promise. Addison-Wesley, 2003. 192 p.
26. **Varro D., Pataricza A.** VPM: A visual, precise and multilevel metamodeling framework for describing mathematical domains and UML // Journal of Software and Systems Modeling. 2003. Vol. 2, Issue. 3. P. 187–210.
27. **Muller P.-A., Fleurey F., Jezequel J.-M.** Weaving Executability into Object-Oriented Metalanguages // ACM/IEEE 8th International Conference on Model Driven Engineering Languages and Systems, Montego Bay, Jamaica, 2005. P. 264–278.
28. **MDA web site.** URL: <http://www.omg.org/mda> (дата обращения 18.04.2017).
29. **Чень Ч., Ли П.** Математическая логика и автоматическое доказательство теорем. Пер. с англ. М.: Наука, 1983. 358 с.
30. **Gribova V. V., Petryaeva M. V., Fedorischev L. A.** Using Artificial Intelligence Methods and 3D Graphics for Implementation a Computer Simulator for Ophthalmology // Proc. of International Conference on Advanced Educational Technology and Information Engineering (AETIE-2015), 17–18 May, 2015, Beijing, PEOPLES R CHINA. USA. Lancaster: Desteck Publicat Inc. 2015. P. 567–573.
31. **Inzartsev A., Pavin A., Kleshev A., Gribova V., Eliseenko G.** Application of Artificial Intelligence Techniques for Fault Diagnostics of Autonomous Underwater Vehicles // Proceedings of the OCEANS 2016 MTS/IEEE Conference & Exhibition, Monterey, California, USA, 19–23 September, 2016. P. 1–6.
32. **Тимченко В. А.** Концепция системы автоматизированного конструирования доказательств теорем, выраженных средствами языка логики первого или более высокого порядка // XXXVIII Дальневосточная математическая школа-семинар им. академика Е. В. Золотова, 1–5 сентября 2014 г., Владивосток: сб. материалов. Владивосток: ИАПУ ДВО РАН, 2014. С. 445–452.
33. **Грибова В. В., Москаленко Ф. М., Окунь Д. Б., Петряева М. В.** Облачная среда для поддержки клинической медицины и образования // Врач и информационные технологии. 2016. № 1. С. 60–66.

A Graph Grammar Managed Development of Intelligent Service Shells on the IACPaaS Cloud Platform

V. V. Gribova, gribova@iacp.dvo.ru, A. S. Kleschev, kleschev@iacp.dvo.ru, Ph. M. Moskalenko, philipmm@iacp.dvo.ru, V. A. Timchenko, vadim@iacp.dvo.ru, L. A. Fedorischev, fleo1987@mail.ru, E. A. Shalfeeva, shalf@iacp.dvo.ru, Federal State Budgetary Institute of Science Institute of Automation and Control Processes, Far Eastern Branch of the Russian Academy of Sciences, Vladivostok, 690041, Russian Federation

Corresponding author:

Moskalenko Philip M., Senior Researcher, Federal State Budgetary Institute of Science Institute of Automation and Control Processes, Far Eastern Branch of the Russian Academy of Sciences, Vladivostok, 690041, Russian Federation, E-mail: philipmm@iacp.dvo.ru

Received on July 27, 2017
Accepted on August 28, 2017

The paper presents the technology for development of intelligent service shells on the IACPaaS cloud platform. The technology is based on extended generative graph grammars. Each grammar specifies an information structure that has the form of a labeled hierarchical digraph that controls the process of solution calculation. The extension of the grammar is a special kind of digraph that sets correspondence between some vertices of the grammar and software agents. The technology is intended to reduce the difficulty and time-complexity of intelligent service shells development and maintenance processes. The use of this technology is focused on a class of problems, process of solving which can be specified in the form of step-by-step execution of individual operations. These operations are initiated either by the user or automatically — by the special interpreter agent. The result of solving the problem is a sequentially formed (in "top-to-bottom" way) information resource (possibly several information resources) — a digraph of information. The technology also simplifies the maintenance process for knowledge-based systems, in which not only the knowledge base changes, but also its structure (knowledge representation model) and the functionality of the problem solver. Simplification occurs due to the explicit indication of the links between the software modules (agents) and the elements of the control structure (ontology). The modification of the functionality of the shell's problem solver comes down to removal or modification of existing correspondences or adding new ones, as well as modification of existing and creation of new agents. Using the proposed technology and tools of the IACPaaS platform, a shell for research in the field of intuitive mathematical proofs verification has been developed and shells for services in practical and educational medicine are being developed.

Keywords: knowledge-based systems, technology for development of knowledge-based systems, knowledge-based system shells, multi-agent systems, hierarchical digraphs, agent-oriented programming, cloud services

For citation:

Gribova V. V., Kleschev A. S., Moskalenko Ph. M., Timchenko D. A., Fedorischev L. A., Shalfeeva E. A. A Graph Grammar Managed Development of Intelligent Service Shells on the IACPaaS Cloud Platform, *Programmnyaya Inzheneriya*, 2017, vol. 8, no. 10, pp. 435—447.

DOI: 10.17587/prin.8.435-447

Acknowledgements: This work was partially supported by the Russian Foundation for Basic Research, project nos. 16-07-00340, 17-07-00299 and 15-07-03193.

References

1. Rybina G. V. *Intellektual'nye sistemy: ot A do Ya. Seriya monografiy v trekh knigakh. Kniga 3. Problemno-spetsializirovannyye intellektual'nye sistemy. Instrumental'nye sredstva postroyeniya intellektual'nykh sistem* (Intelligent systems: from A to Z. A series of monographs in three books. Book 3. Problem-specific intelligent systems. Instrumental tools for building intelligent systems), Moscow, Nauchtekhlitizdat, 2015, 180 p. (in Russian).
2. Kumar S., Prasad R. Importance of Expert System Shell in Development of Expert System, *International journal of innovative research & development*, 2015, vol. 4, issue 3, pp. 128—133.
3. Musen M. The protégé project: a look back and a look forward, *Newsletter AI Matters*, 2015, vol. 1, issue 4., pp. 4—12, doi: 10.1145/2757001.2757003
4. Rajendra M. S. An Enterprise Intelligent System Development and Solution Framework, *International journal of applied science, engineering and technology*, 2007, vol. 4, no. 1, pp. 34—39.
5. Robert J. S. *Intelligent Systems: Principles, Paradigms and Pragmatics*, Sudbury, Mass. Jones and Bartlett Publishers, 2011, 758 p.
6. Zagoru'ko Yu. A. Semanticheskaya tekhnologiya razrabotki intellektual'nykh sistem, oriyentirovannaya na ekspertov predmetnoy oblasti (Semantic technology for the development of intelligent

systems, focused on domain experts), *Ontologiya proyektirovaniya*, 2015, no. 1 (15), pp. 30–46 (in Russian).

7. **Zagorul'ko Yu. A., Borovikova O. I.** Programmnaya obolochka dlya postroyeniya mnogoyazychnykh tezaurusov predmetnykh oblastey, oriyentirovannaya na ekspertov (The software shell for building multilingual domain thesauri, focused on experts), *Trudy 13-y natsional'noy konferentsii po iskusstvennomu intellektu s mezhdunarodnym uchastiyem — KII-2012*, Belgorod, Izd-vo BGTU, 2012, vol. 4, pp. 76–83. (in Russian).

8. **Artem'yeva I. L.** Spetsializirovannyye obolochki intellektual'nykh sistem dlya slozhno-strukturirovannykh predmetnykh oblastey (Specialized shells of intelligent systems for complex structured domains), *Odinnadtsataya nats. konf. po iskusstv. intellektu s mezhdunar. uchastiyem*, Dubna, 28 Sept. — 3 Okt. 2008, sb. tr. v 3-kh tomakh, Moscow, LENAND, 2008, vol. 1, pp. 95–103. (in Russian).

9. **Gribova V. V., Kleshchev A. S.** Tekhnologiya razrabotki intellektual'nykh servisov, oriyentirovannykh na deklarativnye predmetnye bazy znaniy. Chast' 1. Informatsionnye resursy (Technology for the development of intelligent services focused on declarative subject knowledge bases. Part 1. Information resources), *Informatsionnye tekhnologii*, 2013, no. 9, pp. 7–11 (in Russian).

10. **Gribova V. V., Kleshchev A. S.** Tekhnologiya razrabotki intellektual'nykh servisov, oriyentirovannykh na deklarativnye predmetnye bazy znaniy. Chast' 2. Reshatel' zadach. Pol'zovatel'skiy interfeys (Technology for the development of intelligent services focused on declarative subject knowledge bases. Part 2. Problem solver. User interface), *Informatsionnye tekhnologii*, 2013, no. 10, pp. 10–14. (in Russian).

11. **Gribova V. V., Kleshchev A. S., Krylov D. A., Moskalenko F. M., Timchenko V. A., Shalfeyeva E. A.** Bazovaya tekhnologiya razrabotki intellektual'nykh servisov na oblachnoy platforme IACPaaS. Chast' 1. Razrabotka bazy znaniy i reshatelya zadach (A base technology for development of intelligent services with the use of IACPaaS cloud platform. Part 1. A development of knowledge base and problem solver), *Programmnyaya Ingeneriya*, 2015, no. 12, pp. 3–11 (in Russian).

12. **Gribova V. V., Kleshchev A. S., Krylov D. A., Moskalenko F. M., Timchenko V. A., Shalfeyeva E. A.** Bazovaya tekhnologiya razrabotki intellektual'nykh servisov na oblachnoy platforme IACPaaS. Chast' 2. Razrabotka agentov i shablonov soobshcheniy (A base technology for development of intelligent services with the use of IACPaaS cloud platform. Part 2. A development of agents and message templates), *Programmnyaya Ingeneriya*, 2016, vol. 7, no. 1, pp. 14–20 (in Russian).

13. **Gribova V. V., Kleshchev A. S., Krylov D. A., Moskalenko F. M., Timchenko V. A., Fedorischev L. A., Shalfeyeva E. A.** Bazovaya tekhnologiya razrabotki intellektual'nykh servisov na oblachnoy platforme IACPaaS. Chast' 3. Razrabotka interfeysa i primer sozdaniya prikladnykh servisov (A base technology for development of intelligent services with the use of IACPaaS cloud platform. Part 3. A development of interface and an example of applied services creation), *Programmnyaya Ingeneriya*, 2016, vol. 7, no. 3, pp. 99–107 (in Russian).

14. **Gribova V. V., Kleshchev A. S., Krylov D. A., Moskalenko Ph. M., Timchenko V. A., Shalfeyeva E. A., Goldstein M. L.** A software platform for the development of intelligent multi-agent internet-services, *Proceedings of the Distributed Intelligent Systems and Technologies Workshop (DIST'2013)*, 1–4 July 2013, St. Petersburg, Russia, 2013, pp. 29–36.

15. **Gribova V., Kleshchev A., Krylov D., Moskalenko P., Timchenko V., Shalfeyeva E. A.** Cloud Computing Platform for Lifecycle Support of Intelligent Multi-agent Internet-services, *International Conference on Power Electronics and Energy Engineering (PEEE)*, Hong Kong, PEOPLES R CHINA, 19–20 Apr., 2015, USA, Lancaster, Destech Publications, Inc., 2015, pp. 231–235.

16. **Gribova V. V., Kleshchev A. S., Krylov D. A., Moskalenko F. M., Timchenko V. A., Shalfeyeva E. A.** Metody i sredstva razrabotki zhiznesposobnykh intellektual'nykh servisov (Methods and tools for developing viable intelligent services), *Vestnik dal'nevostochnogo otdeleniya Rossiyskoy akademii nauk*, 2016, no. 4, pp. 133–141 (in Russian).

17. **Rozenberg G.**, editor. *Handbook of Graph Grammars and Computing by Graph Transformation, vol. 1: Foundations*, World Scientific Publishing Co. Pte. Ltd., 1997, 572 p.

18. **Ehrig H., Engels G., Kreowski H.-J., Rozenberg G.** (eds.). *Handbook of Graph Grammars and Computing by Graph Transformation, vol. 2: Applications, Languages and Tools*, World Scientific, 1999, 720 p.

19. **Gribova V. V., Kleshchev A. S., Moskalenko F. M., Timchenko V. A.** A Two-Level Model of Information Units with Complex Structure that Correspond to the Questioning Metaphor, *Automatic Documentation and Mathematical Linguistics*, 2015, vol. 49, no. 5, pp. 172–181.

20. **Gribova V. V., Kleshchev A. S., Moskalenko F. M., Timchenko V. A.** A Model for Generation of Directed Graphs of Information by the Directed Graph of Metainformation for a Two-Level Model of Information Units with a Complex Structure, *Automatic Documentation and Mathematical Linguistics*, 2015, vol. 49, no. 6, pp. 221–231.

21. **Gribova V. V., Kleshchev A. S., Moskalenko F. M., Timchenko V. A.** Implementation of a Model of a Metainformation-Controlled Editor of Information Units with a Complex Structure, *Automatic Documentation and Mathematical Linguistics*, 2016, vol. 50, no. 1, pp. 14–25.

22. **Agrawal A., Vizhanyo A., Kalmar Z., Shi F., Narayanan A., Karsai G.** Reusable Idioms and Patterns in Graph Transformation Languages, *Electronic Notes in Theoretical Computer Science*, 2005, vol. 127, no. 1, pp. 181–192.

23. **Klempien-Hinrichs R., Kreowski H.-J., Kuske S.** Rule-Based Transformation of Graphs and the Product Type, *In Transformation of knowledge, information and data: Theory and Applications*, Patrick van Bommel, Information Science Publishing, 2005, pp. 29–51.

24. **Atkinson C., Kuhne T.** Model-driven development: A metamodeling foundation, *IEEE Software*, 2003, vol. 20, issue 5, pp. 36–41.

25. **Kleppe A., Warmer S., Bast W.** *MDA Explained. The Model Driven Architecture: Practice and Promise*, Addison-Wesley, 2003, 192 p.

26. **Varro D., Pataricza A.** VPM: A visual, precise and multilevel metamodeling framework for describing mathematical domains and UML, *Journal of Software and Systems Modeling*, 2003, vol. 2, no. 3, pp. 187–210.

27. **Muller P.-A., Fleurey F., Jezequel J.-M.** Weaving Executability into Object-Oriented Metalanguages, *ACM/IEEE 8th International Conference on Model Driven Engineering Languages and Systems*, Montego Bay, Jamaica, 2005, pp. 264–278.

28. **MDA** web site, available at: <http://www.omg.org/mda> (access date 18.04.2017).

29. **Chen' Ch., Li R.** *Matematicheskaya logika i avtomaticheskoye dokazatel'stvo teorem. Per. s angl.* (Mathematical logic and automatic proof of theorems. Trans.), Moscow, Nauka, 1983, 358 p. (in Russian).

30. **Gribova V. V., Petryayeva M. V., Fedorischev L. A.** Using Artificial Intelligence Methods and 3D Graphics for Implementation a Computer Simulator for Ophthalmology, *Proc. of International Conference on Advanced Educational Technology and Information Engineering (AETIE-2015)*, 17–18 May, 2015, Beijing, PEOPLES R CHINA, USA, Lancaster: Destech Publicat Inc., 2015, pp. 567–573.

31. **Inzartsev A., Pavin A., Kleshchev A., Gribova V., Eliseenko G.** Application of Artificial Intelligence Techniques for Fault Diagnostics of Autonomous Underwater Vehicles, *Proceedings of the OCEANS 2016 MTS/IEEE Conference & Exhibition*, Monterey, California, USA, 19–23 September, 2016, pp. 1–6.

32. **Timchenko V. A.** Kontseptsiya sistemy avtomatizirovannogo konstruirovaniya dokazatel'stv teorem, vyrazhennykh sredstvami yazyka logiki pervogo ili boleye vysokogo poryadka (The concept of a system for automated construction of proofs of theorems expressed by means of the language of logic of the first or higher order), *XXXVIII Dal'nevostochnaya matematicheskaya shkola-seminar im. akademika Ye. V. Zolotova*, 1–5 September, 2014, Vladivostok: sb. materialov, Vladivostok, IAPU DVO RAN, 2014, pp. 445–452 (in Russian).

33. **Gribova V. V., Moskalenko F. M., Okun' D. B., Petryayeva M. V.** Oblachnaya sreda dlya podderzhki klinicheskoy meditsiny i obrazovaniya (Cloud environment for the support of clinical medicine and education), *Vrach i informatsionnye tekhnologii*, 2016, no. 1, pp. 60–66 (in Russian).

В. А. Васенин, д-р физ.-мат. наук, проф., зав. каф., **М. А. Занчурин**, мл. науч. сотр., **А. С. Козицын**, канд. физ.-мат. наук, вед. науч. сотр., **М. А. Кривчиков**, канд. физ.-мат. наук, ст. науч. сотр., e-mail: maxim.krivchikov@gmail.com, **Д. А. Шачнев**, аспирант, МГУ имени М. В. Ломоносова

Архитектурно-технологические аспекты разработки и сопровождения больших информационно-аналитических систем в сфере науки и образования

Механизмы управления научно-технической и образовательной деятельностью являются предметом особого внимания государства. С появлением технологий и средств пакетных коммуникаций, со становлением метасети Интернет и информационных технологий на ее основе, к началу XXI века сложились предпосылки создания систем, способных оперативно собирать, хранить и обрабатывать большие объемы наукометрических данных. В настоящей статье рассмотрены архитектурно-технологические аспекты разработки и сопровождения больших информационно-аналитических систем в сфере науки и образования на примере информационно-аналитической системы "ИСТИНА".

Ключевые слова: информационно-аналитические системы, наука и образование, наукометрия, инженерия программ, распределенные системы, архитектура программного обеспечения

Механизмы управления научно-технической и образовательной деятельностью являются предметом особого внимания любого государства [1]. Причина в том, что от эффективности таких механизмов напрямую зависят темпы научно-технического прогресса и, как следствие, показатели развития национальной экономики [2]. К числу внешних факторов, которые определяют развитие науки и образования, как правило, относят факторы политического, социального и экономического характера. Именно они формируют базовые предпосылки, без которых эффективное управление наукой и образованием невозможно. Не менее значимыми в вопросах такого управления являются факторы внутренние. Такими факторами являются наличие: единства и преемственности между различными стадиями образовательного процесса (школа — вуз — подготовка кадров высшей квалификации); высокого уровня развития издательского дела и других форм оперативного и широкого обмена научными идеями и результатами (семинары, конференции, симпозиумы) ученых и педагогов; регуляторов адекватной оценки и поощрения ученых, инженеров-инноваторов и педагогов.

К числу индикаторов, на основе которых реализуются последние из перечисленных факторов, относятся:

- количественные и качественные показатели "выпускаемой продукции", включая публикации и патенты, число подготовленных кадров различной квалификации и т. п.;

- объемы педагогической деятельности в виде прочитанных курсов лекций и проведенных научных семинаров;

- объемы и качество выполненных научно-исследовательских и опытно-конструкторских работ.

Эти индикаторы и построенные на их основе механизмы регулирования известны и активно использовались в XX веке. Однако отсутствие возможностей сбора и анализа таких показателей в большом объеме, а также их оперативной обработки значительно ограничивало действенность используемых регуляторов.

С появлением технологий и средств пакетных коммуникаций, со становлением метасети Интернет и информационных технологий на ее основе, к началу XXI века сложились предпосылки создания систем, способных оперативно собирать, хранить и обрабатывать большие объемы наукометрических данных. В начале 2000-х гг. появились центры индексирования и анализа библиометрических данных за рубежом — Web of Science (Thomson & Reuters), Scopus (Elsevier), Google Scholar, а также РИНЦ в России. На основе анализа коллекций таких данных стали создаваться информационно-аналитические системы для анализа эффективности (рейтинговых оценок) функционирования субъектов научно-технической деятельности. К их числу относятся Pure, SciVal (Elsevier), Converis (Thomson & Reuters), которые принято именовать Current Research Information Systems — CRIS.

Системы подобного назначения стали востребованы и в России. Многие из создаваемых для этих целей систем ориентированы на сбор и хранение, обработку и анализ данных отдельных вузов и научных центров. В качестве инструментальных средств для их создания и сопровождения использовались либо упомянутые выше зарубежные системы, либо си-

стемы, разрабатываемые в отдельных организациях под собственные, локальные потребности. В качестве проектов создания систем, которые изначально декларировали своей целью ориентацию на различные уровни управления наукой и образованием в России (от отдельных вузов и научных центров до регионального и национального), можно выделить "Карту российской науки" [3, 4] и информационно-аналитическую систему (ИАС) "ИСТИНА" (Интеллектуальная Система Тематического Исследования НАукометрических данных) [6].

Принимая во внимание отмеченную выше высокую востребованность подобных информационно-аналитических систем, их важную роль в управлении отечественной наукой и образованием, в настоящей статье рассматриваются архитектурно-технологические аспекты разработки и сопровождения больших информационно-аналитических систем в сфере науки и образования на примере ИАС "ИСТИНА". Далее применительно к ИАС "ИСТИНА" используется термин Система, что позволяет отделить ее от других упомянутых в тексте систем, в том числе — от используемых в составе ИАС "ИСТИНА" систем.

Общие принципы построения Системы

Теоретическая сторона (математические модели), а также алгоритмика и программные механизмы реализации и вопросы практической автоматизации отдельных бизнес-процессов в области наукометрии достаточно подробно рассмотрены в известных публикациях [4—13]. Однако не менее важными являются архитектурно-технологические особенности создания и сопровождения систем такого рода, а также вытекающие из них требования.

Основные требования к архитектуре Системы приведены далее.

1. Архитектура Системы должна быть модульной и масштабируемой как на макроуровне ее описания, так и на нижележащих уровнях. Такая структура направлена на адекватное отображение процессов, подлежащих автоматизации.

2. Архитектура Системы должна быть иерархически организована (структурирована) и процессно-ориентирована на каждом из уровней структурной иерархии, включая следующие блоки на отдельных уровнях:

- функционально замкнутые блоки (подсистемы) Системы, ориентированные на автоматизированную реализацию макропроцессов, реализующих функций в рамках этого блока;
- функционально обособленные компоненты (модули, приложения) в составе отдельных блоков Системы, ориентированные на реализацию соответствующих этому компоненту функций (процессов);
- компонент Системы, реализующий функции монитора безопасности для разграничения доступа к отдельным модулям Системы, к базам данных и отдельным данным в таких базах;
- компонент Системы, реализующий связанные (передача данных) и интегрирующие функции в Системе.

3. Архитектура Системы должна с достаточной полнотой отражать все автоматизируемые в рамках Системы процессы, которые востребованы практикой наукометрии, а также задачи, поставленные в настоящее время и на прогнозируемую перспективу в сфере подготовки к принятию управленческих решений в этой области.

4. Архитектура Системы должна учитывать процессы взаимодействия (обмен данными, запросы и т. п.) с базами различного рода вспомогательных данных, дополняющих, актуализирующих и конкретизирующих (уточняющих) данные, которыми располагает собственно Система, имея в виду как внутренние, так и внешние по отношению к организации базы.

При разработке, развитии и модернизации ИАС "ИСТИНА", при ее сопровождении по назначению соблюдаются перечисленные выше архитектурные принципы и вытекающие из них общие требования к Системе. Они реализуются в том объеме, который адекватен текущему уровню реализации настоящего проекта.

Далее на примере ИАС "ИСТИНА" кратко сформулируем технологические принципы создания и развития Систем, аналогичных ей по назначению и условиям использования.

1. Соблюдение положений нормативных документов РФ применительно к созданию, эксплуатации и развитию национально значимых систем, в том числе с учетом перспектив использования отечественного программного обеспечения.

2. Соблюдение основных положений и рекомендаций к инженерии программ на всех этапах жизненного цикла Системы. В настоящее время в жизненном цикле отдельных компонентов ИАС "ИСТИНА" предусмотрены и реализуются: проектные исследования; оценка эффективности предлагаемых решений; проектирование; программная реализация; тестирование и т. п. Как правило, каждому ресурсоемкому компоненту в ИАС "ИСТИНА" и Системе в целом соответствуют: модели; алгоритмы; документация в формате Единой системы программной документации (ЕСПД); код с комментариями.

3. При вторичной (по запросу) обработке данных Система должна в максимально возможной степени использовать механизмы взаимодействия с базами данных и содержащимися в них данными (в том числе с точки зрения их конфиденциальности), которые необходимы для выполнения запроса.

Это требование позволит активно использовать данные как "де-факто" существующих баз данных, так и вновь разрабатываемых без ущерба для конфиденциальности части данных в этих базах. Здесь следует отметить то обстоятельство, что существующие и используемые на практике модели и программные средства обеспечения информационной безопасности (в первую очередь, разграничения доступа к ресурсам) таких сложно организованных систем не могут удовлетворить требованиям, которые следуют из положений этого принципа. Однако определенные результаты исследований и практической реализации на этом направлении из-

вестны, в частности, модель, описанная в работе [4], разработанная непосредственно на основе опыта реализации компонентов разграничения доступа в Системе.

4. Механизмы (математическое, алгоритмическое и программное обеспечение) должны учитывать различные уровни конфиденциальности данных, которыми располагает Система, и те базы данных, к которым она может обращаться.

Механизмы логического разграничения доступа к данным в ИАС "ИСТИНА" опираются на формальные модели их описания, которые отличаются от традиционно принятых в "классической" информационной безопасности, аккумулируют мировой опыт создания такого сорта механизмов в социальных сетях и учитывают особенности наукометрии как проблемной области.

5. Система должна поддерживать интеграционные механизмы, позволяющие извлекать необходимые данные по запросу пользователей из других, в том числе, удаленных в сети Интернет баз данных (БД), с их защитой от несанкционированного доступа и верификацией, с установлением их соответствия ("привязкой" или "аффилиацией") к отдельным персонам (далее — персоналиям) в БД Системы.

В настоящее время с той или иной степенью завершенности в ИАС "ИСТИНА" реализованы следующие механизмы интеграции данных:

- механизмы обмена информацией по телекоммуникационным каналам между пользователями Системы и базами данных (БД внутренними и внешними по отношению к Системе) с использованием единой модели логического разграничения доступа к ресурсам Системы;

- механизмы ввода данных в Систему в режимах "снизу вверх" (от конечного пользователя) и "сверху вниз" (из ранее сформированных источников) с их первичной обработкой, верификацией, "привязкой" к персоналиям и размещением в БД Системы под контролем модели логического разграничения доступа;

- механизмы оперативного вывода результатов вторичной обработки и агрегирования данных по запросам пользователей Системы (включая отдельных персоналий, ответственных лиц от структурных подразделений и ректората, отвечающих за сопровождение данных в Системе) под контролем модели логического разграничения доступа.

Архитектура системы

В настоящем разделе представлены основные положения архитектуры ИАС "ИСТИНА", которые реализуют на практике требования и принципы, изложенные в предыдущем разделе.

Архитектура Системы отражает онтологическую модель наукометрии в ее представлении как предметной области, на которую Система ориентирована. Основные понятия (сущности, объекты), которые используются в приложениях наукометрии и составляют ее тезаурус, систематизированы, едины для всех

приложений Системы и хранятся в ее БД. Отношения (связи) между этими объектами, составляющие таксономию нижнего уровня наукометрии, реализуются реляционными механизмами базы данных. Приложения как объекты верхнего уровня таксономии наукометрии (проблемной области) взаимодействуют между собой по заранее принятым правилам, которые реализуются с помощью механизмов шаблона проектирования "модель—представление—поведение" (MVC) и механизмов объектно-реляционного отображения (ORM). При этом используются единообразные объекты и отношения между ними. Таким образом, естественно реализуются механизмы интеграции используемых в Системе данных и приложений.

Интегрирующие механизмы Системы реализованы на основе реляционной БД, структура которой отражает основные аспекты онтологии, а также взаимодействующих с ней приложений. Понятия, используемые Системой, представлены в виде отношений реляционной БД. В коде Системы используются средства объектно-реляционного отображения, которые представляют понятия системы в виде набора объектов и связей между ними.

Основу Системы составляет ядро — набор программных модулей и подсистем, реализующих базовые (системообразующие) функциональные возможности, которые напрямую (во многом) определяют и качество (показатели качества) Системы в целом. Ядро содержит базовые классы, методы, формы и функции, используемые в других модулях. Оно обеспечивает единый интерфейс и базовый каркас добавления, просмотра, редактирования и удаления результатов научно-педагогической деятельности работников. В ядре реализованы механизмы поиска похожих объектов, которые используются для подбора, в частности, похожих работников, журналов и статей. В ядре реализуются также и другие общие функциональные возможности, например, поддерживающие политику безопасности Системы, в частности, механизмы разграничения доступа к различным категориям данных Системы. С позиций описания модели предметной области, ядро Системы содержит базовые классы, определяющие следующие понятия: результат научной или педагогической деятельности (результат); авторство результата или другая связь, описывающая отношение работника к результату (например, "официальный оппонент диссертации").

На следующем уровне архитектуры расположены два базовых приложения: "организации" и "работники", а также подсистема логического разграничения доступа. Модель данных приложения "организация" описывает административную структуру как организации в целом, так и ее структурных подразделений. В состав этого приложения входит также механизм реализации действий должностных лиц, которым делегирована роль ответственных за сопровождение информации в Системе от организации и отдельных ее структурных подразделений. Приложение "работники" описывает понятия, связанные с работниками, включая следующие: профессиональный профиль работника в Системе; аффилиация работника

с организацией и ее структурным подразделением (место работы и должность); сведения об ученой степени и ученом звании работника. Приложения "организации" и "работники" являются базовыми приложениями потому, что с ними связаны все остальные приложения Системы, которые находятся выше по иерархии. Подсистема логического разграничения доступа реализует реляционную модель логического разграничения доступа к объектам системы на основе уже имеющихся в системе отношений.

Основным уровнем архитектуры являются приложения класса "результаты деятельности". Каждое приложение описывает структуру понятий, связанных с отдельным типом результатов научно-инновационной и педагогической деятельности работников (например, публикации), или общую сущность, связанную с таким типом (например, журналы). Эти приложения опираются на общий каркас базовых понятий, классов и функций Системы, которые представлены в ее ядре. На следующем уровне иерархии архитектуры Системы расположены подсистемы, отвечающие за анализ данных, накопленных на предыдущих уровнях, и за их интеграцию с данными, полученными из внешних систем. К таким подсистемам относятся: подсистема верификации данных; подсистема расчета персонального рейтинга; подсистема подготовки отчетных материалов. Подсистема верификации включает механизмы получения и обработки показателей цитирования отдельных статей из Web of Science, из Scopus, а также поиск статей в этих системах. На этом же уровне иерархии расположены подсистемы, реализующие бизнес-процессы по тем или иным аспектам деятельности организации, включая: автоматизацию подготовки и проведения конкурсных процедур; online-подачу документов и конкурсного избрания на научные и профессорско-преподавательские должности; сопровождение деятельности диссертационных советов; учет эффективности использования учебно-научного оборудования.

Приложения и подсистемы интегрированы с ядром Системы и между собой на логическом и программном уровнях путем использования общего тезауруса, описывающего понятия предметной области. Ранее в настоящем разделе было продемонстрировано разделение понятий по отдельным приложениям. В случае, если в одной из подсистем требуется использовать понятия, определяемые в другой подсистеме (например, подсистема расчета персонального рейтинга опирается на определения результатов деятельности), классы, которые описывают эти понятия, импортируются в данную подсистему из приложений, соответствующих таким понятиям. В логической структуре БД такие связи, как правило, реализуются отношениями "многие к одному" (например, несколько статей могут быть опубликованы в одном журнале) и отношениями "многие ко многим" (например, авторство статьи описано отношением "многие ко многим", которое определяет аффилиацию работника — автора с результатом научной деятельности — статьей). На уровне основного кода Системы механизмы объектно-реляционного отображения транслируют такие отношения в связи,

доступные разработчику непосредственно из кода. Приведенный выше пример аффилиации автора со статьей порождает в коде такие связи, как "публикации работника" (а также "статьи работника в журналах", "статьи работника в сборниках" и "тезисы докладов автора", представляющие собой подмножества публикаций работников), "авторы статьи", "записи об авторстве статьи" (дополнительно хранят информацию о порядке упоминания авторов в библиографической ссылке и именах авторов в том виде, в котором они приведены в библиографической ссылке). Механизмы интеграции Системы с внешними источниками данных (такими как библиографические базы данных Web of Science и Scopus) представлены в модели данных в виде отображения между внутренними идентификаторами объектов в Системе и внешними идентификаторами в источниках данных, которые соответствуют тому или иному объекту.

Одной из естественных моделей представления объектов в базе данных Системы является графовая модель. В этой модели различные объекты представляются вершинами графа, а связи между объектами различных типов — ребрами графа с указанием типа. Например, авторы статьи и сама статья связаны отношением "авторство", а статья связана с журналом или сборником отношением "опубликована в". Часто одни и те же два класса могут иметь связи разных типов, например, работник может быть докладчиком на конференции или членом программного комитета этой конференции. Необходимость использовать графовую модель возникает в различных приложениях и модулях Системы, в частности, при вводе данных, при верификации результатов, при расчете персонального рейтинга и при составлении различных отчетов. Другим примером графа, возникающего в Системе, является граф ключевых слов и разделов науки, с помощью которого можно определить семантическую близость двух понятий, классифицировать их по отношению к тому или иному разделу науки, а также осуществить поиск по ключевым словам.

Для удобства работы с графовыми моделями в рамках Системы ведется разработка редактора онтологий. Данный редактор позволяет пользователям совместно работать над пополнением графов большого размера, он основан на стандартах Консорциума Всемирной паутины, что дает возможность соединить графы, редактируемые в редакторе, с графами, уже разработанными вне ИАС "ИСТИНА". Редактор онтологий позволяет каждому пользователю создать несколько собственных рабочих пространств ("страниц онтологии"), создавать и редактировать вершины графа внутри этих страниц и ребра, их соединяющие, и "склеивать" эти страницы между собой. Редактор имеет механизм версионного контроля, что позволяет просматривать изменения, которые вносятся пользователями, и узнавать, кто из них создал тот или иной участок графа.

Отметим, что большинство функций этого модуля пока находятся в стадии исследовательской разработки. Однако эти исследования рассматриваются разработчиками Системы как одно из наиболее перспективных направлений развития систем такого рода.

Технологические аспекты разработки

Ранее в числе основных принципов создания и развития Системы было упомянуто соблюдение основных положений и рекомендаций к инженерии программ на всех этапах ее жизненного цикла. В настоящем разделе описаны отдельные технологические аспекты процессов разработки Системы, которые играют важную роль в реализации этого принципа и в то же время в той или иной степени отражают специфику крупных информационно-аналитических систем.

Управление исходным кодом. В процессах разработки и модернизации ИАС "ИСТИНА" используется система GitLab для управления репозиторием исходного кода. Эта система предоставляет веб-интерфейс к набору репозитория исходного кода ИАС "ИСТИНА". Из числа функциональных возможностей системы GitLab в рамках технологических процессов разработки ИАС "ИСТИНА" следует выделить встроенные средства рецензирования кода. Аналогично популярному проприетарному аналогу — системе GitHub, в GitLab выделено понятие "запросов на слияние" — наборов изменений исходного кода с текстовым обоснованием таких изменений. Содержание каждого такого запроса разрабатывается в отдельной ветке системы контроля версий, независимо от других модификаций кода Системы. После завершения работы над новыми функциональными возможностями или исправлением недостатков Системы автор запроса на слияние передает его на рецензию кому-либо из других членов коллектива, которые не вносили изменений в эту ветку кода, но обладают знаниями об особенностях модифицируемых фрагментов кода. Напрямую (без запросов на слияние и рецензирование кода) вносить изменения в основную ветку запрещено техническими средствами. Такая схема работы имеет следующие достоинства: все члены коллектива получают возможность ознакомиться с новыми изменениями кода Системы; повышается вероятность выявления опечаток и тривиальных ошибок в коде.

В рамках совершенствования технологических процессов разработки в 2017 г. предполагается внедрить практики непрерывной интеграции. Для каждого запроса на слияние (т. е. каждого набора изменений, реализующих те или иные новые функциональные возможности или исправляющих недостатки Системы) средствами системы GitLab будет выполняться набор автоматических тестов, включающий первичное "дымовое" тестирование (*smoke testing*). Такое тестирование заключается в запуске основных компонентов Системы и загрузке главной страницы интерфейса пользователя. Оно предназначено для выявления ошибок в конфигурации Системы, например, для случая, когда разработчик использовал в коде новую программную библиотеку, но при этом не включил ее в список зависимостей. Набор автоматических тестов должен также включать статический анализ кода с использованием средства *pyflakes*, которое позволит выявить потенциальные опечатки, синтаксические ошибки, ошибки в именовании переменных и в импорте библиотечных функций, также он должен включать в себя юнит-тесты и интеграционные тесты.

Перспективные направления для улучшения контроля в этой области включают использование гибридного полигона в сценариях автоматического тестирования, а также проверку на уровень покрытия основного кода Системы тестами.

Обработка обращений пользователей и сообщений об ошибках. Информационно-аналитическая система "ИСТИНА" используется крупнейшей научно-образовательной организацией России — Московским государственным университетом им. М. В. Ломоносова. В настоящее время в Системе работает 15 организаций и более 25 000 пользователей. Учитывая ограниченный размер коллектива разработчиков, традиционные способы взаимодействия с пользователями, такие как личное общение и связь с членами коллектива по электронной почте или по телефону, показали себя неэффективными. В связи с этим было принято решение о развертывании системы обработки обращений пользователей на основе системы Redmine. На каждом экране пользовательского интерфейса ИАС "ИСТИНА" в нижней части находится ссылка "Создать обращение в службу поддержки". При нажатии на эту ссылку открывается диалоговое окно, в котором пользователь может кратко описать суть вопроса, который у него возник, и указать предметную категорию обращения. После отправки обращения пользователем сервер приложений ИАС "ИСТИНА" с использованием интерфейса разработчика системы Remine создает новое обращение в этой системе, в которое, кроме пользовательского текста, включается также дополнительная информация. В состав такой дополнительной информации входит имя и место работы пользователя, адрес страницы, на которой было создано обращение. Настройки и расширения системы Redmine позволяют отправить пользователю email-сообщение с уведомлением о том, что обращение принято. С помощью ответа на это сообщение пользователь может предоставить дополнительную информацию по обращению и прикрепить к нему файлы с дополнительной информацией.

Система Redmine используется также для планирования новых задач по модернизации и сопровождению Системы. Ведутся работы по включению в систему обработки обращений лиц, ответственных за сопровождение информации в ИАС "ИСТИНА" в организациях, использующих Систему. Таким образом, ответственные за сопровождение информации смогут взять на себя первичную фильтрацию обращений и отвечать на ряд обращений организационно-административного характера, а также обрабатывать обращения, которые находятся в области их компетенции.

Для обработки ошибок ИАС "ИСТИНА" использует интегрированную версию компонента непрерывного мониторинга ошибок Sentry. В случае, если в работе Системы возникла ошибка (исключение в терминах языка Python), в журнал непрерывного мониторинга записывается сообщение об ошибке с сохранением параметров HTTP-запроса, который вызвал ошибку, и информацией об исключении, включая трассировку стека и состояние переменных. Кроме того, записи в компоненте Sentry группируются по имени функции, в которой произошла ошибка, что позволяет упростить анализ. На прак-

тике компонент непрерывного мониторинга ошибок позволяет разработчикам оперативно реагировать на возникшие регрессии в коде Системы.

Комплект документации Системы. Одним из аспектов, который часто обходит вниманием при разработке крупных, быстро изменяющихся программных систем, является создание и поддержание в актуальном состоянии комплекта документации по системе. Ранее комплект документации ИАС "ИСТИНА" составлялся с использованием средств LaTeX (для отчетов о научно-исследовательской работе) и редакторов, совместимых с Microsoft Word (документы ЕСПД). Тот факт, что документы Microsoft Word хранятся в репозитории исходного кода "непрозрачно", без возможности просмотра изменений, способствовал рассинхронизации документации с фактическим состоянием Системы. В начале 2017 г. процесс подготовки документов в формате ЕСПД, которые для ИАС "ИСТИНА" включают руководство пользователя, описание программы и руководство системного программиста, был переведен на средство генерации документации Sphinx.

Средство Sphinx принимает на вход легковесный текстовый язык разметки reStructuredText и генерирует по нему комплект документации в различных форматах. Это средство используется рядом крупных программных продуктов с открытым исходным кодом, включая язык Python, платформу Django и ядро операционной системы Linux.

Для руководства пользователя основным преимуществом Sphinx на практике оказалась возможность генерации документов в формате PDF и в виде портала документации на базе статических HTML-страниц с поиском на стороне клиента с использованием кода на языке JavaScript. Руководство пользователя в форме такого портала в настоящее время опубликовано на сайте <http://docs.istina.msu.ru/> и обновляется в полуавтоматическом режиме из той же ревизии, что и основной код Системы. В сочетании с организационными мерами (проверка на предмет обновления документации при рецензировании запросов на слияние) это позволяет поддерживать руководство пользователя в актуальном состоянии. Версия документации в формате PDF использовалась для подготовки комплекта документации ЕСПД, кроме того, одна из версий руководства пользователя была издана ограниченным тиражом в печатном виде. Актуальная PDF-версия доступна на сайте руководства пользователя.

При формировании руководства системного программиста используется возможность генератора Sphinx подгрузки документации непосредственно из кода Системы на языке Python. Такая документация записывается в так называемых строках документации (*docstring*) — комментариях специального вида в коде на языке Python. Формат строк документации представляет собой расширение языка reStructuredText. Более подробно он описан в документе PEP 287, входящем в набор официальных расширений спецификации языка Python. Средство Sphinx предоставляет дополнительные расширения языка разметки, включая директивы `automodule`, `autoclass`, `autofunction` и прочие, которые позволяют интегрировать строки документа-

ции отдельных модулей, классов и функций с общим описанием структуры кода модулей Системы в рамках руководства системного программиста.

Опыт перехода на новую версию платформы. Стандартной практикой для крупных современных программных систем является использование платформ (каркасов, *frameworks*), которые поддерживаются внешними разработчиками. Как правило, такие платформы представляют собой проекты с открытым исходным кодом, которые поддерживаются силами одной или нескольких крупных компаний. Примерами таких платформ с открытым исходным кодом для веб-ориентированных систем, написанных на различных языках программирования, служат ASP.NET Core MVC (язык программирования C#), Spring Framework (Java), Rails (Ruby) и, в случае ИАС "ИСТИНА", Django (Python).

Интерфейс разработчика (API) платформы в той или иной степени используется в значительной части исходного кода системы. Внешний характер развития и поддержки платформ при этом проявляется в том, что при выходе новой версии платформы поведение некоторых функций платформы меняется. Поддержка старой версии, которая включает в себя исправление ошибок, в том числе и потенциально влияющих на безопасность системы, как правило, завершается в достаточно короткие сроки. В таких случаях требуется внесение изменений в код системы для перехода на новую версию платформы. Для языков программирования, не имеющих статической типизации, переход осложняется отсутствием обратной связи от компилятора, которая позволяет оперативно обнаружить фрагменты кода, которые требуют изменений.

Весной 2017 г. коллектив разработчиков ИАС "ИСТИНА" выполнил переход с неподдерживаемой в настоящее время версии платформы Django 1.4 на версию Django 1.8, которая находится в состоянии долгосрочной поддержки (LTS). Учитывая непрерывный характер и высокие темпы внесения изменений в код ИАС "ИСТИНА", переход необходимо было осуществлять в основной ветке (иначе версии кода Системы могли бы разойтись) с сохранением совместимости кода с используемой на тот момент в режиме эксплуатации (*production*) версией Django 1.4. Переход был осуществлен путем поэтапного добавления поддержки для последующих версий платформы (1.5, 1.6, 1.7 и 1.8). После перевода эксплуатационного сервера на Django 1.8 поддержка версий 1.7 и более ранних была исключена.

Когда значительная часть изменений для поддержки Django 1.8 была внесена, был объявлен трехнедельный период бета-тестирования, в рамках которого большинство разработчиков перешли на версию Django 1.8 локально, кроме того, на новую версию был переведен тестовый сервер. На основе журнала HTTP-запросов Системы наиболее часто используемые запросы с эксплуатационного сервера были воспроизведены на тестовом сервере. Этап бета-тестирования позволил выявить и исправить значительную часть ошибок. На заключительном этапе на эксплуатационный сервер параллельно с основным окружением Django 1.4 было добавлено окружение для новой версии и эксплуатационный сервер был переключен на

Django 1.8. На этом этапе был выявлен еще ряд ошибок, которые были исправлены в течение 1–2 дней.

При разработке платформы Django используется двухэтапный период вывода устаревших компонентов интерфейса программиста из эксплуатации. Если компонент отмечен как устаревший в некоторой версии, его поддержка прекращается не ранее чем через две версии платформы. Например, компонент может быть отмечен устаревшим в версии 1.5, тогда он будет удален в версии 1.7.

Значительная часть ошибок на заключительном этапе перехода была выявлена с помощью упомянутой ранее системы Sentry. Данные трассировки стека в этих случаях, как правило, позволяли исправить эти ошибки оперативно и с минимальными объемами изменений. Некоторые ошибки определить было сложнее. В частности, в силу измененной обработки порядка отображения полей ввода для форм, на платформе Django версии 1.8 в редких случаях появлялись нежелательные поля. Ошибки такого рода могли быть обнаружены только путем сравнения результатов обработки Системой одного и того же запроса на платформе Django версии 1.4 и на платформе Django версии 1.8.

Обновление используемой платформы и внешних зависимостей не является однократным мероприятием. В частности, используемая в настоящее время версия платформы будет поддерживаться только до апреля 2018 г. Нужно отметить, что переход на более новую версию платформы следует рассматривать как сокращение технического долга Системы. Ожидается, что последующие обновления будут проходить с меньшими трудозатратами. В ближайшей перспективе предполагается внести в код изменения, устраняющие все случаи использования функций, объявленных устаревшими в платформе Django версий 1.7 и 1.8, а также обновить внешние зависимости до их последних версий. После этого до апреля 2018 г. будет выполнен постепенный переход на платформу Django версии 1.11. Следует отметить, что, начиная с версии 2.0, которая следует за версией 1.11, разработчики платформы Django перешли на новый цикл разработки, который упрощает процесс

перехода для проектов, которые ориентируются на версии платформы с долгосрочной поддержкой.

Список литературы

1. **Налимов В. В., Мульченко З. М.** Наукометрия. Изучение развития науки как информационного процесса. Физико-математическая библиотека инженера. М.: Наука, 1969. 192 с.
2. **Васенин В. А.** Модернизация экономики и новые аспекты инженерии программ // Программная инженерия. 2012. № 2. С. 2–17.
3. **Михайленко И. В.** Информационно-аналитическая система "Карта Российской науки" как инструмент научного мониторинга // Динамика систем, механизмов и машин. 2014. № 4. С. 75–78.
4. **Гончаров М. В., Михайленко И. В.** Наукометрические показатели, используемые в ИАС "Карта Российской науки". Методика расчета // Научные и технические библиотеки. 2016. № 12. С. 37–43.
5. **Васенин В. А., Голомазов Д. Д., Ганкин Г. М.** Архитектура, методы и средства базовой составляющей системы управления научной информацией "ИСТИНА — Наука МГУ" // Программная инженерия. 2014. № 9. С. 3–12.
6. **Васенин В. А., Афонин С. А., Козицын А. С., Голомазов Д. Д.** Система "ИСТИНА" для подготовки принятия решений на основе анализа наукометрической информации // Научный сервис в сети Интернет: Труды XVII Всероссийской научной конференции. ИПМ им. М. В. Келдыша, Москва, 2015. С. 51–62.
7. **Васенин В. А., Иткес А. А., Бухонов В. Ю., Галатенко А. В.** Модели логического разграничения доступа в многопользовательских системах управления наукометрическим контентом // Программная инженерия. 2016. Т. 7, № 12. С. 547–558.
8. **Васенин В. А., Зензинов А. А., Лунев К. В.** Использование наукометрических информационно-аналитических систем для автоматизации проведения конкурсных процедур на примере информационно-аналитической системы "ИСТИНА" // Программная инженерия. 2016. Т. 7, № 10. С. 472–480.
9. **Cobo M. J., López-Herrera A. G., Herrera-Viedma E., Herrera F.** Science mapping software tools: Review, analysis, and cooperative study among tools // Journal of the Association for Information Science and Technology. 2011. Vol. 62, N. 7. P. 1382–1402.
10. **Grivel L., Polanco X., Kaplan A.** A computer system for big scientometrics at the age of the World Wide Web // Scientometrics. 1997. Vol. 40. N. 3. P. 493–506.
11. **Johansson A., Ottosson M. O.** A national current research information system for Sweden // 11th International Conference on Current Research Information Systems — CRIS 2012: Prague, Czech Republic, June 6–9, 2012. Agentura Action M, 2012. P. 67–71.
12. **Clements A., McCutcheon V.** Research data meets research information management: Two case studies using (a) Pure CERIF-CRIS and (b) EPrints repository platform with CERIF extensions // Procedia Computer Science. 2014. Vol. 33. P. 199–206.
13. **Jeffery K. G., Asserson A.** Supporting the Research Process with a CRIS // Enabling Interaction and Quality: Beyond the Hanseatic League. 2006. P. 121–130.

Architectural and Technological Aspects of the Development and Maintenance of Large Information Analysis Systems in the Area of Science and Education

V. A. Vasenin, vasenin@msu.ru, **M. A. Zanchurin**, maxim.zanchurin@gmail.com,
A. S. Kozitsyn, alexanderkz@mail.ru, **M. A. Krivchikov**, maxim.krivchikov@gmail.com,
D. A. Shachnev, mitya57@mitya57.me, Lomonosov Moscow State University, Moscow, 119991, Russian Federation

Corresponding author:

Vasenin Valery A., Dr. Sc., Professor, Head of the Chair, Lomonosov Moscow State University, Moscow, 119991, Russian Federation
E-mail: vasenin@msu.ru

Mechanisms for managing the scientific, technical and educational activity are a subject of a special attention of the government. With the emergence of packet communications technologies and tools, and the Internet meta-network and technologies based on it, at the beginning of 21st century there were all prerequisites ready for creating systems available to collect, store and analyze large amounts of scientometrical data in an immediate manner. The present article considers the architectural and technological aspects of the development and maintenance of such large information analysis systems in the area of science and education, based on the example of the "ISTINA" system (referred to as System throughout the text).

The main requirements to the System architecture include:

- 1) modularity and scalability, on the macro-level of system description as well as on the underlying levels;
- 2) hierarchical organization and orientation on processes;
- 3) completeness of the reflection of all processes currently in demand and for the foreseeable future that are automatized within the System;
- 4) taking processes of interaction with external databases storing various kinds of auxiliary data into account.

The technological principles include:

- compliance with regulations relating to systems of national importance;
- adherence to the main provisions of software engineering at all stages of the System life cycle;
- using the mechanisms of collaboration and integration with the external and internal data bases taking the confidentiality requirements into account.

The System architecture reflects the ontological model of the scientometrics using its representation as the subject area on which the System is oriented. The integrating mechanisms of the System are implemented using a relational database, the structure of which reflects the main aspects of the ontology, and a set of applications using that database.

One of the natural models for representing the objects in the System database is the graph model. A visual ontology editor for working with graph is being developed at the moment. Investigations on using the ontologies in the system architecture are considered as one of the most perspective development directions of such systems by the developers.

With regards to technological aspects this article describes approaches to source code management and continuous integration; processing the users' requests and bug reports; tools for preparing the documentation packages for the System; experience with upgrading to the new version of Django framework.

Keywords: information analysis systems, science and education, scientometrics, software engineering, distributed systems, software architecture

For citation:

Vasenin V. A., Zanchurin M. A., Kozitsyn A. S., Krivchikov M. A., Shachnev D. A. Architectural and Technological Aspects of the Development and Maintenance of Large Information Analysis Systems in the Area of Science and Education, *Programmnaya Ingeneria*, 2017, vol. 8, no. 10, pp. 448–455.

DOI: 10.17587/prin.8.448-455

References

1. Nalimov V. V., Mulchenko Z. M. *Naukometrija. Izuchenie razvitiya nauki kak informacionnogo processa. Fiziko-matematicheskaja biblioteka inzhenera* (Studying the science development as an information process. Physics and mathematics library of the engineer), Moscow, Nauka, 1969, 192 p. (in Russian).
2. Vasenin V. A. Modernizacija jekonomiki i novye aspekty inzhenerii programm (Economics modernization and new aspects of software engineering), *Programmnaya Ingeneria*, 2012, no. 2, pp. 2–17 (in Russian).
3. Mikhailenko I. V. Informacionno-analiticheskaja sistema "Karta Rossijskoj nauki" kak instrument nauchnogo monitoringa (Information analysis system "Russian science map"), *Dinamika sistem, mehanizmov i mashin*, 2014, no. 4, pp. 75–78 (in Russian).
4. Goncharov M. V., Mikhailenko I. V. Naukometricheskie pokazateli. ispol'zuyemye v IAS "Karta Rossijskoj nauki". Metodika raschjota (Scientometrical parameters used in IAS "Russian science map". Methodics of calculation), *Nauchnye i tehniczeskie biblioteki*, 2016, no. 12, pp. 37–43 (in Russian).
5. Vasenin V. A., Golomazov D. D., Gankin G. M. Arhitektura, metody i sredstva bazovoj sostavljajushhej sistemy upravlenija nauchnoj informaciej "ISTINA — Nauka MGU" (Architecture, methods and tools of the base part of the "ISTINA — MSU Science" science information management system), *Programmnaya Ingeneria*, 2014, no. 9, pp. 3–12 (in Russian).
6. Vasenin V. A., Afonin S. A., Kozitsyn A. S., Golomazov D. D. Sistema "ISTINA" dlja podgotovki prinjatija reshenij na osnove analiza naukometricheskoj informaciej (The "ISTINA" decision preparing system based on scientometrical data analysis), *Nauchnyj servis v seti Internet: Trudy XVII Vserossijskoj nauchnoj konferencii*, IPM im. M. V. Keldysha, Moscow, 2015, pp. 51–62 (in Russian).
7. Vasenin V. A., Itkes A. A., Bukhonov V. Yu., Galatenko A. V. Modeli logicheskogo razgranichenija dostupa v mnogopol'zovatel'skikh sistemah upravlenija naukometricheskimi kontentom (Models of logical access restriction in multi-user scientometrical content management systems), *Programmnaya Ingeneria*, 2016, vol. 7, no. 12, pp. 547–558 (in Russian).
8. Vasenin V. A., Zenzinov A. A., Lunev K. V. Ispol'zovanie naukometricheskikh informacionno-analiticheskikh sistem dlja avtomatizacii provedenija konkursnyh procedur na primere informacionno-analiticheskoi sistemy "ISTINA" (Using scientometrical information analysis systems for automating the contest procedures on example of "ISTINA" information analysis system), *Programmnaya Ingeneria*, 2016, vol. 7, no. 10, pp. 472–480 (in Russian).
9. Cobo M. J., Lopez-Herrera A. G., Herrera-Viedma E., Herrera F. Science mapping software tools: Review, analysis, and cooperative study among tools, *Journal of the Association for Information Science and Technology*, 2011, vol. 62, no. 7, pp. 1382–1402.
10. Grivel L., Polanco X., Kaplan A. A computer system for big scientometrics at the age of the World Wide Web, *Scientometrics*, 1997, vol. 40, no. 3, pp. 493–506.
11. Johansson A., Ottosson M. O. A national current research information system for Sweden, *11th International Conference on Current Research Information Systems*, CRIS 2012: Prague, Czech Republic, June 6–9, 2012, Agentura Action M, 2012, pp. 67–71.
12. Clements A., McCutcheon V. Research data meets research information management: Two case studies using (a) Pure CERIF-CRIS and (b) EPrints repository platform with CERIF extensions, *Procedia Computer Science*, 2014, vol. 33, pp. 199–206.
13. Jeffery K. G., Asserson A. Supporting the Research Process with a CRIS, *Enabling Interaction and Quality: Beyond the Hanseatic League*, 2006, pp. 121–130.

В. М. Димитров, преподаватель, e-mail: dimitrov@cs.karelia.ru,
Петрозаводский государственный университет

Повышение производительности языка LaOQL за счет параллельной реализации запросов

Разработанный и реализованный автором язык запросов LaOQL используется на практике сотрудниками вычислительного центра Петрозаводского государственного университета для поиска физического местоположения устройства по его IP-адресу, а также для выполнения ряда других запросов. Язык LaOQL позволяет сократить длину текста запроса на 30...90 % по сравнению с длиной текста запросов на одном из аналогичного назначения языков — на языке HQL. Это способствует, во-первых, повышению производительности исследователя сети и, во-вторых, уменьшению числа ошибок. Вместе с тем производительность выполнения запросов на настоящее время пока не соответствует ожиданиям пользователей, уступая, в том числе языку HQL. Одним из способов улучшения производительности является возможность параллельного выполнения независимых частей алгоритма выполнения запросов. В статье представлены способы параллельной реализации языка LaOQL. Проведенные эксперименты показали, что в определенных случаях можно добиться значительного улучшения производительности по сравнению с аналогичными запросами, выполняемыми последовательно.

Ключевые слова: объектно-ориентированные базы данных, язык запросов, вычислительные сети, параллельные вычисления

Введение

Разрабатываемая в Петрозаводском государственном университете (ПетрГУ) экспериментальная платформа Nest [1, 2] (далее ЭП Nest) предоставляет исследователю моделей и методов сетевого управления средства автоматизированного построения графа Сети, описывающего взаимосвязи аппаратных элементов информационно-коммуникационной инфраструктуры локального поставщика сетевых услуг (лПСУ), механизмы визуализации этого графа и выполнения запросов. Подсистема запросов использует лаконичный язык LaOQL [3] для выполнения запросов к объектно-ориентированной базе данных (ООБД). Объектный граф модели архитектуры лПСУ описан в работе [2]. Эта модель представляет собой композицию трех моделей структур лПСУ: пространственной (*S* — *Spatial*), организационной (*O* — *Organizational*) и аппаратно-сетевой (*N* — *Network*).

Язык LaOQL позволяет проводить выборку объектов, выборку значений атрибутов объектов, фильтрацию объектов по значениям атрибутов, сортировку выборки, объединение результатов запросов [3]. Язык LaOQL сокращает длину текста запроса на 30...90 % по сравнению с длиной текста запросов на одном из аналогичных языков — языке HQL. Это обстоятельство способствует, во-первых, повышению производительности исследователя сети и, во-вторых,

к уменьшению числа ошибок. Например, запрос на поиск маршрутизатора на языке HQL выглядит следующим образом:

```
select d from Device as d
left join d.linkInterfaces as lis
left join lis.networkInterfaces as nis
left join nis.network.networkInterfaces
as nis2 where nis2.linkInterface.device.
forwarding=true and d.id=25
```

Аналогичный запрос на языке LaOQL можно переписать следующим образом:

```
d (ni#(d.forwarding=true && d.id=25))
```

Как видно из текстов запросов, во втором случае пользователю нет необходимости знать конкретные свойства объектов и их последовательность для построения связей, что позволяет сократить время на поиск данной информации, а следовательно и время на построение запроса. Для хранения информации о связях объектов язык LaOQL использует карту предметной области, которая строится автоматически.

Автором были проведены эксперименты по сравнению производительности (в качестве такой характеристики было выбрано время выполнения запроса) и потреблению памяти языков HQL [4] и

LaOQL. Эксперименты проводили для различных ООБД с различным числом элементов на разных видах запросов. К их числу относятся выборки объектов; значений атрибутов объектов; с простым условием, со сложным условием; запросы на соединение, сортировку и др. Результаты этих экспериментов показали, что язык LaOQL в среднем на 15 % уступает языку HQL по данным показателям.

В настоящей публикации предложен параллельный алгоритм выполнения запросов языком LaOQL. Описанные в статье результаты экспериментов свидетельствуют об увеличении примерно на 30 % быстродействия процедуры выполнения запросов по сравнению с последовательной версией, что позволило опередить или приблизиться ко времени выполнения запросов на языке HQL.

1. Параллельное выполнение запросов

Проблема параллельного выполнения запроса в ООБД достаточно широко изучена [5—11]. В контексте результатов, которые представлены в настоящей статье, были исследованы возможности реализации параллельного выполнения для существующего прототипа языка LaOQL в следующих случаях: проход по пути выражения, выполнение условий (проверка условий для списка объектов); выполнение функций; запросы с объединением и пересечением. По результатам анализа производительности выполнения запросов в последовательной версии был сделан вывод о том, что наибольшее отставание от времени выполнения запросов на языке HQL достигается на запросах, в которых требуется проходить по пути выражения. Поэтому было принято решение о параллельной реализации таких запросов. Рассмотрим этот случай более подробно.

В случае запроса, при котором необходимо выполнить проход по ссылкам от одного объекта до другого, существует возможность выполнять эти проходы для множества объектов параллельно. Были исследованы два случая: распараллеливание выполнения проходов только от корневых элементов и распараллеливание выполнения проходов от любого уровня вложенности.

Рассмотрим запрос вида $C_1(C_2(C_3))$, где C — обозначает класс объектной модели. В данном запросе требуется выполнить проходы от корневых объектов класса C_1 до объектов класса C_2 и от объектов класса C_2 до объектов класса C_3 . В первом случае выполняется параллельный обход от множества C_1 до объектов класса C_3 . Во втором случае выполняется параллельный обход от множества объектов класса C_1 до объектов класса C_2 . Затем в этом же процессе для множества объектов класса C_2 выполняется параллельный обход до объектов класса C_3 .

Этот подход проиллюстрирован на рис. 1 и 2 для первого и второго случаев соответственно. На этих рисунках параллельный алгоритм пред-

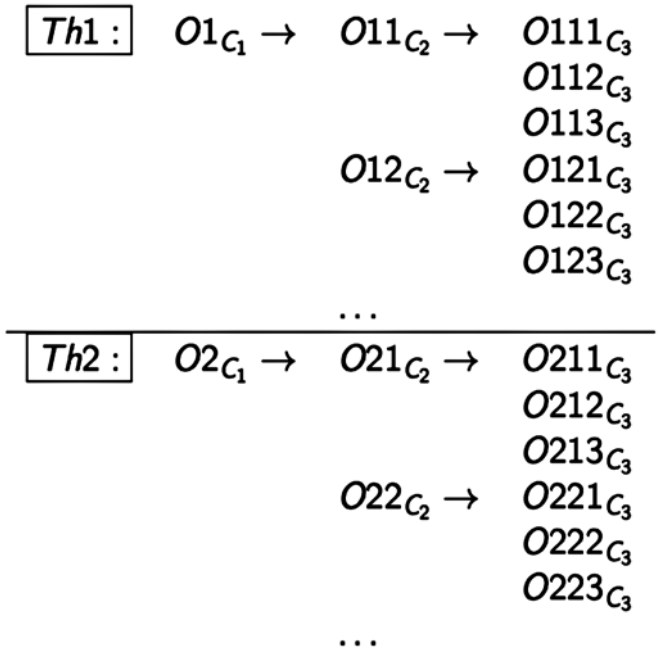


Рис. 1. Параллельный проход только от корневых элементов

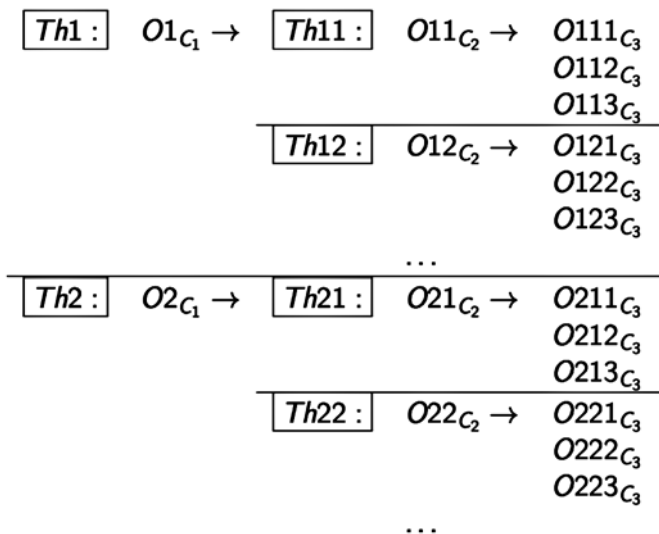


Рис. 2. Параллельный обход на любом уровне вложенности

ставлен в ярусно-параллельной форме (ЯПФ) [12]. На рисунках использованы следующие обозначения:

- Th — параллельный процесс;
- C — класс объектной модели;
- O_C — объект класса C .

Как видно на рис. 2, для объектов $O11_{C_2}$ и $O12_{C_2}$ внутри процесса $Th1$ запускаются отдельные процессы $Th11$ и $Th12$.

Параллельный алгоритм реализован с помощью программного каркаса Fork/Join из состава пакета `java.util.concurrent JDK7`.

2. Система генерации объектного графа модели SON

Таблица 1

Весы для элементов модели SON

SON-элемент	Вес
<i>Building</i>	1
<i>Floor</i>	5
<i>Room</i>	50
<i>Occupancy</i>	5
<i>SimpleOU</i>	20
<i>CompositeOU</i>	5
<i>Device</i>	300
<i>Network</i>	5
<i>NetworkInterface</i>	170
<i>EthernetInterface</i>	150
<i>LinkInterface</i>	150
<i>IPv4Interface</i>	170
<i>IPNetwork</i>	5
<i>VLANInterface</i>	20

Для проведения экспериментов по анализу производительности реализованного прототипа была разработана система генерации объектного графа модели SON, который затем мог загружаться в любую из необходимых для экспериментов ООБД. Это необходимо было сделать по следующим причинам:

- возможность указания необходимого числа элементов в БД;
- возможность заполнения свойств определенными значениями (для запросов на условия);
- необходимость проведения экспериментов на различных конфигурациях программного и аппаратного обеспечения.

Для получения модели, более приближенной к реальности, для каждого класса модели SON были введены веса, которые приведены в табл. 1.

Под инициализацией объекта будем понимать создание объекта с заполнением его свойств. Для каждого созданного объекта заполняются свойства *description* и *name* (они наследуются от общего класса *SonElement*) из заранее определенного набора значений.

Кроме того, для некоторых объектов заполняются следующие их уникальные и отличительные свойства:

- *Floor* — *number*;
- *Room* — *number*;
- *IPv4Interface* — *inetAddress*;
- *EthernetInterface* — *MACAddress*;
- *IPNetwork* — *address u mask*.

Генерация объектного графа происходит в соответствии со следующим алгоритмом.

Шаг 1. Перед началом генерации создается массив из классов модели SON. Число элементов в массиве для класса определяется введенными выше весами.

Шаг 2. Также перед началом генерации для каждого класса инициализируется один объект, который приобретает статус текущего объекта. Между объектами в полученном наборе устанавливаются связи.

Шаг 3. С помощью счетчика псевдослучайных чисел выбирается класс из массива, для которого инициализируется объект. При необходимости устанавливаются связи для нового объекта с текущими, а затем текущий объект данного класса заменяется на новый, приобретая, в свою очередь, статус текущего. Правило для установления связи выглядит следующим образом:

- если связь один-ко-многим с некоторым другим классом, то новый объект добавляется в список текущего объекта этого другого класса (например, если выбранным классом является класс *Floor*, то для объекта этого класса устанавливается связь с текущим объектом *Building*, отметим, что на данном этапе с объектами класса *Room* никаких связей не устанавливается);

- если связь один-к-одному с некоторым другим классом, то связь устанавливается только в случае отсутствия у текущего объекта этого другого класса ссылки.

3. Схема организации экспериментов

Общая архитектура проведения экспериментов представлена на рис. 3.

Система предоставляет пользователю командный интерфейс (консоль). Модуль "Конфигуратор

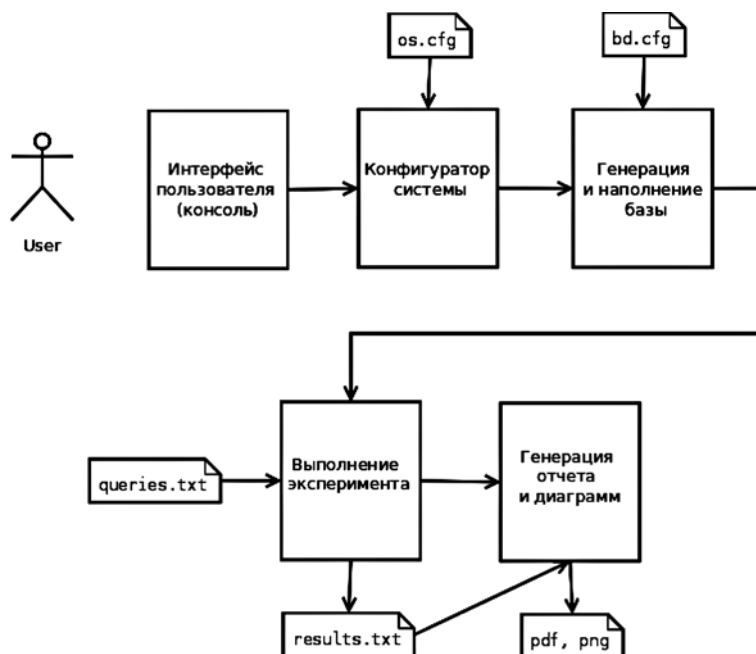


Рис. 3. Архитектура системы проведения экспериментов

системы" отвечает за настройку системы (число ядер, язык запросов проведения эксперимента, типы отчетов, выбор базы данных и др.). Модуль "Генерация и наполнение базы" отвечает за генерацию модели SON (более подробно описано в разд. 2). Модуль "Генерации отчетов и диаграмм" строит отчеты и диаграммы на основе настроек системы.

Для проведения экспериментов использовали следующие БД: H2 [4], Derby [13], HSQLDB [14], а для преобразования ОО-модели — систему Hibernate [15]. Так как выбранные СУБД позволяют создавать БД в оперативной памяти, без сохранения их в постоянную память, то эксперименты проводили как с БД в оперативной памяти, так и с БД в постоянной памяти на дисках.

Для обнаружения тенденций изменения производительности с ростом числа элементов генерировали и сохраняли модель с различным числом элементов: 1000, 5000, 10 000, 15 000, 20 000, 50 000 и 100 000.

Эксперименты проводили как на отдельных видах запросов, так и на заданном списке запросов. Для отслеживания тенденций по изменению производительности с добавлением новых возможностей или с изменением существующего кода использовался список, состоящий из 411 запросов. Такая возможность позволила на отдельных этапах разработки

предотвратить появление серьезных ухудшений в производительности. В частности, предотвращать падение производительности после добавления в язык запросов возможности по получению результата только с уникальными объектами.

Сравнение на отдельных видах запросов. Для отдельных видов запросов были отобраны следующие типы запросов:

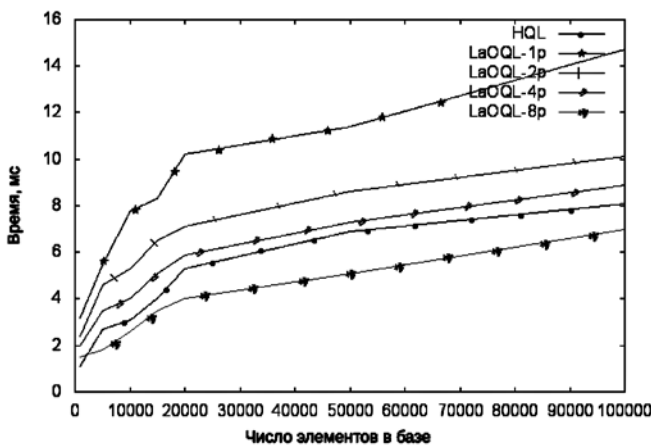
- 1) выборка связанных элементов (один уровень вложенности);
- 2) выборка связанных элементов (один уровень вложенности, проход должен осуществляться в обратном направлении по сравнению с предыдущим запросом);
- 3) выборка связанных элементов (два уровня вложенности).

Приведенный выше список запросов был реализован на языке LaOQL следующим образом:

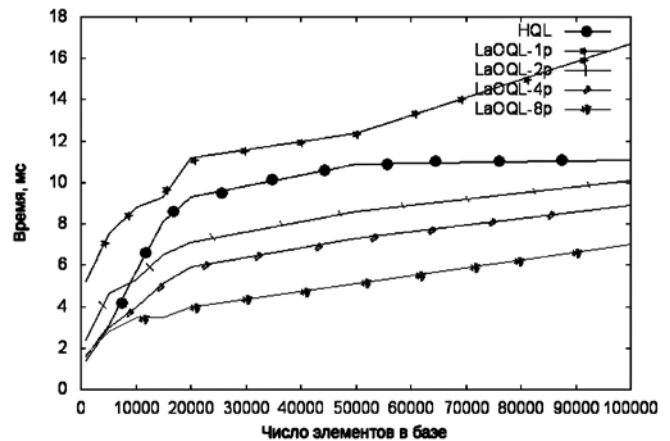
- 1) device (building)
- 2) building (device)
- 3) li (n (d))

Соответствующие им аналоги на языке HQL были реализованы следующим образом:

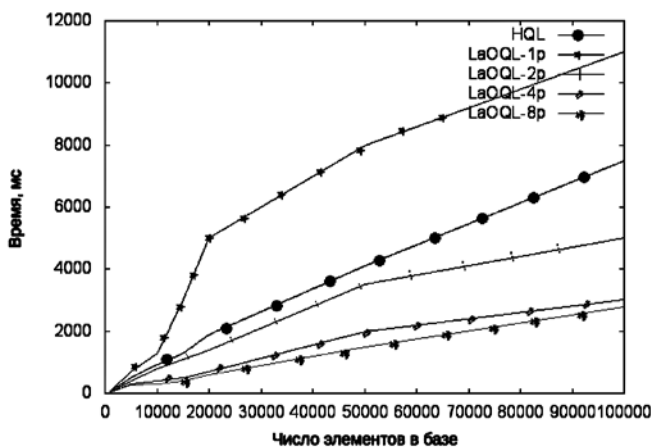
- 1) select d, d.occupancy.room.floor.building from Device d



а)



б)



в)

Рис. 4. Зависимость времени выполнения запросов device (building) (а), building (device) (б) и li (n (d)) (в) от числа элементов в базе при работе на различном числе ядер:

1p — параллельная реализация; 2p, 4p, 8p — 2, 4 и 8 ядер соответственно

```

2) select b, d from Building b
   left join b.floors f left join f.rooms r
   left join r.occupancies o left join o.devices d
3) select distinct li, ni.network, d from
LinkInterface li
   join li.networkInterfaces ni join ni.network n
   join n.networkInterfaces ni2 join ni2.
linkInterface.device d

```

На рис. 4 представлены результаты выполнения запросов языков HQL и LaOQL на различном числе вычислительных ядер. Запросы на языке HQL проводились на максимально доступном количестве ядер (в нашем случае это 8). График LaOQL-1p представляет собой последовательную реализацию языка LaOQL. Как видно из графиков, удалось значительно улучшить время выполнения запросов по сравнению с последовательной реализацией языка LaOQL.

Сравнение на сценариях. Помимо отдельных видов запросов, производительность их выполнения сравнивали на списках запросов с целью отслеживания работы разных запросов. В ЭП Nest существует ряд задач, для решения которых требуется выполнить определенный набор запросов для получения некоторого результата. К их числу относятся:

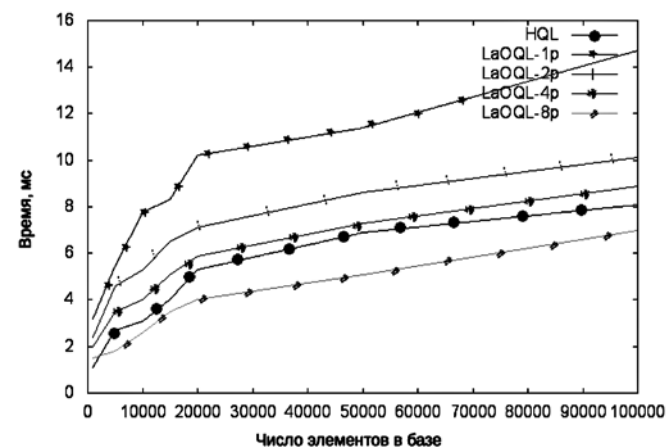
1) выполнение скрипта по автоматизированному заполнению организационной и пространственной структур по заданному файлу с шаблоном и собранной сетевой структуре (SonEnlivener.js, 132 запроса);

2) поиск информации о сетевом адресе (AddressInfo, 6 запросов);

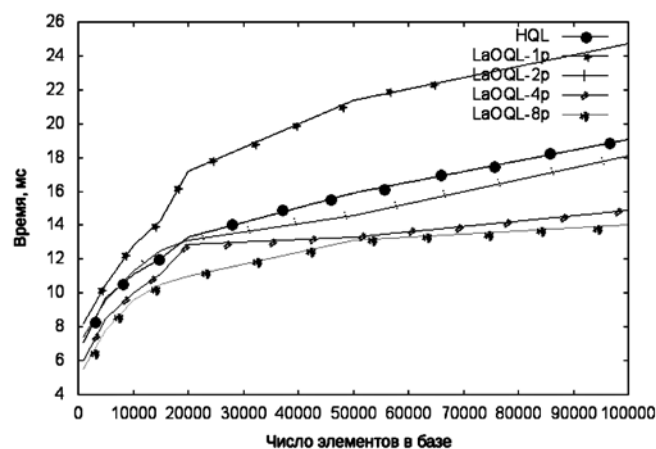
3) построение дерева SON-элементов в графическом пользовательском интерфейсе (SonNestTree, 4 запроса).

На рис. 5 представлены результаты выполнения сценариев на языках HQL и LaOQL. Как видно на графиках, для сценария SonEnlivener.js для пула из 8 вычислительных ядер реализация языка LaOQL опережает время выполнения сценариев с использованием языка HQL. Для сценариев AddressInfo и SonNestTree такое можно сказать для пула 2, 4 и 8 ядер.

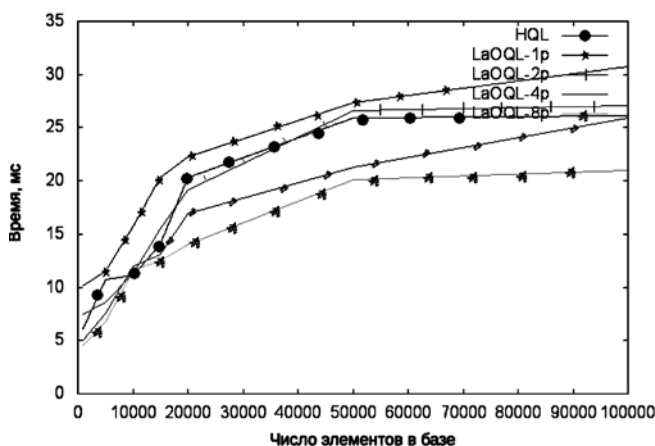
В табл. 2 представлены общие результаты изменения времени выполнения запросов для последовательной и параллельной реализаций языка LaOQL и языка HQL в процентах. Например, для 1000 элементов в базе и числе ядер 2 значение 25 означает, что время выполнения запросов на параллельной версии языка LaOQL на 25 % выше по сравнению со временем выполнения запросов на последовательной



а)



б)



в)

Рис. 5. Зависимость времени выполнения сценариев SonEnlivener (а), AddressInfo (б) и SonNestTree (в) от числа элементов в базе при работе на различном числе ядер:

1p — параллельная реализация; 2p, 4p, 8p — 2, 4 и 8 ядер соответственно

Результаты изменения времени выполнения запросов

Число элементов в базе	LaOQL (последовательная версия) против LaOQL (параллельная версия), %			LaOQL против HQL, %			
	Число ядер						
	2	4	8	1	2	4	8
1000	25	37	53	-190	-118	-81	-36
5000	16	36	67	-104	-70	-30	33
10000	32	48	66	-152	-71	-29	16
15000	21	38	57	-108	-63	-28	13
20000	30	42	60	-92	-34	-11	25
50000	24	35	55	-65	-25	-6	26
100000	31	39	52	-81	-25	-10	14

версии языка LaOQL. Как видно из данных таблицы, параллельная версия реализации опережает по времени выполнения последовательную версию языка LaOQL на любом пуле ядер, а реализацию языка HQL — на пуле из 8 ядер.

Заключение

Представленная параллельная реализация лаконичного языка запросов LaOQL имеет более высокую производительность, чем ее последовательная версия, что подтверждается экспериментами. Так, время выполнения некоторых сложных запросов при использовании пула из 8 потоков примерно в 3 раза меньше, чем время выполнения тех же запросов с помощью последовательной реализации.

Автор благодарит научного руководителя Ю. А. Богоявленского, старших преподавателей ПетрГУ кафедры ИМО М. А. Крышени и А. С. Колосова и магистранта А. А. Андреева за полезные советы и помощь в подготовке статьи.

Список литературы

1. Богоявленский Ю. А., Колосов А. С. Организация и автоматизированная поддержка объектной базы данных графа ИКТ-инфраструктуры поставщика услуг Интернета // Научно-технические ведомости СПбГПУ. 2011. № 3 (126). С. 27—36.
2. Богоявленский Ю. А. Прототип экспериментальной платформы Nest для исследования моделей и методов управления ИКТ-инфраструктурами локальных поставщиков услуг Интернет // Программная инженерия. 2013. № 2. С. 11—20.
3. Димитров В. М., Воронин А. В., Богоявленский Ю. А. Лаконичный язык запросов LaOQL на основе связей в объ-

ектной модели данных // Программная инженерия. 2017. № 3. С. 112—119.

4. H2 database engine. URL: <http://www.h2database.com/html/main.html>.

5. Sampaio S., Paton N., Smith J., Watson P. Measuring and modelling the performance of a parallel odmg compliant object database server // Concurrency and Computation: Practice and Experience. 2006. Vol. 18, No. 1. P. 63—109.

6. Jaedicke M., Mitschang B. A framework for parallel processing of aggregate and scalar functions in object-relational DBMS. TUM-I 9741, SFB-Bericht Nr. 342/25/97 A, September 1997. URL: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.55.1437&rep=rep1&type=pdf>.

7. Thakore A., Su S. Performance analysis of parallel object-oriented query processing algorithms // Distributed and Parallel Databases. 1994. Vol. 2, No. 1. P. 59—100.

8. Smith J., Watson P., Sampaio S., Paton N. Polar: An architecture for a parallel odmg compliant object database // Proceedings of the ninth International conference on Information and knowledge management, ACM. 2000. P. 352—359.

9. Taniar D., Leung C. Query execution scheduling in parallel object-oriented databases // Information and Software Technology. 1999. Vol. 41, No. 3. P. 163—178.

10. Sampaio S., Smith J., Paton N., Watson P. Experimenting with object navigation in parallel object databases // 12th International Workshop on Database and Expert Systems Applications (DEXA 2001), 3—7 September 2001, Munich, Germany; IEEE Computer Society. 2001. P. 103—109.

11. Tavares F., Victor A., Mattoso M. Parallel processing evaluation of path expressions // Proc XV Brazilian Symposium on Databases, SBC, João Pessoa, Brazil. 2000. P. 49—63.

12. Воеводин В. В., Воеводин Вл. В. Параллельные вычисления. СПб.: БВХ-Петербург, 2002. 608 с.

13. Apache derby. URL: <http://db.apache.org/derby/>.

14. Hsqldb. URL: <http://hsqldb.org/>.

15. Bauer C., King G. Hibernate in Action (In Action series). Greenwich, CT, USA: Manning Publications Co., 2004. 400 p.

Laconic Object Query Language Using Features of Object Model

V. M. Dimitrov, dimitrov@cs.petrstu.ru, Petrozavodsk State University, Petrozavodsk, 185910, Republic of Karelia, Russia

Corresponding author:

Dimitrov Vjacheslav M., Senior Lecturer, dimitrov@cs.petrstu.ru, Petrozavodsk State University, Petrozavodsk, 185910, Republic of Karelia, Russia,
E-mail: dimitrov@cs.petrstu.ru

Received on July 17, 2017

Accepted on July 24, 2017

The LaOQL query language is used by staff of computing centre of PetrSU, for example, to find the physical location of the device by its IP address. However, the query performance does not meet user's expectations, leaving LaOQL behind its analogues, for example, the HQL language. One way to improve performance is the possibility of parallel execution of independent parts of the algorithm perform queries.

The report presents methods of parallel implementation of the LaOQL language. Experiments have shown that in certain cases it is possible to achieve significant performance improvements in comparison with the same queries that are executed sequentially.

Keywords: *object-oriented database system, query language, network, parallel queries*

For citation:

Dimitrov V. M. Laconic Object Query Language Using Features of Object Model, *Programmnaya Ingeneria*, 2017, vol. 8, no. 10, pp. 456–462.

DOI: 10.17587/prin.8.456-462

References

1. **Bogoiavlenskii Yu. A., Kolosov A. S.** Organizacija i avtomatizirovannaja podderzhka ob#ektnoj bazy dannyh grafa IKT-infrastruktury postavshhika uslug Interneta (Organization method for storing an IT-infrastructure (Network) graph of a service provider based on the object-oriented model), *Nauchno-tehnicheskie vedomosti SPbGPU*, 2011, no. 3 (126), pp. 27–36 (in Russian).
2. **Bogoiavlenskii Yu. A.** Prototip eksperimental'noj platformy Nest dlja issledovanija modelej i metodov upravljenija IKT-infrastrukturami lokal'nyh postavshhikov uslug Internet (Prototype of the Testbed Nest for Research of Network Management Methods and Models at the Enterprise Network Level), *Programmnaya Ingeneria*, 2013, no. 2, pp. 11–20 (in Russian).
3. **Dimitrov V. M., Voronin A. V., Bogoiavlenskii Yu. A.** Laconichnyj jazyk zaprosov LaOQL na osnove svjazej v ob#ektnoj modeli dannyh (Laconic Object Query Language Using Features of Object Model), *Programmnaya Ingeneria*, 2017, vol. 8, no. 3, pp. 112–119 (in Russian).
4. **H2 database engine**, available at: <http://www.h2database.com/html/main.html>.
5. **Sampaio S., Paton N., Smith J., Watson P.** Measuring and modelling the performance of a parallel odmng compliant object database serve, *Concurrency and Computation: Practice and Experience*, 2006, vol. 18, no. 1, pp. 63–109.
6. **Jaedicke M., Mitschang B.** A framework for parallel processing of aggregate and scalar functions in object-relational DBMS, TUM-I 9741, SFB-Bericht Nr. 342/25/97 A, September 1997, available at: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.55.1437&rep=rep1&type=pdf>.
7. **Thakore A., Su S.** Performance analysis of parallel object-oriented query processing algorithms, *Distributed and Parallel Databases*, 1994, vol. 2, no. 1, pp. 59–100.
8. **Smith J., Watson P., Sampaio S., Paton N.** Polar: An architecture for a parallel odmng compliant object database, *Proceedings of the ninth International conference on Information and knowledge management*, ACM, 2000, pp. 352–359.
9. **Taniar D., Leung C.** Query execution scheduling in parallel object-oriented databases, *Information and Software Technology*, 1999, vol. 41, no. 3, pp. 163–178.
10. **Sampaio S., Smith J., Paton N., Watson P.** Experimenting with object navigation in parallel object databases, *Database and Expert Systems Applications, 2001, Proceedings, 12th International Workshop on*, IEEE, 2001, pp. 103–109.
11. **Tavares F., Victor A., Mattoso M.** Parallel processing evaluation of path expressions, *Proc. XV Brazilian Symposium on Databases, SBC, Joao Pessoa, Brazil, 2000*, pp. 49–63.
12. **Voevodin V. V., Voevodin VI. V.** *Parallel'nye vychislenija* (Parallel calculation), Saint-Petersburg, BVX-Peterburg, 2002, 608 p. (in Russian).
13. **Apache derby**, available at: <http://db.apache.org/derby/>.
14. **Hsqldb**, available at: <http://hsqldb.org/>.
15. **Bauer C., King G.** *Hibernate in Action (In Action series)*, Greenwich, CT, USA, Manning Publications Co., 2004, 400 p.

К. Ф. Иванова, канд. техн. наук, докторант, e-mail: klara.i2010@yandex.ru,
Санкт-Петербургский государственный университет

Интервальное допустимое решение задачи линейного программирования в системе MATLAB

Предложено решение задачи линейного программирования (ЗЛП) при наличии неопределенности параметров целевой функции (ЦФ) и ограничений-неравенств путем ее сведения к обычным детерминированным системам. Алгоритм расчета моделируемой оптимальной задачи базируется на использовании симплексного метода, встроенного в систему MATLAB. Интервальное решение стандартной задачи оптимизации представляется как интервальное расширение оптимального точечного плана, полученного на базе номинальной канонической системы уравнений. С использованием интервального представления коэффициентов полученной номинальной системы решается интервальная задача внутреннего оценивания множества решений интервальной системы линейных алгебраических уравнений (ИСЛАУ) и вычисляется интервальный вектор (брус), соответствующий оптимальному допустимому плану.

Допустимость найденного интервального оптимального плана проверяется его подстановкой в точечные системы уравнений, определяющие внешнее оценивание множества решений ИСЛАУ и сконструированные на основе интервальной канонической системы. Для оптимальных угловых матриц формализованных точечных систем их произведение на найденный вектор не должно выходить за границы интервального вектора ограничений \mathbf{b} .

В интервальной постановке симметричной двойственной ЗЛП целью исследования является получение границ оптимизируемых ЦФ, не выходящих за пределы возможных отклонений, вызванных неопределенностью входных параметров задачи и отвечающих заданной системе ограничений.

Ключевые слова: задача линейного программирования, каноническая система, внешняя, внутренняя оценки множества решений ИСЛАУ, оптимальный допустимый план, расширение точечного решения, ограничения, ЦФ

Введение

Интервальная задача линейного программирования как предмет исследования. Традиционная задача линейного программирования (ЗЛП), состоящая в оптимизации эффективности ЦФ, формулируется как задача поиска на допустимом множестве оптимального решения, отвечающего заданному критерию. В условиях неопределенности, вызванной неточными измерениями и неучтенными факторами воздействия, ошибки моделирования можно компенсировать интервальным заданием следующих параметров: коэффициентов матриц неравенств-ограничений и компонент вектора их правых частей. В этом случае точное решение попадает в интервал предельных значений оптимальной ЦФ, гарантируя надежность ее оценки.

Задача идентификации интервальных матрично-векторных систем операций [1–3] в интервальном представлении записывается следующим образом:

$$\Xi_{tol}(\mathbf{A}, \mathbf{b}) = \{x \in \mathbb{R}^n \mid (\mathbf{A}x \subseteq \mathbf{b})\}, \quad (1)$$

где Ξ_{tol} — множество допустимых значений;

$\mathbf{A} \in \mathbb{IR}^{n \times n}$ — интервальная матрица размера $n \times n$,
 $\mathbf{b} \in \mathbb{IR}^n$ — интервальный вектор правой части размера n ; x — вектор неизвестных системы уравнений.

Задача построения множества вида (1) называется линейной задачей о допусках, пришедшей из задачи расчета межотраслевого экономического баланса при неточных данных [4].

В работе Н. А. Хлебалина [5] предложено брать в качестве наиболее вероятного допустимого множества решений решение средней точечной системы $(\text{mid } \mathbf{A})x = \text{mid } \mathbf{b}$, которое затем тестируется на включение $\mathbf{A}x \subseteq \mathbf{b}$. Описанный тест средней системы, как показывает практика, работает лишь когда матрица \mathbf{A} достаточно узка в сравнении с вектором правой части \mathbf{b} . В 1985 г. И. Рон показал [6], что $x \in \Xi_{tol}$ тогда и только тогда, когда $x = x' - x''$, когда x'' и x' дают решение систем линейных неравенств:

$$\begin{cases} \bar{\mathbf{A}}x'' - \underline{\mathbf{A}}x' \leq \bar{\mathbf{b}}, \\ -\underline{\mathbf{A}}x'' + \bar{\mathbf{A}}x' \leq -\underline{\mathbf{b}}, \\ x'' \geq 0, \quad x' \geq 0. \end{cases} \quad (2)$$

В системе (2) вещественные матрицы $\underline{\mathbf{A}}$ и $\bar{\mathbf{A}}$ и векторы $\underline{\mathbf{b}}$ и $\bar{\mathbf{b}}$ являются гранями соответствующих интервальных объектов $[\underline{\mathbf{A}}, \bar{\mathbf{A}}]$ и $[\underline{\mathbf{b}}, \bar{\mathbf{b}}]$. Как следствие, И. Рон добавил утверждение, что допустимое решение является выпуклым многогранником. Если же это условие не выполнено, то делается заключение

о практической неразрешимости линейной задачи о допусках. Постановка и решение интервальных задач о допустимом решении широко освещено в работах [7–9]. В этих работах доказательство существования допустимого решения сводится к обоснованию выпуклости множества Ξ_{tol} интервальными методами и к выписыванию системы линейных уравнений с каждым ортантом. В работе [8] показано, что из описания Ξ_{tol} в виде множества уровня функционала следует, что его подграфиком является выпуклое многогранное множество. В работе [9] И. А. Шарая доказала, что допустимое множество представимо пересечением не более чем $n \cdot 2^n$ гиперполос (n -размер системы). Гиперполосой в \mathbf{R}^n считается множество $\Xi_{tol}(\mathbf{A}, \mathbf{b}) = \{x \in \mathbf{R}^n | \mathbf{A} \cdot x \subseteq \mathbf{b}\}$. Из приведенных доказательств следует, что задача отыскания допустимого множества решений и задача о пересечении конечного числа гиперполос эквивалентны, и множества решений этих задач можно оценивать одинаковыми методами.

Следует отметить, что во всех методиках оценок влияния интервальности матриц и правых частей уравнений сопутствующим недостатком решения задачи является возникновение систем больших размерностей. Этот недостаток характерен для всех интервальных систем с интервальными параметрами, что приводит к чрезмерным вычислительным трудностям.

Постановка задачи линейного программирования

Наиболее распространенным типом задач линейного программирования является производственная задача максимизации суммарной прибыли путем выпуска определенной номенклатуры продукции при ограниченных ресурсах [4]. Количество ресурсов m обозначается вектором $\mathbf{b} = (b_1, b_2, \dots, b_m)$, технологические коэффициенты a_{ij} , $i = 1, m, j = 1, n$ показывают норму расхода i -го ресурса на производство единицы j -й продукции. Эффективность выпуска единицы j -й продукции определяется прибылью c_j , стоимостная оценка единицы ресурса — y_j .

В общем виде целью постановки симметричной пары взаимно двойственных ЗЛП является решение как прямой, так двойственной задач. В прямой задаче определяется вектор выпуска продукции $\mathbf{x} = (x_1, x_2, \dots, x_n)$, удовлетворяющий функциональным условиям $\sum_{j=1}^n a_{ij}x_j \leq b_i$, ограничениям $x_i \geq 0$ и достав-

ляющий максимум линейной форме $F(\mathbf{x}) = \sum_{j=1}^n c_j x_j$.

В двойственной задаче требуется найти вектор, выражающий стоимостную цену расхода ресурса $\mathbf{y} = (y_1, y_2, \dots, y_m)$, удовлетворяющий функциональным условиям $\sum_{i=1}^m a_{ij}y_i \leq c_j$, ограничениям $y_j \geq 0$ и мини-

муму линейной формы $Z(\mathbf{y}) = \sum_{i=1}^m b_i y_i$. Все допустимые решения образуют область определения задачи

линейного программирования. Допустимое решение, оптимизирующее целевую функцию $F(\mathbf{x})$ или $Z(\mathbf{y})$, является оптимальным планом задачи.

Интервальная математическая модель ЗЛП задается интервальными параметрами. К их числу относятся интервалы значений коэффициентов технологической матрицы \mathbf{a}_{ij} и вектора правых частей $\mathbf{b} = (b_1, b_2, \dots, b_m)$ и аналогичных коэффициентов двойственной задачи, которые определяют погрешность входных данных.

Причины, порождающие погрешности входных параметров, могут быть самыми различными. Однако в качестве основных можно назвать: изменение качества сырья, удорожание или удешевление производства одного или нескольких видов продукции; модификации технологических коэффициентов, переоценки, повышение требований к качеству и многие другие причины. В свете продекларированных требований достоверность результатов решения задачи оптимизации может соблюдаться, если при формализации задачи учитываются входные параметры системы в интервальном виде с вариациями коэффициентов матрицы, компонент вектора ограничений и коэффициентов ЦФ или их совместного воздействия.

Множества допустимых решений

Нахождение решения стандартной ЗЛП состоит в определении угловых точек (вершин) выпуклого множества G , представляющего собой многогранник с границей dG , состоящей из плоских граней, полученного как оболочка непустого пересечения конечного числа полупространств. Каждое точечное неравенство ЗЛП геометрически определяет полуплоскость с границей $a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n \leq b_i$, $i = 1, m$. Условия неотрицательности соблюдаются, когда все $x_i \geq 0$. В том случае, если область G — ограниченный замкнутый многогранник, линейная функция достигает минимума в одной из вершин, а допустимыми решениями служат все вершины многогранника.

В геометрическом представлении точки пересечения четырех полуплоскостей обусловлены двумя границами коэффициентов уравнений и компонент их правых частей. При использовании унифицированной методики формируются выпуклые многогранники с вершинами в точках пересечения интервальных полуплоскостей, определяющих допустимые решения. Вершины, в которых достигается интервальный минимум (максимум) оптимизируемой функции, отвечают определению максимальных по включению (неулучшаемых) внутренних оценок множества решений интервальной системы и являются координатами интервального допустимого плана ЗЛП.

Подход к нахождению допустимого интервального плана распадается на решение двух угловых точечных систем с матрицами $\mathbf{A} = [\mathbf{A}_d^-, \mathbf{A}_d^+]$, совместно определяющих допустимое множество решений:

$$\begin{aligned} \Sigma_{\forall \exists} &= \{x \in \mathbf{R}^n | (\forall \mathbf{A}_d^- \in \mathbf{A}) (\exists \mathbf{b} \in \mathbf{b}) (\mathbf{A}_d^- x \subseteq \mathbf{b})\}, \\ \Sigma_{\forall \exists} &= \{x \in \mathbf{R}^n | (\forall \mathbf{A}_d^+ \in \mathbf{A}) (\exists \mathbf{b} \in \mathbf{b}) (\mathbf{A}_d^+ x \subseteq \mathbf{b})\}. \end{aligned} \quad (3)$$

Факт того, что вектор x удовлетворяет допустимому множеству решений (3), определяет оптимальный допустимый план, так как для каждой матрицы $A_d^-, A_d^+ \in \mathbf{A}$ существует соответствующее ей значение правой части \underline{b} , $\bar{b} \in \mathbf{b}$ и выполняется включение $Ax \subseteq b$.

Интервальная постановка двойственных задач линейного программирования

В интервальном виде постановка задачи заключается в оптимизации ЦФ $\min(\max) F = \sum_{j=1}^n c_j x_j$. Параметры неравенств-ограничений и коэффициенты функции F являются заданными интервальными величинами. Оптимизация F рассматривается в ограниченной области, определяемой системой неравенств:

$$\begin{cases} \sum_{j=1}^n a_{ij} x_j \leq b_i, \\ x_j \geq 0; \quad j = \overline{1, n}; \quad i = \overline{1, m}, \end{cases} \quad (4)$$

где $\mathbf{A} = \{a_{ij}\}$, $\mathbf{A} \in \mathbb{R}^{m \times n}$, $\mathbf{b} = \{b_i\}$, $\mathbf{b} \in \mathbb{R}^m$, $i = \overline{1, m}$ и $\mathbf{c} \in \mathbb{R}^n$, $\mathbf{c} = \{c_j\}$, $j = \overline{1, n}$.

Для описания двойственной задачи формируются тождественные уравнения с оптимизируемой функцией цели Z : $\max(\min) Z = \sum_{j=1}^n b_j y_j$

$$\begin{cases} \sum_{j=1}^n a_{ij}^T y_j \leq c_i, \\ y_j \geq 0; \quad j = \overline{1, n}; \quad i = \overline{1, m}, \end{cases} \quad (5)$$

где $\mathbf{A}^T = \{a_{ij}^T\}$, $\mathbf{A}^T \in \mathbb{R}^{m \times n}$, $\mathbf{c} = \{c_i\}$, $\mathbf{c} \in \mathbb{R}^m$, $i = \overline{1, m}$ и $\mathbf{b} \in \mathbb{R}^n$, $\mathbf{b} = \{b_j\}$, $j = \overline{1, n}$.

При соблюдении корректности задачи вариации внешних параметров (интервалов их изменения) не должны вызывать нарушения устойчивости решения. Предлагаемый алгоритм определения допустимого оптимального множества решений интервальной ЗЛП строится на реализации решения трех задач. Первая задача — определение номинального точечного плана x_0 допустимого оптимального решения; вторая задача — внешняя оценка множества решений (4), (5) в окрестности x_0 ; третья задача — внутренняя оценка множества решений (4), (5).

Определение x_0 проводится с использованием симплекс-метода, встроенного в программные коды системы MATLAB.

Внешней оценкой множества решений систем интервальных равенств (4) и (5) является решение точечных систем, сформированных на основании унифицированной методики, разработанной автором и представленной в предыдущих работах [10]:

$$\begin{cases} A^- x \leq b^+ \rightarrow (x^-, x^+), \\ A^+ x \leq b^- \rightarrow (x^-, x^+), \end{cases} \\ x = [x^-, x^+], \quad x^- = (x_1^-, x_2^-, \dots, x_n^-)^T, \quad (6) \\ x^+ = (x_1^+, x_2^+, \dots, x_n^+)^T,$$

$$\begin{cases} A^T y \geq c^+ \\ A^T y \geq c^- \end{cases} \rightarrow (y^-, y^+),$$

$$y = [y^-, y^+], \quad y^- = (y_1^-, y_2^-, \dots, y_n^-)^T, \quad (7) \\ y^+ = (y_1^+, y_2^+, \dots, y_n^+)^T.$$

Найденные компоненты векторов двойной размерности решения $x = [x^-, x^+]$ и $y = [y^-, y^+]$ систем (6) и (7) являются внешней оценкой объединенного множества решений в угловых точках пересечения двойных гиперполос.

Допустимое оптимальное решение определяется на основе решения дуальной интервальной системы, сконструированной на основании найденного оптимального плана x_0 . Допустимым оптимальным решением (базисом, планом) ЗЛП является интервальный вектор (брус), который удовлетворяет соответствующей системе неравенств и реализует экстремум ЦФ. Следует определить максимальный интервальный вектор, включающий допустимое множество решений.

Таким образом, интервальное решение стандартной задачи оптимизации представляется как интервальное расширение допустимого оптимального точечного решения, полученного на базе точечной системы уравнений. Предполагается, что вокруг найденной угловой точки оптимального допустимого решения x_0 существует выпуклый многогранник непустого пересечения конечного числа полупространств интервальных полос неравенств, образующих выпуклое многогранное непустое множество.

Определение оптимального допустимого плана каждой из сопряженных задач проводится при выполнении следующей последовательности операций:

1) симплекс-методом находится оптимальный план x_0 для номинальной системы с первоначально заданными элементами матрицы и ограничениями-неравенствами;

2) на основании полученного вектора оптимального плана x_0 фиксируется соответствующая ему квадратная матрица и правые части, определяющие точечную систему равенств;

3) задаются погрешности ε и δ для элементов полученной точечной матрицы и правых частей, и формируется интервальная система с нижними и верхними границами $\mathbf{A} = [\underline{\mathbf{A}}, \overline{\mathbf{A}}]$, $\mathbf{b} = [\underline{\mathbf{b}}, \overline{\mathbf{b}}]$;

4) в соответствии с унифицированной методикой из интервальной матрицы конструируются две оптимальные угловые точечные матрицы с заданными границами правых частей $A^-, A^+ \in \mathbb{R}^{n \times n}$;

5) определяется вектор двойной размерности $x = \{x^-, x^+\}$, $x^-, x^+ \in \mathbb{R}^n$ с максимальными отклонениями в положительном и отрицательном направлениях, соответствующий внешней оценке множества решения;

6) из решения дуальной системы с интервальной матрицей $\text{dualA} = [\overline{\mathbf{A}}, \underline{\mathbf{A}}]$ находится вектор $x_d = \{x_d^-, x_d^+\}$, определяющий внутреннее оценивание множества решений, соответствующее оптимальному допустимому решению ЗЛП [11];

7) проверяется допустимость найденного интервального оптимального вектора $x_d = \{x_d^-, x_d^+\}$ в интервале $b = [b^-, b^+]$ его подстановкой в системы (6) и (7) с оптимальными угловыми матрицами A^- и A^+ ;

8) из скалярного произведения интервальных компонент оптимального вектора $x_d = \{x_d^-, x_d^+\}$ на интервальные коэффициенты целевой функции $c = [c, \bar{c}]$ находится двухсторонний экстремум целевой функции $F = \sum_{j=1}^n c_j x_{dj}$.

Порядок выполнения операций по определению оптимального допустимого плана двойственной задачи аналогичен алгоритму прямой задачи.

Примеры решения двойственных интервальных задач линейного программирования

Пример 1. Рассматривается задача с ограничениями, содержащими две переменные. Математическая модель описывается системой неравенств размерности 5×2 с двумя переменными x_1, x_2 для точечных значений коэффициентов.

$$Z(x) = c_1 x_1 + c_2 x_2 \rightarrow \max;$$

$$\begin{cases} -1x_1 + 3x_2 \leq 10 \\ 1x_1 + 1x_2 \leq 6 \\ 1x_1 - 1x_2 \leq 2 \\ 1x_1 + 3x_2 \geq 6 \\ 2x_1 + 1x_2 \geq 4, \\ x_1, x_2 \geq 0. \end{cases} \quad (8)$$

Из решения номинальной системы с помощью системы MATLAB получен оптимальный допустимый план, выраженный вектором $x_0 = \begin{pmatrix} 2 \\ 4 \end{pmatrix}$.

На втором этапе задаются погрешности коэффициентов матрицы, правых частей и коэффициентов ЦФ системы (8): $\varepsilon = 0,05, \delta = 0,12$.

С использованием унифицированной методики [10] находим объединенное решение задачи с интервальными коэффициентами:

$$x = (x^-, x^+) = \begin{pmatrix} 1,104 & 2,989 \\ 3,311 & 4,776 \end{pmatrix},$$

определяющее прямоугольную область внешнего интервального оценивания в окрестности ранее найденного оптимального плана $x_0 = \begin{pmatrix} 2 \\ 4 \end{pmatrix}$.

Область оптимального решения находится из дуальной интервальной матрицы системы уравнений:

$$\text{dual}A = \text{dual} \left[\begin{matrix} [-1 - \varepsilon, -1 + \varepsilon], [3 - \varepsilon, 3 + \varepsilon] \\ [1 - \varepsilon, 1 + \varepsilon], [1 - \varepsilon, 1 + \varepsilon] \end{matrix} \right] \rightarrow$$

$$\rightarrow \bar{A} = \begin{pmatrix} -1 + \varepsilon, & 3 + \varepsilon \\ 1 + \varepsilon, & 1 + \varepsilon \end{pmatrix}, \underline{A} = \begin{pmatrix} -1 - \varepsilon, & 3 - \varepsilon \\ 1 - \varepsilon, & 1 - \varepsilon \end{pmatrix}$$

с заданными погрешностями элементов матрицы. Найденный брус

$$x_d = (x_d^-, x_d^+) = \begin{pmatrix} 1,805 & 2,185 \\ 3,753 & 4,215 \end{pmatrix}$$

определяет максимальное внутреннее оценивание решения, соответствующее оптимальному допустимому интервальному плану.

Двойственную задачу, сопряженную задаче (8), можно записать как

$$Z(y) = 10y_1 + 6y_2 + 2y_3 + 6y_4 + 4y_5 \rightarrow \min;$$

$$\begin{cases} -y_1 + 1y_2 + 1y_3 + 1y_4 + 2y_5 \geq 1 \\ 3y_1 + 1y_2 - y_3 + 3y_4 + 1y_5 \geq 2, \end{cases} \quad (9)$$

где $y_1, y_2 \geq 0, \varepsilon = 0,12, \delta = 0,05$.

Номинальное, объединенное и оптимальное допустимое решения имеют следующие значения:

$$y_0 = \begin{pmatrix} 0,25 \\ 1,25 \end{pmatrix}, y = (y^-, y^+) = \begin{pmatrix} 0,152 & 0,358 \\ 1,032 & 1,496 \end{pmatrix},$$

$$y_d = (y_d^-, y_d^+) = \begin{pmatrix} 0,225 & 0,273 \\ 1,175 & 1,314 \end{pmatrix}.$$

В табл. 1 и 2 показаны результаты расчетов объединенного и оптимального допустимого решений двойственных задач (8) и (9).

Относительные отклонения интервальной ЦФ по отношению к среднему значению составляют ~18 %, что превосходит заданные погрешности элементов матрицы и компонент вектора правой части, равные $\varepsilon = 5 \%$ и $\delta = 12 \%$.

Таблица 1

Номинальное решение x_0 , максимальные отклонения x^-, x^+ и допустимые оптимальные значения интервального вектора (оптимального допустимого плана) $x_d = (x_d^-, x_d^+)$ прямой задачи (8)

x	x_0	x^-	x^+	x_d^-	x_d^+
x_1	2	1,104	2,989	1,805	2,185
x_2	4	3,311	4,776	3,753	4,215
$F(x)$	10	6,8	14,05	8,19	11,88

Таблица 2

Номинальное решение y_0 , максимальные отклонения y^-, y^+ и допустимые оптимальные значения вектора решения y_d^-, y_d^+ (оптимального допустимого плана сопряженной задачи)

y	y_0	y^-	y^+	y_d^-	y_d^+
y_1	0,250	0,152	0,358	0,225	0,273
y_2	1,250	1,032	1,496	1,175	1,314
$Z(y)$	10	6,8	14,06	8,19	11,88

С целью наглядного представления полученного допустимого оптимального решения задачи оптимизации рассматривается внешняя оценка множества решений интервальной системы в угловой точке оптимизации и соотношения внутренней и внешних оценок между собой.

На рис. 1 показаны прямоугольники, внешние из которых соответствуют оценке объединенного решения интервальных систем uni , внутренние $opt\ tol$ — оптимальной допустимой оценке (двумерными брусами) для прямой и сопряженной задач. Точка в центре прямоугольников отображает оптимальный допустимый план номинальной системы.

Пример 2. Рассмотрим ЗЛП с ограничениями, содержащими три переменные. Математическая модель средней системы размерности 4×3 с тремя переменными x_1, x_2, x_3 и с оптимизируемой ЦФ $F(x)$ описывается номинальной системой неравенств:

$$F(x) = 3x_1 + 4x_2 + 2x_3 \rightarrow \max;$$

$$\begin{cases} x_1 + 2x_2 + x_3 \leq 18 \\ 2x_1 + x_2 + x_3 \leq 16 \\ x_1 + x_2 \leq 8 \\ x_2 + x_3 \leq 6 \\ x_j \geq 0, j = 1, 2, 3 \\ x_1, x_2, x_3 \geq 0. \end{cases} \quad (10)$$

Сопряженная система записывается в следующем виде:

$$Z(y) = 16y_1 + 8y_2 + 6y_3 \rightarrow \min;$$

$$\begin{cases} 2y_1 + y_2 + y_3 \geq 3 \\ y_1 + y_2 \geq 4 \\ y_2 + y_3 \geq 2 \\ y_j \geq 0, j = 1, 2, 3. \end{cases} \quad (11)$$

Применение симплекс-метода к точечным матрицам (10) и (11) приводит исходную прямоугольную систему к системам с квадратными матрицами, определяющими оптимальные допустимые решения x_0 и y_0 .

Для систем (10), (11) задаются погрешности коэффициентов матриц, правых частей и коэффициентов ЦФ $\varepsilon = 0,05, \delta = 0,08$.

На основании систем уравнений с квадратными матрицами конструируются системы с интервальными матрицами для прямой и сопряженных задач. Из каждой интервальной матрицы формализуются две граничные оптимальные угловые матрицы и дуальные матрицы $dual(A) = (\bar{A}, \underline{A})$. Сформированные системы позволяют получить внешнюю и внутреннюю оценки допустимого решения.

Найденные решения для двойственных задач примут следующий вид:

$$x_0 = \begin{pmatrix} 5 \\ 3 \\ 3 \end{pmatrix}, \quad x = (x^-, x^+) = \begin{pmatrix} 3,752 & 6,321 \\ 1,345 & 5,022 \\ 1,196 & 4,852 \end{pmatrix},$$

$$x_d = (x_d^-, x_d^+) = \begin{pmatrix} 4,842 & 5,143 \\ 2,905 & 3,086 \\ 2,905 & 3,086 \end{pmatrix},$$

$$y_0 = \begin{pmatrix} 0,5 \\ 2,0 \\ 1,5 \end{pmatrix}, \quad y_d = (y_d^-, y_d^+) = \begin{pmatrix} 0,085 & 1,030 \\ 1,349 & 2,611 \\ 1,033 & 2,065 \end{pmatrix},$$

$$y_d = (y_d^-, y_d^+) = \begin{pmatrix} 0,484 & 0,514 \\ 1,937 & 2,057 \\ 1,453 & 1,543 \end{pmatrix}.$$

Результаты вычислений векторов обеих двойственных задач с соответствующими точечными и интервальными значениями ЦФ приведены в табл. 3, 4.

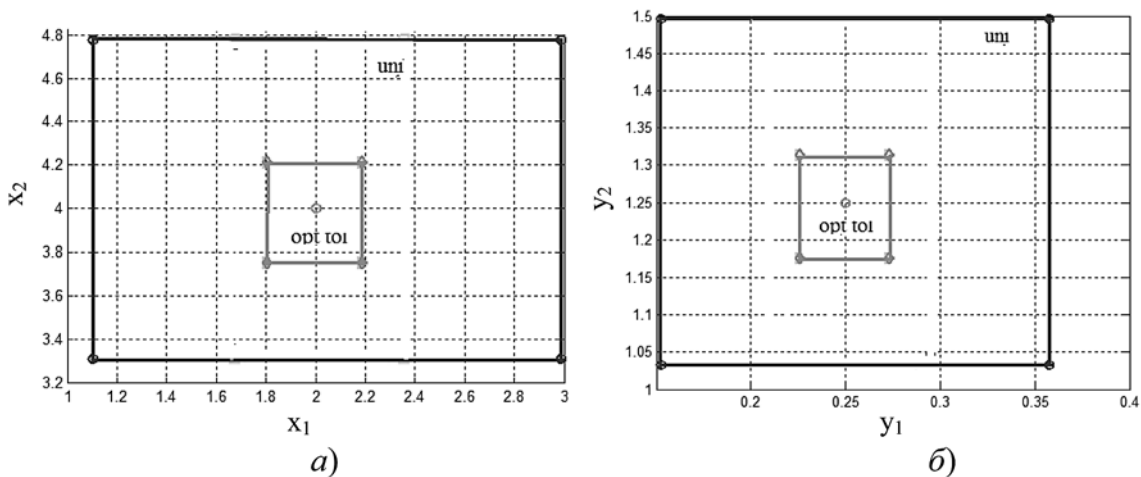


Рис. 1. Внешняя и максимальная внутренняя оболочки множества решений объединенного uni и оптимально допустимого плана $opt\ tol$ плоской ЗЛП:

a — прямой задачи; b — двойственной задачи

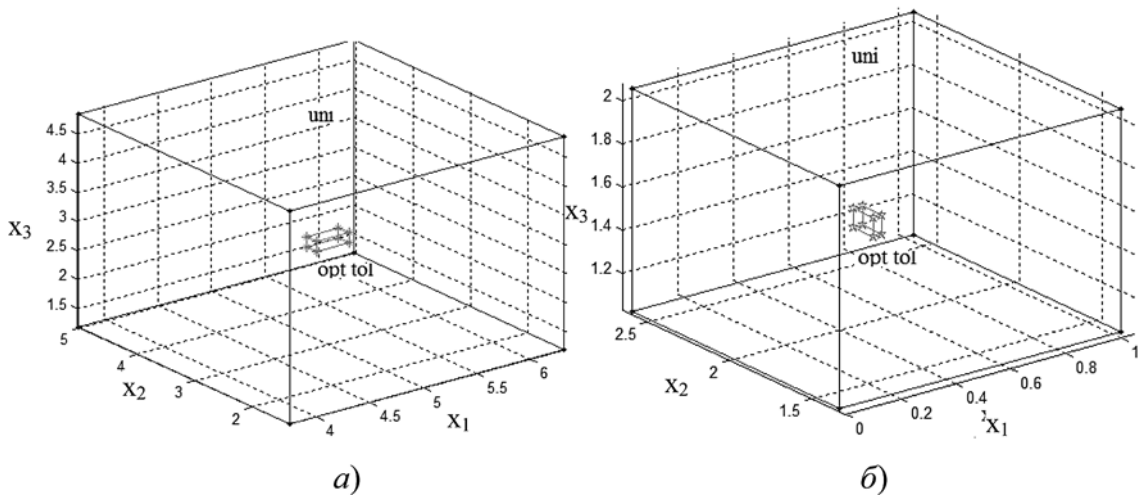


Рис. 2. Внешняя и максимальная внутренняя оболочки множества решений объединенного uni и оптимально допустимого плана opt tol ЗЛП с тремя неизвестными:

a — прямой задачи; *б* — двойственной задачи

Таблица 3

Допустимое решение точечной ЗЛП, объединенное множество решений, оптимальные допустимые планы и соответствующие им ЦФ

x	x_0	x^-	x^+	x_d^-	x_d^+
x_1	5	3,752	6,321	4,842	5,143
x_2	3	1,345	5,022	2,905	3,086
x_3	3	1,196	4,852	2,905	3,086
$F(x)$	33	17,51	52,65	29,40	36,66

По теореме двойственности точечные значения $\max Z(y) = \min F(x)$ для сопряженных задач совпадают. Для допустимых оптимальных интервальных планов необходимо, чтобы их значения также совпадали. Из сравнения вычисленных значений ЦФ в табл. 1 и 2, а также в табл. 3 и 4 видно их полное совпадение в диапазоне заданных погрешностей ϵ и δ .

На рис. 2 показаны параллелограммы, внешние из которых соответствуют, как и в случае двумерного случая, оценке объединенного решения интервальных систем uni, а внутренние параллелограммы opt tol — оптимальной допустимой оценке для прямой и сопряженной задач.

Заключение

Представленные в настоящей публикации результаты позволяют сделать следующие выводы.

- Предложен новый подход, ранее не применяемый в практике интервального оценивания оптимального допустимого решения ЗЛП с неопределенными коэффициентами, как интервальное расширение полученного точечного оптимального допустимого решения.
- Предложен способ проверки интервального оптимального допустимого решения ЗЛП на осно-

Таблица 4

Допустимое решение точечной ЗЛП, внешняя оценка и оптимальный допустимый план (внутренняя оценка) объединенного множества решений и соответствующие им ЦФ

y	y_0	y^-	y^+	y_d^-	y_d^+
y_1	0,5	0,085	1,030	0,484	0,514
y_2	2,0	1,349	2,611	1,937	2,057
y_3	1,5	1,033	2,065	1,453	1,543
$Z(y)$	33	17,51	52,65	29,40	36,66

вании формализованных уравнений, которые получаются по ранее разработанной автором методике. Контроль принадлежности интервального вектора допустимому оптимальному плану, не выводящему решение задачи за пределы задаваемых ограничений, проверяется его подстановкой в системы с оптимальными угловыми матрицами.

- Показано, что при условии существования решения точечной ЗЛП возможно определение интервальных оптимальных допустимых планов симметричных сопряженных ЗЛП в пределах задаваемых погрешностей коэффициентов неравенств, правых частей и коэффициентов ЦФ.
- Рассмотренные примеры подтвердили, что интервал допустимых оптимальных планов, представляющий собой нижнюю, внутреннюю оценку, значительно более узкий по сравнению с внешней оценкой объединенного решения приведенной квадратной системы.
- Определено, что интервал изменения ЦФ под воздействием неопределенности параметров системы превосходит задаваемые входные погрешности.
- Вычисление интервальных ЦФ обеспечивает двухстороннюю гарантированную оценку решения оптимальной задачи ЛП.

Список литературы

1. Давыдов Д. В. Интервальные методы и модели принятия решений в экономике: Автореф. дис. д-ра экон. наук. Владивосток, 2009. 42 с.
2. Уоткинс Д. Основы матричных вычислений. Перевод с английского. М.: Бином, 2006. 684 с.
3. Шарый С. П., Джаныбеков Б. С. Модификация подхода Оеттли для оптимального оценивания множеств АЕ-решений интервальных линейных систем // Вычислительные технологии. 2005. Т. 10, № 3. С. 117—126.
4. Замков О. О., Черемных Ю. А., Тостопяченко А. В. Математические методы в экономике. М.: Изд-во МГУ, 2001. 368 с.
5. Хлебалин Н. А. Аналитический метод синтеза регуляторов в условиях неопределенности параметров объекта // Аналитические методы синтеза регуляторов. Межвуз. научн. сб. Саратов: Саратов. политехнический институт, 1981. С. 107—123.

6. Rohn J. NP-hardness results for linear algebraic problems with interval data // Topics in Validated Numerics / Eds. J. Herzberger. Amsterdam: North-Holland, 1994. P. 463—471.
7. Лакеев А. В., Носков С. И. Описание множества решений линейного уравнения с интервально заданными оператором и правой частью // Доклады РАН. 1993. Т. 330, № 4. С. 430—433.
8. Шарая И. А. Строение допустимого множества решений интервальной линейной системы // Вычислительные технологии. 2005. Т. 10, № 5. С. 103—119.
9. Шарый С. П., Шарая И. А. Распознавание разрешимости интервальных уравнений и его приложения к анализу данных // Вычислительные технологии. 2013. Т. 18, № 3. С. 80—109.
10. Иванова К. Ф. Унификация точечных алгебраических методов внешней оценки решений интервальных систем // Программная инженерия. 2016. Т. 7, № 4. С. 181—190.
11. Kupriyanova L. Inner estimation of the united solution set of interval linear algebraic system // Reliable Computing. 1995. Vol. 1, No. 1. P. 15—31.

Interval Admissible Solution of a Linear Programming Problem in MATLAB System

K. F. Ivanova, Klara.I2010@yandex.ru, Saint Petersburg State University, Saint Petersburg, 195252, Russian Federation

Corresponding author:

Ivanova Klara F., Doctorant, Saint Petersburg State University, Saint Petersburg, 195252, Russian Federation,
E-mail: Klara.I2010@yandex.ru

Received on July 11, 2017

Accepted on July 20, 2017

In article the solution of a linear programming problem (LPP) in presence of uncertainty of parameters of target function and inequality restrictions by its reduction to the usual determined systems is offered. The algorithm of calculation of a mod-eled optimal problem is based on use of the simplex method which is built into the MATLAB system. The interval solution of a standard problem of optimization is represented as interval expansion of the optimal pointwise plan received on the basis of a nominal canonical set of equations. With use of interval representation of factors of the received nominal system the interval problem of an interior estimation of set of solutions of interval system of the linear algebraic equations (ISLAE) is solved and the interval vector (brus) corresponding to the optimal admissible plan is calculated.

The admissibility of the discovered interval optimal plan is checked by its substitution in the pointwise sets of equations defining an exterior estimation of set of solutions ISLAE constructed on the basis of interval canonical system. For optimal angular matrices of the formalized pointwise systems their product on the discovered vector should not overstep the bounds of an interval vector of restrictions.

The purpose of this research is an objective deriving of boundaries of the optimal target function, not exceeding the limit of possible deviations called by uncertainty of a problem data and conforming to inequality restrictions.

Keywords: a linear programming problem, canonical system, exterior, interior estimations of solutions set ISLAE, the optimal admissible plan, expansion of a pointwise solution, inequality restrictions

For citation:

Ivanova K. F. Interval Admissible Solution of a Linear Programming Problem in MATLAB System, *Programmnyaya Ingeneriya*, 2017, vol. 8, no. 10, pp. 463—469.

DOI: 10.17587/prin.8.463-469

References

1. Davydov D. V. *Interval'nye metody i modeli prinyatiya resheniy v ehkonomike* (Interval methods and decision making models in economy), Vladivostok, 2009, 42 p. (in Russian).
2. Uotkins D. *Osnovy matrichnykh vychisleniy* (Bases of matrix evaluations), Moscow, Binom, 2006, 684 p. (in Russian).
3. Sharyj S. P., Dzhanibekov B. S. Modifikatsiya podhoda Oettli dlya optimal'nogo ocenivaniya mnozhestv AE-resheniy interval'nykh lineynykh sistem (Modification of the approach of Oettli for an optimum estimation of sets AE-solutions of interval linear systems), *Vychislitel'nye tehnologii*, 2005, vol. 10, no. 3, pp. 117—127 (in Russian).
4. Zamkov O. O., Cheremnykh Yu. A., Tostopyatenko A. V. *Matematicheskie metody v ehkonomike* (Mathematical methods in economy), Moscow, Izd-vo MSU, 2001, 368 p. (in Russian).
5. Hlebalin N. A. Analiticheskij metod sinteza regulatorov v usloviyah neopredelyonnosti parametrov ob'ekta. (Analytical method of synthesis of regulators in the conditions of uncertainty of parameters of a subject), *Analytical methods of synthesis of regulators*, Saratov, Sarat. politekhnicheskij institut, 1981, pp. 107—123 (in Russian).

6. Rohn J. NP-hardness results for linear algebraic problems with interval data, *Topics in Validated Numerics* / Eds. J. Herzberger, Amsterdam, North-Holland, 1994, pp. 3463—471.

7. Lakeev A. V., Noskov S. I. Opisanie mnozhestva resheniy linejnogo uravneniya s interval'no zadannymi operatorom i pravoy chast'yu (Description of set of solutions of a linear equation with interval operator and the right-hand part), *Doklady RAN*, 1993, vol. 330, no. 4, pp. 430—433 (in Russian).

8. Sharyj S. P., Sharaya I. A. Raspoznavanie razreshimosti interval'nykh uravneniy i ego prilozheniya k analizu dannykh. (Recognition of resolvability of the interval equations and its application to the analysis of the data), *Vychislitel'nye tehnologii*, 2013, vol. 18, no. 3, pp. 80—109 (in Russian).

9. Sharaya I. A. Stroenie dopustimogo mnozhestva resheniy interval'noj lineinoj sistemy (Structure of admissible set of solutions of interval linear system), *Vychislitel'nye tehnologii*, 2005, vol.10, no. 5, pp. 102—119 (in Russian).

10. Ivanova K. F. Unifikatsiya tochechnykh algebraicheskikh metodik vneshnej ocenki resheniy interval'nykh sistem (Unification of Point Algebraic Techniques of an Exterior estimation of Set Solutions Interval Systems), *Programmnyaya Ingeneriya*, 2016, vol. 7, no. 4, pp. 181—189 (in Russian).

11. Kupriyanova L. Inner estimation of the united solution set of interval linear algebraic system, *Reliable Computing*, 1995, vol. 1, no. 1, pp. 15—31.

А. В. Шокуров, канд. физ.-мат. наук, ст. науч. сотр., **К. А. Кривчикова**, студент,
e-mail: k.krivchikova@gmail.com, МГУ имени М. В. Ломоносова

Оценка абсолютной погрешности при поиске внешних параметров группы видеокамер в приложении к задаче локализации и сопровождения объекта

Рассмотрена задача калибровки группы видеокамер как один из этапов локализации и сопровождения целевого объекта. Предложен алгоритм поиска внешних параметров группы камер на основе нескольких существующих подходов. Рассмотрена задача оценки точности при калибровке, рассчитаны оценки абсолютных погрешностей для трех наиболее распространенных алгоритмов поиска фундаментальной матрицы, а также оценка погрешности предложенного алгоритма поиска внешних параметров сети видеокамер. Представлены результаты программной реализации нескольких шагов предложенного алгоритма на синтезированной сцене.

Ключевые слова: внешние параметры видеокамеры, калибровка видеокамеры, оценка абсолютной погрешности, фундаментальная матрица, сопровождение объекта

Актуальность исследования и постановка задачи

Задачи определения положения (локализации) объектов, а также отслеживания и сопровождения движущихся объектов по изображениям с нескольких видеокамер всегда представляли интерес для специалистов в области компьютерного зрения. Однако особенную актуальность эти задачи приобрели в последнее десятилетие, когда вычислительные ресурсы компьютерной техники стали позволять реализовать их аппаратно-программное решение в режиме реального времени. Системы автоматического обнаружения, локализации и сопровождения движущихся объектов с использованием сети видеокамер широко применяют в рамках видеонаблюдения в задачах обеспечения безопасности, а также во многих других прикладных областях. К их числу, например, относятся создание 3D-моделей, в частности, создание интерфейсов дополненной реальности [1]; робототехника [2]; оптимизация использования пространств с высоким трафиком через создание моделей передвижения автомобилей или людей. При этом применение видеокамер с возможностью управления углами поворота и наклона, а также изменения фокусного расстояния (так называемых Pan-Tilt-Zoom или PTZ-камер) или только с изменением фокусного расстояния (так называемых масштабируемых камер) в составе группы камер позволяет обеспечить высокую детализацию наблюдаемых небольших или отдаленных объектов с сохранением возможности наблюдения за сценой в целом за счет других камер в составе группы.

Использование нескольких видеокамер вместо одной при решении рассматриваемых задач позволяет расширить область обзора, получить более полную информацию о положении наблюдаемых объектов в трехмерном пространстве. Такой подход дает возможность избежать выхода движущегося объекта за пределы кадра сопровождающей его камеры (например, при высокой угловой скорости объекта или при сложной траектории движения) за счет использования информации о положении объекта на основе не только видеоизображения сопровождающей камеры, но и данных от других камер. Однако точность локализации и сопровождения объектов на изображениях, полученных с различных камер, в значительной степени зависит от точности калибровки системы таких камер, т. е. определения их внешних параметров — положения и поворота в пространстве. В настоящей работе предложен вариант алгоритма автоматического определения таких внешних параметров группы камер и проведена аналитическая оценка абсолютных погрешностей этих параметров, которые получаются при использовании алгоритма.

Практическая значимость получения таких оценок обусловлена возможностью расчета оценки погрешности местоположения наблюдаемого объекта на основе полученных абсолютных погрешностей внешних параметров камер, а учет погрешности местоположения объекта, в свою очередь, позволяет избежать потери объекта системой слежения в случае неточного определения параметров PTZ-видеокамер и масштабируемых видеокамер. В ряде таких случаев указанная неточность может быть компенсирована увеличением угла обзора сопровождающей камеры

с учетом полученной оценки погрешности. Оценки погрешности местоположения объекта, полученные аналитически для различных комбинаций размера объекта и его положения, позволяют также оценить ограничения применимости рассматриваемой системы видеокамер для локализации и слежения за объектами различного вида.

В рамках настоящего исследования определение внешних параметров видеокамер и оценка абсолютных погрешностей таких параметров выполняется в следующих условиях:

- используется группа видеокамер, состоящая более чем из одной видеокамеры и содержащая не менее одной PTZ-видеокамеры, причем области наблюдения видеокамер попарно пересекаются;

- наблюдение проводится во внешней среде (вне помещения, например, на улице), причем нет физического доступа к наблюдаемой сцене, либо действия в рамках доступа к наблюдаемой сцене трудно осуществить на практике (например, специализированные мишени для взаимной калибровки камер затруднительно разместить в пересечении областей видимости, либо подготовка мишеней крупного размера для калибровки удаленных друг от друга камер не может быть выполнена);

- обнаружение и сопровождение ведется только для одного целевого объекта;

- известны внутренние параметры всех видеокамер с заранее определенной погрешностью, включая фокусное расстояние, параметры растяжения пикселей вдоль осей изображения, координаты точки пересечения плоскости изображения и направления обзора видеокамеры (так называемой главной точки), коэффициенты дисторсии;

- видеокамеры неподвижны во внешней среде.

Такой набор условий выбран для моделирования ситуации, близкой к решению задач видеонаблюдения на крупных объектах вне помещений с ограниченным уровнем контроля за наблюдаемой сценой. Такие ситуации представляют интерес, например, для объектов транспортной инфраструктуры, для которых актуальны задачи локализации и сопровождения отдельных объектов в поле зрения видеокамер. При этом следует отметить, что использование более точных методов калибровки группы камер может быть затруднено вследствие особенностей внешней среды.

С учетом изложенного выше, целью исследования, результаты которого представлены в настоящей работе, является нахождение метода, осуществляющего поиск внешних параметров сети видеокамер, а также расчет абсолютной погрешности найденных параметров с учетом всех условий общей постановки задачи. Внешние параметры видеокамер — поворот и положение в трехмерном пространстве — рассматриваются в рамках решаемой задачи в виде матрицы поворота и вектора переноса. Для основных объектов исследования — видеокамер — с определенными погрешностями заранее известны их внутренние параметры, а также приближение их координат в на-

блюдаемой сцене во внешней среде (например, на основе данных геопозиционирования).

Далее в работе представлены известные подходы к определению внешних параметров видеокамер, изложена исследуемая модель видеокамеры, предложен метод автоматического определения внешних параметров для группы камер на основе комбинации нескольких известных подходов, получены оценки абсолютной погрешности внешних параметров. Кроме того, для отдельных элементов метода приведены результаты программных экспериментов на имитационной модели.

Подходы к поиску внешних параметров видеокамер

Известные подходы к поиску внешних параметров видеокамер условно делят на две группы: с использованием специального объекта, так называемой калибровочной мишени, и без него. Калибровочная мишень — это объект, находящийся в поле зрения калибруемых видеокамер, про который заранее известны некоторые свойства: геометрическая форма, либо координаты его отдельных точек в трехмерной системе координат, либо траектория перемещения. Использование калибровочной мишени позволяет находить внешние параметры более точно. Обзоры с более подробными результатами экспериментальной проверки представлены в работах [3, 4].

К первой группе относятся классические работы Zhang [5] и Sturm, Maybank [6], в которых в качестве калибровочной мишени используется плоский объект, раскрашенный по типу шахматной доски. К настоящему времени известно несколько модификаций данного метода. К их числу относятся, например, работа Deng, Lu и Shao [7], где для решения подзадачи оптимизации авторы скомбинировали метод роя частиц (*Particle Swarm Optimization*) [8], и метод дифференциальной эволюции (*Differential Evolution*) [9]. В работах Ueshiba и Tomita [10] используется плоский объект с тремя отмеченными точками; Oyamada, Fallavollita и Navab [11] используют плоскость с точками, поставленными в случайном порядке. Существуют подходы, в которых в качестве калибровочной мишени используется поверхность вращения [12]. Обзор методов из этой группы представлен в работе [11]. Вместе с тем для решения поставленной в настоящей работе задачи перечисленные методы не подходят, так как условия не предполагают доступа к сцене наблюдения.

Вторая группа методов не предполагает использование калибровочного объекта. В англоязычной литературе такие методы называют *self-calibration*. Классическим считается метод, предложенный Faugeras, Luong, Maybank [13], он основывается на нахождении фундаментальной матрицы и решении уравнений Круппа. Его современные модификации — методы Svoboda, Martinec, Pajdla [1], Senior, Hampapur, Lu [14], метод А. Шокурова, А. Хропова, Д. Иванова [15], в которых проводится поиск фундаментальной матрицы,

а также метод поиска трифокального тензора [16], используемый в работе А. Хропова, А. Шокурова, В. Лемпицкого и Д. Иванова [17]. Во всех перечисленных подходах искомая матрица или тензор вычисляется через минимизацию ошибки обратного проецирования (*reprojection error*) с учетом дополнительных ограничений.

Существует также группа методов одновременной калибровки и построения 3D-модели (*Simultaneous Localization and Mapping, SLAM*), например, проект MonoSLAM [2]. В таких подходах видеокамеры подвижны, их траектории неизвестны, а калибровка осуществляется как один из этапов 3D-реконструкции. Следует отметить проект PhotoTourism [18], в котором фундаментальная матрица используется только для исключения выбросов, а внешние параметры находят из минимизации ошибки обратного проецирования, выраженной через внутренние и внешние параметры видеокамер. Этот проект, как и ImagiCAD [19], используется для автоматического создания 3D-модели по набору фотографий, сделанных с разных ракурсов.

Отметим, что все современные подходы к решению задачи калибровки сводятся к оптимизации, так как имеют дело с переопределенными системами линейных и, реже, нелинейных уравнений [20].

С учетом перечисленных выше особенностей различных подходов в рамках настоящего исследования перспективной представляется комбинация методов второй группы, в частности, схемы, предложенной Svoboda, Martinec, Pajdla [1] и алгоритма калибровки PTZ-камер, предложенного Senior, Namprapur, Lu [14], а также отдельных способов исключения выбросов при построении фундаментальных матриц.

Модель видеокамеры и основные понятия

В области компьютерного зрения обычно используют однородные координаты в проективном пространстве \mathbf{RP}^n , $n = 2, 3$ [16], видеокамера рассматривается как отображение из \mathbf{RP}^3 в \mathbf{RP}^2 . Будем использовать CCD-модель видеокамеры (видеокамеры с матрицей типа Charged Coupled Device), в которой видеокамера отождествляется с верхнетреугольной матрицей размера 3×4 :

$$\mathbf{K} = \begin{pmatrix} \alpha_x & 0 & x_0 & 0 \\ 0 & \alpha_y & y_0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

где (x_0, y_0) — координаты главной точки; $\alpha_x = f \cdot m_x$, $\alpha_y = f \cdot m_y$; f — фокусное расстояние объектива видеокамеры, m_x и m_y — растяжение (в пикселях) вдоль осей изображения. Параметры растяжения позволяют смоделировать ситуацию, когда пиксели не квадратны, что отвечает реальным особенностям видеокамер. Стоит пояснить, что фокусное расстояние является физическим параметром оптической системы видеокамеры. Пропорциональное изменение фокусного расстояния соответствует масштабированию изображения, получаемого с видеокамеры.

Кроме того, в рассматриваемой модели видеокамеры будем учитывать так называемую дисторсию — искажение изображения по мере удаленности от центра:

$$x' = x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) + 2p_1 xy + p_2(r^2 + 2x^2),$$

$$y' = y(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) + p_1(r^2 + 2y^2) + 2p_2 xy,$$

где (x, y) — координаты точки на изображении при отсутствии дисторсии, т. е. при проекции проективных пространств; (x', y') — искаженные координаты, $r = \sqrt{x^2 + y^2}$; k_1, k_2, k_3 — коэффициенты радиальной дисторсии; p_1, p_2 — коэффициенты тангенциальной дисторсии; начало координат — в центре изображения.

Пять коэффициентов дисторсии и четыре параметра матрицы \mathbf{K} составляют в рамках настоящей работы набор внутренних параметров видеокамеры.

Система координат изображения, связанного с камерой, и фиксированная объемлющая система координат связаны аффинным преобразованием: $\mathbf{R} \in \mathbf{O}(3)$ — матрица поворота; $\mathbf{t} \in \mathbf{R}^3$ — вектор переноса. Матрица \mathbf{R} и вектор \mathbf{t} составляют набор внешних параметров видеокамеры.

Таким образом, отображение проективных пространств $\mathbf{P}: \mathbf{RP}^3 \rightarrow \mathbf{RP}^2$, осуществляемое видеокамерой, представляется в виде матрицы 4×3 :

$$\mathbf{P} = \mathbf{K}[\mathbf{R} \mid -\mathbf{Rt}].$$

Далее будем использовать понятие особых точек (локальных точечных особенностей, *feature points*). В компьютерном зрении они используются для восстановления внутренних и внешних параметров видеокамер, а также для 3D-реконструкции путем триангуляции. Согласно работе [21], можно дать предельное далее определение особых точек сцены и изображения.

Под особой точкой сцены понимается такая точка сцены, лежащая на плоском участке поверхности сцены, изображение окрестности которой можно отличить от изображений окрестностей всех других точек сцены из некоторой другой окрестности этой точки. Особая точка изображения — это такая точка изображения, окрестность которой можно отличить от окрестности любой другой точки изображения в некоторой другой окрестности этой точки.

Возникает задача сопоставления особых точек двух и более изображений, которые являются проекциями одной и той же особой точки сцены. Неверно определенные соответствия точек изображений будем называть выбросами.

Описание метода

В предыдущем разделе представлены обзоры современных методов нахождения внешних параметров группы видеокамер [3, 4]. По результатам их анализа был сделан вывод, что для решения поставленной

выше задачи подходит класс методов, основанных на последовательном исключении выбросов с помощью фундаментальной матрицы и ошибки обратного проецирования. Для их реализации используются алгоритмы, описанные в работах Svoboda, Martines, Pajdla [1], А. Шокурова, А. Хропова, Д. Иванова [15], в цикле работ по проекту Photo Tourism авторов Snavely, Seitz, Szeliski (основная статья [18]), а также в работе Senior, Hamprug, Lu [14].

Несмотря на то что во многих рассмотренных выше исследованиях решается похожая задача, рассмотренные методы не в полной мере применимы к решению задач настоящей работы. В частности, известные методы, применяемые по отдельности, могут не удовлетворять некоторым дополнительным условиям, в особенности следующим: неизвестны 3D-координаты особых точек; нельзя выделить плоские объекты; большое количество выбросов из-за наблюдения во внешней среде; возможно, очень малая область пересечения областей обзоров некоторых пар видеокамер. По этой причине в настоящей работе используется комбинация указанных выше методов.

Представим краткое описание алгоритма, предлагаемого для решения поставленной задачи. Отметим, что пункты, отмеченные далее звездочкой, являются необязательными и вводятся для повышения точности.

Шаг 1. Поиск особых точек, описание каждой особой точки с помощью вектора дескриптора (n -мерный вектор из \mathbf{R}^n или $\{0, 1\}^n$, значение n зависит от выбранного типа дескриптора).

Шаг 2. Нахождение попарного соответствия особых точек с изображений от разных видеокамер с помощью минимизации расстояния между дескрипторами.

Шаг 3. Исключение выбросов с помощью алгоритма RANSAC по фундаментальной матрице, поиск которой проводится по 8-точечному алгоритму [16]. Отметим, что на данном шаге находится подмножество множества всех пар особых точек, имеющее два свойства:

а) всем элементам этого подмножества соответствует одна фундаментальная матрица \mathbf{F} ; это означает, что \mathbf{F} и каждая пара особых точек $(\mathbf{x}, \mathbf{x}')$ в подмножестве удовлетворяют соотношению: $\mathbf{x}'\mathbf{F}\mathbf{x} = 0$ с некоторой заранее заданной погрешностью;

б) найденное подмножество максимально по мощности среди всех, обладающих свойством (а).

Пары точек, лежащие вне найденного подмножества, объявляются выбросами и исключаются из дальнейшего рассмотрения.

Комментируя неформально, сначала выполняется поиск такой матрицы, которая была бы "фундаментальной" для большинства пар особых точек, т. е. для которой большинство особых точек удовлетворяло бы соотношению $\mathbf{x}'\mathbf{F}\mathbf{x} = 0$. Такая матрица хорошо приближает настоящую фундаментальную матрицу для данной пары изображений [16]. Как следствие, выбросами являются те соответствия, у которых точ-

ки внутри пары не переводятся друг в друга найденной матрицей.

Шаг 4. Вычисление глубин (расстояний от плоскости изображения вдоль направляющего вектора камеры), соответствующих особым точкам, с помощью фундаментальной матрицы, полученной на предыдущем шаге [1]. Вычисление внешних параметров видеокамер с помощью найденных глубин и известных внутренних параметров.

Шаг 5*. Изменение параметров PTZ-видеокамер и масштабируемых видеокамер, аналитический расчет новых внутренних и внешних параметров [14], расчет матрицы аффинного преобразования точек между изображениями с разных конфигураций одной и той же видеокамеры.

Шаг 6*. Поиск соответствия особых точек между изображениями с одной видеокамеры с помощью аффинного преобразования, найденного на предыдущем шаге, а также расстояния между дескрипторами [14].

Шаг 7*. Повторение шагов 1—6 с учетом уже полученных результатов, составление "треков": для каждой 3D-точки, которой соответствует как минимум две особые точки на изображениях, запись в отдельное множество координат всех ее проекций на изображения при всех конфигурациях видеокамер, а также индикатора видимости данной точки на изображениях.

Шаг 8. Исключение выбросов через ошибку обратной проекции: для каждой 3D-точки осуществляется триангуляция: 3D-координаты точки находятся через преобразование, обратное проекции, которую осуществляет видеокамера. Затем для каждой конфигурации видеокамер вычисляется величина $ind(i, j) \| \mathbf{x}_{i,j} - \mathbf{K}_i \mathbf{R}_i (\mathbf{X}_j - \mathbf{t}_i) \|^2$, где i — номер видеокамеры; j — номер точки, $ind(i, j)$ — индикатор видимости j -й точки i -й видеокамерой; \mathbf{X}_j — j -я особая точка сцены; $\mathbf{x}_{i,j}$ — ее проекция на изображение с i -й видеокамеры. Выбросами объявляются те особые точки, для которых значение данной величины больше заранее выбранного порогового значения.

Шаг 9. Повторение шагов 3—4, 8, пока есть выбросы.

Шаг 10. Повторение шагов 3—4, 8, 9, пока ошибка обратной проекции больше заданного порогового значения.

В качестве дескриптора на шаге 1 может использоваться SIFT-дескриптор [22], а в качестве начальных параметров для вычисления внешних параметров на шаге 4 — параметры, рассчитанные исходя из известных данных геопозиционирования (например, GPS-координат) сети видеокамер и внутренних параметров видеокамер.

Аналитические оценки погрешностей

В контексте рассматриваемых задач аналитический расчет оценки сверху косвенных погрешностей измерений проводится для внешних параметров $\mathbf{R}_i, \mathbf{t}_i, i = 1, \dots, M$ группы из M видеокамер. Здесь \mathbf{R}_i — матрица

поворота, \mathbf{t}_i — вектор переноса системы координат, отвечающей i -й видеокамере.

Расчет проводили по алгоритму, представленному в работах [23, 24]. В частности, были использованы следующие способы расчета для явных и неявных зависимостей величин. Если величина g явно зависит от x , то абсолютная погрешность g рассчитывается по дифференциальной формуле:

$$\Delta g = \frac{\partial g}{\partial x} \Delta x.$$

Эта формула следует из разложения функции g в ряд Тейлора в окрестности точки x :

$$g(x + \Delta x) = g(x) + \frac{\partial g}{\partial x} \Delta x.$$

Если есть неявная зависимость, например, на каком-то шаге алгоритма решается задача $G(x, \mathbf{F}) \rightarrow \min_{\mathbf{F}}$, то погрешность $\Delta \mathbf{F}$ рассчитывается по формуле

$$G(x + \Delta x, \mathbf{F} + \Delta \mathbf{F}) = \varepsilon,$$

где $\varepsilon = G(x, \mathbf{F})$ — точность решения задачи $G(x, \mathbf{F}) \rightarrow \min_{\mathbf{F}}$.

Расчет оценок погрешностей проводится сначала для фундаментальной матрицы, а затем полученные соотношения используются непосредственно для определения оценок погрешностей внешних параметров видеокамер.

Фундаментальная матрица: краткое описание методов поиска и оценка погрешности

Согласно схеме предложенного алгоритма поиска внешних параметров камер, первый вычислительный шаг — это расчет фундаментальной матрицы для каждой пары видеокамер. Согласно обзору Armatange и Salvi [25], существующие подходы к нахождению фундаментальной матрицы можно разделить на прямые (например, RANSAC и Least-Median-Squares) и итерационные, а итерационные, в свою очередь, на две группы по виду функционала, который используется для поиска минимума. Отметим, что для аналитической оценки не имеет значения, как именно ведется оптимизация. Значение имеет только вид функционала, для которого решается задача оптимизации. Далее рассмотрим сначала итерационные (п. I), а затем прямые (п. II) методы.

Пусть выбрана фиксированная пара видеокамер, \mathbf{F} — фундаментальная матрица перехода от первой камеры ко второй камере, найдено N особых точек: $\mathbf{x}_i \in \mathbf{R}^2$, $i = 1, \dots, N$ — особые точки изображения с первой видеокамеры, $\mathbf{x}'_i \in \mathbf{R}^2$, $i = 1, \dots, N$ — особые точки изображения со второй видеокамеры. Особые точки из разных изображений с одинаковыми индексами — это точки, между которыми установлено соответствие. Далее будем использовать терминологию из теории погрешностей. Найденные соот-

ветствия особых точек изображений будем называть измеренными, а точные проекции точек сцены на плоскости изображений будем называть точными соответствиями.

Пусть соответствие особых точек измерено с погрешностью $\Delta \mathbf{x} \in \mathbf{R}^2$. Тогда во всех расчетных формулах фигурируют не точные значения \mathbf{x}_i , \mathbf{x}'_i , а измеренные значения $\mathbf{x}_i + \Delta \mathbf{x}$, $\mathbf{x}'_i + \Delta \mathbf{x}$. Если \mathbf{F} — точная фундаментальная матрица, то $\mathbf{F} + \Delta \mathbf{F}$ — фундаментальная матрица, которая найдена по измеренным соответствиям особых точек, $\Delta \mathbf{F}$ — абсолютная погрешность.

I. Суть итерационных методов заключается в следующем. Пусть $G(\mathbf{x}, \mathbf{F})$ — функционал, действующий на \mathbf{F} . Будем решать задачу минимизации $G(\mathbf{x}, \mathbf{F}) \rightarrow \min_{\mathbf{F}}$.

Пусть $\gamma = G(\mathbf{x} + \Delta \mathbf{x}, \mathbf{F} + \Delta \mathbf{F})$, $\varepsilon = G(\mathbf{x}, \mathbf{F})$. Параметр γ характеризует точность решения задачи оптимизации для измеренных \mathbf{x}_i , \mathbf{x}'_i , параметр ε — для точных значений \mathbf{x}_i , \mathbf{x}'_i и заданного γ .

Рассмотрим функционал, соответствующий расстоянию между особыми точками и их образами при отображении с помощью фундаментальной матрицы. В зависимости от формы записи этого функционала удобно использовать разные итерационные методы. Следуя обзору [25], рассмотрим две формы записи. Далее для каждого вида функционала представлены оценки сверху абсолютной погрешности фундаментальной матрицы.

1. Рассмотрим функционал

$$G(\mathbf{x}, \mathbf{F}) = \sum_{i=1}^N \text{dist}^2(\mathbf{x}_i, \mathbf{F}\mathbf{x}'_i) + \text{dist}^2(\mathbf{x}'_i, \mathbf{F}\mathbf{x}_i).$$

Для каждой особой точки в паре \mathbf{x}_i , \mathbf{x}'_i осуществляется проекция на другое изображение пары с помощью фундаментальной матрицы. В результате на каждом изображении для каждой особой точки сцены есть два образа. Функционал представляет собой сумму квадратов расстояний между этими образами для каждого изображения в паре.

Для функционала в таком виде можно применять стандартные методы минимизации, например, метод Ньютона, метод Левенберга—Марквардта.

Из выражений $\gamma = G(\mathbf{x} + \Delta \mathbf{x}, \mathbf{F} + \Delta \mathbf{F})$, $\varepsilon = G(\mathbf{x}, \mathbf{F})$ можно оценить норму абсолютной ошибки фундаментальной матрицы. Получим:

$$\|\Delta \mathbf{F}\| \leq \frac{2\|\Delta \mathbf{x}\|\|\mathbf{F} - \mathbf{I}\|}{\sqrt{\|\mathbf{x}\|^2 + \|\mathbf{x}'\|^2}} + \frac{\varepsilon - \gamma}{N\sqrt{\varepsilon}} \frac{1}{\sqrt{2(\|\mathbf{x}\|^2 + \|\mathbf{x}'\|^2)}},$$

где $\|\mathbf{x}\| = \max_{i=1, \dots, N} \|\mathbf{x}_i\|$, $\|\mathbf{x}'\| = \max_{i=1, \dots, N} \|\mathbf{x}'_i\|$, \mathbf{I} — единичная матрица 3×3 .

Второе слагаемое в правой части мало, и если его значением пренебречь, то получим

$$\|\Delta \mathbf{F}\| \leq \frac{2\|\Delta \mathbf{x}\|\|\mathbf{F} - \mathbf{I}\|}{\sqrt{\|\mathbf{x}\|^2 + \|\mathbf{x}'\|^2}}.$$

2. Рассмотрим функционал $G(\mathbf{x}, \mathbf{F}) = \sum_{i=1}^N \frac{(x_i^T \mathbf{F} \mathbf{x}'_i)^2}{g_i^2}$,

$$g_i^2 = (\mathbf{F}^T \mathbf{x}'_1)^2 + (\mathbf{F}^T \mathbf{x}'_2)^2 + (\mathbf{F}^T \mathbf{x}'_1)^2 + (\mathbf{F}^T \mathbf{x}'_2)^2,$$

где $(\mathbf{v})_j$ обозначена j -я компонента вектора \mathbf{v} .

Данный функционал можно получить из предыдущего [25]. Выбор такого вида функционала обусловлен удобством использования градиентных методов минимизации, а также метода максимального правдоподобия (*Fundamental Numerical Scheme*, *Constrained Fundamental Numerical Scheme*).

Проводя аналогичные рассуждения, получим оценку для $\|\Delta \mathbf{F}\|$:

$$\|\Delta \mathbf{F}\|^2 \leq \|\mathbf{F}\| - \frac{c^2}{\|\mathbf{x}\|^2 \|\mathbf{x}'\|^2} + \|\mathbf{F}\|^2 \times \left(\frac{c}{4} \|\mathbf{F}\| \left(\frac{\|\mathbf{x}\|}{\|\mathbf{x}'\|} + 1 + \max \left\{ \|\mathbf{x}\|, \frac{\|\mathbf{x}'\|^2}{\|\mathbf{x}\|} \right\} \right) + \|\Delta \mathbf{x}\| \left(\frac{1}{\|\mathbf{x}'\|} + \frac{1}{\|\mathbf{x}\|} \right) \right)^2,$$

где $c = \max_{i=1, \dots, N} |\langle \mathbf{x}_i, \mathbf{F} \mathbf{x}'_i \rangle| \approx \frac{(\gamma - \varepsilon) \max_{i=1, \dots, N} g_i^2}{N}$, $\|\mathbf{x}\|$ и

$\|\mathbf{x}'\|$ определяются аналогично предыдущему случаю. Заметим, что константа c мала, и если ее значением пренебречь, то формула примет вид

$$\|\Delta \mathbf{F}\|^2 \leq \|\mathbf{F}\| + \|\mathbf{F}\|^2 \|\Delta \mathbf{x}\|^2 \left(\frac{1}{\|\mathbf{x}'\|} + \frac{1}{\|\mathbf{x}\|} \right)^2.$$

II. Для прямых методов обычно используется 8-точечный метод [16], в котором решается линейная система уравнений:

$$\mathbf{x}'_i^T \mathbf{F} \mathbf{x}'_i = 0, \quad i = 1, \dots, 8.$$

Восьми точек достаточно, так как фундаментальная матрица \mathbf{F} определена с точностью до умножения на ненулевой скаляр, значит, можно положить $F_{33} = 1$. Отметим, что фундаментальная матрица имеет ранг 2, а значит, для ее определения достаточно семи точек, но широкое практическое применение получил именно 8-точечный метод.

Абсолютная погрешность $\|\Delta \mathbf{F}\|$ находится из выражения для абсолютной погрешности для вектора, составленного из восьми компонент \mathbf{F} :

$$\mathbf{f} = (F_{11}, F_{12}, F_{13}, F_{21}, F_{22}, F_{23}, F_{31}, F_{32})^T;$$

$$\Delta \mathbf{f} = -\mathbf{A}^{-1} \Delta \mathbf{A} \mathbf{f};$$

$$\mathbf{A} = \begin{pmatrix} x_1 x'_1 & x_1 y'_1 & x_1 & y_1 x'_1 & y_1 y'_1 & y_1 & x'_1 & y'_1 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ x_8 x'_8 & x_8 y'_8 & x_8 & y_8 x'_8 & y_8 y'_8 & y_8 & x'_8 & y'_8 \end{pmatrix}.$$

Итоговые оценки погрешности алгоритма

Пусть N — число особых точек; M — число видеокамер. Расчет погрешностей проводился согласно методу, предложенному Svoboda, Martines и Pajdla [1]. В нем существенно используются фундаментальные

матрицы: в качестве первого шага алгоритма рассчитываются глубины λ_{ij} для каждой i -й пары точек в j -й паре видеокамер. В результате была получена следующая оценка для нормы абсолютных погрешностей векторов-строк матрицы поворота $\mathbf{R}_1, \mathbf{R}_2, \mathbf{R}_3$ и компонент вектора переноса \mathbf{t} :

$$\|\Delta \mathbf{R}_1\| \leq \frac{3 \|\Delta \mathbf{P}\| \kappa^3(\mathbf{P})}{f} \leq \frac{3N \|\Delta \mathbf{x}\| \|\lambda\|_{\mathbf{IR}^M} \kappa^3(\mathbf{P})}{f}.$$

Точно такая же оценка верна и для $\mathbf{R}_2, \mathbf{R}_3$, компонент t_1, t_2, t_3 вектора переноса. В формуле использованы следующие обозначения:

f — фокусное расстояние;

\mathbf{P} — матрица, полученная путем сингулярного разложения матрицы измерений

$$\mathbf{W} = \begin{pmatrix} \lambda_{11} x_{11} & \dots & \lambda_{1N} x_{1N} \\ \dots & \dots & \dots \\ \lambda_{M1} x_{M1} & \dots & \lambda_{MN} x_{MN} \end{pmatrix} = \begin{pmatrix} P_1 \\ \dots \\ P_M \end{pmatrix} (X_1 \dots X_N),$$

здесь x_{ij} — проекция j -й особой точки на изображение с i -й камеры, X_j — j -я особая точка сцены;

$\kappa(\mathbf{P})$ — число обусловленности матрицы \mathbf{P} ;

$$\|\lambda^i\|_{\mathbf{IR}^M} = \sqrt{\sum_{k=1}^M (\lambda_k^i)^2} \text{ — норма столбца, составлен-$$

ного из глубин λ_k^i , $k = 1, \dots, M$, $i = 1, \dots, N$;

$$\|\lambda\|_{\mathbf{IR}^M} = \max_{i=1, \dots, N} \|\lambda^i\|_{\mathbf{IR}^M}.$$

Эксперименты на имитационной модели

Для проверки на практике полученных аналитических оценок был частично программно реализован предложенный выше алгоритм, проведен эксперимент со смоделированными группами видеокамер и сценой. С этой целью была создана 3D-модель сцены во внешней среде — фрагмента аэродрома — и сети видеокамер с различными параметрами в программе Blender. Алгоритм был реализован на языке программирования Python. Была использована модель SIFT особых точек, в качестве дескрипторов были использованы векторы длины 128 также из модели SIFT [22]. Для оценки корректности полученных соответствий использовали визуальную оценку. В будущем планируется использовать свойства 3D-модели для точной оценки корректности.

Особое внимание при реализации предлагаемых алгоритмов уделялось исключению выбросов, в частности, с использованием алгоритма RANSAC. Отметим, что исключение выбросов — это та часть метода, от которой напрямую зависит корректность и точность результата, а погрешность нельзя оценить теоретически. Эта часть единственная, которая зависит от особенностей наблюдаемой сцены и свойств видеокамер.

До исключения выбросов было найдено 1237 соответствий особых точек, после исключения осталось

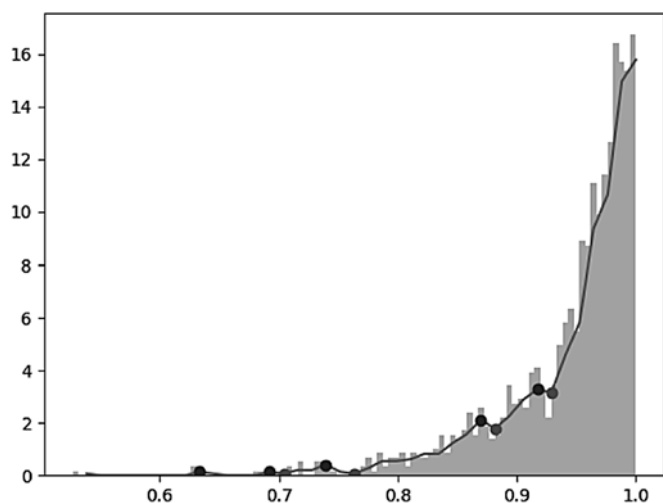


Рис. 1.

89 соответствий. Однако по визуальной оценке лишь несколько из оставшихся соответствий являются корректными.

В ходе испытания были исследованы и другие методы исключения выбросов, которые используются в современном компьютерном зрении. Наиболее надежным среди них считается метод пороговой фильтрации гистограммы (*ratio-test*), предложенный Lowe [22]. На рис. 1 и рис. 2 показаны гистограммы сравнения двух кадров с разных видеокамер в 3D-модели. По оси абсцисс отложены отношения расстояний между особой точкой и двумя ближайшими к ней особыми точками другого изображения из пары. В качестве метрики используется евклидово расстояние в пространстве векторов-дескрипторов \mathbf{R}^{128} . По оси ординат отложено количество особых точек с данным значением отношения расстояний.

Ломаная линия — приближение по формуле Sturges, отдельные точки — локальные минимумы и максимумы.

В рассматриваемом нами случае для изображений с разных видеокамер гистограмма имеет лишь один выраженный пик (см. рис. 1, рис. 2), и часто приближается монотонной функцией, что делает невозможным использование данного метода. Приближение гистограммы гладкими функциями также не позволяет выделить пороговые значения (см. рис. 1). Однако вычисления показали, что даже при наличии двух пиков после проведения фильтрации остается небольшое количество корректных соответствий.

На рис. 3 можно видеть соответствия особых точек до исключения выбросов, на рис. 4 — после исключения. В качестве порогового значения для фильтрации гистограммы взят второй локальный минимум (см. рис. 2). Линиями соединены особые точки, между которыми установлено соответствие. Отдельными кружками отмечены отдельные особые точки, к которым не удалось найти соответствия на другом изображении.

Задача предполагает возможность ручной обработки данных, в том числе на этапе поиска соответствия особых точек. Применялась следующая модификация: до запуска алгоритма на паре изображений вручную размечаются проекции одной и той же прямой линии. Выбросами объявляются те соответствия особых точек, у которых одна точка лежит в окрестности отмеченной линии, а другая лежит вне окрестности линии. Были проведены тесты с одной, двумя и тремя линиями, но гистограмма незначительно изменила вид после такой предобработки (рис. 2).

На рис. 2, а показана гистограмма до разметки линий, ломаная линия 1 — приближение гистограммы ломаной по формуле Sturges [26, 27], линией 2 показано приближение распределения гауссовым распре-

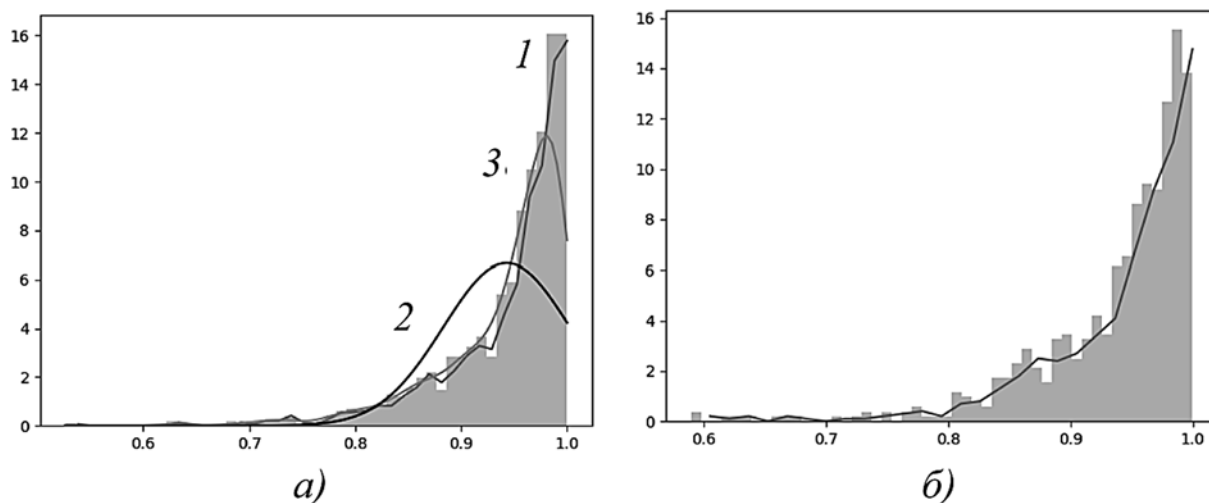


Рис. 2.

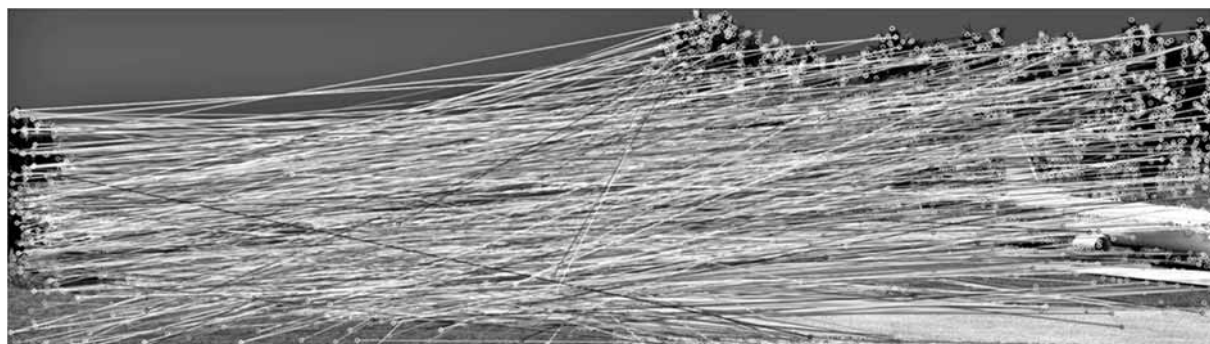


Рис. 3.

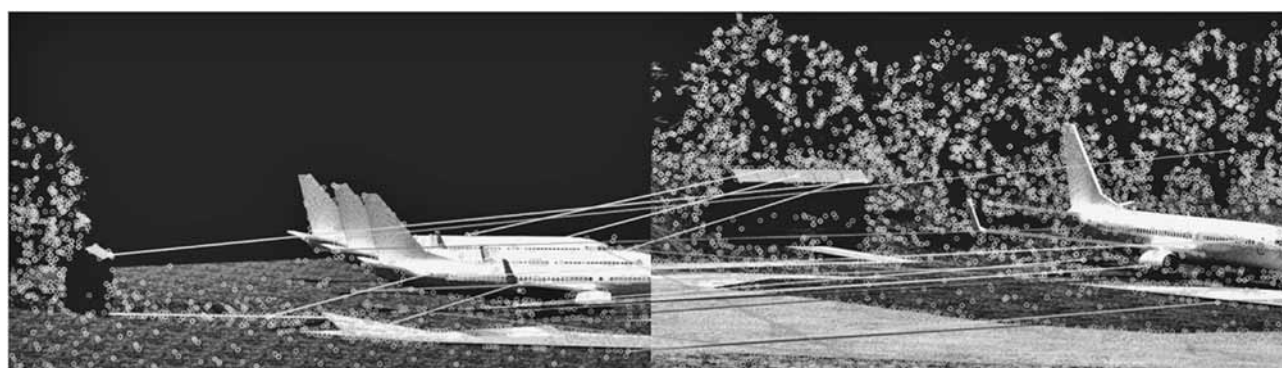


Рис. 4.

делением (*kernel density estimation*) по методу Scott [28], а линией 3 — приближение по оценке максимального правдоподобия. На рис. 2, б показана гистограмма после разметки трех линий. Для всех методов была использована готовая реализация в составе пакетов *numpy* и *scipy*. Процесс разметки линий и определение их окрестностей показан на рис. 5.

Таким образом, одним из направлений дальнейшей доработки алгоритма является адаптация алгоритмов исключения выбросов к рассматриваемой задаче.



Рис. 5.

Выводы

В настоящей работе аналитически рассчитаны оценки сверху абсолютных погрешностей определения внешних параметров сети видеочамер, а также погрешностей определения фундаментальной матрицы между парой видеочамер тремя наиболее распространенными способами. Предложен алгоритм практической проверки вычисленных погрешностей, который скомбинирован из существующих методов определения внешних параметров сети видеочамер. Реализовано и апробировано в рамках имитационной модели несколько шагов работы алгоритма и намечены пути его модификации.

В качестве дальнейших работ в выбранном направлении исследования планируется автоматизировать проверку корректности сопоставления особых точек изображений, найти подходящий метод исключения выбросов, реализовать его и весь алгоритм определения внешних параметров группы видеочамер и проверить полученные теоретические оценки погрешностей на практике.

В будущем планируется также исследовать влияние взаимного расположения видеочамер в группе на

точность определения внешних параметров, а также найти критические расположения, при которых работа алгоритма невозможна.

В качестве другого возможного направления дальнейших исследований стоит отметить поиск трифакторного тензора — аналога фундаментальной матрицы при рассмотрении не пар видеокамер, а троек. Кроме того, представляется перспективным использовать дополнительные данные о наблюдаемой сцене, в частности, информацию о форме, размерах и взаимном расположении специфических объектов, которые можно было бы использовать в качестве калибровочных мишеней.

Список литературы

1. **Svoboda T., Martinec D., Pajdla T.** A convenient multicamera self-calibration for virtual environments // PRESENCE: Teleoperators and Virtual Environments. 2005. Vol. 14, No. 4. P. 407–422.
2. **Davison A., Reid I., Molton N. D., Stasse O.** MonoSLAM: Realtime single camera SLAM // IEEE Trans. Pattern Analysis and Machine Intelligence, 2007. Vol. 29, No. 6. P. 1052 — 1067.
3. **Dima E., Sjöström M., Olsson R.** Assessment of Multi-Camera Calibration Algorithms for Two-Dimensional Camera Arrays Relative to Ground Truth Position and Direction // In 3D Video (3DTV-CON), 2016 3DTV-Conference: The True Vision - Capture, Transmission and Display of 3D Video, Hamburg, Germany. 2016. P. 1–4.
4. **Hodlmoser M., Zollner H., Kampel M.** An Evaluation of Camera Calibration Methods Using Digital Low Cost Cameras // In Computer Vision Winter Workshop 2010, February 2010. URL: https://www.researchgate.net/profile/Martin_Kampel/publication/228774958_An_Evaluation_of_Camera_Calibration_Methods_Using_Digital_Low_Cost_Cameras/links/0fcfd50cb86569a4ed000000.pdf
5. **Zhang Z.** Flexible Camera Calibration by Viewing a Plane from Unknown Orientations // In Proc. 7th International Conference on Computer Vision. 1999. Vol. 1. P. 666–673.
6. **Sturm P. F., Maybank S. J.** On Plane-Based Camera Calibration: A General Algorithm, Singularities, Applications // In Proc. Computer Vision and Pattern Recognition, 1999. Vol. 1. P. 432–437.
7. **Deng L., Lu G., Shao Y. et al.** A novel camera calibration technique based on differential evolution particle swarm optimization algorithm // Neurocomputing. 2016. Vol. 174. P. 456–465.
8. **Kennedy J., Eberhart R.** Particle Swarm Optimization // Proceedings of IEEE International Conference on Neural Networks. 1995. Vol. IV. P. 1942–1948.
9. **Storn R., Price K.** Differential evolution — a simple and efficient heuristic for global optimization over continuous spaces // Journal of Global Optimization. 1997. Vol. 11. P. 341–359.
10. **Ueshiba T., Tomita F.** Plane-based calibration algorithm for multi-camera systems via factorization of homography matrices // In Proc. Int. Conf. Computer Vision, 2003. Vol. 2. P. 966–973.
11. **Oyamada Y., Fallavollita P., Navab N.** Single camera calibration using partially visible calibration objects based on random dots marker tracking algorithm // In IEEE and ACM International Symposium on Mixed and Augmented Reality, Workshop on Tracking Methods and Applications (TMA), Atlanta, USA, 2012. P. 1–7.
12. **Wong K. Y., Mendonca R. S. P., Cipolla R.** Camera calibration from surfaces of revolution // IEEE Trans. Pattern Anal. Machine Intell. 2003. Vol. 25. P. 147–161.
13. **Faugeras O., Luong Q.-T., Maybank S.** Camera self-calibration: Theory and experiments // In Proc. European Conference on Computer Vision. 1992. P. 321–334.
14. **Senior A., Hampapur A., Lu M.** Acquiring multiscale images by pan-tilt-zoom control and automatic multicamera calibration // In IEEE Workshop on Application of Computer Vision (WACV/MOTION'05). 2005. Vol. 1. P. 433–438.
15. **Shokurov A., Khropov A., Ivanov D.** Feature Tracking in Image and Video // Proc. of GraphiCon-2003. 2003. P. 177–179.
16. **Hartley R., Zisserman A.** Multiple View Geometry in computer vision. Cambridge University Press, 2003.
17. **Khropov A., Shokurov A., Lempitsky V., Ivanov D.** Reconstruction of projective and metric cameras for image triplets // Proc. of GraphiCon-2004. 2004. P. 143–146.
18. **Snively N., Seitz S. M., Szeliski R.** Photo Tourism: Exploring Photo Collections in 3D // ACM Transactions on Graphics. 2006. Vol. 25, No. 3. P. 835–846.
19. **Ivanov D., Lempitsky V., Khropov A., Shokurov A., Kuzmin Ye.** Computer Vision at CGG MSU // UK-RUSSIA WORKSHOP "PROACTIVE COMPUTING", University of Nizhny Novgorod, 2005. P. 44–58.
20. **Weng J., Cohen P., Herniou M.** Camera calibration with distortion models and accuracy evaluation // IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI. 1992. Vol. 14, No. 10. P. 965–980.
21. **Конущин А.** Слежение за точечными особенностями сцены // Компьютерная графика и мультимедиа. 2003. № 1 (5). URL: <http://cgm.computergraphics.ru/content/view/54> (дата обращения 05.03.2017).
22. **Lowe D. G.** Distinctive Image Features from Scale-Invariant Keypoints // In Proceedings of the 7th International Conference on Computer Vision, Kerkyra, Greece. 1999. P. 1150–1157.
23. **Корнев А. А., Чижонков Е. В.** Упражнения по численным методам. Часть II. М.: Изд-во ЦПИ при механико-математическом факультете МГУ им. М. В. Ломоносова. 2003. 200 с.
24. **Бирюков С. В.** Метрология: тексты лекций. Омск: ОмГТУ, 2000. 110 с.
25. **Armangue X., Salvi J.** Overall view regarding fundamental matrix estimation // Image Vis. Comput. 2003. Vol. 21, No. 2. P. 205–220.
26. **Sturges H. A.** The choice of a class interval // Journal of the American Statistical Association. 1926. Vol. 21. P. 65–66. doi: 10.1080/01621459.1926.10502161.
27. **Doane D. P.** Aesthetic frequency classification // American Statistician, 1976. Vol. 30. P. 181–183.
28. **Scott D. W.** Multivariate Density Estimation: Theory, Practice, and Visualization. New York, Chichester: John Wiley & Sons, 1992. 336 p.

Absolute Error Estimation when Reconstructing the Extrinsic Parameters of a Camera Group in Application to Object Localization and Tracking

A. V. Shokurov, shokurov.anton.v@yandex.ru, K. A. Krivchikova, k.krivchikova@gmail.com, Lomonosov Moscow State University, Moscow, 119234, Russian Federation

Corresponding author:

Krivchikova Kseniya A., Student, Lomonosov Moscow State University, Moscow, 119234, Russian Federation
E-mail: k.krivchikova@gmail.com

Received on June 28, 2017

Accepted on July 10, 2017

There are many approaches to object localization and tracking, and the most common way to achieve accurate results is to use a camera group instead of a single camera. Correct mapping between frames from distinct cameras helps to achieve more reliable tracking and more accurate object motion estimation, and calibration is the only way to find these mappings in practice.

In this article we propose a new technique of extrinsic self calibration based principally on [1, 12] and absolute error upper-estimates for extrinsic parameters and fundamental matrix reconstruction. We assume that the scene is located outdoor and there is no physical access to the scene (so it is not possible to use special calibration targets). We also assume that cameras' intrinsic parameters and global positioning data are known with some given error.

Error estimation improves accuracy of object localization and determines the range of applications of the whole tracking algorithm. Since methods of fundamental matrix reconstruction are very specific and depend on many additional constraints, we derive error estimates for three methods: two iterative and one robust. For each method the upper estimate of absolute error is calculated on the assumption that feature point projections are determined with fixed error Δx . The absolute error of extrinsic camera parameters is estimated as a function of the fundamental matrix and feature point projections.

Finally, we examine a few methods for removing outliers of feature matching on the synthetic data obtained by rendering an outdoor 3D scene from multiple viewpoints. Poor performance of these methods — Lowe ratio-test with several automated techniques of threshold determination and one novel half-automated technique — is discovered under certain conditions showing the need to improve the proposed algorithm and adapt other methods of finding outliers.

Keywords: external camera parameters, camera calibration, absolute error estimation, fundamental matrix, object tracking

For citation:

Shokurov A. V., Krivchikova K. A. Absolute Error Estimation when Reconstructing the Extrinsic Parameters of a Camera Group in Application to Object Localization and Tracking, *Programmnyaya Inzheneriya*, 2017, vol. 8, no. 10, pp. 470—480.

DOI: 10.17587/prin.8.470-480

References

1. Svoboda T., Martinec D., Pajdla T. A convenient multicamera self-calibration for virtual environments, *PRESENCE: Teleoperators and Virtual Environments*, 2005, vol. 14, no. 4, pp. 407—422.
2. Davison A., Reid I., Molton N. D., Stasse O. MonoSLAM: Realtime single camera SLAM, *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2007, vol. 29, no. 6, pp. 1062—1067.

3. Dima E., Sjostrom M., Olsson R. Assessment of Multi-Camera Calibration Algorithms for Two-Dimensional Camera Arrays Relative to Ground Truth Position and Direction, *3D Video (3DTV-CON), 2016 3DTV-Conference: The True Vision — Capture, Transmission and Display of 3D Video*, Hamburg, Germany, Jul. 4—8, 2016, pp. 1—4.

4. Hodlmoser M., Zollner H., Kampel M. An Evaluation of Camera Calibration Methods Using Digital Low Cost Cameras, *Computer Vision Winter Workshop 2010*, Nove Grade, Czech

- Republic, Feb. 3–5, 2010, available at: https://www.researchgate.net/profile/Martin_Kampel/publication/228774958_An_Evaluation_of_Camera_Calibration_Methods_Using_Digital_Low_Cost_Cameras/links/0fcfd50cb86569a4ed000000.pdf
5. **Zhang Z.** Flexible Camera Calibration by Viewing a Plane from Unknown Orientations, *Proc. 7th International Conference on Computer Vision*, 1999, vol. 1, pp. 666–673.
 6. **Sturm P. F., Maybank S. J.** On Plane-Based Camera Calibration: A General Algorithm, Singularities, Applications, *Proc. Computer Vision and Pattern Recognition*, 1999, vol. 1, pp. 432–437.
 7. **Deng L., Lu G., Shao Y., Fei M., Hu H.** A novel camera calibration technique based on differential evolution particle swarm optimization algorithm, *Neurocomputing*, 2016, vol. 174, pp. 456–465.
 8. **Kennedy J., Eberhart R.** Particle Swarm Optimization, *Proceedings of IEEE International Conference on Neural Networks*, 1995, vol. IV, pp. 1942–1948.
 9. **Storn R., Price K.** Differential evolution — a simple and efficient heuristic for global optimization over continuous spaces, *Journal of Global Optimization*, 1997, vol. 11, pp. 341–359.
 10. **Ueshiba T., Tomita F.** Plane-based calibration algorithm for multi-camera systems via factorization of homography matrices, *Proc. Int. Conf. Computer Vision*, 2003, vol. 2, pp. 966–973.
 11. **Oyamada Y., Fallavollita P., Navab N.** Single camera calibration using partially visible calibration objects based on random dots marker tracking algorithm, *IEEE and ACM International Symposium on Mixed and Augmented Reality, Workshop on Tracking Methods and Applications (TMA)*, Atlanta, USA. Nov. 5–8, 2012, pp. 1–7.
 12. **Wong K. Y., Mendonca R. S. P., Cipolla R.** Camera calibration from surfaces of revolution, *IEEE Trans. Pattern Anal. Machine Intell.*, 2003, vol. 25, pp. 147–161.
 13. **Faugeras O., Luong Q.-T., Maybank S.** Camera self-calibration: Theory and experiments, *Proc. European Conference on Computer Vision*, 1992, pp. 321–334.
 14. **Senior A., Hampapur A., Lu M.** Acquiring multiscale images by pan-tilt-zoom control and automatic multicamera calibration, *IEEE Workshop on Application of Computer Vision (WACV/MOTION'05)*, 2005, vol. 1, pp. 433–438.
 15. **Shokurov A., Khropov A., Ivanov D.** Feature Tracking in Image and Video, *Proc. of GraphiCon-2003*, 2003, pp. 177–179.
 16. **Hartley R., Zisserman A.** Multiple View Geometry in computer vision, Cambridge, Cambridge University Press, 2003, 673 p.
 17. **Khropov A., Shokurov A., Lempitsky V., Ivanov D.** Reconstruction of projective and metric cameras for image triplets, *Proc. of GraphiCon-2004*, 2004, pp. 143–146.
 18. **Snaveley N., Seitz S. M., Szeliski R.** Photo Tourism: Exploring Photo Collections in 3D, *ACM Transactions on Graphics*, 2006, vol. 25(3), pp. 835–846.
 19. **Ivanov D., Lempitsky V., Khropov A., Shokurov A., Kuzmin Ye.** Computer Vision at CGG MSU, *UK-RUSSIA WORKSHOP "PROACTIVE COMPUTING"*, Nizhny Novgorod, Russia, Feb. 3–6, 2005, pp. 44–58.
 20. **Weng J., Cohen P., Herniou M.** Camera calibration with distortion models and accuracy evaluation, *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 1992, vol. 14 (10), pp. 965–980.
 21. **Konushin A.** Slejeniyе za tochechnymi osobennostyami sceny (Point feature tracking), *Computernaya Grafika i Multimedia*, 2011, no. 1 (5), available at: <http://cgm.computergraphics.ru/content/view/54> (in Russian).
 22. **Lowe D. G.** Distinctive Image Features from Scale-Invariant Keypoints, *Proceedings of the 7th International Conference on Computer Vision*, Kerkyra, Greece, Sept. 20–27, 1999, pp. 1150–1157.
 23. **Kornev A.A., Chigokov E. V.** *Upragneniya po chislennym metodam. Chast II* (Numerical method exercises. Part II), Moscow, Izdatelstvo CPI pri mehaniko-matematicheskoy fakultete MGU im. M. V. Lomonosova, 2003, 200 p. (in Russian).
 24. **Birykov S. V.** *Metrologiya: teksty lekcii* (The Metrology: lecture conspects), Omsk, OmSTU, 2000, 110 p. (in Russian).
 25. **Armangue X., Salvi J.** Overall view regarding fundamental matrix estimation, *Image Vis. Comput.*, 2003, vol. 21, no. 2, pp. 205–220.
 26. **Sturges H. A.** The choice of a class interval, *Journal of the American Statistical Association*, 1926, vol. 21, pp. 65–66. doi:10.1080/01621459.1926.10502161.
 27. **Doane D. P.** Aesthetic frequency classification, *American Statistician*, 1976, vol. 30, pp. 181–183.
 28. **Scott D. W.** *Multivariate Density Estimation: Theory, Practice, and Visualization*, New York, John Wiley & Sons, 1992, 336 p.

ООО "Издательство "Новые технологии". 107076, Москва, Стромьинский пер., 4
Технический редактор *Е. М. Патрушева*. Корректор *И. Е. Назарова*

Сдано в набор 09.08.2017 г. Подписано в печать 19.09.2017 г. Формат 60×88 1/8. Заказ П11017
Цена свободная.

Оригинал-макет ООО "Авансед солюшнз". Отпечатано в ООО "Авансед солюшнз".
119071, г. Москва, Ленинский пр-т, д. 19, стр. 1. Сайт: www.aov.ru
