

# Программная инженерия

Том 7  
№ 3  
2016  
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., Крылов Д. А., Москаленко Ф. М., Тимченко В. А., Федорищев Л. А., Шалфеева Е. А.** Базовая технология разработки интеллектуальных сервисов на облачной платформе IASPaas. Часть 3. Разработка интерфейса и пример создания прикладных сервисов ..... 99
- Стенников В. А., Барахтенко Е. А., Соколов Д. В.** Применение концепции Model-Driven Engineering в программном комплексе для определения оптимальных параметров теплоснабжающих систем ..... 108
- Галатенко В. А.** Кибербезопасность в здравоохранении ..... 117
- Быкова Н. М., Зайнагабдинов Д. А., Беялов Т. Ш., Мешков И. В.** Технологические основы программно-аппаратного сопровождения автоматизированного мониторинга деформаций железнодорожных тоннелей ..... 126
- Юрушкин М. В.** Двойное блочное размещение данных в оперативной памяти при решении задачи умножения матриц ..... 132
- Свердлов С. З.** Настройка насыщенности при обработке цифровых изображений ..... 140

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.ru 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", 2016

# SOFTWARE ENGINEERING

## PROGRAMMAYA INGENERIA

Vol. 7

N 3

2016

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  
CHETVERTUSHKIN 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)  
SEREBRJKOV 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

<b>Gribova V. V., Kleschev A. S., Krylov D. A., Moskalenko Ph. M., Timchenko V. A., Fedorischev L. A., Shalfeyeva E. A.</b> The Base Technology for Intelligent Services Development with the Use of IACPaaS Cloud Platform. Part 3. An Interface Development and an Example of Applied Services Creation .....	99
<b>Stennikov V. A., Barakhtenko E. A., Sokolov D. V.</b> Model-Driven Engineering in the Software for Parameter Optimization of Heat Supply Systems .....	108
<b>Galatenko V. A.</b> Cybersecurity in Healthcare .....	117
<b>Bykova N. M., Zainagabdinov D. A., Belyalov T. Sh., Meshkov I. V.</b> Technological Bases of Software-Instrumental Accompaniment Electronic Deformation Monitoring of Railway Tunnels .....	126
<b>Yurushkin M. V.</b> Double Block Data Layout in High Performance Matrix Multiplication Algorithm .....	132
<b>Sverdlov S. Z.</b> Saturation Adjustment in the Digital Images Processing .....	140

Information about the journal is available online at:  
<http://novtex.ru/prin/eng> e-mail: [prin@novtex.ru](mailto: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: kleshev@iacp.dvo.ru,  
**Д. А. Крылов**, канд. техн. наук, вед. инженер-программист, e-mail: dmalkr@gmail.com,  
**Ф. М. Москаленко**, канд. техн. наук, науч. сотр., e-mail: philipmm@iacp.dvo.ru,  
**В. А. Тимченко**, канд. техн. наук, 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

# Базовая технология разработки интеллектуальных сервисов на облачной платформе IACPaaS.

## Часть 3. Разработка интерфейса и пример создания прикладных сервисов

*Настоящая статья завершает цикл работ, посвященных описанию базовой технологии разработки интеллектуальных мультиагентных сервисов и их компонентов с использованием инструментальных сервисов облачной платформы IACPaaS. Данная часть посвящена описанию технологии разработки пользовательского интерфейса интеллектуального сервиса и описанию примера использования базовой технологии для разработки прикладных интеллектуальных сервисов в области офтальмологии.*

**Ключевые слова:** интеллектуальные системы, мультиагентные системы, технология разработки интеллектуальных систем, иерархические орграфы, агентно-ориентированное программирование, облачные платформы, облачные сервисы

### Введение

В настоящей статье, завершающей цикл работ [1, 2], посвященных описанию базовой технологии разработки интеллектуальных сервисов на платформе IACPaaS, описываются технология разработки пользовательского интерфейса интеллектуального сервиса, а также методология расширения базы интерфейсных элементов платформы. Рассмотрен пример использования описанной в цикле работ базовой технологии для разработки прикладных интеллектуальных сервисов в области офтальмологии: *Виртуальная среда тренажера для обучения определению остроты зрения по таблицам и Виртуальная среда тренажера для обучения определению остроты зрения по оплотипам Б. Л. Поляка.*

## 1. Разработка интерфейса интеллектуального сервиса

### 1.1. Общие сведения

Под разработкой интерфейса интеллектуального сервиса на облачной платформе IACPaaS подразумевается разработка web-интерфейса. Множество современных web-интерфейсов создают с использованием различных систем управления контентом (CMS), которые позволяют разработчикам абстрагироваться от низкоуровневых деталей web-программирования. Для разработки интерфейсов на платформе IACPaaS используют CMS

MediaWiki [3]. Существенным преимуществом MediaWiki является развитая система расширений, благодаря которым можно значительно повысить функциональные возможности данной CMS, добавив в нее множество различных дополнительных возможностей.

Основной сущностью CMS MediaWiki является *wiki-страница*, на которой могут быть размещены различные визуальные объекты — текст с гиперссылками, изображения, таблицы, а также интерфейсные элементы дополнительных подключаемых модулей (плагинов) — вплоть до проигрываемых Flash-скриптов. Процессор пользовательского интерфейса (ППИ) платформы IACPaaS интегрирован в среду MediaWiki именно в виде плагина.

Пользовательский web-интерфейс сервиса может быть представлен тройкой вида  $UI = \langle a_{ui}, \{wiki\_page_1, \dots, wiki\_page_n\}, \{flash_1, \dots, flash_m\} \rangle$ ,  $n \geq 1$ ,  $m \geq 0$ , где:

- $a_{ui}$  — агент *Интерфейсный контроллер* [1, 2];
- $\{wiki\_page_1, \dots, wiki\_page_n\}$  — множество wiki-страниц на web-сайте платформы IACPaaS, одна из которых должна быть указана в качестве *стартовой* при интеграции решателя задач с пользовательским интерфейсом [1];
- $\{flash_1, \dots, flash_m\}$  — множество Flash-скриптов, запускаемых и выполняющихся на wiki-страницах  $wiki\_page_1, \dots, wiki\_page_p$ ,  $p \in [0, n]$ , где  $wiki\_page_j \in \{wiki\_page_1, \dots, wiki\_page_n\}$ ,  $j \in [0, p]$  (данный компонент может отсутствовать, если разработка пользо-

вательского интерфейса не предполагает разработку Flash-скриптов).

Каждая wiki-страница  $wiki\_page_i$  ( $i = 1, \dots, n$ ) представляет собой тройку вида  $(content\_descr, \{ui\_request_1, \dots, ui\_request_k\}, [CSS])$ ,  $k \geq 1$ , элементы которой описаны далее.

- Элемент  $content\_descr$  — описание содержательной части интерфейса сервиса на языке разметки MediaWiki. Синтаксис данного языка является простым (в сравнении с синтаксисом языка HTML) и вместе с тем расширяемым.

- Элемент  $\{ui\_request_1, \dots, ui\_request_k\}$  — непустое множество  $ui$ -запросов к агенту Интерфейсный контроллер ( $a_{ui}$ ) интегрированного решателя задач разрабатываемого сервиса. Каждый запрос  $ui\_request_j$  ( $j = 1, \dots, k$ ) представляет собой множество пар  $parametr = значение$ , среди которых обязательно должна присутствовать пара вида  $service = "предметная область/раздел/название решателя задач"$ , идентифицирующая интегрированный решатель задач, к агенту Интерфейсный контроллер которого выполняется запрос. Каждый такой запрос с помощью специального тега  $\langle ui\_... \rangle$  встраивается в описание  $content\_descr$ . Данный тег указывает транслятору языка разметки MediaWiki на то, что необходимо обратиться к ППИ платформы IACPaaS, передав ему параметры запроса. Затем, получив от ППИ результат (обычно это описание фрагмента содержательной части интерфейса, которая формируется соответствующим агентом Интерфейсный контроллер), поместить этот результат в место расположения данного тега.

- Элемент CSS — описание таблицы стилей CSS (множества правил CSS), определяющей внешний вид wiki-страницы — дизайнерскую часть интерфейса сервиса (может отсутствовать, если разработчиком принимается дизайн, определяемый оформлением (skin) CMS MediaWiki, или, например, дизайн целиком определяется в проигрываемом Flash-скрипте).

Схема взаимодействия агента  $Vid$  с установленной на web-сервере CMS MediaWiki и агентом Интерфейсный контроллер (входящим в состав некоторого решателя задач) представлена на рис. 1. Процесс взаимодействия состоит в передаче запроса из браузера агентам сервиса и в возврате результата в браузер, между которыми выполняется обработка запроса. Нумерация стрелок на рис. 1 задает также порядок взаимодействий. Сплошные стрелки черного цвета означают передачу данных (уникальных для каждого запроса) между системными компонентами платформы; штриховые стрелки черного цвета — передачу служебных данных между системными компонентами платформы; стрелки серого цвета — передачу данных между агентами сервиса.

Устройство web-интерфейса сервиса согласуется с моделью представления пользовательского интерфейса сервисов платформы IACPaaS, в основу которой положена концепция MVC (Модель-Представление-Контроллер) [4]. Ее основополагающий принцип состоит в разделении данных, логики их обработки и способа представления в целях обеспечения возможности независимого изменения каждого из компонентов. Проекция данной концепции

на модель интерфейса сервисов платформы IACPaaS представлена следующим образом.

**Модель.** Данный компонент включает в себя перечисленные далее составляющие.

- *Модель абстрактного пользовательского интерфейса* — информационный ресурс Фонда, представляющий метайнформацию и содержащий описание структуры стандартных интерфейсных элементов (простых и интерфейсных элементов-контейнеров) WIMP-интерфейсов и способа их рекурсивной организации в единую вложенную структуру. Множество информационных ресурсов, порожденных по метайнформации Модель абстрактного пользовательского интерфейса, таким образом, содержательно является описаниями абстрактных пользовательских интерфейсов.

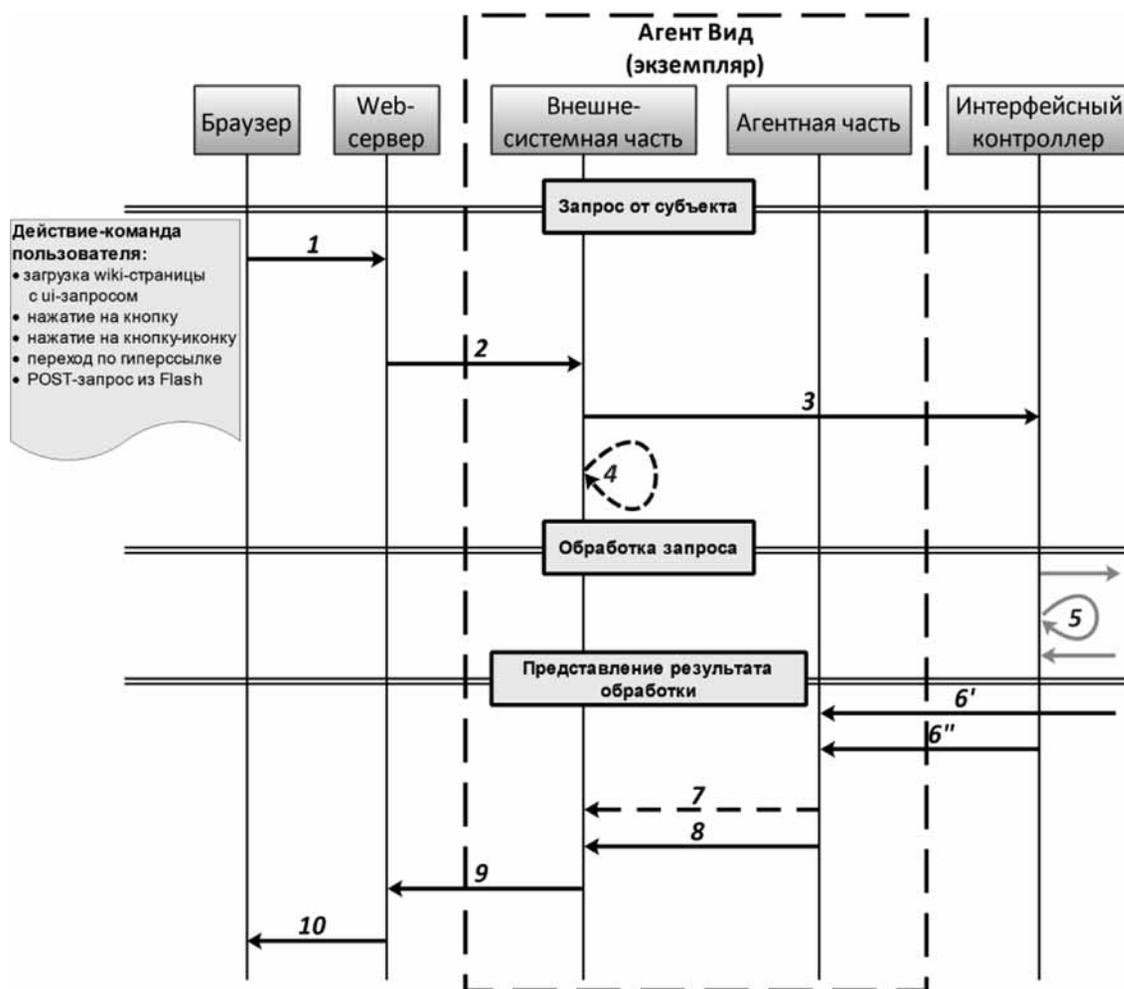
- *Программный интерфейс для формирования абстрактных интерфейсов* — набор высокоуровневых функций для создания фрагментов абстрактных интерфейсов. Посредством вызовов этих функций с соответствующими аргументами значительно повышается уровень абстракции, на котором формируется информационный ресурс, который представляет собой описание некоторого абстрактного интерфейса. Это происходит за счет того, что формирование выполняется не в терминах отдельных вершин и дуг соответствующего абстрактному интерфейсу орграфа, а в терминах конструирования этого орграфа из подграфов, каждый из которых есть описание интерфейсного элемента или его атрибута, соответствующего вызванной функции. Таким образом, формирование описания нужного абстрактного интерфейса (фрагмента) состоит в построении соответствующей суперпозиции вызовов функций данного программного интерфейса.

**Представление.** Данный компонент представлен системным агентом платформы Вид, который является составной частью ППИ платформы IACPaaS. Данный агент (точнее, множество его экземпляров) является *гибридным*. Он служит посредником между частью ППИ, реализованной в виде расширения CMS MediaWiki (с ней взаимодействует *Внешнесистемная часть* этого агента), и множеством агентов платформы, являющихся Интерфейсными контроллерами в составе решателей задач (с ними взаимодействует его Агентная часть). Взаимодействие с некоторым Интерфейсным контроллером происходит посредством посылки ему сообщений по шаблону *Запрос от агента Вид*. Помимо функции сопряжения MediaWiki и остальной части виртуальной машины платформы IACPaaS, в его задачи входят:

- формирование описания конкретного интерфейса (HTML-код) на основе описания абстрактного интерфейса и правил отображения представлений интерфейсных элементов и их атрибутов в терминах модели абстрактного пользовательского интерфейса в их представлении на языке HTML;

- обеспечение возможности запуска Flash-скриптов<sup>1</sup> на wiki-страницах (посредством внедре-

<sup>1</sup> Под запуском Flash-скриптов понимается воспроизведение \*.swf-файлов, содержащих Flash-анимацию, URL которых на web-сервере указываются в формируемом HTML-коде в специальном проигрывателе Flash Player (существующем в виде плагина к web-браузеру).



**Рис. 1.** Схема взаимодействия агента Вид с web-сервером и агентом Интерфейсный контроллер:

1 — *ui*-запрос; 2 — параметры запроса; 3 — сообщение по шаблону Запрос от агента Вид (содержащее параметры запроса); 4 — ожидание уведомления; 5 — сообщения, которыми обмениваются агенты сервиса при обработке запроса; 6' или 6'' — сообщение по одному из входных шаблонов для агента Вид (Отобразить окно, Вернуть инфоресурс в окно, Вернуть строку в окно); 7 — уведомление о готовности результата обработки; 8, 9, 10 — HTML-код, строка, JSON-представление

ния Flash-анимации с помощью соответствующей стандартам кроссбраузерной HTML-разметки);

- формирование представления информационных ресурсов Фонда (представляющих информацию) в формате JSON для последующей их обработки (в частности, Flash-скриптами);

- загрузка и выгрузка бинарных данных на web-сервер.

**Контроллер.** Данный компонент представлен агентами, играющими роль Интерфейсного контроллера в составе различных решателей задач. Эти агенты взаимодействуют с агентом Вид посредством обмена сообщениями по определенным шаблонам и реализуют (взаимодействуя, возможно, с другими агентами решателя задач) логику обработки:

- *ui*-запросов, размещенных на wiki-страницах и выполняемых в процессе трансляции описания *content\_descr* с языка разметки MediaWiki в HTML-код;

- событий, генерируемых интерфейсными элементами в ответ на действия пользователей;

- POST-запросов из Flash-скриптов.

Результатом обработки является один из трех возможных объектов:

- информационный ресурс, представляющий описание абстрактного пользовательского интерфейса, фрагменты которого (или все описание полностью) могут быть сформированы другими агентами решателя задач (передается Агентной части агента Вид в сообщении, созданном по шаблону *Отобразить окно*);

- произвольный информационный ресурс, представляющий информацию (передается Агентной части агента Вид в сообщении, созданном по шаблону *Вернуть инфоресурс в окно*);

- произвольная строка символов (передается Агентной части агента Вид в сообщении, созданном по шаблону *Вернуть строку в окно*).

При разработке пользовательского интерфейса сервиса в Фонде должен присутствовать агент Интерфейсный контроллер, который реализует логику обработки:

— всех  $ui$ -запросов для множества тех  $wiki$ -страниц сервиса  $\{wiki\_page_1, \dots, wiki\_page_s\}$ ,  $s \in [1, n]$ , на которых они присутствуют ( $\{wiki\_page_1, \dots, wiki\_page_s\} \subseteq \subseteq \{wiki\_page_1, \dots, wiki\_page_n\}$ );

— событий, генерируемых интерфейсными элементами (если таковые присутствуют на  $wiki$ -страницах  $wiki\_page_1, \dots, wiki\_page_s$ );

— запросов из Flash-скриптов, выполняемых на  $wiki$ -страницах  $wiki\_page_1, \dots, wiki\_page_p$ ,  $p \in [0, n]$  (если разработка пользовательского интерфейса предполагает разработку Flash-скриптов).

Агент Интерфейсный контроллер может быть повторно используемым, например, в том случае, если у сервисов одинаковые с точки зрения структуры и поведения интерфейсы, но различный дизайн. Для этого на  $wiki$ -страницах сервиса необходимо использовать разные таблицы стилей CSS (с одноименными CSS-классами).

Если подходящий агент Интерфейсный контроллер отсутствует в Фонде, то его необходимо разработать по технологии, описанной в работе [2].

## 1.2. Порядок разработки интерфейса интеллектуального сервиса

Разработка пользовательского интерфейса включает следующие этапы.

**Этап 1.** Создание стандартным для CMS MediaWiki способом множества  $wiki$ -страниц  $\{wiki\_page_1, \dots, wiki\_page_n\}$ ,  $n \geq 1$ , на сайте платформы IACPaas и формирование содержимого каждой  $wiki\_page_i$  ( $i = 1, \dots, n$ ) страницы следующим образом:

- описать содержательную часть интерфейса сервиса на языке разметки MediaWiki с размещением в нужных местах  $ui$ -запросов вида  $\langle ui\ solver = "предметная\ область/раздел/название\ решателя\ задач" параметр_1 = значение_1 \dots параметр_m = значение_m / \rangle$ ;

- если необходимо определить собственные стили оформления элементов интерфейса, то внутри блока  $\{\{css: \dots\}$  поместить описание таблицы стилей CSS, определяющей внешний вид  $wiki$ -страниц — дизайнерскую часть интерфейса сервиса.

**Этап 2.** Разработка по технологии, описанной в работе [2], агента Интерфейсный контроллер, обрабатывающего размещенные на  $wiki$ -страницах  $ui$ -запросы и события, генерируемые интерфейсными элементами (если в Фонде отсутствует подходящий для этого агент).

**Этап 3.** Разработка Flash-скриптов  $\{flash_1, \dots, flash_m\}$ , выполняемых на  $wiki$ -страницах  $wiki\_page_1, \dots, wiki\_page_p$ , где  $p \in [0, n]$  и  $wiki\_page_j \in \{wiki\_page_1, \dots, wiki\_page_n\}$ ,  $j \in [0, p]$ , если разработка пользовательского интерфейса предполагает разработку Flash-скриптов.

Исходный код Flash-скрипта разрабатывается на объектно-ориентированном языке программирования ActionScript и сохраняется в  $*.as$ -файле. В результате компиляции  $*.as$ -файла формируется исполняемый код (байт-код), который сохраняется в  $*.swf$ -файл. После этого  $*.swf$ -файл размещается по заданному пути на  $web$ -сервере, который агент Интерфейсный контроллер помещает в описание абстрактного пользовательского интерфейса и передает агенту Вид (в ответ на соответствующий  $ui$ -запрос).

## 1.3. Расширение набора интерфейсных элементов платформы

Набор интерфейсных элементов в модели интерфейса сервисов платформы IACPaas является расширяемым. Добавление нового элемента состоит в добавлении:

- в Модель абстрактного пользовательского интерфейса — декларативного описания (в виде орграфа метайнформации) структуры интерфейсного элемента;

- в Программный интерфейс для формирования абстрактных интерфейсов — функции, формирующей декларативное представление (в виде орграфа информации) интерфейсного элемента как фрагмента информационного ресурса, представляющего описание абстрактного интерфейса;

- в агент Вид — реализации правила отображения декларативного представления интерфейсного элемента в его представление на языке HTML.

Выделение в структуре  $wiki$ -страниц содержательной и дизайнерской частей интерфейса позволяет:

- к одной содержательной части интерфейса применять разные таблицы стилей CSS, равно как и наоборот, одну и ту же таблицу стилей CSS использовать в интерфейсах разных сервисов. За счет этого повышается гибкость и адаптивность разрабатываемых интерфейсов, а также упрощается процесс их сопровождения;

- разделить процесс разработки и сопровождения содержательной и дизайнерской частей и привлечь к работам над каждой частью соответствующих специалистов. Разработчикам этих частей необходимо согласовать между собой лишь имена классов в описании таблицы стилей CSS.

## 2. Пример разработки прикладных сервисов для офтальмологии

В качестве примера использования базовой технологии опишем разработку с ее помощью нескольких прикладных сервисов.

Начнем с сервиса *Виртуальная среда тренажера для обучения определению остроты зрения по таблицам* (рис. 2, см. вторую сторону обложки). Он состоит из следующих компонентов [1]:

- $app$  (интегрированный решатель задач) = *Интерпретатор виртуальных сред*;

- $\langle in_1, \dots, in_k \rangle$  — входные фактические параметры ( $k = 2$ ):

- $in_1$  = информационный ресурс *Проект виртуальной среды по определению остроты зрения по таблицам* (сформирован по метайнформации *Онтология виртуального мира*);
- $in_2$  = информационный ресурс *Графический редактор. Стандартный интерфейс* (сформирован по метайнформации *Онтология интерфейса виртуальной среды*);
- $\langle out_1, \dots, out_p \rangle$  — выходные фактические параметры =  $\emptyset$  (отсутствуют,  $p = 0$ ).

<sup>2</sup> Орграфы информации, представляющие информационные ресурсы, являющиеся фактическими входными параметрами, а также собственные информационные ресурсы и соответствующие им орграфы метайнформации в настоящей работе не приводятся.

Интегрированный решатель Интерпретатор виртуальных сред предназначен для отображения и функционирования виртуальной среды в соответствии с декларативной моделью, представленной в виде информационного ресурса. При этом в функции Интерпретатора виртуальных сред входят:

- загрузка ресурсов виртуальной среды — трехмерных моделей объектов, текстур, иконок, анимаций;
- обеспечение интерактивного взаимодействия с объектами среды;
- инициализация логических параметров виртуальной среды;
- сохранение и загрузка анимационных роликов;
- отображение объяснений и результатов действий пользователя.

Интерпретатор виртуальных сред (*app*) представлен в Фонде платформы IASaaS информационным ресурсом Интерпретатор виртуальных сред (рис. 3). и состоит из следующих компонентов [1]:

- *name* = Интерпретатор виртуальных сред;
- *descr* = Интерпретатор предназначен для запуска и работы созданных виртуальных сред;
- *a<sub>root</sub>* = Корневой агент для интерпретатора;
- *a<sub>ui</sub>* = Интерфейсный агент для интерпретатора;
- *wiki\_page* = ИРВИС;
- $\langle IN_1, \dots, IN_k \rangle$  ( $k = 2$ ):
- *IN<sub>1</sub>* = информационный ресурс *Онтология виртуального мира*;
- *IN<sub>2</sub>* = информационный ресурс *Онтология интерфейса виртуальной среды*;
- $\langle OUT_1, \dots, OUT_p \rangle = \emptyset$  (отсутствуют,  $p = 0$ );
- $\langle own_1, \dots, own_q \rangle$  ( $q = 3$ ):
- *own<sub>1</sub>* = информационный ресурс *Ресурсы виртуальных сред* (построен по метаинформации *Структура ресурсов виртуальных сред*);
- *own<sub>2</sub>* = информационный ресурс *Временный проект виртуальной среды* (построен по метаинформации *Онтология виртуального мира*);
- *own<sub>3</sub>* = информационный ресурс *Ролики виртуальной среды* (построен по метаинформации *Структура записей виртуальных сред*);
- $\langle log_1, \dots, log_s \rangle = \emptyset$  (отсутствуют на начальном этапе,  $s = 0$ ).

Агент Корневой агент для интерпретатора представлен в Фонде платформы IASaaS информационным ресурсом Корневой агент для интерпретатора (рис. 4) и состоит из следующих компонентов [2]:

- *agent\_name* = Корневой агент для интерпретатора;
- *agent\_internal\_name* = InterpreterVRInitAgent;
- *agent\_descr* = инициализирующий агент для интерпретатора виртуальных сред;
- *local\_structure* =  $\emptyset$  (отсутствует);
- $\{production\_block_1, \dots, production\_block_n\}$  ( $n = 1$ ):
- *production\_block<sub>1</sub>* (Инициализация):
  - *production\_block\_descr* = обработка инициализирующего сообщения;
  - *inMsgTemplate* = Шаблон Инициализирующее сообщение;
  - $\{outMsgTemplate_1, \dots, outMsgTemplate_m\} = \emptyset$  (отсутствуют,  $m = 0$ );
- *agent\_code* — байт-код, полученный путем компиляции класса агента, написанного на языке программирования Java (здесь не приводится).

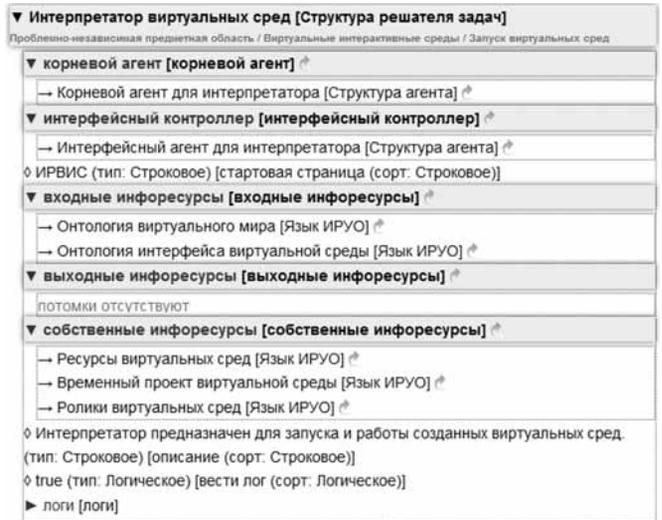


Рис. 3. Интерфейс Редактора решателей задач платформы IASaaS, в котором сформирован оргграф информации, представляющий интегрированный решатель задач Интерпретатор виртуальных сред

Основные функциональные возможности Интерпретатора реализованы в агенте *Интерфейсный контроллер для интерпретатора*. Он выполняет отображение пользовательского веб-интерфейса на стартовой странице ИРВИС. Интерфейс включает возможность отображения и работы Flash-скрипта. Интерпретатор также использует: средство платформы для преобразования декларативной модели в формат JSON и средства платформы для асинхронной загрузки с нее и на нее различных ресурсов (файлы моделей, текстур и др.). Интерфейсный контроллер для интерпретатора также может отправлять сообщения другим (произвольным) агентам, указанным в декларативной модели, для реализации дополнительных функциональных возможностей. Агент Интерфейсный контроллер для интерпретатора представлен в Фонде платформы IASaaS информационным ресурсом Интерфейсный контроллер для интерпретатора (рис. 5) и состоит из [2]:

- *agent\_name* = Интерфейсный контроллер для интерпретатора;
- *agent\_internal\_name* = InterpreterVRInterfaceAgent;

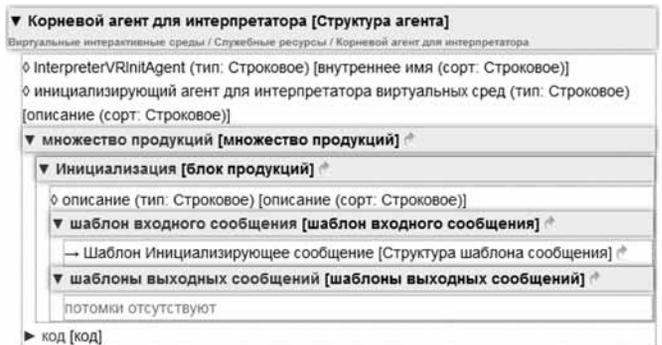


Рис. 4. Интерфейс Редактора агентов платформы IASaaS, в котором сформирован оргграф информации, представляющий агент Корневой агент для интерпретатора

- *agent\_descr* = Интерфейсный агент выполняет отображение и поддержку работы веб-клиента интерпретатора;
  - *local\_structure* = см. рис. 6;
  - {*production\_block*<sub>1</sub>, ..., *production\_block*<sub>*n*</sub>} (*n* = 2):
    - *production\_block*<sub>1</sub> (обработать сообщение от агента Вид):
      - *production\_block\_descr* = выполняется отрисовка окна с Flash-скриптом;
      - *inMsgTemplate* = Шаблон Запрос от агента Вид;
      - {*outMsgTemplate*<sub>1</sub>, ..., *outMsgTemplate*<sub>*m*</sub>} (*m* = 4):
        - ◆ *outMsgTemplate*<sub>1</sub> = Шаблон Отобразить окно;
        - ◆ *outMsgTemplate*<sub>2</sub> = Шаблон Сообщение обработчика виртуальной среды;
        - ◆ *outMsgTemplate*<sub>3</sub> = Шаблон Вернуть строку в окно;
        - ◆ *outMsgTemplate*<sub>4</sub> = Шаблон Вернуть инфоресурс в окно;
    - *production\_block*<sub>2</sub> (обработать ответ от другого агента):
      - *production\_block\_descr* = выполняется обработка ответа, приходящего от вспомогательного агента;
  - *inMsgTemplate* = Шаблон Сообщение обработчика виртуальной среды;
  - {*outMsgTemplate*<sub>1</sub>, ..., *outMsgTemplate*<sub>*m*</sub>} (*m* = 2):
    - ◆ *outMsgTemplate*<sub>1</sub> = Шаблон Отобразить окно;
    - ◆ *outMsgTemplate*<sub>2</sub> = Шаблон Вернуть строку в окно;
  - *agent\_code* — байт-код, полученный путем компиляции класса агента, написанного на языке программирования Java (здесь не приводится).
- В данном примере генерация шаблонов исходного кода, сам исходный код, процесс его компиляции в байт-код и загрузки в Фонд платформы IACPaaS, а также процесс тестирования агентов не описываются.
- Как видно, взаимодействие агентов Интерпретатора состоит в обмене в основном сообщениями, сформированными по системным шаблонам платформы IACPaaS. Требуется лишь разработать шаблон сообщений Шаблон Сообщение обработчика виртуальной среды. Он представлен в Фонде платформы IACPaaS информационным ресурсом *Шаблон Сообщение обработчика виртуальной среды* (рис. 7) и состоит из следующих компонентов [2]:

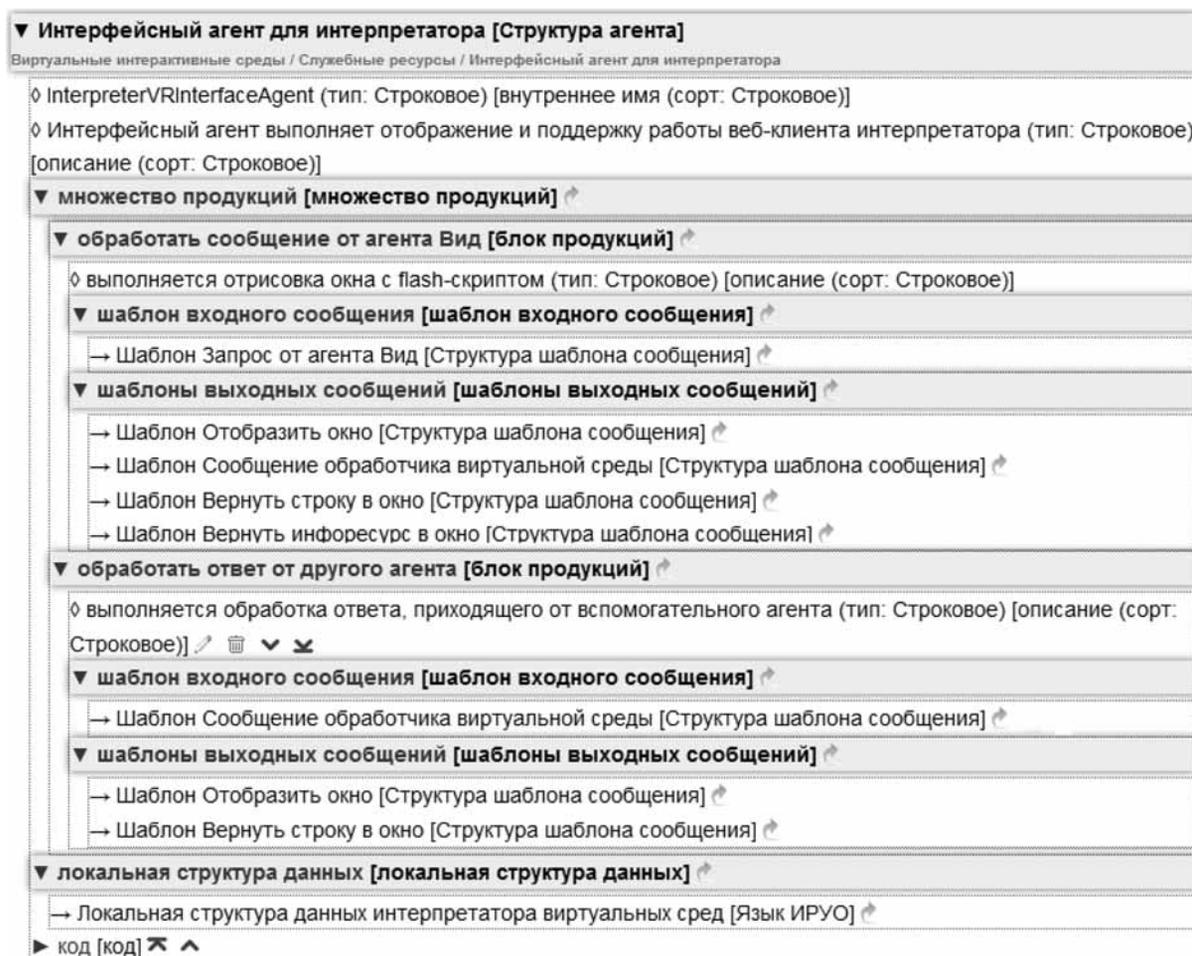


Рис. 5. Интерфейс Редактора агентов платформы IACPaaS, в котором сформирован оргграф информации, представляющий агент Интерфейсный контроллер для интерпретатора

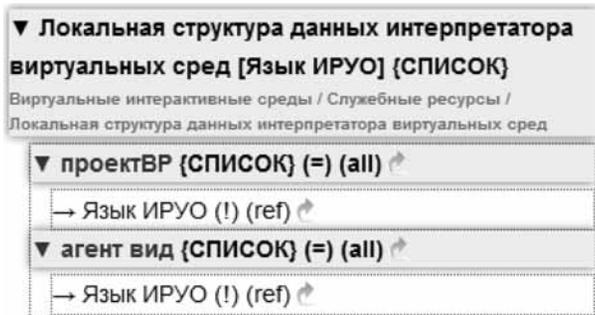


Рис. 6. Интерфейс Редактора орграфов метайнформации платформы IACPaaS, в котором сформирован орграф метайнформации, представляющий локальную структуру данных для агента Интерфейсный контроллер для интерпретатора

- $template\_name$  = Шаблон Сообщение обработчика виртуальной среды;
- $template\_internal\_name$  = ProcessVR;
- $template\_descr$  = Сообщение обработчика виртуальной среды;
- $message\_structure$  — см. рис. 8;
- $template\_code$  — байт-код, полученный путем компиляции класса агента, написанного на языке программирования Java (здесь не приводится).

Интерфейс сервиса устроен следующим образом:

- $a_{ui}$  = Интерфейсный агент для интерпретатора;
- $\{wiki\_page_1, \dots, wiki\_page_n\}$  ( $n = 1$ ):
  - $wiki\_page_1$  (ИРВИС):
    - $content\_descr = \emptyset$  (отсутствует);
    - $\{ui\_request_1, \dots, ui\_request_k\}$  ( $k = 1$ ):
      - ◆  $ui\_request_1 = \langle ui\_solver = \text{Проблемно-независимая предметная область/Виртуальные интерактивные среды/Интерпретатор виртуальных сред, action} = \text{"init"} \rangle$ ;
    - $CSS = \emptyset$  (отсутствует);
  - $\{flash_1, \dots, flash_m\}$  ( $m = 1$ ):
    - $flash_1$  = флеш-скрипт VMProject.swf (будет исполняться на  $wiki\_page_1$ ).

С использованием разработанного интегрированного решателя задач выполним сборку еще одного интеллектуального сервиса — *Виртуальная среда тренажера для обучения определению остроты зрения по оптотипам Б. Л. Поляка*, предварительно сформировав с использованием Редактора орграфов информации информационный ресурс "Проект виртуальной среды по определению остроты зрения по оптотипам Поляка". Он состоит из следующих компонентов [1]:

- $app$  (интегрированный решатель задач) = Интерпретатор виртуальных сред;

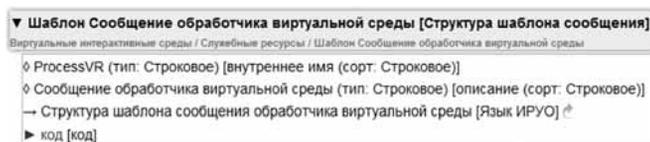


Рис. 7. Интерфейс Редактора шаблонов сообщений платформы IACPaaS, в котором сформирован орграф информации, шаблон сообщений Интерфейсный контроллер для интерпретатора



Рис. 8. Интерфейс Редактора орграфов метайнформации платформы IACPaaS, в котором сформирован орграф метайнформации, представляющий структуру содержимого сообщений для шаблона сообщений Шаблон Сообщение обработчика виртуальной среды

- $\langle in_1, \dots, in_k \rangle$  — входные фактические параметры ( $k = 2$ ):
  - $in_1$  = информационный ресурс Проект виртуальной среды по определению остроты зрения по оптотипам Поляка (сформирован по метайнформации Онтология виртуального мира);
  - $in_2$  = информационный ресурс Графический редактор. Стандартный интерфейс (сформирован по метайнформации Онтология интерфейса виртуальной среды);
- $\langle out_1, \dots, out_p \rangle$  (выходные фактические параметры) =  $\emptyset$  (отсутствуют,  $p = 0$ ).

## Заключение

В отличие от облачных платформ, предоставляющих сервисы разных типов по моделям SaaS, PaaS, IaaS, DaaS и поддерживающих различные технологии создания облачных сервисов, в том числе для научных исследований (таких как CLAVIRE [5], NanoCloud [6], Unihub [7], Персональный виртуальный компьютер [8]), платформа IACPaaS ориентирована на создание интеллектуальных сервисов, в архитектуре которых имеется база знаний, технология разработки сервисов на платформе IACPaaS поддерживает проектирование специфического для интеллектуальной системы компонента — баз знаний — с помощью Редактора орграфов информации. Поскольку пользовательский интерфейс данного редактора генерируется по соответствующей метайнформации, созданной инженером знаний, эксперты и специалисты предметной области могут создавать и сопровождать (поддерживать в актуальном состоянии) базы знаний в привычной для них системе понятий без посредников или дополнительного обучения. В этом состоит одно из его основных отличий от редактора Protégé [9], в котором отсутствует четкое разделение между терминологиями инженеров знаний и экспертов предметной области, что делает практически невозможной самостоятельную работу последних. Представленная в работе технология поддерживает интеграцию баз знаний в интеллектуальный сервис.

Особенности предлагаемой технологии и технологии OSTIS [10, 11] состоят в том, что при разработке интеллектуальных систем акцент делается на их поэтапную сборку (сверху вниз) из повторно используемых совместимых компонентов, хранящихся в непрерывно развиваемом Фонде информационных ресурсов платформы IACPaaS и базе компонентов платформы OSTIS соответственно. Однако ключевой особенностью предлагаемой технологии является единое унифицированное декларативное представление всех компонентов, которое в технологии OSTIS обеспечивается за счет надстройки в виде абстрактной унифицированной логико-семантической модели и ее интерпретатора, которые не только снижают эффективность работы интеллектуальной системы, но также затрудняют повторное использование компонентов и создание инструментальных сервисов поддержки технологии разработки.

С использованием предложенной технологии и базовых инструментальных сервисов платформы созданы следующие облачные специализированные инструментальные и прикладные сервисы: комплекс средств для разработки профессиональных виртуальных облачных сред [12], виртуальная модель городского района, виртуальная химическая лаборатория и прототип компьютерного обучающего тренажера по классическим методам исследования в офтальмологии [13]. В процессе разработки находятся прикладные сервисы по автоматизированному конструированию доказательств математических теорем и медицинской диагностике.

*Работа выполнена при частичной финансовой поддержке РФФИ (проект 13-07-00024, проект 14-07-00299 и проект 15-07-03193) и программы ФНИ (Дальний Восток).*

#### Список литературы

1. Грибова В. В., Клешев А. С., Крылов Д. А. и др. Базовая технология разработки интеллектуальных сервисов на облачной

платформе IACPaaS. Часть 1. Разработка базы знаний и решателя задач // Программная инженерия. 2015. № 12. С. 3—11.

2. Грибова В. В., Клешев А. С., Крылов Д. А. и др. Базовая технология разработки интеллектуальных сервисов на облачной платформе IACPaaS. Часть 2. Разработка агентов и шаблонов сообщений // Программная инженерия. 2016. № 1. С. 14—20.

3. MediaWiki. URL: <https://www.mediawiki.org/> (дата обращения 18.02.2015).

4. Reenskaug Trygve. MVC. Xerox PARC 1978-79. URL: <http://heim.ifi.uio.no/~trygver/themes/mvc/mvc-index.html> (дата обращения: 18.02.2015).

5. Бухановский А. В., Васильев В. Н., Виноградов В. Н. и др. CLAVIRE: перспективная технология облачных вычислений второго поколения // Известия вузов. Приборостроение. 2011. № 10. С. 7—13.

6. NanoCloud — Вычислительные сервисы для нанотехнологий. URL: <http://cloud7.grid.kiae.ru/about.do> (дата обращения: 15.01.2015).

7. Крапошин М. В., Самоваров О. И., Стрижак С. В. Особенности реализации Web-лаборатории механики сплошной среды на базе технологической платформы "Университетский кластер" // Научный сервис в сети Интернет: эксафлопное будущее. Труды Международной суперкомпьютерной конференции. 19—24 сентября 2011 г., г. Новороссийск. М.: Изд-во МГУ, 2011. С. 473—475.

8. Костенецкий П. С., Семенов А. И., Соколинский Л. Б. Создание образовательной платформы "Персональный виртуальный компьютер" на базе облачных вычислений // Научный сервис в сети Интернет: эксафлопное будущее. Труды Международной суперкомпьютерной конференции. 19—24 сентября 2011 г., г. Новороссийск. М.: Изд-во МГУ, 2011. С. 374—377.

9. The Protégé Ontology Editor and Knowledge Acquisition System. URL: <http://protege.stanford.edu/> (дата обращения 18.02.2015).

10. Голенков В. В., Гулякина Н. А. Проект открытой семантической технологии компонентного проектирования интеллектуальных систем. Часть 1: Принципы создания // Онтология проектирования. 2014. № 1 (11). С. 42—64.

11. Голенков В. В., Гулякина Н. А. Проект открытой семантической технологии компонентного проектирования интеллектуальных систем. Часть 2: Унифицированные модели проектирования // Онтология проектирования. 2014. № 4 (14). С. 34—53.

12. Грибова В. В., Федоричев Л. А. Программный комплекс для разработки облачных виртуальных сред // Программные продукты и системы. 2015. № 2. С. 60—64.

13. Грибова В. В., Петряева М. В., Федоричев Л. А. Компьютерный обучающий тренажер с виртуальной реальностью для офтальмологии // Открытое образование. 2013. № 6. С. 45—51.

## The Base Technology for Intelligent Services Development with the Use of IACPaaS Cloud Platform. Part 3. An Interface Development and an Example of Applied Services Creation

V. V. Gribova, gribova@iacp.dvo.ru, A. S. Kleshev, kleshev@iacp.dvo.ru, D. A. Krylov, dmalkr@gmail.com, Ph. M. Moskalenko, philipmm@iacp.dvo.ru, V. A. Timchenko, vadim@iacp.dvo.ru, L. A. Fedorischev, fedorischev@iacp.dvo.ru, E. A. Shalfeyeva, shalf@iacp.dvo.ru, Federal State Budgetary Institution of Science "Institute for Automation and Control Processes" (Far Eastern Branch of the Russian Academy of Sciences), Vladivostok, 690041, Russian Federation

*Corresponding author:*

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

The paper concludes a series of works which present a base technology for development of intelligent multi-agent cloud services and their components with the use of system tools of IACPaaS platform. The technology is put to reduce the labour-intensiveness of development and first of all support for intelligent cloud services and proposes involvement of domain experts (without mediators or additional training) in these processes, whose task is to create and maintain information components of platforms' applied services in actual state during their lifecycle. This part describes a technology for development of user interface of an intelligent cloud service and describes an example of usage of the base technology for development of applied intelligent services in the field of ophthalmology.

**Keywords:** intelligent system, multi-agent system, intelligent software development technology, hierarchical semantic network, agent-oriented programming, cloud platform, cloud service

**Acknowledgements:** This work was supported by the Russian Foundation for Basic Research, project nos. 13-07-00024, 14-07-00299 and 15-07-03193 and by the Fundamental Scientific Research, Far East program.

For citation:

Gribova V. V., Kleshev A. S., Krylov D. A., Moskalenko Ph. M., Timchenko V. A., Fedorischev L. A., Shalfeyeva E. A. The Base Technology for Intelligent Services Development with the Use of IACPaaS Cloud Platform. Part 3. An Interface Development and an Example of Applied Services Creation, *Programmnaya Ingeneria*, 2016, vol. 7, no. 3, pp. 99-107.

DOI: 10.17587/prin.7.99-107

## References

1. Gribova V. V., Kleshev A. S., Krylov D. A., Moskalenko Ph. M., Timchenko V. A., Shalfeyeva E. A. Bazovaya tekhnologiya razrabotki intellektual'nykh servisov na oblachnoj 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), *Programmnaya Ingeneria*, 2015, no. 12, pp. 3—11 (in Russian).
2. Gribova V. V., Kleshev A. S., Krylov D. A., Moskalenko Ph. M., Timchenko V. A., Shalfeyeva E. A. Bazovaya tekhnologiya razrabotki intellektual'nykh servisov na oblachnoj platforme IACPaaS. Chast' 2. Razrabotka agentov i shablonov soobschenij (A base technology for development of intelligent services with the use of IACPaaS cloud platform. Part 2. A development of agents and message templates), *Programmnaya Ingeneria*, 2016, no. 1, pp. 14—20 (in Russian).
3. **MediaWiki**, available at: <https://www.mediawiki.org/>
4. **Reenskaug Trygve**. MVC. Xerox PARC 1978-79, available at: <http://heim.ifi.uio.no/~trygver/themes/mvc/mvc-index.html>.
5. **Bukhanovskij A. V., Vasil'ev V. N., Vinogradov V. N., Smirnov D. K., Sukhorukov S. A., Yapparov T. G.** CLAVIRE: perspektivnaya tekhnologiya oblachnykh vychislenij vtotogo pokoleniya (CLAVIRE: a perspective technology of cloud computing of the second generation), *Izvestiya Vuzov. Priborostroenie*, 2011, no. 10, pp. 7—13 (in Russian).
6. **NanoCloud** — Vychislitelnye servisy dlya nanotekhnologij (NanoCloud — Computational services for nanotechnologies), available at: <http://cloud7.grid.kiae.ru/about.do> (in Russian).
7. **Kraposhin M. V., Samovarov O. I., Strizhak S. V.** Osobennosti realizatsii Web-laboratorii mekhaniki sploshnoj sredy na baze tekhnologicheskoy platformy "Universitetskij klaster" (Special issues in implementation of a Web-laboratory of mechanics of continuum

on the basis of the "University cluster" platform), *Nauchnij servis v seti Internet: Ekzaflopsnoe budusheye: Trudy Mejdunarodnoy superkompyuternoy konferencii*. 19—24 September, 2011, Novorossiysk. Moscow, Izdatelstvo MGU, 2011, pp. 473—475 (in Russian).

8. **Kostenetskij P. S., Semenov A. I., Sokolinskij L. B.** Sozdanie obrazovatel'noj platformy "Personal'nyj virtual'nyj kompyuter" na baze oblachnykh vychislenij (Creation of educational platform "Personal virtual computer" on the basis of cloud computing), *Nauchnij servis v seti Internet: Ekzaflopsnoe budusheye: Trudy Mejdunarodnoy superkompyuternoy konferencii*. 19—24 September, 2011, Novorossiysk. Moscow, Izdatelstvo MGU, 2011, pp. 374—377 (in Russian).

9. **The Protege** Ontology Editor and Knowledge Acquisition System available at: <http://protege.stanford.edu/>

10. **Golenkov V. V., Gulyakina N. A.** Proekt otkrytoj semanticheskoy tekhnologii komponentnogo proektirovaniya intellektual'nykh sistem. Chast' 1: Principy sozdaniya (A project of an open semantic technology for component design of intelligent systems. Part 1: Principles of creation), *Ontologiya Proektirovaniya*, 2014, no. 1 (11), pp. 42—64 (in Russian).

11. **Golenkov V. V., Gulyakina N. A.** Proekt otkrytoj semanticheskoy tekhnologii komponentnogo proektirovaniya intellektual'nykh sistem. Chast' 2: Unificirovannye modeli proektirovaniya (A project of an open semantic technology for component design of intelligent systems. Part 2: Unified models of design), *Ontologiya Proektirovaniya*, 2014, no. 4 (14), pp. 34—53 (in Russian).

12. **Gribova V. V., Fedorischev L. A.** Programmnij kompleks dlya razrabotki oblachnykh virtualnykh sred (A software complex for development of cloud virtual environments), *Programmnye Produkty i Sistemy*, 2015, no. 2, pp. 60—64 (in Russian).

13. **Gribova V. V., Petryaeva M. V., Fedorischev L. A.** Kompyuternij obuchayuschij trenazhor s virtualnoj realnostyu dlya oftalmologii (A computer-based teaching simulator of virtual reality for ophthalmology), *Otkrytoe Obrazovaniye*, 2013, no. 6, pp. 45—51 (in Russian).

**В. А. Стенников**, д-р техн. наук, зам. директора, e-mail: sva@isem.irk.ru,  
**Е. А. Барахтенко**, канд. техн. наук, ст. науч. сотр., e-mail: barakhtenko@isem.irk.ru,  
**Д. В. Соколов**, канд. техн. наук, ст. науч. сотр., e-mail: sokolov\_dv@isem.irk.ru,  
Федеральное государственное бюджетное учреждение науки Институт систем энергетики им. Л. А. Мелентьева Сибирского отделения Российской академии наук (ИСЭМ СО РАН), г. Иркутск

## Применение концепции Model-Driven Engineering в программном комплексе для определения оптимальных параметров теплоснабжающих систем

*Изложен опыт применения концепции Model-Driven Engineering при разработке программного обеспечения нового поколения СОСНА для определения оптимальных параметров теплоснабжающих систем. СОСНА представляет собой сложный прикладной программный комплекс, который в автоматизированном режиме создается на основе компьютерной модели исследуемой теплоснабжающей системы, формального описания решаемой задачи и знаний о предметной области в виде онтологий.*

**Ключевые слова:** Model-Driven Engineering, методика, метапрограммирование, онтология, программный комплекс, автоматизация программирования, архитектура программного обеспечения, теплоснабжающая система, методы оптимизации

### Введение

Теплоснабжающие системы (ТСС) в условиях сурового российского климата имеют высокую социальную и экономическую значимость. В настоящее время они превратились в сложные пространственно-распределенные трубопроводные системы централизованного снабжения потребителей тепловой энергией. Сложность ТСС связана с их замкнутыми двухлинейными схемами, многокольцевой структурой, наличием большого числа теплоисточников и управляющих элементов (насосных станций, регуляторов давления и расхода теплоносителя, тепловых пунктов). В современных условиях возрастают требования к эффективности, качеству теплоснабжения потребителей и надежности ТСС. Такие требования приводят к необходимости разработки и применения эффективных методов и программного обеспечения (ПО) для решения задач проектирования ТСС.

Важное значение для обеспечения эффективной работы ТСС имеет задача определения ее оптимальных параметров. Эта задача может иметь самостоятельное значение, а также рассматриваться в качестве подзадачи общего процесса, направленного на проектирование ТСС [1, 2]. Большие размеры ТСС и вычислительная сложность используемых моделей, методов и алгоритмов не позволяют без применения специализированного ПО решать задачи поиска оптимальных параметров ТСС.

В статье изложен опыт применения концепции Model-Driven Engineering (управляемая моделью разработка, MDE) при построении ПО нового поколения для определения оптимальных параметров ТСС. Эта концепция представляет собой совокупность методических подходов автоматизированного построения сложных программных систем на основе предварительно разработанных моделей [3, 4]. На базе MDE реализован программный комплекс (ПК) СОСНА (Синтез Оптимальных Систем с учетом Надежности), предназначенный для определения оптимальных параметров ТСС. Программный комплекс СОСНА создан в автоматизированном режиме на основе компьютерной модели исследуемой ТСС, формального описания решаемой задачи и знаний о предметной области, хранение которых организовано в виде онтологий [5, 6]. Предложенная в статье методика имеет универсальный характер и может применяться при разработке ПО для решения широкого спектра задач проектирования ТСС и других трубопроводных систем.

При построении ПК используются современные технологии метапрограммирования [7]. Применение технологий метапрограммирования позволяет настраивать ПО на расчет систем с разным оборудованием (стальные и металлопластиковые трубы разных диаметров, насосы и пр.) и различными способами его сооружения.

## Задача определения оптимальных параметров теплоснабжающих систем и методы ее решения

Постановка задачи оптимизации параметров ТСС представляется в следующем виде. Заданными являются:

- схема тепловой сети из  $m$  узлов и  $n$  ветвей, представляемая в виде ориентированного графа  $G_{ТС} = (J, I)$ , где  $J$  — множество вершин (узлов);  $I$  — множество дуг (ветвей);  $J = J_{\Pi} \cup J_{И} \cup J_{Р}$ , где  $J_{\Pi}$ ,  $J_{И}$  и  $J_{Р}$  — множества потребителей, источников и точек разветвления на схеме соответственно;  $I = I_{Л} \cup I_{И} \cup I_{\Pi}$ , где  $I_{Л} = I_{Н} \cup I_{С}$  — множество ветвей линейной части сети, состоящее из множеств существующих  $I_{С}$  и новых  $I_{Н}$  ветвей;  $I_{\Pi}$  и  $I_{И}$  — множества ветвей-потребителей и ветвей-источников соответственно;  $I_{НС} \subset I_{И} \cup I_{Л}$  — ветви, на которых установлены или разрешены насосные станции (НС);

- длины трубопроводов на ветвях сети  $L_i, i \in I_{Л}$ ;
- множество диаметров стандартных трубопроводов  $D$ ;
- множество номеров всех возможных для сети способов сооружения (реконструкции) трубопроводов  $U$ ;
- множества номеров допустимых для каждой ветви способов сооружения (реконструкции) трубопровода  $U_i \subset U, i \in I_{Л}$ ;

- нижние  $(P_j^H, j \in J)$  и верхние  $(P_j^B, j \in J)$  ограничения на давление;
- нижние  $(v_i^H, i \in I_{Л})$  и верхние  $(v_i^B, i \in I_{Л})$  ограничения на скорость течения теплоносителя на ветвях;
- вектор узловых отборов и притоков теплоносителя  $\mathbf{Q} = (Q_1, \dots, Q_m)^T$ ;
- множество напоров НС  $T$ ;
- ограничения на минимальный располагаемый напор у потребителей  $\Delta H_i^H, i \in I_{\Pi}$ .

В результате решения задачи необходимо определить оптимальные параметры ТСС:

- вектор диаметров трубопроводов  $\mathbf{d} = (d_1, \dots, d_n)^T$ ;
- вектор  $\mathbf{u} = (u_1, \dots, u_n)^T$ , компоненты  $(u_i \in U_i, i \in I_{Л})$  которого содержат номера оптимальных способов сооружения (реконструкции) трубопроводов;
- вектор напоров на НС  $\mathbf{H} = (H_1, \dots, H_n)^T$ ;
- вектор расходов теплоносителя на ветвях

$\mathbf{x} = (x_1, \dots, x_n)^T$ ;

- вектор давлений в узлах  $\mathbf{P} = (P_1, \dots, P_m)^T$ .

Требуется минимизировать функцию общих затрат в ТСС, имеющую следующий вид:

$$Z(\mathbf{d}, \mathbf{u}, \mathbf{H}, \mathbf{x}, \mathbf{P}) = \sum_{i \in I_{Л}} Z_i^C(d_i, u_i) + \sum_{i \in I_{НС}} Z_i^{HC}(H_i, x_i) + \sum_{i \in I_{Л}} Z_i^{\exists}(x_i, d_i) + \sum_{i \in I_{Л}} Z_i^T(d_i) + \sum_{\substack{i \in I_{\Pi} \\ \varphi(i)=(j,k)}} Z_i^{\Pi}(x_i, P_j, P_k) \rightarrow \min, \quad (1)$$

где  $Z_i^C$  — затраты на сооружение и эксплуатацию трубопровода на ветви сети;  $Z_i^{HC}$  — затраты на со-

оружение и эксплуатацию НС;  $Z_i^{\exists}$  — стоимость электроэнергии, расходуемой на перекачку теплоносителя;  $Z_i^T$  — стоимость тепловых потерь;  $Z_j^{\Pi}$  — затраты на электроэнергию, расходуемую на подачу теплоносителя потребителю;  $\varphi(i)$  — функция, сопоставляющая каждой ветви  $i \in I$  пару инцидентных узлов  $(j, k)$ .

Модель потокораспределения в ТСС имеет следующий вид:

$$\mathbf{A}\mathbf{x} - \mathbf{Q} = 0, \quad (2)$$

$$\mathbf{A}^T \mathbf{P} + \mathbf{H} - \mathbf{f}(\mathbf{x}, \mathbf{s}) = 0, \quad (3)$$

где  $\mathbf{A}$  — матрица инцидентности узлов и ветвей расчетной схемы размера  $m \times n$ ;  $\mathbf{s} = (s_1, \dots, s_n)^T$  — вектор гидравлических сопротивлений ветвей с элементами  $s_i = s_i(d_i)$ ,  $i \in I$ ;  $\mathbf{f}(\mathbf{x}, \mathbf{s})$  —  $n$ -мерная вектор-функция с элементами  $f_i(s_i, x_i) = s_i x_i |x_i|^{\beta-1}$ ,  $i = \overline{1, n}$ , отражающими законы падения давления на ветвях сети;  $\beta$  — показатель степени, зависящий от типа трубопровода и режима течения теплоносителя.

Система условий и ограничений включает:

- ограничение на давление в узлах

$$P_j^H \leq P_j \leq P_j^B, \quad j \in J; \quad (4)$$

- ограничение на скорость течения теплоносителя по ветвям

$$v_i^H \leq v_i(x_i) \leq v_i^B, \quad i \in I_{Л}; \quad (5)$$

- условие дискретности диаметров трубопроводов

$$d_i \in D, \quad i \in I_{Л}; \quad (6)$$

- условие дискретности видов реконструкции трубопроводов

$$u_i \in U_i \subset U, \quad i \in I_{Л}; \quad (7)$$

- условие дискретности напоров НС

$$H_i \in T, \quad i \in I_{НС}; \quad (8)$$

- ограничение на располагаемый напор у потребителей

$$P_j - P_k \geq \Delta H_i^H, \quad \varphi(i) = (j, k), \quad i \in I_{\Pi}. \quad (9)$$

В результате решения задачи минимизации функции (1) при условиях (2)-(9) необходимо определить оптимальные параметры ТСС.

В рамках основанной в ИСЭМ СО РАН теории гидравлических цепей (ТГЦ) разработаны эффективные методы решения задачи определения оптимальных параметров ТСС. Для разветвленных сетей разработан метод пошаговой оптимизации на базе динамического программирования (ДП) [1, 2]. Для кольцевых (многоконтурных) сетей разработан метод многоконтурной оптимизации (МКО), основанный

на принципе последовательного улучшения решения [1, 2, 8]. Важное свойство перечисленных методов состоит в том, что они позволяют гибко настраивать вычислительную процедуру на математические модели используемого набора оборудования.

Аддитивная целевая функция (1) позволяет применять ДП для решения задачи определения оптимальных параметров разветвленных ТСС: для сетей этого типа расходы на ветвях  $x$  однозначно определяются древовидной структурой и узловыми расходами  $Q$ . Вычислительная процедура, основанная на ДП, состоит в многошаговом процессе определения параметров элементов сети (ветвей и узлов) путем их последовательного подбора в направлении от потребителей к источнику [1, 2].

Идея метода МКО состоит в декомпозиции сложной задачи оптимизации параметров кольцевой ТСС на две менее сложные подзадачи [2, 8]:

1) определение оптимальных параметров (диаметров  $d$  и напоров  $H$ ) выделенного дерева сети методом ДП при фиксированных расходах  $x$ ;

2) расчет потокораспределения в кольцевой сети (расходов  $x$  и давлений  $P$ ) при фиксированных диаметрах  $d$  и напорах  $H$ .

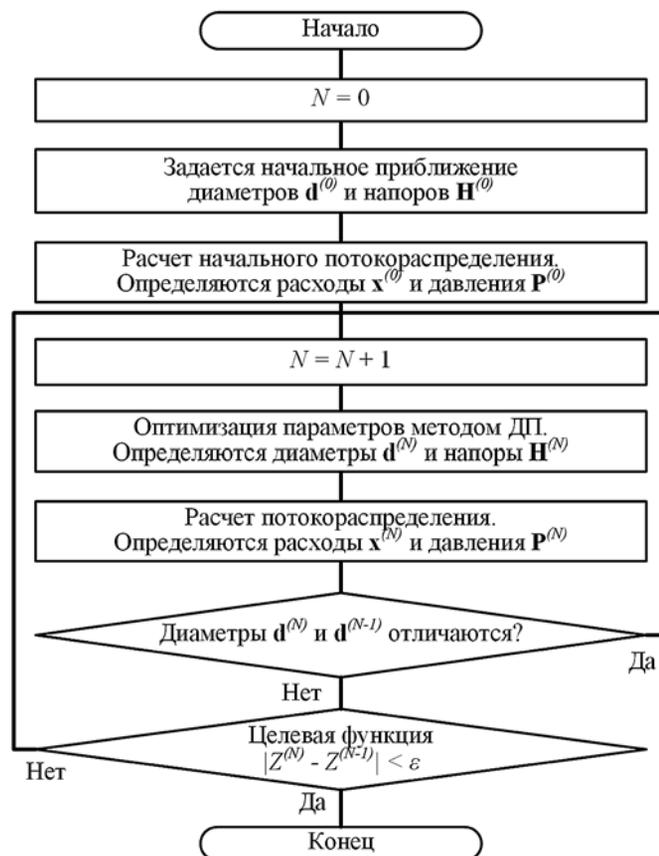


Рис. 1. Блок-схема алгоритма определения оптимальных параметров кольцевой тепловой сети методом многоконтурной оптимизации:

$N$  — номер итерации;  $\epsilon$  — число, задающее точность вычислительного процесса

Метод МКО предполагает организацию итерационного вычислительного процесса последовательного улучшения параметров ТСС, в ходе которого попеременно решаются перечисленные подзадачи (рис. 1). Критерием останова вычислительного процесса является прекращение уменьшения целевой функции или совпадение диаметров трубопроводов на соседних итерациях. Таким образом, последовательное решение выделенных подзадач приведет к получению решения, которое уже не может быть улучшено в результате применения метода МКО.

Задача расчета потокораспределения в ТСС сводится к решению системы линейных и нелинейных алгебраических уравнений, причем различают две основные формы записи таких систем — узловую и контурную модели потокораспределения [1, 2]. Постановка задачи в случае использования узловой модели потокораспределения состоит в следующем. Заданы: матрица инцидентности  $A$ , векторы узловых расходов  $Q$  и напоров  $H$ , вектор-функция гидравлических характеристик  $f(x, s)$  и давление  $P'_m$  в узле  $m$ . Необходимо определить векторы расходов на ветвях  $x$  и давлений в узлах  $P$ , удовлетворяющие системе уравнений (2), (3). Для решения этой задачи используют метод узловых давлений (МД) [1, 2]. В случае постановки задачи в контурной форме для решения используют метод контурных расходов (МКР) [1, 2].

### Описание проблемы и постановка задач исследования

Развитие рынка теплопроводов, оборудования и технологий, применяемых при сооружении ТСС, значительно расширяет возможности реализации тех или иных технических решений. В свою очередь, расширение этих возможностей требует их учета при решении задачи определения оптимальных параметров ТСС, так как каждый тип применяемого оборудования имеет свои отличительные характеристики и представляется своим набором математических моделей, описывающих его параметры и технико-экономические зависимости.

Конкретные математические модели для расчета составляющих функции затрат (1) ( $Z_i^C$ ,  $Z_i^{HC}$ ,  $Z_i^{\Delta}$ ,  $Z_i^T$ ,  $Z_i^{\Pi}$ ) и гидравлических характеристик ветвей ( $s_i$ ,  $f_i(s_i, x_i)$ ,  $v_i(x_i)$ ) выбирают в зависимости от свойств используемого оборудования (материал, компонентный состав, способ сооружения, срок использования и др.).

Приведем несколько примеров математических моделей, используемых при решении задачи определения оптимальных параметров ТСС.

Функция  $f_i(s_i, x_i)$ , отражающая закон падения давления на ветви сети для стального трубопровода, имеет следующий вид ( $\beta = 2$ ):

$$f_i(s_i, x_i) = s_i x_i |x_i|.$$

Гидравлическое сопротивление ветви  $s_i$ , на которой установлен стальной трубопровод, может быть рассчитано по формуле

$$s_i = \lambda_i \frac{(1 + \alpha_i)L_i}{d_i^{5,25}},$$

где  $\lambda_i$  — коэффициент, зависящий от эквивалентной шероховатости трубопровода на ветви;  $\alpha_i$  — коэффициент местных потерь на ветви.

Затраты на сооружение и эксплуатацию нового трубопровода  $Z_i^C$  рассчитывают по формуле

$$Z_i^C(d_i) = (a + f_c)K(d_i)L_i,$$

где  $a$  — коэффициент дисконтирования;  $f_c$  — доля амортизационных отчислений на трубопровод;  $K(d_i)$  — удельные капиталовложения в трубопровод.

В ИСЭМ СО РАН на протяжении 40 лет разрабатывается ПО для решения задач определения оптимальных параметров ТСС [1, 2]. Модули ПО предыдущего поколения, реализующие методы и алгоритмы решения этой задачи, ориентированы на конкретный набор оборудования, что значительно затрудняет их настройку под решение задач расчета сетей с другим набором оборудования. В связи с этим обстоятельством возникла необходимость в разработке и применении новых адаптивных подходов, которые позволяют настраивать ПО на моделирование реальных ТСС с широким набором используемого оборудования. Для реализации ПК нового поколения для определения оптимальных параметров ТСС необходимо разработать новую методику, которая позволит гибко адаптировать ПО под особенности развития и состав оборудования исследуемой ТСС.

### Методика модельно-управляемого построения программного комплекса

В результате проведенных авторами исследований предложена новая методика модельно-управляемого построения ПК на базе концепции MDE [9]. Эта методика позволяет выполнять автоматизированное построение ПО как для определения оптимальных параметров ТСС, так и для решения других, смежных задач проектирования ТСС. Приведем основные положения этой методики.

1. Объединение компьютерной модели конкретной ТСС, моделей элементов ТСС (т. е. отдельных подсистем) и методов (алгоритмов) происходит только в контексте решения прикладной задачи (рис. 2).

2. Компоненты ПК, зависящие от свойств моделируемой ТСС (набор компонентов-моделей элементов ТСС), создаются в автоматизированном режиме при помощи метапрограммирования на основе компьютерной модели этой ТСС и онтологий, содержащих описание оборудования и решаемых прикладных задач.

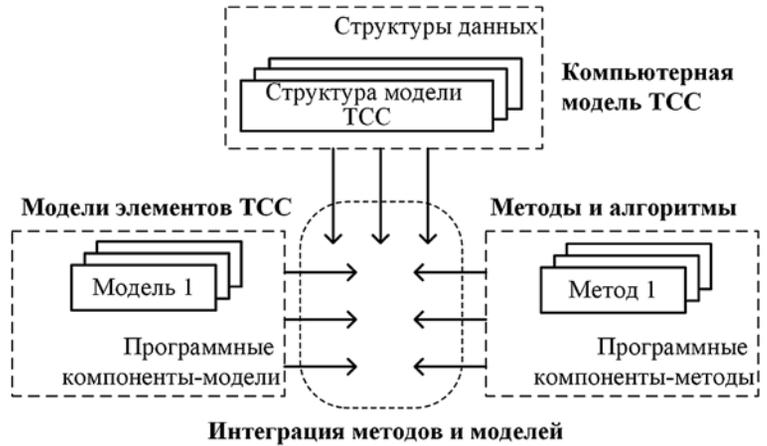


Рис. 2. Процесс интеграции методов и моделей элементов ТСС во время решения прикладной задачи

3. Программный комплекс, ориентированный на моделирование конкретной ТСС, строится в контексте решения прикладной задачи с помощью метапрограммирования на основе компонентов-моделей элементов ТСС, построенных в автоматизированном режиме, и компонентов, реализующих методы и алгоритмы решения прикладных задач. Процесс построения ПК управляется знаниями, хранящимися в онтологиях: онтологии ТСС, онтологии задач и онтологии ПО.

В качестве средства формального представления онтологий используется язык XML [5], на основе которого разработан собственный предметно-ориентированный язык. Для хранения математических моделей элементов ТСС используется формат MathML<sup>1</sup>.

Онтология ТСС включает следующие составляющие: описание иерархической структуры построения ТСС, классификацию используемого оборудования тепловых сетей и его технические характеристики, описание элементов ТСС и их параметров (технические характеристики, гидравлические параметры и граничные условия) и классификацию и описание используемых математических моделей (например, уравнения, описывающие законы падения напора на ветвях сети, формулы расчета сопротивлений и др.). На рис. 3 представлен фрагмент онтологии ТСС.

Онтология задач содержит описание прикладных задач (например, оптимизации параметров многоконтурной сети) и методов их решения (например, метода МКО), описание алгоритмов, перечисление параметров, являющихся исходными данными, и параметров, получаемых в результате решения задачи.

Онтология ПО предназначена для хранения знаний, необходимых для автоматизации построения и использования ПО. Эта онтология содержит описание программных компонентов и их свойств, метаданные (входные и выходные параметры, описание форматов данных), описание технологий и интерфейсов доступа к программным компонентам.

<sup>1</sup> Mathematical Markup Language. Сайт проекта: <http://www.w3.org/Math/>



Рис. 3. Фрагмент онтологии ТСС

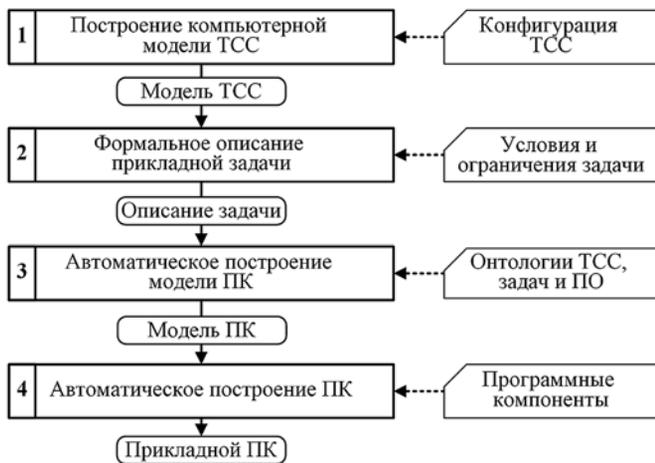


Рис. 4. Методика модельно-управляемого построения ПК

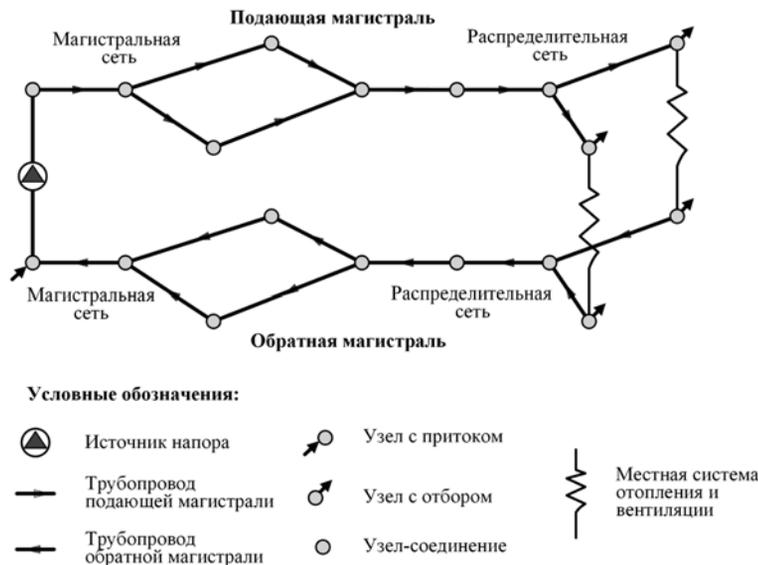


Рис. 5. Модель ТСС в двухлинейном изображении

Разработанная методика ориентирована на применение Java [10] в качестве основного языка и платформы программирования.

Методика модельно-управляемого построения ПК включает четыре этапа. Схема, отражающая взаимосвязь этих этапов, представлена на рис. 4.

**Этап 1. Построение компьютерной модели конкретной ТСС.** На этом этапе инженером создается компьютерная иерархическая модель конкретной ТСС, представляющая собой ориентированный граф, вершины которого соответствуют узлам (источники, потребители, узлы-соединения), а дуги — ветвям (пассивные ветви, активные ветви с НС). Для каждого элемента модели указывают установленное на нем оборудование, его характеристики и способы сооружения. Построенная модель сохраняется в базе данных (БД) для многократного использования. На рис. 5 приведена модель ТСС в двухлинейном изображении, содержащая один источник тепла, магистральную и распределительную сети, двух потребителей тепловой энергии. Для создания моделей реальных ТСС, состоящих из тысяч элементов, используется графический редактор схем. Этот редактор позволяет создавать расчетные схемы ТСС на плане местности и работать с ними.

**Этап 2. Формальное описание прикладной задачи** На этом этапе инженером осуществляется формальное описание прикладной задачи, состоящее в том, что задаются условия и ограничения задачи определения оптимальных параметров ТСС, включающие в себя: набор стандартных трубопроводов и их характеристики; ограничения для каждой ветви сети на способы сооружения (реконструкции) трубопровода; ограничения на давление в узлах; ограничения на скорость течения теплоносителя; проектные значения отборов теплоносителя у потребителей и его притоков на источниках; набор стандартных НС и их характеристики.

**Этап 3. Автоматическое построение модели программного комплекса.** На этом этапе автоматически создается модель ПК, ориентированного на решение прикладной задачи. Эта модель представляет собой совокупность структур данных, отражающих следующие свойства ПК: структуру и состав программных компонентов, описание вычислительного процесса в виде ориентированного графа, описание математических моделей оборудования, список используемых методов и алгоритмов. Построение модели ПК осуществляется машиной вывода, которая во время своей работы использует компьютерную модель ТСС, формальное описание задачи и онтологии. Для логического вывода на онтологиях, представленных в формате XML, используются правила вывода, реализованные на языках XPath и XSLT [5, 11].

Алгоритм автоматизированного построения модели ПК предполагает выполнение следующих шагов.

**Шаг 1.** На основе компьютерной модели ТСС формируется список использованного при построении тепловой сети оборудования и способов его сооружения.

**Шаг 2.** На основе формального описания решаемой задачи определения оптимальных параметров ТСС формируется список допустимого к установке при реконструкции сети нового оборудования и способов его сооружения.

**Шаг 3.** На основе сформированных на предыдущих шагах списков оборудования и способов его сооружения машина вывода, используя онтологию ТСС, строит структуры данных, содержащие описание моделей элементов ТСС, необходимых для решения задачи.

**Шаг 4.** На основе компьютерной модели ТСС, формального описания прикладной задачи и онтологии задач выбирается конкретная методика решения, определяется список подзадач, выбираются методы и алгоритмы.

**Шаг 5.** На основе онтологии задач строится ориентированный граф, описывающий ход вычислительного процесса при решении задачи определения оптимальных параметров ТСС, узлы которого соответствуют этапам решения задачи, а дуги — связям между ними.

**Шаг 6.** На базе списка методов и алгоритмов машина вывода, используя онтологию ПО, создает список программных компонентов, необходимых для решения задачи определения оптимальных параметров ТСС.

**Шаг 7.** Формируются структуры данных, описывающие связи "задача—метод—компонент".

**Шаг 8.** На основе построенного на предыдущих шагах описания компонентов, реализующих методы и алгоритмы, и онтологии ПО определяются интерфейсы компонентов, реализующих математические модели элементов ТСС.

**Шаг 9.** На основе онтологии задач и онтологии ПО формируются структуры данных, описывающие входные и выходные параметры для каждого программного компонента, типы данных и способы их передачи.

Схема процесса построения структур данных, описывающих модели элементов ТСС, представлена на рис. 6.

**Этап 4. Автоматическое построение программного комплекса на основе его модели.** На этом этапе выполняется автоматическое

построение ПК на основе его модели при помощи технологий метапрограммирования. В процессе построения решаются три следующие подзадачи.

**Подзадача 1.** Создание набора программных компонентов, реализующих математические модели элементов ТСС (рис. 7). Ниже приведен алгоритм построения набора компонентов-моделей элементов ТСС, состоящий из трех шагов.

**Шаг 1.** Компонент управления построением вызывает XSLT-процессор для создания моделей элементов, необходимых для решения прикладной задачи. XSLT-процессор на основе списка оборудования из модели ПК, математических моделей в формате MathML и правил преобразования в формате XSLT создает набор структур данных, содержащих описание моделей элементов ТСС.

**Шаг 2.** Управляющий компонент вызывает построитель кода, который на основе структур данных, описывающих модели элементов ТСС, и описания интерфей-

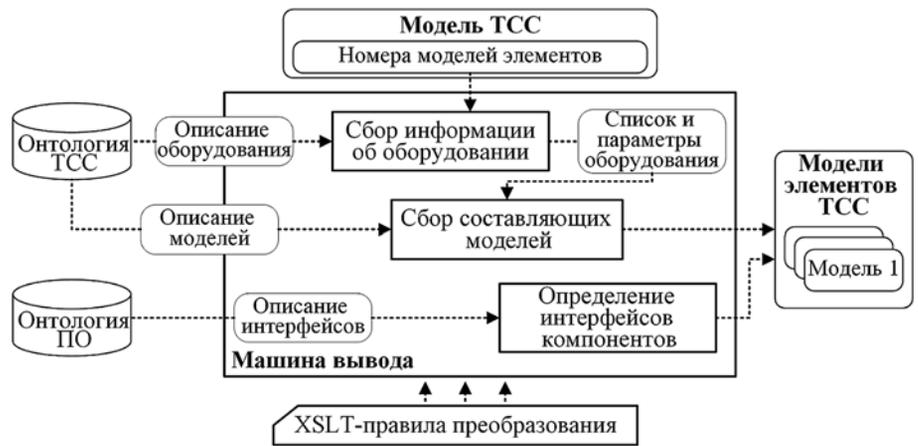


Рис. 6. Схема построения моделей элементов ТСС

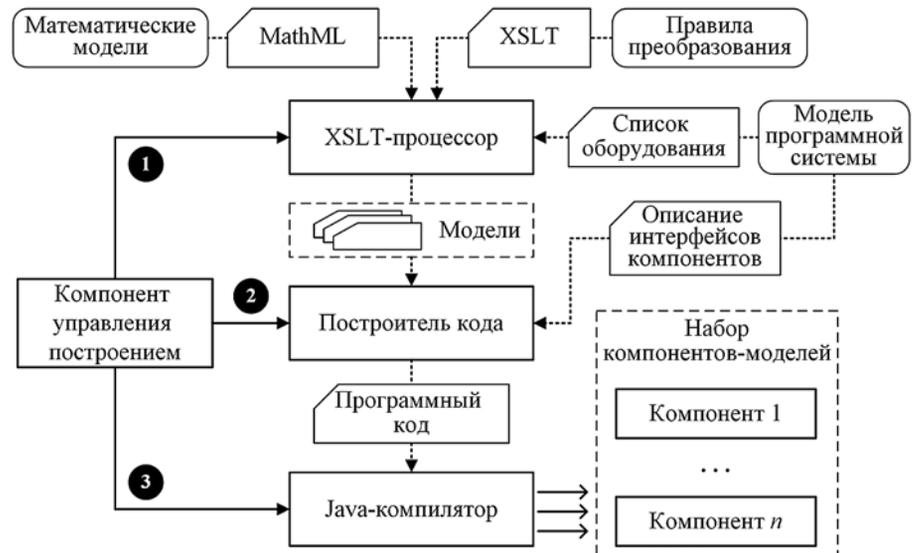


Рис. 7. Динамическое создание набора компонентов-моделей

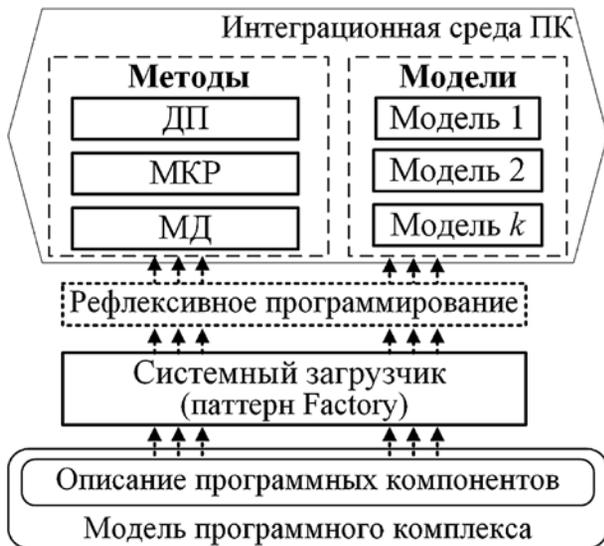


Рис. 8. Схема загрузки компонентов в интеграционную среду ПК при его построении

сов программных компонентов из модели ПК строит программный код на языке программирования Java.

**Шаг 3.** Управляющий компонент вызывает компилятор языка программирования Java, который выполняет компиляцию созданного на предыдущем шаге программного кода для создания набора необходимых компонентов-моделей.

**Подзадача 2.** Загрузка в память программных компонентов, реализующих методы и математические модели, средствами рефлексивного программирования, которое является одним из видов метапрограммирования [12]. Интеграция этих компонентов в единый ПК осуществляется через стандартизированные

интерфейсы, обеспечиваемые паттернами проектирования [13].

Системный загрузчик, реализующий паттерн проектирования Factory, выполняет следующие действия (рис. 8): получает из модели ПК описание программных компонентов; находит компоненты, соответствующие решаемой задаче; подготавливает необходимые для их вызова структуры данных; загружает в память компоненты, используя средства рефлексии языка программирования Java; передает ссылки на компоненты в интеграционную среду ПК.

**Подзадача 3.** Заполнение хеш-таблиц парами "номер — ссылка". Эти хеш-таблицы используются программными компонентами-методами, которые по номерам математических моделей элементов, содержащимся в модели ТСС, получают ссылки на соответствующие программные компоненты-модели.

Программный комплекс, получаемый в результате автоматизированного построения в соответствии с предложенной методикой, состоит из трех архитектурных слоев (рис. 9): подсистемы управления вычислением (супервизора), которая содержит компоненты, управляющие ходом вычисленного процесса; вычислительной подсистемы, которая решает прикладную задачу путем использования программных компонентов, реализующих методы, алгоритмы и модели; подсистемы хранения данных, которая обеспечивает обмен информацией между БД и локальной памятью.

### Практическое применение программного комплекса СОСНА

Разработанный ПК СОСНА успешно применяется в ИСЭМ СО РАН для решения задач определения оптимальных параметров ТСС при решении вопросов проектирования, развития и реконструкции

ТСС. Предложенная методика модельно-управляемого построения ПК обеспечивает возможность его гибкого использования при моделировании ТСС с широким набором оборудования.

На основе ПК СОСНА были выполнены многовариантные расчеты, которые позволили определить оптимальные параметры и получить рекомендации по рациональной реконструкции и развитию ТСС Центрального и Адмиралтейского районов Санкт-Петербурга, Братска (рис. 10) и поселка городского типа Магистральный. Предложенные рекомендации по реконструкции ТСС обеспечивают их адаптацию к возросшим перспективным тепловым нагрузкам и будут способствовать реализации оптимальных режимов их функционирования.

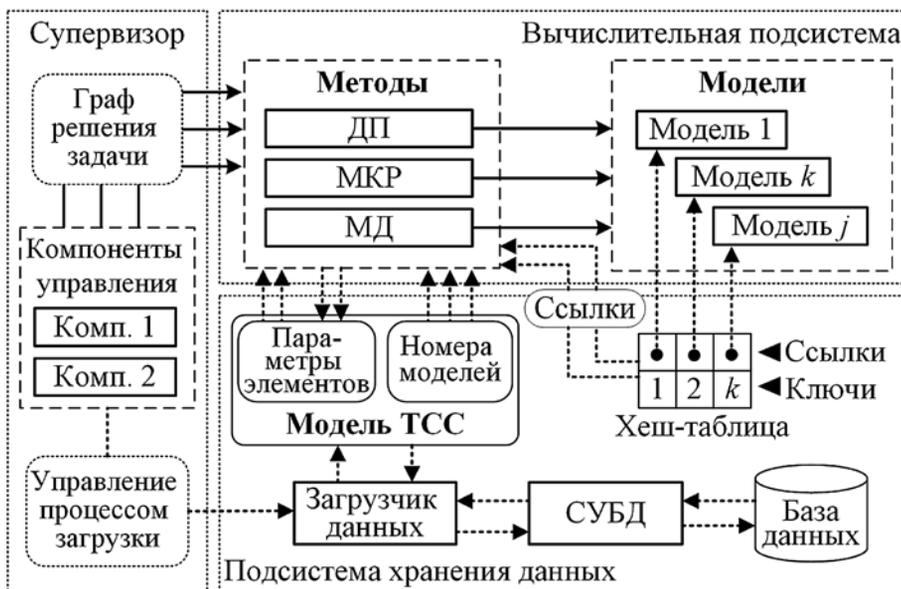


Рис. 9. Архитектура ПК для оптимизации ТСС

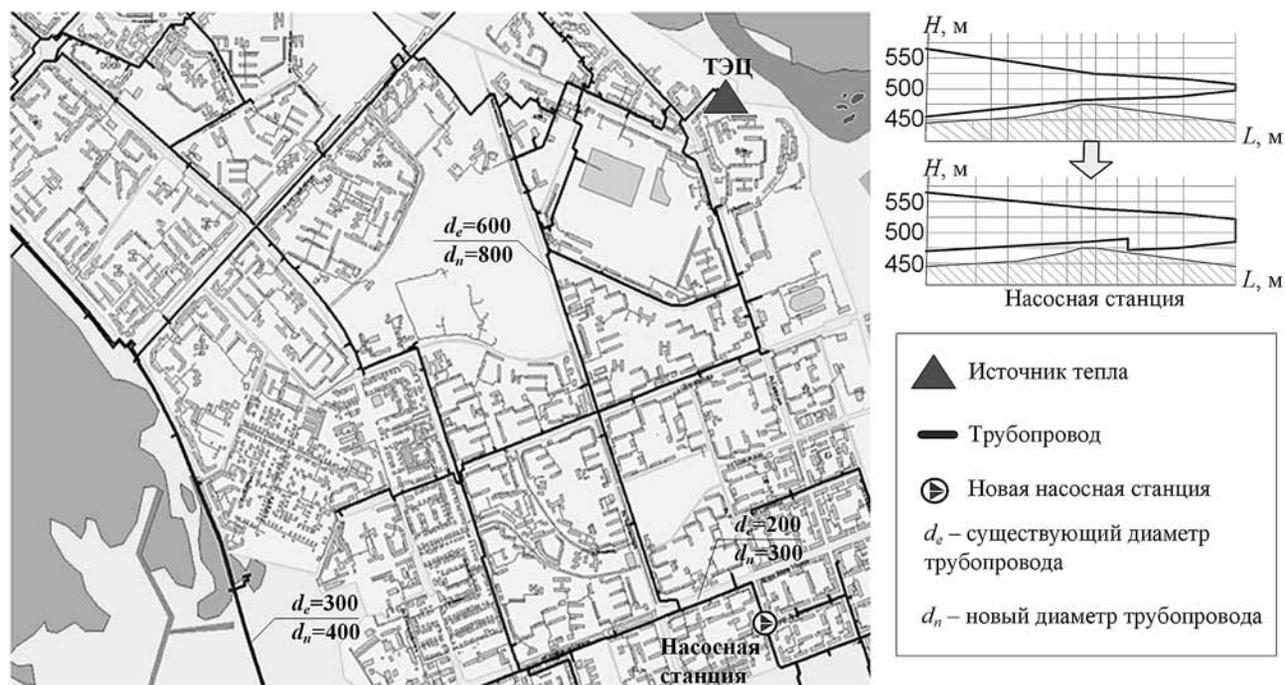


Рис. 10. Фрагмент схемы ТСС Братска с обозначенными рекомендуемыми мероприятиями по ее реконструкции

### Заключение

Предложена оригинальная методика модельно-управляемого построения ПК СОСНА для решения сложной инженерной задачи определения оптимальных параметров ТСС. Эта методика основана на применении концепции MDE, которая позволяет в контексте решения прикладной задачи создавать сложные ПК, ориентированные на особенности построения и рекомендуемый состав оборудования исследуемой ТСС. Программный комплекс может применяться в научно-исследовательских, проектных и эксплуатационных организациях, занимающихся вопросами теплоснабжения, и позволяет получать рекомендации по реконструкции и развитию ТСС, обеспечивающие повышение эффективности их работы и необходимое качество снабжения потребителей тепловой энергией.

*Исследования выполняются при частичной финансовой поддержке Российского фонда фундаментальных исследований (грант № 13-07-00297).*

### Список литературы

1. Меренков А. П., Хасилев В. Я. Теория гидравлических цепей. М.: Наука, 1985. 280 с.
2. Меренков А. П., Сеннова Е. В., Сумароков С. В. и др. Математическое моделирование и оптимизация систем тепло-, водо-, нефте- и газоснабжения. Новосибирск: Наука, 1992. 407 с.

3. Brambilla M., Cabot J., Wimmer M. Model Driven Software Engineering in Practice: Synthesis Lectures on Software Engineering N. 1. San Rafael: Morgan & Claypool, 2012. 182 p.
4. Völter M., Stahl T., Bettin J. et al. Model-Driven Software Development: Technology, Engineering, Management. New York: Wiley, 2006. 446 p.
5. Коголовский М. Р. Перспективные технологии информационных систем. М.: ДМК-Пресс, 2003. 288 с.
6. Стенников В. А., Барахтенко Е. А., Соколов Д. В. Применение онтологий при реализации программного комплекса для решения задач оптимального проектирования теплоснабжающих систем // Информационные технологии. 2013. № 3. С. 2–7.
7. Bartlett J. The art of metaprogramming, Part 1: Introduction to metaprogramming. IBM homepage. URL: <http://www.ibm.com/developerworks/library/l-metaprogl/>
8. Сумароков С. В. Метод решения многоэкстремальной сетевой задачи // Экономика и математические методы. 1976. Том 12, № 5. С. 1016–1018.
9. Стенников В. А., Барахтенко Е. А., Соколов Д. В. Методический подход на основе концепции Model-Driven Engineering и онтологий к разработке программного обеспечения для проектирования теплоснабжающих систем // Информационные технологии. 2015. Том 21, № 3. С. 201–209.
10. Gosling J., Joy B., Steele G. et al. Java Language Specification, Java SE 8 Edition. Boston: Addison-Wesley, USA, 2014. 792 p.
11. Kay M. XSLT 2.0 and XPath 2.0 Programmer's Reference. 4th Edition. New York: Wrox, 2008. 1376 p.
12. Forman I., Forman N. Java Reflection in Action. Greenwich: Manning Publications Co., 2005. 273 p.
13. Gamma E., Helm R., Johnson R., Vlissides J. Design Patterns: Elements of Reusable Object-Oriented Software. Boston: Addison-Wesley, USA, 1994. 395 p.

# Model-Driven Engineering in the Software for Parameter Optimization of Heat Supply Systems

V. A. Stennikov, sva@isem.irk.ru, E. A. Barakhtenko, barakhtenko@isem.irk.ru,  
D. V. Sokolov, sokolov\_dv@isem.irk.ru, Melentiev Energy Systems Institute of Siberian Branch of the Russian Academy of Sciences, Irkutsk, 664033, Russian Federation

Corresponding author:

**Barakhtenko Evgeny A.**, Senior Researcher, Melentiev Energy Systems Institute of Siberian Branch of the Russian Academy of Sciences, Irkutsk, 664033, Russian Federation, e-mail: barakhtenko@isem.irk.ru

Received on June 04, 2015  
Accepted on December 21, 2015

*This paper presents a new methodology for the creation of software for parameter optimization of heat supply systems. The methodology is based on the Model-Driven Engineering paradigm. The essence of this paradigm is that the software is generated on the basis of formal description represented by models. This methodology allows us to automate the process of software creation. The ontologies of heat supply systems, problems, and software are a means of representing the models.*

*The authors suggest metaprogramming to create a software architecture flexibly adjustable to the problem of parameter optimization of a concrete heat supply system in the course of the problem-solving process. Metaprogramming technologies allow us to develop the software that changes or creates software components when solving the problem.*

*The proposed methodology of software development includes four stages.*

*Stage 1. Development of a computer model of the heat supply system.*

*Stage 2. Formalization of the applied problem.*

*Stage 3. Automatic building of the software model.*

*Stage 4. Automatic building of the software on the basis of the model.*

*The suggested methodology underlies the software SOSNA intended for solving the parameter optimization problems of heat supply systems. The software makes it possible to calculate large-scale systems which have a complex structure with any set of nodes, sections, and circuits. The use of the software for the control of heat supply system development will enhance their energy efficiency and cost-effectiveness. The software was applied to solving the optimal reconstruction problems of urban heat supply systems.*

**Keywords:** Model-Driven Engineering, methodology, metaprogramming, ontology, software, automation of programming, software architecture, software engineering, heat supply system, optimization methods, nonlinear programming

**Acknowledgements:** This work was supported by the Russian Foundation for Basic Research, project no. 13-07-00297.

*For citation:*

**Stennikov V. A., Barakhtenko E. A., Sokolov D. V.** Model-Driven Engineering in the Software for Parameter Optimization of Heat Supply Systems, *Programmnaya Ingeneria*, 2016, vol. 7, no. 3, pp. 108–116.

DOI: 10.17587/prin.7.108-116

## References

1. **Merenkov A. P., Khasilev V. Ya.** *Teoriya gidravlicheskih tsepei* (The theory of hydraulic circuits). Moscow, Nauka, 1985, 280 p. (in Russian).
2. **Merenkov A. P., Sennova E. V., Sumarokov S. V., Sidler V. G., Novitskii N. N.** *Matematicheskoe modelirovanie i optimizatsiya sistem teplo-, vodo-, nefte- i gazosnabzheniya* (Mathematical modeling and optimization of heat, water, oil and gas supply systems). Novosibirsk, Nauka, 1992, 407 p. (in Russian).
3. **Brambilla M., Cabot J., Wimmer M.** *Model Driven Software Engineering in Practice. Synthesis Lectures on Software Engineering* N. 1. San Rafael, Morgan & Claypool Publishers, 2012. 182 p.
4. **Volter M., Stahl T., Bettin J., Haase A., Helsen S.** *Model-Driven Software Development: Technology, Engineering, Management*. New York, Wiley, 2006, 446 p.
5. **Kogalovskii M. R.** *Perspektivnye tekhnologii informatsionnykh sistem* (Perspective technologies of information systems). Moscow, DMK-Press, 2003, 288 p. (in Russian).
6. **Stennikov V. A., Barakhtenko E. A., Sokolov D. V.** Prime-nenie ontologii pri realizatsii programmogo kompleksa dlya resheniya zadach optimal'nogo proektirovaniya teplosnabzhayushchikh system (Ontologies application for implementation of the software for solving optimal design problems of heat supply systems), *Informacionnye Tehnologii*, 2013, no. 3, pp. 2–7 (in Russian).
7. **Bartlett J.** The art of metaprogramming, Part 1: Introduction to metaprogramming. October 20, 2005. Website of IBM Corporation, available at: <http://www.ibm.com/developerworks/library/l-metaprogl/>.
8. **Sumarokov S. V.** Metod resheniya mnogoekstremal'noi sete-voi zadachi (The method of solving the multi-extremal network problem), *Ekonomika i Matematicheskie Metody*, 1976, vol. 12, no. 5, pp. 1016–1018 (in Russian).
9. **Stennikov V. A., Barakhtenko E. A., Sokolov D. V.** Metodicheskiy podkhod na osnove kontseptsii Model-Driven Engineering i ontologii k razrabotke programmogo obespecheniya dlya proektirovaniya teplosnabzhayushchikh system (A methodological approach on the basis of Model-Driven Engineering and ontologies for the development of a software for heat supply system design), *Informacionnye Tekhnologii*, 2015, vol. 21, no. 3, pp. 201–209 (in Russian).
10. **Gosling J., Joy B., Steele G., Bracha G., Buckley A.** *Java Language Specification, Java SE 8 Edition*. Boston, Addison-Wesley Professional, 2014, 792 p.
11. **Kay M.** *XSLT 2.0 and XPath 2.0 Programmer's Reference. 4th Edition*. New York, Wrox, 2008, 1376 p.
12. **Forman I., Forman N.** *Java Reflection in Action*. Greenwich: Manning Publications Co., 2005, 273 p.
13. **Gamma E., Helm R., Johnson R., Vlissides J.** *Design Patterns: Elements of Reusable Object-Oriented Software*. Boston, Addison-Wesley, 1994, 395 p.

**В. А. Галатенко**, д-р физ.-мат. наук, ст. науч. сотр., зав. сектором, e-mail: galat@niisi.msk.ru, Федеральное государственное учреждение "Федеральный научный центр Научно-исследовательский институт системных исследований Российской академии наук", Москва

## Кибербезопасность в здравоохранении

*Применение информационных технологий в здравоохранении способно улучшить качество лечения, облегчить труд медицинского персонала, сократить издержки. В то же время следует учитывать возникающие при этом вопросы кибербезопасности, которые носят как технический, так и поведенческий характер. В статье сделана попытка систематизировать эти вопросы и предложить возможные пути их решения.*

**Ключевые слова:** кибербезопасность, приватность, здравоохранение, электронная медицинская карта, поведенческая информационная безопасность, кибербезопасность управления медицинским оборудованием

### Введение

Здравоохранение относится к числу сфер применения информационных технологий с повышенными требованиями к кибербезопасности. Эти требования существенно различаются для следующих предметных областей:

- медицинское делопроизводство (концентрируется вокруг электронных медицинских карт);
- коммуникации между врачом и пациентом (в настоящее время осуществляются при поддержке мобильных приложений);
- киберфизические системы, включающие в себя медицинское оборудование (такие, например, как настенные сети).

При рассмотрении вопросов кибербезопасности наряду со спецификой предметных областей следует учитывать особенности современных информационных технологий, а именно:

- массовое использование мобильных устройств (по этому показателю здравоохранение занимает одну из ведущих позиций);
- распространение облачных сервисов, кибербезопасность которых особенно критична с точки зрения приватности персональных данных пациентов.

Характерная особенность здравоохранения — доминирование человеческого фактора, поэтому основными угрозами кибербезопасности являются не только криминальные атаки, но и упущения и халатность медицинского персонала [1, 2].

Угрозы кибербезопасности в здравоохранении носят не только теоретический, потенциальный характер, они вполне реальны. По данным на 2014 г., в США около 40 % случаев "кражи личности" пришлось на сферу здравоохранения [2]. Было украдено более 8 млн записей.

Представленные выше соображения определили круг рассматриваемых далее проблемных вопросов. Акцент сделан на специфику информационных систем в здравоохранении. Выполнение стандартных требований кибербезопасности подразумевается. Их изложение можно найти, например, в работах [3, 4].

### 1. Кибербезопасность медицинского делопроизводства

Нарушение кибербезопасности медицинского делопроизводства (в частности — электронных медицинских карт, ЭМК) способно нанести вред здоровью пациентов, скомпрометировать их приватность и идентифицирующие атрибуты, что чревато моральными и/или материальными издержками. Особенно уязвимы ЭМК, попавшие на мобильные устройства. К сожалению, зачастую в организациях здравоохранения мобильные устройства начинают использовать в медицинском делопроизводстве до того, как оценены риски, продуманы защитные меры и выстроена архитектура безопасности. Причина — простота и привычность использования подобных устройств.

Характерный эпизод описан в статье [5]. 28 января 2014 г., во время снежной бури на юго-востоке США, сделавшей автомобильное движение невозможным, нейрохирург Зенко Хрынкив прошел шесть миль от дома до больницы и выполнил сложнейшую операцию на мозге, спасшую пациенту жизнь. Потом он пояснил, что по пути изучал результаты компьютерной томографии пациента, переданные на его телефон по текстовому запросу, так что по приходу в больницу он имел готовый план операции и приступил к делу, не теряя времени. Здесь видны как очевидные плюсы использования информационных технологий в здравоохранении, такие как высокая доступность информации, так и новые (по сравнению с бумажным медицинским делопроизводством) риски, связанные с возможностью несанкционированного доступа к этой информации. Здоровье и жизнь пациентов — высший приоритет. Медикам зачастую приходится действовать в стрессовых, непредсказуемых условиях, поэтому определение верного баланса между простотой использования и безопасностью медицинских информационных систем — сложная, ответственная задача.

## 1.1. Архитектура безопасности систем медицинского делопроизводства

Защита медицинского делопроизводства в его современной трактовке детально и конструктивно рассматривается в серии специальных публикаций Национального института стандартов и технологий (NIST) США [6–10].

Типична ситуация, когда врач-терапевт направляет пациента к специалисту в узкой области. Предполагается, что ЭМК хранятся на центральном сервере, а медицинский персонал обращается к централизованным сервисам с мобильных устройств. Тогда врач-терапевт отправляет на сервер заявку, по которой узкоспециализированный врач получает соответствующую ЭМК.

В подобной конфигурации можно выделить три составляющие:

- центральный сервер;
- сетевая инфраструктура;
- мобильные устройства.

Центральный сервер должен содержать:

- ♦ сервер системы управления мобильным устройством;
- ♦ систему обнаружения вторжений;
- ♦ шифрование памяти;
- ♦ антивирус;
- ♦ средства протоколирования и аудита мобильных устройств;
- ♦ средства управления идентификацией мобильных устройств.

Сетевая инфраструктура должна включать в себя:

- межсетевой экран;
- средства поддержки виртуальных собственных сетей;
- сервер аутентификации, авторизации и учета;
- доверенный центр управления сертификатами.

На мобильном устройстве должны присутствовать следующие защитные средства и сервисы:

- ♦ клиент системы управления мобильным устройством;
- ♦ система обнаружения вторжений;
- ♦ программный межсетевой экран;
- ♦ шифрование памяти;
- ♦ антивирус.

Конфигурация в целом должна обеспечивать:

- управление доступом;
- протоколирование и аудит событий безопасности;
- контроль целостности устройств;
- коммуникационную безопасность;
- нейтрализацию нарушений безопасности и последующее восстановление нормального функционирования.

Перечисленные защитные средства и сервисы могут реализовываться с помощью следующего программного инструментария с открытыми исходными текстами на платформе Linux:

- ♦ OpenEMR (система поддержки ЭМК с web-интерфейсом);
- ♦ ModSecurity (межсетевое экранирование web-приложений с поддержкой OpenEMR);
- ♦ iptables;
- ♦ Bind9;
- ♦ Fedora PKI Manager;

♦ Security Onion (сервер системы обнаружения вторжений).

Кроме того, целесообразно использование коммерческих защитных средств от компаний RSA (корпоративные системы управления кибербезопасностью) и Cisco (виртуальные собственные сети, точки доступа wi-fi).

## 1.2. Привнесение собственных устройств в корпоративные сети

Согласно приведенным в работе [5] данным, "осмысленное использование" информационных технологий в здравоохранении улучшило результаты лечения пациентов. На практике "осмысленное использование" включает в себя как перенос домой части полученной в результате деятельности на рабочем месте информации, которая доставляется на производственных мобильных устройствах, так и привнесение собственных устройств в корпоративные сети. Последний феномен получил в англоязычной литературе название "*bring your own device*" (BYOD). Примечательно, что в этой области здравоохранения входит в тройку лидеров: в 2013 г. в США более 88 % медицинских работников использовали личные смартфоны для производственных целей (на первом месте здесь — образование, на втором — технологическая сфера).

В результате привнесения собственных устройств в корпоративные сети размывается граница между производственным и личным использованием информационных технологий. Более половины организаций в США признают, что они не в состоянии контролировать соблюдение работниками требований действующего законодательства и принятой политики безопасности применительно к защите данных [5].

Привнесение собственных устройств в корпоративные сети делает миссию обеспечения кибербезопасности в здравоохранении невыполнимой. Попадая на личные смартфоны, данные ограниченного доступа оказываются вне контролируемого периметра безопасности. На собственных мобильных устройствах смешивается производственное и личное, доверенное и вредоносное.

Когда на персональном мобильном устройстве смешивается производственное и личное, это создает почву для конфликта интересов. Организация вправе считать программы и данные своей собственностью, но у владельца устройства, куда эти программы и данные попали, может быть другое мнение. Получается, что при увольнении сотрудника необходимо санитировать его личный смартфон.

Особую опасность представляет вредоносное программное обеспечение (ПО). Согласно данным, приведенным в работе [11], за полтора года с июля 2012 г. по январь 2014 г. число известных образцов вредоносного ПО на платформе Android выросло более чем на порядок: с 45 до 650 тыс. Это весьма актуально для России, уверенно лидирующей по количеству вредоносного ПО для мобильных устройств (см., например, [12]).

Чтобы как-то поправить ситуацию, организация здравоохранения может выбрать одну из следующих стратегий [13]:

- запретить привнесение собственных устройств в корпоративные сети;

- ограничить множество ресурсов, доступных с собственных мобильных устройств;
- реализовать удаленную зачистку устройств, содержащих информацию ограниченного доступа;
- реализовать специальную песочницу или иной защищенный контейнер для производственных программ и данных;
- контролировать целостность программной конфигурации устройств, чтобы можно было считать их доверенными.

Наиболее многообещающим направлением представляется применение технологии виртуализации.

Может быть реализовано сочетание концепции тонкого клиента и облачной модели вычислений с внутренним для организации облаком, когда персональные мобильные устройства используются только для доступа, а программы и данные не покидают пределов корпоративной информационной системы.

Такие технические решения, как AirWatch от компании VMWare [13], позволяют централизованно управлять мобильными устройствами (*Mobile Device Management*, MDM), мобильными приложениями (*Mobile Application Management*, MAM) и мобильным информационным наполнением (*Mobile Content Management*, MCM). Тем самым у организации появляется возможность распространить корпоративную политику безопасности на персональные мобильные устройства. Имеется также возможность при необходимости удалить с персонального устройства все корпоративные данные.

Компания VMWare предлагает целостное решение Mobile Workspace с возможностью установить гипервизор поверх первоначальной операционной системы (ОС) мобильного устройства, а затем поместить все корпоративное в виртуальную машину над этим гипервизором. Для владельца устройства это, вероятно, самое простое и удобное решение, но компрометация мобильной ОС оказывается угрозой и для корпоративного рабочего пространства.

Отметим, однако, что только техническими мерами проблемы, которые порождает BYOD, не решить. Необходимо учитывать человеческий фактор, сформировать и поддерживать поведенческую информационную безопасность.

### 1.3. Поведенческая информационная безопасность

В рамках поведенческой информационной безопасности изучают поведение людей, действия которых влияют на конфиденциальность, целостность и доступность информации и информационных систем в целом. В поведенческом контексте термин "информационная безопасность", на взгляд автора, предпочтительнее "кибербезопасности".

Поведение людей, использующих личные смартфоны в производственных целях, само по себе не является плохим или нетипичным: такова психологическая и технологическая реальность. Вопрос в том, что многие из них игнорируют даже простейшие меры безопасности. Например, в здравоохранении США 41 % опрошенных не защитили устройства паролем, а 53 % сознательно подключаются к небезопасным или неизвестным сетям.

Для медицинского персонала характерно не просто равнодушное, но откровенно враждебное отношение к информационной безопасности: "Боюсь, в один ужасный день ваши требования безопасности убиты кого-нибудь из моих пациентов!".

В здравоохранении все шире внедряют и используют информационные технологии, ЭМК. Однако в области информационной безопасности наблюдается отставание даже от розничной торговли [14]. И причины тому, в первую очередь, поведенческие.

Если поведенческие нарушения информационной безопасности, как правило, можно разделить на случайные или намеренные, безобидные или вредоносные, то действия работников здравоохранения в большинстве своем попадают в категорию "намеренные безобидные". Медицинский персонал упрощает себе жизнь, создавая при этом уязвимости, делающие возможными вредоносные действия.

Подобное поведение можно попытаться исправить информированием и обучением (разумеется, умышленные вредоносные действия необходимо выявлять и пресекать, а виновных — наказывать). Однако главное здесь — не усложнять и без того непростую жизнь медиков, не подталкивать их к поиску путей обхода неудобных в использовании средств безопасности.

В качестве примера рассмотрим известные механизмы аутентификации. Сильные пароли трудно запоминать и долго вводить. Средства распознавания отпечатков пальцев, лиц и т. п. могут быть неприменимы, поскольку на руках у врачей и медсестер могут быть перчатки, а на лицах — медицинские маски.

По сравнению с другими пользователями информационных систем в поведении медицинских работников должно быть больше свободы. Следует предусмотреть механизмы экстренного расширения прав доступа из категории "в случае опасности разбить стекло". Дополнительные права следует балансировать усиленным протоколированием и автоматическим аудитом, реализуя принцип подотчетности.

Еще одна показательная ситуация [15]. В одной из больниц США внедрили систему протоколирования применяемых лекарств на основе штрихкодов. Каждому пациенту выдавали браслет со штрихкодом, штрихкод имелся и на лекарствах. По протоколу медсестра перед применением медикамента была обязана просканировать штрихкоды на браслете пациента и на лекарстве. Однако иногда со сканированием возникали трудности, поэтому медсестры заранее запасались копиями штрихкодов, и в трудных случаях, чтобы не терять драгоценное время, сканировали копии, а не оригиналы, что, естественно, являлось нарушением протокола.

Медицинский персонал в больницах находится в постоянном движении. По причине этого абстрактно правильные технические решения оказываются неудачными. Например, механизмы единого входа в сеть, казалось бы, должны облегчать жизнь врачам и медсестрам, привыкшим к традиционной многократной аутентификации при заходе на разные серверы. Тем не менее единый вход, открывая сразу множество ресурсов для доступа, требует выхода или блокировки экрана, если медработник отходит от компьютера. Получается "единый вход с многочисленными, частыми выходами", на которые тратится

время. По-особому должны обслуживаться системы на колесах: медсестра вроде бы рядом, но пока каталка движется по коридору, экран компьютерного устройства с информацией о пациенте могут увидеть посторонние. Чтобы не блокировать экран, медсестры набрасывают на него свитер или что-то еще из одежды. То есть медицинская специфика проявляется везде и во всем, порой неожиданным образом. Фактическое поведение персонала отличается от предполагаемого или предписываемого, что необходимо учитывать при разработке и реализации средств безопасности. Предварительные консультации разработчиков с медицинским персоналом не способны исправить ситуацию, поскольку первоначально врачи и медсестры могут говорить одно, а потом вести себя по-другому. Единственный выход — итерационный характер проектирования, разработки и реализации с пилотными внедрениями.

Еще один вывод состоит в неизбежности отступлений от политики безопасности и необходимости управлять небезопасностью. Для этого защита должна быть эшелонированной, с применением средств активного аудита.

Актуальными представляются кросс-культурные исследования в области поведенческой информационной безопасности.

## 2. Кибербезопасность мобильных медицинских приложений

По данным на 2014 г., в США количество мобильных медицинских приложений (mHealth) превысило 100 тыс. [5]. Ежегодное число их скачиваний измеряется миллионами. Число владельцев смартфонов, установивших и использующих на своих устройствах мобильные медицинские приложения, в 2015 г. оценивается в 500 млн. Предполагается, что в 2017 г. оно увеличится более чем в 3 раза.

Мобильные медицинские приложения способны кардинально улучшить обслуживание хронических больных, изменить привычные способы общения врача и пациента. Появилась возможность расширить спектр собираемых данных и увеличить длительность процесса сбора. Число параметров жизнедеятельности, накапливаемых каждый месяц, оценивается в сотни миллионов.

При желании врач может получить полное представление об образе жизни пациента. В работе [16] приводится классификация самых популярных свободных mHealth-приложений Google Play. На верхнем уровне их можно разбить на две группы: для пациентов и для медицинских работников.

Приложения для пациентов подразделяются на пять категорий:

- управление образом жизни (подсчет калорий, отслеживание гастрономических пристрастий, периодов сна и бодрствования, физической активности и т. д.);
- мониторинг физиологических параметров (сердечный ритм, кровяное давление, содержание сахара в крови и т. п.);
- управление лечением и болезнями (управление рецептами, идентификация лекарств, покупки медицинских препаратов, поиск по симптомам, управление хроническими болезнями и т. д.);

- медицинские коммуникации (взаимодействие с медицинскими учреждениями, врачами и медсестрами);
- управление ЭМК (хранение и/или синхронизация версий с учреждениями здравоохранения).

Для медицинских работников выделены три категории:

- получение справочной информации (медицинские словари, справочники лекарств, анатомические справочники и т. п.);
- учебные приложения для студентов-медиков (помощь в изучении медицинских теорий);
- профессиональные коммуникации (взаимодействие в экстренных случаях и обычные профессиональные контакты).

Увеличив объем собираемых персональных данных, мобильные медицинские приложения могут представлять угрозу кибербезопасности, передавая данные ограниченного доступа в открытом виде или даже направляя их на рекламные сайты [16].

Если на мобильном устройстве выполняется несколько приложений, то приватные данные от mHealth могут быть доступны другим приложениям. Еще одна угроза — попадание данных ограниченного доступа в системные журналы при протоколировании. Приведем несколько примеров.

Пациентское приложение Doctor Online из категории "медицинские коммуникации" передает по сети в открытом виде имя пользователя, адрес электронной почты и пароль.

Приложение CVS/pharmacy помещает в регистрационные журналы персональную медицинскую информацию, вводимую пользователем в дополнение к рецептам, а также атрибуты входа в систему. У злоумышленника появляется возможность "кражи медицинской личности", воспользовавшись которой, он способен не только восстановить историю болезни пациента, но и покупать лекарства от его имени.

Мобильное приложение Sleep Talk Recorder записывает звуки, издаваемые человеком во сне, и сохраняет их на SD-карте памяти, доступной другим приложениям. Ничто не мешает пересылать соответствующие .wav-файлы на внешние серверы.

В исследовании [17] утверждается, что более 95 % mHealth-приложений на платформах iOS и Android имеют те или иные трудности с информационной безопасностью, причем у 11 % приложений эти трудности можно назвать серьезными.

Еще одна характеристика — критичность информации, доступной приложению, размер возможного ущерба от нарушений безопасности. Почти 20 % приложений оперируют информацией, представляющей высокую ценность для сторонних злоумышленников. Пользователю (пациенту) будет нанесен серьезный ущерб из-за нарушения конфиденциальности персональных данных для более чем 20 % mHealth-приложений (приложению доступна нестандартная информация, такая, например, как сведения о душевных или генетических заболеваниях). Нарушение целостности информации чувствительно для пользователей более чем 30 % mHealth-приложений, утеря данных — более чем для 10 %. Иными словами, слабости в защите сочетаются с высоким размером потенциального ущерба.

В статье [18] предложена архитектура безопасности для мобильных приложений, используемых

медиками в чрезвычайных ситуациях. Врачи должны передать общие сведения об обстановке, а также результаты первичного осмотра пострадавших перед их отправкой в больницы, чтобы больницы могли лучше подготовиться к их приему.

Идентификация/аутентификация субъектов в устройствах, которые используют медики, как правило, сводится к PIN-кодам. На ввод длинных входных имен и сильных паролей у них нет времени. Файлы на мобильном устройстве шифруются эффективными симметричными методами (предпочтителен AES). Ключи симметричного шифрования (короткие) защищаются асимметричными методами. Это позволяет сбалансировать эффективность и защищенность. Коммуникации шифруются на уровне SSL/TLS. Задействуется механизм удаленной зачистки данных. Успешно переданные с мобильного устройства файлы зачищаются немедленно. В случае вопросов коммуникационного характера файлы удаляются через некоторое (конфигурируемое) время, например через 24 часа. Тем самым обеспечивается подстраховка на случай потери или кражи мобильного устройства.

В работе [19] исследован важный вопрос тестирования безопасности мобильных медицинских приложений, предложена методика тестирования, ориентированная в первую очередь на приложения для пациентов, страдающих гипертонией и/или диабетом. Применение методики для тестирования мобильных медицинских приложений на платформе Android дало следующие неутешительные результаты:

- шифрование практически не используется;
- проверка корректности вводимых пользователем данных носит фрагментарный характер, по ошибке могут быть введены летальные или бессмысленные данные;
- афишируются имена приложений, что нарушает приватность их использования в случае пассивного прослушивания сетевых коммуникаций;
- политика приватности отсутствует или неадекватна, потенциальные пользователи лишены необходимой информации.

Проблемные вопросы с кибербезопасностью можно назвать предсказуемыми, поскольку рынок мобильных медицинских приложений быстро растет, на нем появляются новые игроки, не имеющие опыта в данной области. По данным на первый квартал 2014 г., около 35 % приложений опубликовано людьми или организациями, разработавшими свое первое mHealth-приложение в течение последних 15 месяцев [19].

### **3. Кибербезопасность управления медицинским оборудованием**

Современное компьютеризованное медицинское оборудование, обладающее сетевым доступом, помогает медикам и пациентам, но создает большие трудности с решением вопросов кибербезопасности. У подобного оборудования длительный срок службы (10 и более лет), в течение которого не только выявляются уязвимости в ПО, но базовое ПО остается без поддержки. Модернизация ОС (таких как неподдерживаемые разновидности MS Windows) при этом может быть сопряжена со сложностями. В результате медицинское оборудование, критичное с точки зре-

ния безопасности, оказывается беззащитным против вредоносного ПО и других современных угроз.

Отметим, что с точки зрения кибербезопасности медицинские устройства могут оказаться слабым звеном компьютерной сети организации здравоохранения и, будучи зараженными вредоносным ПО, позволить злоумышленникам скомпрометировать всю сеть.

Для программного обеспечения медицинского оборудования идеи контролируемого выполнения [20] представляются особенно актуальными. Простота архитектуры, самоконтроль, самолечение — это свойства, которые медицинское ПО должно иметь.

Далее будут рассмотрены вопросы кибербезопасности, отражающие специфику медицинского оборудования, которое включает в себя аппаратно-программное обеспечение и средства сетевого взаимодействия (беспроводного или через сеть Интернет). Подобное медицинское оборудование обладает всеми свойствами информационных систем и разделяет с ними киберугрозы: несанкционированный доступ, несанкционированное изменение настроек, утечки данных, вредоносное ПО и т. п. [21].

Особенно критичны с точки зрения безопасности имплантируемые устройства. Во всем мире все больше больных полагаются на них, поэтому нарушение их нормального функционирования представляет угрозу здоровью и жизни пациентов.

Если у медицинского устройства нарушается нормальное функционирование, его отзывают. Сбои аппаратно-программного обеспечения составляют примерно 20...25 % среди причин отзыва. В основном (90 %) виновником сбоев оказывается ПО. Лишь в 3 % случаев его коррекцию можно провести в оперативном режиме. Массовая модернизация занимает несколько месяцев, т. е. окно опасности открыто слишком долго.

Доказано, что внешние атаки на медицинские устройства принципиально возможны. Например, вмешаться в работу инсулиновой помпы, имеющей беспроводной интерфейс, можно с расстояния до 50 метров.

Характерным примером современных медицинских устройств, существующей вокруг них инфраструктуры и потенциальных угроз безопасности может служить кардиостимулятор. Дома у пациента (обычно в спальне) располагается устройство, отслеживающее через беспроводной интерфейс работу кардиостимулятора и передающее полученные данные через сеть Интернет в клинику. Там эти данные анализируют и при необходимости подстраивают кардиостимулятор (опять через беспроводной интерфейс) с помощью программирующего устройства. В настоящее время программирующее устройство управляется вручную, так что для подстройки кардиостимулятора пациент должен прибыть в клинику, но, несомненно, в недалеком будущем и программирующее устройство будет управляться удаленно, через сеть.

В подобной конфигурации под угрозой оказываются все три основных аспекта кибербезопасности: конфиденциальность, целостность и доступность. В частности, конфиденциальность и целостность могут быть нарушены в любой точке маршрута передачи

и анализа данных, начиная от беспроводного канала связи между кардиостимулятором и устройством мониторинга и заканчивая сервером в клинике. Нарушение доступности (нормального функционирования) кардиостимулятора особенно опасно. Оно может не только стать следствием дефектов в самом имплантируемом устройстве, но и быть вызвано нештатной работой программирующего устройства при подстройке.

Для пациентов традиционно важно соблюдение приватности. В статье [22] сформулированы следующие цели для этого аспекта информационной безопасности применительно к персональным медицинским устройствам:

- факт использования персонального медицинского устройства считается приватной информацией;
- если факт использования скрыть невозможно, тип устройства должен быть приватным;
- если тип устройства скрыть невозможно, приватным должен быть уникальный идентификатор;
- приватными должны быть результаты измерений и регистрационная информация, ассоциированная с устройством;
- по используемому устройству не должно быть возможности установить личность пациента;
- не должно быть возможности с помощью персонального медицинского устройства отследить положение и перемещения пациента.

Несмотря на ограниченность ресурсов имплантируемых медицинских устройств, в их программном обеспечении могут выявляться ошибки и уязвимости. С самого начала процесса проектирования устройств следует предусмотреть возможность необременительной для пациентов, оперативной установки программных коррекций. Это сложная задача, но решать ее необходимо.

Важной современной категорией медицинского оборудования являются нательные сенсорные сети, которые отличаются от беспроводных сенсорных сетей общего характера еще большей ограниченностью аппаратных ресурсов и энергопотребления. Например, нагревающиеся батарейки не должны оказывать негативного воздействия на ткани пациентов, смена батареек может быть неприятной процедурой и т. п. В то же время беспроводные коммуникации требуют защиты. В статье [23] предложены криптографические механизмы с открытыми ключами, основанные на использовании эллиптических кривых. Они позволяют реализовать такие необходимые для медицинского оборудования сервисы безопасности, как конфиденциальность, аутентификация (в том числе аутентификация источника данных), управление доступом, невозможность отказаться от совершенных действий и контроль целостности. Утверждается, что при этом требования реального времени и ограничения по ресурсам оказываются выполненными. Многообещающим выглядит также сочетание симметричного шифрования и контроля целостности со сжатием данных [24].

Еще одно возможное решение — использование внешнего устройства, принимающего на себя максимально возможную часть нагрузки и тем самым разгружающее имплантируемые и переносимые пациентом медицинские устройства, а также продлевающее срок службы их батареек.

Слабым звеном в архитектуре нательных сетей является концентратор, собирающий, обрабатывающий и передающий данные дальше, а также отображающий их для пациента. В роли концентратора может выступать персональный смартфон с недоверенным медицинским приложением. Ввиду сложности программного обеспечения концентратора он потенциально уязвим для таких атак, как внедрение вредоносного ПО. Может быть скомпрометирована ОС концентратора, под управлением которой выполняются все приложения. Для решения подобных задач рекомендуется создавать защищенные области для выполнения критичных по безопасности приложений (основываясь, например, на технологии виртуализации или на аппаратной поддержке).

Вопросы обеспечения кибербезопасности актуальны и для медицинского оборудования в больницах. Пример — прикроватные мониторы, передающие данные на центральную станцию (пост, ординаторскую) в открытом виде. Более того, в ответ на широковещательный запрос прикроватные мониторы сообщают, кто они, какие у них сетевые и несетевые параметры, после чего по запросам выдают данные пациента. Имеется также возможность удаленно менять параметры монитора, как сетевые, например IP-адреса, так и медицинские, например, граничные значения физиологических параметров, при выходе за которые возбуждается сигнал тревоги.

Небезопасность медицинского оборудования усугубляется поведенческой небезопасностью медицинского персонала. В ряде случаев врачи хотят наблюдать за всем, что происходит с конкретными пациентами, в любое время дня и ночи, из любой точки мира. Интересует врачей все — от выполнения назначений медсестрами до текущих физиологических параметров, отображаемых прикроватным монитором, а удаленный доступ осуществляется с персонального мобильного устройства через недоверенные сети.

Специфика медицинских устройств такова, что их кибербезопасность должна обеспечиваться превентивными, а не реактивными мерами. Первым звеном в цепочке превентивных мер должна стать формально верифицируемая разработка аппаратного и программного обеспечения. По данным на первую половину 2010 г., причинами более четверти отзывов медицинского оборудования стали программные дефекты [24].

В то же время задача эта крайне сложна: программы взаимодействуют с низкоуровневым оборудованием, функционируют под управлением прерываний, окружение не может быть формализовано и т. п. Следовательно, формальную верификацию необходимо дополнить архитектурой в духе контролируемого выполнения.

#### **4. Кибербезопасность облачных вычислений в здравоохранении**

Использование облачных вычислений в здравоохранении является логическим дополнением сетей из персональных имплантируемых и/или носимых медицинских устройств с ограниченными ресурсами. В перспективе подобные устройства полезно представлять себе как узлы "Интернета вещей" [25],

которые в разные моменты времени могут быть частью домашней, корпоративной или публичной сети. Медицинские данные пациента передаются в облачный сервис, содержащий базу данных персональных медицинских карт. Медицинский персонал также имеет доступ к этому сервису и этой базе, так что связь между пациентом и его лечащим врачом потенциально возможна в любой момент времени, из любого места.

Специфичный для здравоохранения характер носит решение задачи обеспечения защищенного взаимодействия облачных сервисов с медицинским оборудованием. Одно из возможных решений — разделять некоторую базовую информацию при установке персональных медицинских устройств, например базовые значения медицинских параметров пациента, а в дальнейшем передавать с устройства в облако только разность текущих и базовых значений, как предложено в работе [25]. Впрочем, перехват последовательности разностей все равно ставит под угрозу приватность пациентов.

Облачная архитектура в здравоохранении может носить пациенто-центрический или организационно-центрический характер. В первом случае, используя облака общего пользования, пациент помещает туда свою медицинскую информацию, чтобы специалисты, услугами которых он пользуется, могли с ней ознакомиться. Во втором случае облачные конфигурации используют медицинские учреждения как средство разделения информации.

В любом случае у медицинской информации должен быть менеджер, управляющий доступом к ней (в пациенто-центрической модели это пациент). Облачная среда должна поддерживать доступность, целостность и конфиденциальность данных, аутентификацию (в том числе аутентификацию источника данных); отсутствие возможности отказаться от совершенных действий; протоколирование и резервное копирование. Основой облачной безопасности служат криптографические механизмы и протоколы [26].

Безопасность медицинской информации должна обеспечиваться как при передаче, так и при хранении. Для защиты передаваемых данных имеются стандартные решения, основанные на протоколах SSL/TLS и поддерживаемые как медицинскими устройствами, так и облачными конфигурациями. С защитой данных при хранении сложилась ситуация более сложная. Хранимые данные должны быть зашифрованы. В настоящее время нет общепринятых стандартов криптографической защиты данных в системах управления базами данных (СУБД). Обычно поставщики облачных сервисов предполагают, что данные в облако поступают уже зашифрованными, т. е. оставляют криптографию пользователям. Такой подход усложняет разделение данных между медицинскими работниками и доступ к ним в экстренных ситуациях. В работе [27] предложено разделять медицинские данные на общедоступные и персонифицированные, хранить их и управлять доступом к ним отдельно. Например, персонифицированные данные могут быть доступны только медицинским работникам и храниться в частном облаке медицинской организации.

Можно констатировать, что проблема кибербезопасности облачных вычислений в здравоохранении пока не получила целостного решения.

## 5. Кибербезопасность работы с большими данными в здравоохранении

Работа с большими данными в здравоохранении в перспективе должна обеспечить переход от реактивной к превентивной медицине, призвана сделать медицинскую помощь персонифицированной. Вклад в накопление больших данных вносят персональные медицинские устройства, хранение и обработка таких данных может осуществляться облачными конфигурациями.

У больших данных в здравоохранении, помимо медицинских организаций, потенциально имеется большое число других поставщиков и потребителей. К их числу относятся административные, финансовые, экологические и другие органы и организации. Это усложняет доступ к большим данным и обеспечение безопасности этого доступа.

В статье [28] сформулированы следующие необходимые условия применения подходов, характерных для работы с большими данными:

- оцифровка данных (в основном это касается унаследованных данных);
- семантическое аннотирование (данные могут поступать из разных источников, необходимо неким универсальным образом пояснить их смысл);
- разделение данных (посредством универсальных средств описания структуры и кодирования данных);
- информационная безопасность (критически важное требование в здравоохранении, включающее в себя, в частности, анонимизацию данных перед их агрегированием);
- качество данных (требование естественное, но практически нереализуемое).

Обеспечение информационной безопасности в крупномасштабной, распределенной, разнородной среде — сложная проблема. Основная идея ее решения состоит в ассоциировании с данными правил политики безопасности в целях управления доступом [29], т. е. данные должны быть самодостаточны не только синтаксически и семантически, но и в плане безопасности.

Проведение в жизнь политики безопасности и, в частности, управление доступом осуществляются криптографическими методами. Данные хранятся зашифрованными ключом владельца, а выдаются (если доступ разрешен) зашифрованными ключом получателя. С доступом могут ассоциироваться обязательные действия, такие как анонимизация данных.

## Заключение

Применение информационных технологий в здравоохранении в принципе позволяет улучшить качество оказываемых медицинских услуг, облегчает работу специалистов, снижает издержки, т. е. оно выгодно всем заинтересованным сторонам. В то же время многие вопросы, особенно связанные с обеспечением кибербезопасности, пока далеки от решения. Необходимы научные исследования, разработка технических решений и их внедрение в повседневную практику.

## Список литературы

1. **Fifth Annual Benchmark Study on Privacy & Security of Healthcare Data.** Ponemon Institute. May 2015. 42 p.
2. **Fifth Annual Study on Medical Identity Theft.** Ponemon Institute. February 2015. 38 p.
3. **Галатенко В. А.** Основы информационной безопасности / Под ред. В. Б. Бетелина. М.: ИНТУИТ.РУ, 2003. 280 с.
4. **Галатенко В. А.** Стандарты информационной безопасности / Под ред. В. Б. Бетелина. М.: ИНТУИТ.РУ, 2004. 328 с.
5. **Burns A. J., Johnson M. E.** Securing Health Information // IT Pro. 2015. January/February. P. 23–29.
6. **Practice Guide.** Health IT. Securing Electronic Health Records on Mobile Devices. Executive Summary. U. S. Department of Commerce, National Institute of Standards and Technology, National Cybersecurity Center of Excellence, NIST Special Publication 1800-1a. NIST. July 2015. 3 p.
7. **O'Brien G., Lesser N., Pleasant B.** et al. Health IT Sector. Securing Electronic Health Records on Mobile Devices. Approach, Architecture, and Security Characteristics. For CIOs, CISOs, and Security Managers / Ed. L. Kauffman. U. S. Department of Commerce, National Institute of Standards and Technology, National Cybersecurity Center of Excellence, NIST Special Publication 1800-1b, Draft. NIST. July 2015. 31 p.
8. **O'Brien G., Lesser N., Pleasant B.** et al. Health IT Sector. Securing Electronic Health Records on Mobile Devices. How-To Guides. For Security Engineers / Ed. L. Kauffman. U. S. Department of Commerce, National Institute of Standards and Technology, National Cybersecurity Center of Excellence, NIST Special Publication 1800-1c, Draft. NIST. July 2015. 91 p.
9. **O'Brien G., Lesser N., Pleasant B.** et al. Health IT Sector. Securing Electronic Health Records on Mobile Devices. Standards and Controls Mapping / Ed. L. Kauffman. U. S. Department of Commerce, National Institute of Standards and Technology, National Cybersecurity Center of Excellence, NIST Special Publication 1800-1d, Draft. NIST. July 2015. 22 p.
10. **O'Brien G., Lesser N., Pleasant B.** et al. Health IT Sector. Securing Electronic Health Records on Mobile Devices. Risk Assessment and Outcomes / Ed. L. Kauffman. U. S. Department of Commerce, National Institute of Standards and Technology, National Cybersecurity Center of Excellence, NIST Special Publication 1800-1e, Draft. NIST. July 2015. 82 p.
11. **Chang J. M., Ho P.-C., Chang T.-C.** Securing BYOD // IT Pro. 2014. September/October. P. 9–11.
12. **Svajcer V.** Sophos Mobile Security Threat Report. Launched at Mobile World Congress. 2014. 10 p.
13. **Souppaya M., Scarfone K.** Guidelines for Managing the Security of Mobile Devices in the Enterprise. U. S. Department of Commerce, National Institute of Standards and Technology, Special Publication 800-124. Revision 1. NIST. June 2013. 30 p.
14. **Kotz D., Fu K., Gunter C., Rubin A.** Security for Mobile and Cloud Frontiers in Healthcare // Communications of the ACM. 2015. Vol. 58, N 8. P. 21–23.
15. **Heckle R. R.** Security Dilemma: Healthcare Clinicians at Work // IEEE Security & Privacy. 2011. November/December. P. 14–19.
16. **He D., Naveed M., Gunter C. A., Nahrstedt K.** Security Concerns in Android mHealth Apps // Proceedings of the AMIA Symposium. Washington DC. November 2014. P. 645–654.
17. **Dehling T., Gao F., Schneider S., Sunyaev A.** Exploring the Far Side of Mobile Health: Information Security and Privacy of Mobile Health Apps on iOS and Android // JMIR mHealth uHealth. 2015. Vol. 3, Issue. 1, e8.
18. **Murad A., Schooley B., Abed Y.** Demonstration Paper: A Secure mHealth Application for EMS: Design and Implementation // Proceedings of the Wireless Health'13. Baltimore, MD, USA. ACM. November 1–3, 2013. P. 15–16.
19. **Knorr K., Aspinall D.** Security Testing for Android mHealth Apps // Proceedings of the 2015 IEEE Eighth International Conference on Software Testing, Verification and Validation Workshops (ICSTW), 6th International Workshop on Security Testing (SECTEST 2015). 2015. P. 1–8.
20. **Бетелин В. Б., Галатенко В. А., Костюхин К. А.** Основные понятия контролируемого выполнения сложных систем // Информационные технологии. 2013. № 3. Приложение. С. 1–32.
21. **Sametinger J., Rozenblit J., Lysecky R., Ott P.** Security Challenges for Medical Devices // Communications of the ACM. 2015. Vol. 58, N 4. P. 74–82.
22. **Rushanan M., Rubin A. D., Kune D. F., Swanson C. M.** SoK: Security and Privacy in Implantable Medical Devices and Body Area Networks // Proceedings of the 2014 IEEE Symposium on Security and Privacy. 2014. P. 524–539.
23. **Le X. H., Sankar R., Khalid M., Lee S.** Public Key Cryptography — based Security Scheme for Wireless Sensor Networks in Healthcare // Proceedings of the ICUIMC'10. January 14–15, 2010. Suwon, Korea. ACM. URL: [http://uclab.khu.ac.kr/resources/publication/C\\_203.pdf](http://uclab.khu.ac.kr/resources/publication/C_203.pdf)
24. **Zhang M., Raghunathan A., Jha N. K.** Trustworthiness of Medical Devices and Body Area Networks // Proceedings of the IEEE. 2014. Vol. 102, N 8. P. 1174–1188.
25. **Mohan A.** Cyber Security for Personal Medical Devices Internet of Things // Proceedings of the 2014 IEEE International Conference on Distributed Computing in Sensor Systems. 2014. P. 372–374.
26. **Zhang R., Liu L.** Security Models and Requirements for Healthcare Application Clouds // Proceedings of the 2010 IEEE 3rd International Conference on Cloud Computing. 2010. P. 268–275.
27. **Lee Y. S., Bruce N., Non T.** et al. Hybrid Cloud Service based Healthcare Solutions // Proceedings of the 2015 29th International Conference on Advanced Information Networking and Applications Workshops, IEEE. 2015. P. 25–30.
28. **Zillner S., Oberkamp H., Bretschneider C.** et al. Towards a Technology Roadmap for Big Data Applications in the Healthcare Domain // Proceedings of the IEEE IRI 2014. August 13–15, 2014. San Francisco, California, USA. 2014. P. 291–296.
29. **Li S., Zhang T., Gao J., Park Y.** A Sticky Policy Framework for Big Data Security // Proceedings of the 2015 IEEE First International Conference on Big Data Computing Service and Applications. 2015. P. 130–137.

# Cybersecurity in Healthcare

**V. A. Galatenko** galat@niisi.ras.ru, Federal State Institution "Scientific Research Institute of System Analysis of the Russian Academy of Science", Moscow, 117218, Russian Federation

*Corresponding author:*

**Galatenko Vladimir A.**, Head of Sector, Federal State Institution "Scientific Research Institute of System Analysis of the Russian Academy of Science", Moscow, 117218, Russian Federation  
e-mail: galat@niisi.ras.ru

*Received on December 01, 2015*

*Accepted on December 11, 2015*

*The widespread use of information technologies in healthcare may improve the quality of care, facilitate the work of medical staff, reduce costs. At the same time it is necessary to pay attention to the problems of cybersecurity which can be of either technical or behavioral character.*

*Security problems may affect confidentiality and integrity of data, availability of medical services. The issue of privacy is of special importance in the area of healthcare. Standard cybersecurity means may be inapplicable in*

healthcare due to specificities of this area and stress working conditions of medical personnel. Further complexities are caused by use of legacy medical equipment which may be vulnerable to potential security attacks. This paper is an attempt to systematize the problems of cybersecurity in healthcare and propose possible solutions.

The issues discussed are cybersecurity of Electronic Health Records (EHR), the problem of BYOD (bring your own device) in healthcare, mobile healthcare applications (mHealth) cybersecurity, medical devices cybersecurity, cloud and big data cybersecurity in healthcare.

**Keywords:** cybersecurity, privacy, healthcare, Electronic Health Record, behavioral information security, medical devices cybersecurity

For citation:

**Galatenko V. A.** Cybersecurity in Healthcare, *Programmnaya Ingeneria*, 2016, vol. 7, no. 3, pp. 117–125.

DOI: 10.17587/prin.7.117-125

## References

1. **Fifth Annual Benchmark Study on Privacy & Security of Healthcare Data.** Presented by Ponemon Institute, May 2015, 42 pp.
2. **Fifth Annual Study on Medical Identity Theft.** Ponemon Institute, February 2015, 38 pp.
3. **Galatenko V. A., Osnovy informacionnoj bezopasnosti (Information Security Basics)** / Ed. V. B. Betelin. Moscow, INTUIT.RU, 2003, 280 p. (in Russian).
4. **Galatenko V. A. Standartny informacionnoj bezopasnosti (Information Security Standards)** / Ed. V. B. Betelin. Moscow, INTUIT.RU, 2004, 328 p. (in Russian).
5. **Burns A. J., Johnson M. E.** Securing Health Information. *IT Pro*, 2015, January/February, pp. 23–29.
6. **Practice Guide. Health IT. Securing Electronic Health Records on Mobile Devices. Executive Summary.** U. S. Department of Commerce, National Institute of Standards and Technology, National Cybersecurity Center of Excellence, NIST Special Publication 1800-1a. NIST. July 2015, 3 p.
7. **O'Brien G., Lesser N., Pleasant B., Wang S., Zheng K., Bowers C., Kamke K.** *Health IT Sector. Securing Electronic Health Records on Mobile Devices. Approach, Architecture, and Security Characteristics.* For CIOs, CISOs, and Security Managers / Ed. L. Kauffman. U. S. Department of Commerce, National Institute of Standards and Technology, National Cybersecurity Center of Excellence, NIST Special Publication 1800-1b, Draft. NIST. July 2015, 31 p.
8. **O'Brien G., Lesser N., Pleasant B., Wang S., Zheng K., Bowers C., Kamke K.** *Health IT Sector. Securing Electronic Health Records on Mobile Devices. How-To Guides. For Security Engineers* / Ed. L. Kauffman. U. S. Department of Commerce, National Institute of Standards and Technology, National Cybersecurity Center of Excellence, NIST Special Publication 1800-1c, Draft, NIST, July 2015, 91 p.
9. **O'Brien G., Lesser N., Pleasant B., Wang S., Zheng K., Bowers C., Kamke K.** *Health IT Sector. Securing Electronic Health Records on Mobile Devices. Standards and Controls Mapping* / Ed. L. Kauffman. U. S. Department of Commerce, National Institute of Standards and Technology, National Cybersecurity Center of Excellence, NIST Special Publication 1800-1d, Draft, NIST, July 2015, 22 p.
10. **O'Brien G., Lesser N., Pleasant B., Wang S., Zheng K., Bowers C., Kamke K.** *Health IT Sector. Securing Electronic Health Records on Mobile Devices. Risk Assessment and Outcomes* / Ed. L. Kauffman. U. S. Department of Commerce, National Institute of Standards and Technology, National Cybersecurity Center of Excellence, NIST Special Publication 1800-1e, Draft. NIST. July 2015. 82 p.
11. **Chang J. M., Ho P.-C., Chang T.-C.** Securing BYOD. *IT Pro*, 2014, September/October, pp. 9–11.
12. **Svajcer V.** Sophos Mobile Security Threat Report. *Launched at Mobile World Congress.* 2014, 10 p.
13. **Souppaya M., Scarfone K.** *Guidelines for Managing the Security of Mobile Devices in the Enterprise.* U. S. Department of Commerce, National Institute of Standards and Technology, Special Publication 800-124. Revision 1, NIST, June 2013, 30 p.
14. **Kotz D., Fu K., Gunter C., Rubin A.** Security for Mobile and Cloud Frontiers in Healthcare, *Communications of the ACM*, 2015, vol. 58, no. 8, pp. 21–23.
15. **Heckle R. R.** Security Dilemma: Healthcare Clinicians at Work, *IEEE Security & Privacy*, 2011, November/December, pp. 14–19.
16. **He D., Naveed M., Gunter C. A., Nahrstedt K.** Security Concerns in Android mHealth Apps, *Proceedings of the AMIA Symposium*, Washington DC, November 2014, pp. 645–654.
17. **Dehling T., Gao F., Schneider S., Sunyaev A.** Exploring the Far Side of Mobile Health: Information Security and Privacy of Mobile Health Apps on iOS and Android, *JMIR mHealth*, 2015, vol. 3, issue 1, e8.
18. **Murad A., Schooley B., Abed Y.** Demonstration Paper: A Secure mHealth Application for EMS: Design and Implementation, *Proceedings of the Wireless Health'13.* November 1–3, 2013. Baltimore, MD, USA, ACM, 2013, pp. 15–16.
19. **Knorr K., Aspinall D.** Security Testing for Android mHealth Apps, *Proceedings of the 2015 IEEE Eighth International Conference on Software Testing, Verification and Validation Workshops (ICSTW), 6th International Workshop on Security Testing (SECTEST 2015)*, 2015, pp. 1–8.
20. **Betelin V. B., Galatenko V. A., Kostukhin K. A.** Osnovnye ponjatija kontroliruemogo vypolnenija slozhnyh sistem (Controlled Execution of Complex Systems: Main Terms), *Informacionnye Tehnologii*, 2013, no. 3, Supl., pp. 1–32 (in Russian).
21. **Sametinger J., Rozenblit J., Lysecky R., Ott P.** Security Challenges for Medical Devices, *Communications of the ACM*, 2015, vol. 58, no. 4, pp. 74–82.
22. **Rushanan M., Rubin A. D., Kune D. F., Swanson C. M.** SoK: Security and Privacy in Implantable Medical Devices and Body Area Networks, *Proceedings of the 2014 IEEE Symposium on Security and Privacy*, 2014, pp. 524–539.
23. **Le X. H., Sankar R., Khalid M., Lee S.** Public Key Cryptography – based Security Scheme for Wireless Sensor Networks in Healthcare, *Proceedings of the ICUIMC'10.* January 14–15, 2010. Suwon, Korea. ACM, available at: [http://uclab.khu.ac.kr/resources/publication/C\\_203.pdf](http://uclab.khu.ac.kr/resources/publication/C_203.pdf)
24. **Zhang M., Raghunathan A., Jha N. K.** Trustworthiness of Medical Devices and Body Area Networks, *Proceedings of the IEEE*, 2014, vol. 102, no. 8, pp. 1174–1188.
25. **Mohan A.** Cyber Security for Personal Medical Devices Internet of Things, *Proceedings of the 2014 IEEE International Conference on Distributed Computing in Sensor Systems*, 2014, pp. 372–374.
26. **Zhang R., Liu L.** Security Models and Requirements for Healthcare Application Clouds, *Proceedings of the 2010 IEEE 3rd International Conference on Cloud Computing*, 2010, pp. 268–275.
27. **Lee Y. S., Bruce N., Non T., Alasarela E., Lee H. J.** Hybrid Cloud Service based Healthcare Solutions, *Proceedings of the 2015 29th International Conference on Advanced Information Networking and Applications Workshops*, IEEE, 2015, pp. 25–30.
28. **Zillner S., Oberkamp H., Bretschneider C., Zaver A., Faix W., Neururer S.** Towards a Technology Roadmap for Big Data Applications in the Healthcare Domain, *Proceedings of the IEEE IRI 2014.* August 13–15, 2014. San Francisco, California, USA, 2014, pp. 291–296.
29. **Li S., Zhang T., Gao J., Park Y.** A Sticky Policy Framework for Big Data Security, *Proceedings of the 2015 IEEE First International Conference on Big Data Computing Service and Applications*, 2015, pp. 130–137.

**Н. М. Быкова**, канд. техн. наук., доц., зав. каф., e-mail: bikovanm@mail.ru,

**Д. А. Зайнагабдинов**, ст. препод., e-mail: damirmt@mail.ru,

**Т. Ш. Белялов**, аспирант, e-mail: belyalov\_timur@mail.ru,

**И. В. Мешков**, аспирант, e-mail: nerig@rambler.ru,

Иркутский государственный университет путей сообщения

## Технологические основы программно-аппаратного сопровождения автоматизированного мониторинга деформаций железнодорожных тоннелей

*Представлены разработанные авторами способы построения программно-аппаратного комплекса для автоматизированного мониторинга деформаций (АМД) железнодорожных тоннелей. Рассмотрены технологическая схема работы АМД, общее описание комплекса и основные положения, которые используются при разработке программного обеспечения. Сформированы правила анализа показателей датчиков деформаций тоннелей в процессе мониторинга и критерии назначения категорий геодинамической безопасности.*

**Ключевые слова:** программное обеспечение, деформационный мониторинг, геодинамическая безопасность, железнодорожные тоннели

### Введение

Автоматизированный мониторинг деформаций (АМД) транспортных тоннелей является частью комплексной системы мониторинга геодинамической безопасности железнодорожных тоннелей. Система АМД предназначена для контроля значений величин деформаций в обделке тоннеля, возникающих в результате трудно предсказуемых на стадии проектирования геодинамических и геодеформационных воздействий горно-геологической среды, а также для анализа изменения жесткостных свойств конструкций вследствие развития скрытых повреждений. Целью АМД является оперативное предупреждение о внезапном возникновении нештатных ситуаций или о начале развития скрытых опасных процессов в конструкциях тоннеля.

Задачами АМД являются:

- сбор, обработка и хранение данных, поставляемых инструментальными средствами системы, и результатов их первичной обработки;
- анализ и оперативная оценка текущей информации с назначением категорий опасности и формированием сообщений заинтересованным службам;
- создание и поддержка математических и численных моделей конструкций тоннеля с возможностью проведения анализа напряженно-деформированного состояния по результатам обработки измерений, регистрируемых подсистемами мониторинга;
- оценка технического состояния конструкций с назначением категорий геодинамической безопасности.

Эффективность работы АМД зависит как от инструментальной базы, так и от программных

разработок, обеспечивающих выполнение этих задач.

### Схема работы АМД

Схема, отражающая технологию автоматизированного деформационного мониторинга, приведена на рис. 1.

Схема предусматривает использование волоконно-оптических датчиков для регистрации изменения деформаций со сбором и передачей показаний через интеррогатор на сервер первичной обработки данных. Блок численных моделей работает независимо. Однако он связан с сервером первичных данных, откуда поступает информация о внешних показателях других подсистем (геодезических, геодинамических, деформационных), а также с аналитическим контуром, куда поступают результаты расчетов конструкций. В аналитическом контуре выполняются проверки критериев и назначаются категории геодинамической безопасности. В зависимости от работы аналитического контура формируются сигналы для блока сигнализации с выходом на управляющий контур. В задачи управляющего контура входит проверка работы самой системы, а также принятие решений с учетом категорий опасности по режимам пропуска транспорта и стабилизации ситуации.

Далее рассмотрены технологии, которые используют для решения задач мониторинга деформаций тоннеля в плане увязки аппаратно-технической, аналитической и программной составляющих систем.

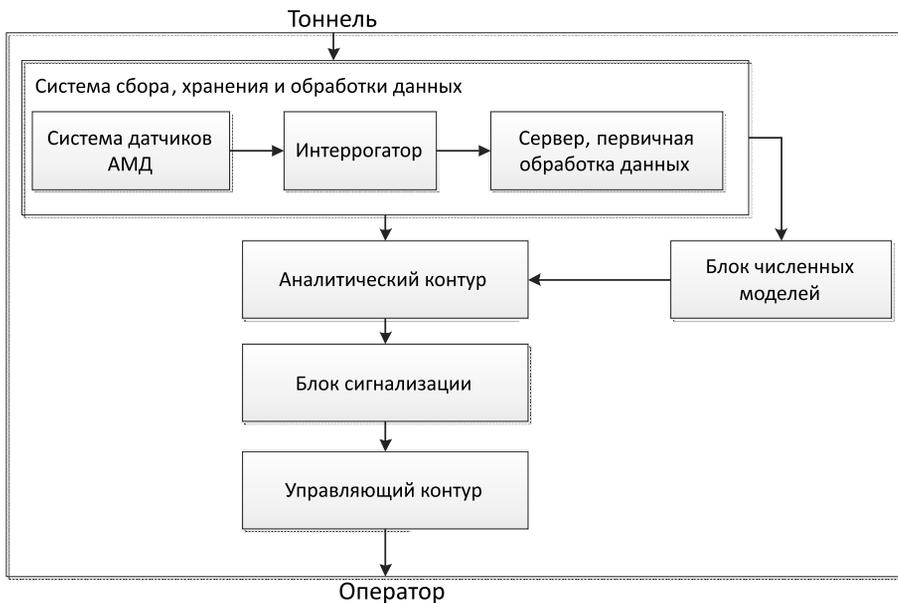


Рис. 1. Схема работы АМД

### Общее описание программно-аппаратного комплекса

Система АМД состоит из инструментальной и программной составляющих. Инструментальная часть включает системы датчиков, регистрирующих деформации, блоки сбора информации и систему передачи информации на сервер. Для электрофицированных протяженных железнодорожных тоннелей можно рекомендовать волоконно-оптические датчики деформаций, позволяющие с высокой точностью фиксировать изменение напряженно-деформированного состояния элементов тоннеля, исключая при этом влияние электромагнитных помех. Контур оптоволоконного кабеля с датчиками подключается к системе сбора данных (интеррогатор и мультимплексор), откуда информация передается на сервер.

Программная часть представляет собой комплекс программных средств, осуществляющих сбор, хранение, передачу, анализ и обработку данных, сигнализацию и визуализацию работы тоннеля, вывод оператору всей необходимой информации о работе системы и проведении плановых мероприятий.

Общая схема работы аппаратно-программного комплекса представлена на рис. 2.

Программное обеспечение комплекса представляет собой совокупность взаимодействующих друг с другом программных моделей, выполняющих разные функции и решающих разные задачи. Разделение программного обеспечения на несколько блоков позволяет располагать центры мониторинга за состоянием тоннелей удаленно от объекта исследования.

**Сбор, обработка и хранение информации.** Сбор и обработка данных с датчиков осуществляется с помощью специального программного обеспечения, входящего в комплект поставляемого приемо-

регистрирующего устройства. С его помощью исходные сигналы с датчиков конвертируются в данные, отображающие деформации, и представляются в текстовом и графическом видах. Установка этого программного обеспечения осуществляется на сервер, а передача информации к нему от системы сбора данных может выполняться с помощью оптоволоконного кабеля.

Хранение данных осуществляется на сервере в текстовом виде. Это необходимо для более удобной и простой работы. Предусматривается передача сохраненных данных по локальной сети или по сети Интернет на любой компьютер, где будет располагаться центр мониторинга за состоянием тоннеля. С помощью данного программного обеспечения выполняется также настройка

функций блока приема информации и диагностика работы датчиков.

**Анализ и оперативная оценка текущей информации АМД.** Текущая информация содержит данные о деформациях участков тоннельных конструкций, которые характеризуют их реакцию на статические и динамические внешние воздействия. Эти данные представляют собой важные сигналы о диапазонах работы тоннеля в сложных геодинамических условиях. Для использования этих сигналов при выработке управляющих решений программное обеспечение на основе принятых данных проводит анализ деформаций на разных участках обделки тоннеля, сопоставление полученных значений с заданными критериями и назначение определенной категории безопасности в соответствии с заданными правилами [1].

Предложены три категории оценки геодинамической безопасности [1]:

- **первая категория** соответствует безопасному штатному режиму эксплуатации;
- **вторая категория** предупреждает о пограничном состоянии сооружения;
- **третья категория** характеризует наступление нештатной ситуации, опасной для эксплуатации сооружения.

Ниже приведена технология назначения категорий по количественным показателям.

Принимаемые данные по деформациям тоннельных обделок можно разделить на два типа:

- **мгновенные всплески** — отражают текущую реакцию обделки тоннеля при кратковременном событии (например, сейсмические события, сдвиг породы);
- **тренды** — представляют собой длительные направленные деформации, данные о которых очищены от случайных флуктуаций, циклических и других

## Система сбора информации

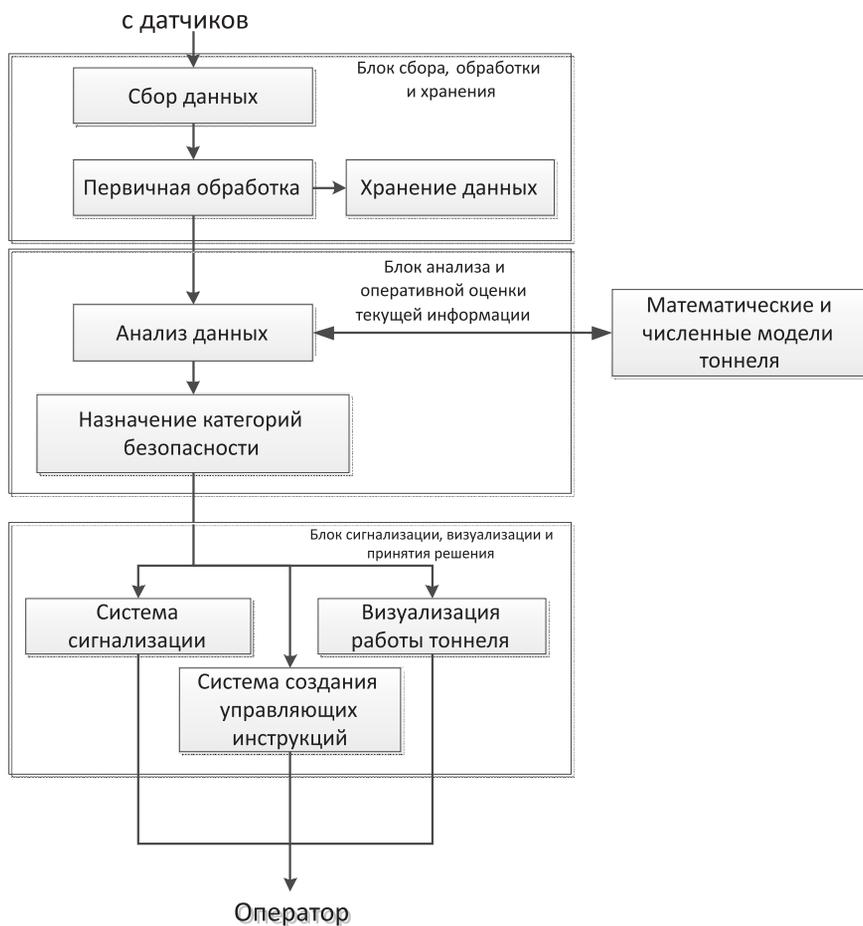


Рис. 2. Схема работы аппаратно-программного комплекса АМД

особенностей, отражающих работу обделки тоннеля в течение длительного периода времени.

Рассматривается наиболее часто принимаемая бетонная и железобетонная обделка тоннеля.

Технология анализа мгновенных скачкообразных деформаций заключается в том, что при трехкратной разнице соседних текущих показаний деформаций  $3a_{i-1} < a_i > 3a_{i+1}$  выполняется условие

$$\sigma_t = \sigma_0 \pm (a_i E_b) \leq R_i,$$

где  $a_i$  — показание относительных деформаций по текущим данным;  $E_b$  — модуль упругости бетона;  $\sigma_0$  — начальные напряжения в обделке в месте установки датчика;  $\sigma_t$  — текущее напряжение в обделке в месте установки датчика;  $R_i$  — контролируемый параметр расчетного сопротивления материала.

Начальные напряжения в обделке на период установки системы мониторинга определяют численными методами с выборочным экспериментальным контролем.

Знаки "плюс" принимают для растянутых зон конструкций, знаки "минус" — для сжатых зон.

Модуль упругости бетона определяется с учетом инструментально определенного класса бетона по прочности на сжатие.

Поскольку тоннельная обделка работает преимущественно в условиях сжатия, то в качестве критерия предельного сжатия при кратковременных динамических событиях принято расчетное сопротивление бетона осевому сжатию  $R_i = R_b$  [2]. Если обделка в каких-то зонах переходит в область растяжения, то критерием является расчетное сопротивление бетона растяжению  $R_i = R_{bt}$  [2].

При анализе скачков деформаций используют только две категории геодинамической безопасности. Вторую категорию присваивают, если текущее напряжение с учетом скачка напряжений больше или равно критерию  $R_i$ , в остальных случаях назначают первую категорию, соответствующую штатному режиму эксплуатации. Третья категория не назначается. Это связано с тем, что скачки вызываются, как правило, сейсмическими быстро проходящими волнами. При всплеске деформаций выше предельной нормы, которая определена для бетона, может появиться трещина, и участок должен быть немедленно осмотрен. Прочие остаточные явления попадают в область анализа трендовых деформаций, а возврат к условиям до прохода волны соответствует штатному режиму первой категории. В случае невозврата к первоначальным условиям рассматривается контроль трендовых процессов.

При анализе трендовых процессов условия назначения первой категории для бетонных и железобетонных обделок в зонах сжатия следующие:

$$\sigma_0 \pm (a_i - a_{\min})E_b < R_{b, mc2}. \quad (1)$$

Условия назначения второй категории:

$$R_b > \sigma_0 \pm (a_i - a_{\min})E_b \geq R_{b, mc2}. \quad (2)$$

Аналогичные условия для третьей категории:

$$R_b \leq \sigma_0 \pm (a_i - a_{\min})E_b. \quad (3)$$

В формулах (1)–(3) приняты следующие обозначения:

$a_{\min}$  — последний зафиксированный минимум деформаций по данным отслеживаемого датчика;

$R_{b, mc2}$  — расчетные сопротивления бетона сжатию при условии недопущения продольных трещин [2];

$R_b$  — расчетные сопротивления бетона осевому сжатию [2].

Бетонные обделки не должны работать на растяжение, поэтому для них с учетом погашения начальных напряжений сжатия первая категория не предусматривается.

Условия назначения третьей категории в зонах растяжения:

$$\sigma_0 \pm (a_i - a_{\min})E_b > R_{bt,ser} \quad (4)$$

Аналогичные условия для второй категории:

$$0 < \sigma_0 \pm (a_i - a_{\min})E_b < R_{bt,ser} \quad (5)$$

В формулах (4), (5)  $R_{bt,ser}$  — расчетное сопротивление бетона осевому растяжению при расчетах по предельным состояниям второй группы [2].

В железобетонных обделках в растянутых зонах работает арматура. Поэтому условия назначения первой категории в зонах растяжения (при условии обеспечения гидроизоляции) следующие:

$$0 < \sigma_0 \pm (a_i - a_{\min})E_b \leq R_{bt}$$

Условия назначения для второй категории:

$$\sigma_0 \pm (a_i - a_{\min})E_b > R_{bt}$$

Третья категория может быть назначена с учетом расчетных проверок участков обделки, испытывающих предельно растянутые напряжения  $R_{bt}$  [2].

Для чугунных и стальных обделок проверяют условия непревышения расчетных сопротивлений материала на сжатие и растяжение. При этом предупреждающую вторую категорию можно назначить при достижении 75 % от значений расчетных сопротивлений материалов.

Кроме анализа мгновенных и трендовых деформационных процессов, контролируются деформации тоннеля в плане. Их анализ основывается на сравнении синхронности показаний двух горизонтальных датчиков на разных стенах. Если показания датчиков двух стен

одного знака, ситуация рассматривается как штатная и соответствует первой категории, если знаки разные, то очевидно, что наблюдается изгиб тоннеля. В этом случае присваивают вторую категорию опасности. Третья категория может быть назначена при дополнительном контроле на участках со второй категорией геометрических параметров положения пути в плане. В таблице приведены условия контроля напряжений, соответствующих назначению определенных категорий геодинамической безопасности (ГДБ). В качестве допущений используют предпосылки об упругой работе материала и принцип независимости действия сил. Контроль с использованием математических и численных моделей позволяет на отдельных участках использовать методы учета физической и контактной нелинейности конструкций.

**Сигнализация и вывод текущей информации оператору.** Отдельный блок программного обеспечения собирает полученную и обработанную информацию и выводит ее оператору в режиме реального времени. Он состоит из системы сигнализации, текстового блока и визуальной модели тоннеля.

Система сигнализации необходима для своевременного предупреждения оператора о развитии опасных процессов в тоннельных конструкциях. Сигнализация представляет собой визуальную систему типа "светофор" с дублирующими звуковыми сигналами и визуальную модель обделки тоннеля с расположенными на ней датчиками.

Светофор показывает интегральную оценку геодинамической безопасности в категориях.

Первая категория соответствует зеленому верхнему сигналу светофора и означает режим штатной работы. При этом контролируемые параметры, ха-

Условия назначения категорий геодинамической безопасности по контролю показаний АМД

Тип обделок	Условия контроля	Зона сжатия	Зона растяжения	Категории ГДБ
Бетонные и железобетонные обделки	Скачки показаний одного датчика	$\sigma_0 \pm a_i E_b \leq R_b$	$\sigma_0 \pm a_i E_b \leq R_{bt}$	1
		$\sigma_0 \pm a_i E_b > R_b$	$\sigma_0 \pm a_i E_b > R_{bt}$	2
	Тренды показаний одного датчика	$\sigma_0 \pm (a_i - a_{\min})E_b < R_{b,mc2}$	—	1
		$R_b > \sigma_0 \pm (a_i - a_{\min})E_b \geq R_{b,mc2}$		2
		$\sigma_0 \pm (a_i - a_{\min})E_b \geq R_b$		3
Бетонные обделки	Тренды показаний одного датчика	—	$0 < \sigma_0 \pm (a_i - a_{\min})E_b < E_{bt,ser}$	2
			$\sigma_0 \pm (a_i - a_{\min})E_b > R_{bt,ser}$	3
Железобетонные обделки		—	$0 < R_0 \pm (a_i - a_{\min})E_b \leq R_{bt}$	1
			$\sigma_0 \pm (a_i - a_{\min})E_b > R_{bt}$	2
Чугунные и стальные обделки	Скачки и тренды одного датчика	$\sigma_0 \pm (a_i - a_{\min})E_s^* < 0,75R_s^{**}$	$\sigma_0 \pm (a_i - a_{\min})E_s < 0,75R_s$	1
		$0,75R_s < \sigma_0 \pm (a_i - a_{\min})E_s < R_s$	$0,75R_s < \sigma_0 \pm (a_i - a_{\min})E_s < R_s$	2
		$\sigma_0 \pm (a_i - a_{\min})E_s > R_s$	$\sigma_0 \pm (a_i - a_{\min})E_s > R_s$	3
Любые обделки	Синхронность горизонтальных датчиков двух стен	Сжатие — сжатие	Растяжение — растяжение	1
		Сжатие — растяжение	Растяжение — сжатие	2

\*  $E_s$  — модуль упругости материала.

\*\*  $R_s$  — расчетное сопротивление материала на сжатие и растяжение.

рактизирующие работу тоннельных конструкций, не выходят за границы допустимых критериев.

Вторая категория соответствует желтому среднему сигналу светофора и имеет значение предупреждающего характера. Оператору выводится сообщение о зонах тоннельных обделок и условия, по которым назначена вторая категория геодинамической безопасности.

Третья категория соответствует красному нижнему сигналу светофора и означает близость конструктивной системы к нештатной ситуации. Сигнал сообщает оператору о расположении критических зон и показаниях, соответствующих назначению третьей категории.

Для красного и желтого сигналов светофора по запросу "показать" на визуальной схеме обделки тоннеля по очереди демонстрируются деформирующиеся участки тоннеля с датчиками, передающими опасные сигналы, параллельно отражается участок тоннеля в цветах, соответствующих разным категориям геодинамической безопасности.

Оператору формируются сообщения диагностической карты и инструкции для выполнения определенных действий.

Визуальная модель тоннеля представляет собой трехмерную модель обделки тоннеля, всегда отображающуюся на окне электронного деформационного мониторинга (рис. 3, см. вторую сторону обложки). Модель масштабируется и позволяет увидеть каждое сечение с установленными датчиками. Каждый датчик имеет индикацию работы и его технического состояния. На модели в режиме реального времени отображается рабочее/нерабочее состояние датчиков и их показания. Участок тоннеля, соответствующий определенной категории безопасности, цветом (1 категория — обычный серый цвет бетона, 2 категория — желтый цвет, 3 категория — красный цвет) отображается на экране.

### **Создание и поддержка математических и численных моделей конструкций тоннеля с возможностью проведения анализа напряженно-деформированного состояния на внешние показатели, регистрируемые подсистемами мониторинга**

На основе получаемых показателей внешних воздействий из систем мониторинга можно смоделировать работу тоннеля в долгосрочной перспективе и оценить оставшийся ресурс тоннеля. Для этого требуется создание математических и численных моделей конструкций тоннеля с возможностью проведения анализа напряженно-деформированного состояния на внешние показатели.

Моделирование конструкций тоннеля возможно с помощью ПК Midas/GTS, Mark или других программных продуктов, использующих численные методы расчета. Начальное напряженно-деформированное состояние может быть определено с использованием экспериментальных методов. Проверка адекватности математических моделей может быть

проверена проведением тестовых статических и динамических испытаний. Исходными данными для построения численных моделей служат материалы проектной, исполнительной документации и материалы обследований. Математическое и численное моделирование позволяет создавать арсенал моделей различной сложности для решения различных задач оценки напряженно-деформированного состояния [3].

### **Оценка технического состояния конструкций с назначением категорий геодинамической безопасности**

Оценка технического состояния тоннельных конструкций с использованием автоматизированных систем мониторинга направлена в первую очередь на оперативное выявление степени опасности в условиях непредсказуемых геодинамических и геодинамических факторов. В статье [1] предложена оценка категорий геодинамической безопасности транспортных тоннелей, которая выполняется по результатам анализа показателей инструментальных систем мониторинга и по результатам численных моделей. В таблице сведены условия назначения категорий геодинамической безопасности по контролю показаний автоматизированного деформационного мониторинга. По результатам расчетов численных моделей выявляются опасные значения внешних геодинамических и геодинамических показателей, контролируемых геодинамическими и динамическими подсистемами мониторинга, при превышении которых назначается третья категория. В аналитическом контуре схемы (см. рис. 1) сопоставляются категории безопасности для всех участков тоннелей, полученные по данным оперативного контроля показаний программно-аппаратного комплекса и результатам численного моделирования.

### **Заключение**

Сформулированы технологические основы программно-аппаратного комплекса для автоматизированного мониторинга деформаций железнодорожных тоннелей. Структура программного обеспечения связана с задачами выполнения блока сбора, хранения и обработки данных, блока анализа напряженно-деформированного состояния тоннельных конструкций с помощью численных моделей, аналитического блока назначения категории геодинамической безопасности, блока сигнализации, визуализации и формирования управляющих решений.

Разработаны принципы построения аппаратной части, включающей в себя датчики контроля деформаций конструкций, приемо-регистрирующее устройство, блоки сбора информации и передачи на сервер. Эти блоки связаны со специализированным программным обеспечением и дополнительными блоками, реализующими функции сбора, обработки, хранения и передачи данных на удаленные рабочие места.

Сформирован аналитический блок, отвечающий за текущий контроль деформаций тоннеля. Он ос-

новывается на работе программного обеспечения, предназначенного для назначения трех категорий геодинамической безопасности. Такое назначение выполняется в оперативном режиме по сформулированным правилам контроля поступающих данных о деформациях обделки тоннеля и в текущем режиме с использованием программ математического моделирования работы тоннеля.

Разработаны основы системы сигнализации и визуализации результатов работы аналитического блока и программного обеспечения для этого блока. Блок системы сигнализации и визуализации предназначен для информирования оператора о степени опасности тоннеля на различных участках с учетом полученных категорий. С учетом трех категорий гео-

динамической безопасности используется сигнализация типа "светофор", трехмерная модель тоннеля и текстовый блок с сообщениями диагностических карт и управляющими решениями.

#### Список литературы

1. **Быкова Н. М., Зайнагабдинов Д. А., Белялов Т. Ш.** Критерии контроля параметров электронного деформационного мониторинга в транспортных тоннелях // Транспортные сооружения, 2015. Т. 2. № 2. URL: <http://t-s.today/PDF/02TS215>.
2. **СП 35.13330.2011.** Мосты и трубы. Актуализир. ред. СНиП 2.05.03-84\*; введ. 2011-05-20. М.: ЦНИИС, 2011. 341 с.
3. **Зайнагабдинов Д. А., Май Дык Минь.** Математические модели при автоматизированном мониторинге тоннелей в зонах разломов // Современные технологии. Системный анализ. Моделирование. 2013. № 4 (40). С. 66–72.

## Technological Bases of Software-Instrumental Accompaniment Electronic Deformation Monitoring of Railway Tunnels

**N. M. Bykova**, [bikovanm@mail.ru](mailto:bikovanm@mail.ru), **D. A. Zainagabdinov**, [damirmt@mail.ru](mailto:damirmt@mail.ru),  
**T. Sh. Belyalov**, [belyalov\\_timur@mail.ru](mailto:belyalov_timur@mail.ru), **I. V. Meshkov**, [nerig@rambler.ru](mailto:nerig@rambler.ru),  
Irkutsk State University of Railway Transport, Irkutsk, 664074, Russian Federation

*Corresponding author:*

**Meshkov Ivan V.**, Postgraduate Student, State University of Railway Transport, Irkutsk, 664074, Russian Federation, e-mail: [nerig@rambler.ru](mailto:nerig@rambler.ru)

*Received on October 10, 2015  
Accepted on November 19, 2015*

*In the article the authors developed the methods of constructing the hardware and software complex for automated monitoring of deformations (AMD) railway tunnels. The technological scheme of AMD, general description of the software-instrumental complex and basics of software development are discussed. The principles of creation of the hardware including sensors of the control strain of construction, receiving and recording device, the information acquisition unit and transfer to the server were developed. The relationship between acquisition units and the data with specialized software, and optional units developmental functions of collection, processing, storage and transmission of data to remote workstations are considered. The software structure associated with a task execution unit for collection, storage and processing of data, unit of analysis of the stress-strain state of tunnel structures using numerical models, the analytical unit category assignments of geodynamic safety, alarm, visualization and the generation of control solutions unit are describe. The analytical unit responsible for the monitoring of deformations of tunnel and intended to assign three categories of geodynamic safety, development the fundamentals of alarm systems and visualization of the results of the analytical unit and the software for this unit are also discussed in this paper.*

**Keywords:** software, deformation monitoring, geodynamic safety, railway tunnels

*For citation:*

**Bykova N. M., Zainagabdinov D. A., Belyalov T. Sh., Meshkov I. V.** Technological Bases of Software-Instrumental Accompaniment Electronic Deformation Monitoring of Railway Tunnels, *Programmnyaya Inzheneriya*, 2016, vol. 7, no 3, pp. 127–131.

DOI: 10.17587/prin.7.127-131

#### References

1. **Bykova N. M., Zainagabdinov D. A., Belyalov T. Sh.** Kriterii kontrolja parametrov jelektronnogo deformacionnogo monitoringa (JeDM) v transportnyh tonneljah (Criteria for control parameters of the electronic deformation monitoring in transportation tunnels, *Transportnye Sooruzhenija*, 2015, vol. 2, no. 2, available at: <http://t-s.today/PDF/02TS215> (in Russian).

2. **СП 35.13330.2011.** Mosty i truby. Aktualizir. red. SNiP 2.05.03-84\*; vved. 2011-05-20. Moscow, ЦНИИС, 2011. 341 p.

3. **Zainagabdinov D. A., Maj Dyk Min'.** Matematicheskie modeli pri avtomatizirovannom monitoringe tonnej v zonah razlomov (Mathematical models with computer-aided monitoring of tunnels in fault zones), *Sovremennye Tehnologii. Sistemnyj Analiz. Modelirovanie*, 2013, no. 4 (40), pp. 66–72 (in Russian).

М. В. Юрушкин, инженер-программист, e-mail: m.yurushkin@gmail.com,  
Южный федеральный университет, Ростов-на-Дону

## Двойное блочное размещение данных в оперативной памяти при решении задачи умножения матриц

*Представлен алгоритм умножения матриц, использующий двойное блочное размещение данных в оперативной памяти. Такое размещение данных позволяет существенно сократить число промахов к данным кеш-памяти, к кеш-памяти TLB и добиться производительности 97 % от пиковой. В заключительной части статьи приведено сравнение результатов работы представленного алгоритма с результатами работы существующих высокопроизводительных аналогов (MKL, PLASMA, OpenBLAS). Автор подчеркивает, что реализованный алгоритм на вход принимает матрицы, изначально размещенные нестандартным образом, в отличие, например, от алгоритмов, реализованных в пакетах MKL и OpenBLAS, которые принимают матрицы, размещенные по столбцам либо по строкам согласно стандарту. Как следствие, предлагаемый алгоритм не заменяет существующие высокопроизводительные алгоритмы, а только их дополняет.*

**Ключевые слова:** кеш-память, блочное размещение матриц, блочные алгоритмы, высокопроизводительные вычисления

### Введение

На протяжении последних 40 лет активно развиваются различные алгоритмы умножения матриц [1, 2]. Это, в частности, связано с тем обстоятельством, что к решению задачи умножения матриц сводятся алгоритмы решения других задач (например, один из алгоритмов решения СЛАУ [3], QR-разложение матрицы [4, 5], LU-разложение матрицы [6]).

На настоящее время различные алгоритмы умножения матриц реализованы во многих библиотеках решения задач линейной алгебры (Lapack, MKL, OpenBLAS, PLASMA и т.д.). Такое разнообразие библиотек обусловлено тем обстоятельством, что сложно охватить все множество существующих вычислительных систем. Учитывая тот факт, что вычислительные системы с каждым годом стремительно усложняются, особый интерес представляют новые высокопроизводительные алгоритмы, а также используемые ими методы оптимизации.

Одним из главных факторов, сдерживающих достижение высокой производительности, является скорость доступа к оперативной памяти, которая существенно ниже скорости выполнения арифметических операций. По этой причине некоторые алгоритмы, которые использовались 20–30 лет назад [7, 8], сейчас практически не находят применения [9, 10]. В работе [11] показано, что алгоритм Штрассена, по сравнению со стандартным алгоритмом умножения матриц, требует большее число операций с памятью, что делает его неприменимым на практике. Кроме

того, очевидно, что он хуже по сравнению со стандартным алгоритмом распараллеливается.

В данной работе представлено описание нового алгоритма умножения матриц, в основе которого лежит использование нестандартного размещения матриц в оперативной памяти. В программной реализации задействованы различные методы оптимизации для процессора, имеющего кеш-память и поддержку векторных вычислений. Исследуемая программа была протестирована на процессоре с поддержкой 256-битных векторных регистров AVX. В заключительной части статьи представлены результаты численных экспериментов, в которых проводилось сравнение реализованного алгоритма с функцией DGEMM из библиотек MKL, OpenBLAS, PLASMA.

### 1. Блочное размещение данных

*Блочное размещение матрицы* — это способ хранения матрицы в памяти, при котором матрица разбивается на блоки одинакового размера. Блоки матрицы хранятся в памяти последовательно без промежутков. Элементы, находящиеся внутри одного блока, хранятся в памяти стандартным образом, например по строкам.

В некоторых блочных алгоритмах блочное размещение матриц дает существенное увеличение производительности. Рассмотрим блочный алгоритм умножения квадратных матриц  $C = AB$  (рис. 1).

Если размер матриц  $N$  больше размера виртуальной страницы, то соседние по вертикали элементы

матрицы будут находиться в различных виртуальных страницах. Нетрудно подобрать такой размер блока  $d$ , при котором кеш-память TLB уже не будет способна хранить физические адреса всех используемых виртуальных страниц и в программе будет происходить большое число промахов к кеш-памяти TLB. Ситуация усугубляется еще больше, так как в случае промаха к кеш-памяти TLB процессор вынужден простаивать (в отличие от ситуации с промахом к кеш-памяти). Возникшую в связи с этим задачу можно решить, если разместить матрицы **A**, **B**, **C** блочно с размером блока, равным  $d$ . В этом случае при умножении блоков задействуется минимальное количество виртуальных страниц.

Данные между оперативной памятью и кеш-памятью пересылаются небольшими блоками — кеш-линейками. Размер кеш-линейки обычно равен 32 либо 64 байт и зависит от процессора. В момент времени, когда происходит обращение к данным кеш-линейки, сначала производится проверка на наличие этой кеш-линейки в кеш-памяти. Если кеш-линейка находится в кеш-памяти, то обращение к оперативной памяти не осуществляется. Такая ситуация называется попаданием в кеш-память (*cache hit*). В противном случае происходит кеш-промах (*cache miss*) — считывание (подкачка) кеш-линейки из оперативной памяти.

Если размер блока не кратен размеру кеш-линейки, то блочное размещение матриц позволяет эффективнее использовать кеш-память, чем стандартное размещение матриц, так как при стандартном размещении матриц в кеш-память будут попадать не только элементы перемножаемых блоков, но также элементы и соседних блоков. Кеш-память будет засоряться неиспользуемыми данными, что

отрицательно скажется на вычислительной производительности. Напротив, при блочном размещении матриц в кеш-память будут подкачиваться только элементы перемножаемых блоков.

## 2. Высокопроизводительные пакеты программ линейной алгебры

Статья [2] посвящена построению высокопроизводительного алгоритма умножения матриц. В ней рассмотрено семейство блочных алгоритмов умножения матриц и с помощью аналитических выкладок найден оптимальный алгоритм, при котором время подкачки блоков сравнимо со временем умножения блоков. Исходная реализация алгоритма представлена в библиотеке GotoBLAS [12].

Библиотека OpenBLAS [13] является библиотекой с открытым программным кодом, в основе которой лежит библиотека GotoBLAS. Она оптимизирована под процессоры архитектуры Sandy Bridge, Haswell и Loongson. В частности, в функции умножения матриц данной библиотеки реализована методика упаковки регистров, позволяющая эффективно действовать векторные вычисления.

Библиотека MKL [14] является коммерческой библиотекой, разрабатываемой и поддерживаемой компанией Intel. В ее составе присутствуют модули для решения задач линейной алгебры, функции быстрых преобразований Фурье, генераторы случайных чисел. Библиотека оптимизирована для процессоров Intel. Допускается ее использование в программах на языках C/C++ и Fortran.

Библиотека PLASMA [15] является библиотекой решения задач линейной алгебры, специально оптимизированной для работы на современных многоядерных процессорах. Библиотека PLASMA отличается от своих аналогов (OpenBLAS, MKL) тем, что все алгоритмы, входящие в ее состав, оптимизированы для эффективного использования памяти. Одной из причин высокой производительности этих алгоритмов является то обстоятельство, что они работают только с матрицами, размещенными в памяти блочным образом. На настоящее время поддерживаются алгоритмы решения нескольких наиболее важных задач линейной алгебры, таких как QR-разложение матрицы, LU-разложение матрицы, умножения матриц. Следует, однако, отметить, что в этой библиотеке нет поддержки эффективной работы с разреженными матрицами (*sparse matrices*) и матрицами-полосами (*band matrices*).

```
void matrix_mult(int N, int d) {
    ...
    double *A, *B, *C;
    A = malloc(sizeof(double)*N*N);
    B = malloc(sizeof(double)*N*N);
    C = malloc(sizeof(double)*N*N);

    // инициализация матриц A, B
    ...

    for(di = 0; di < N; i+=d)
    for(dj = 0; dj < N; j+=d)
    for(dk = 0; dk < N; k+=d)
        for(i = di; i < MIN(N, di+d); i++)
        for(k = dk; k < MIN(N, dk+d); k++)
        for(j = dj; j < MIN(N, dj+d); j++)
            C[i*N+j] += A[i*N+k]*B[k*N+j];
    ...
}
```

Рис. 1. Блочный алгоритм умножения квадратных матриц

### 3. Высокопроизводительный алгоритм умножения матриц, использующий двойное блочное размещение данных

Будем предполагать, что перемножаются матрица **A** размера  $M \times K$  и матрица **B** размера  $K \times N$ , а результат умножения сохраняется в матрицу **C** размера  $M \times N$ . В основе программно реализованного автором алгоритма умножения лежит следующий блочный алгоритм умножения матриц, представленный на листинге на рис. 2.

Внутри функции `BlockMult` проводится перемножение блоков матрицы **A** и матрицы **B**, результат сохраняется в блок матрицы **C**. Матрицы при реализации такого алгоритма умножения разбиваются на блоки согласно схеме, представленной на рис. 3.

Значение  $D_k$  является одновременно шириной полосы матрицы **A** и высотой полосы матрицы **B**. Для того чтобы ускорить время подкачки данных из оперативной памяти в кеш-память, данные в матрицах изначально размещают блочно. Матрица **A** размещается в оперативной памяти блоками размера  $D_i \times D_k$ , которые хранятся по столбцам (рис. 4).

Соответственно, формула вычисления адреса блока матрицы **A** имеет вид

$$\text{BlockAAddr}(di, dk) = dkM + diD_k.$$

Матрица **B** размещается в оперативной памяти блоками размера  $D_k \times D_j$ , которые хранятся по строкам (рис. 5).

Соответственно, формула вычисления адреса блока матрицы **B** имеет вид

$$\text{BlockBAddr}(dk, dj) = dkN + djD_k.$$

Матрица **C** разбивается на блоки размера  $D_i \times D_j$ , которые хранятся по строкам (рис. 6).

Соответственно, формула вычисления адреса блока матрицы **C** имеет вид

$$\text{BlockCAddr}(di, dj) = diN + djD_i.$$

Для того чтобы эффективно использовать кеш-память, размер блока матрицы **A** должен принимать максимальное значение и помещаться целиком в кеш-память, в то время как  $D_j \ll \min(D_i, D_k)$ . В реализованном алгоритме  $D_j = 8$ .

Алгоритм умножения блоков матриц **A** и **B** является также блочным (рис. 7).

Перемножаемые блоки при таком размещении разбиваются на блоки меньшего размера согласно схеме, представленной на рис. 8.

Данные внутри перемножаемых блоков матрицы **A** хранятся вертикальными полосами ширины  $L$ . Внутри полосы матрицы **A** элементы хранятся блоками размера  $4 \times L$ . Будем называть такие блоки блоками второго уровня. На рис. 9 продемонстрировано размещение элементов, лежащих внутри блоков матрицы **A**.

Элементы, лежащие внутри блоков матрицы **B** и матрицы **C**, хранятся по строкам. Высота блока

```
void MatrixMult(double* A, double* B, double* C)
{
    for (dk=0; dk<K; dk+=Dk)
    for (di=0; di<M; di+=Di)
    for (dj=0; dj<N; dj+=Dj)
        BlockAAddr = ... // вычисление адреса блока матрицы A
        BlockBAddr = ... // вычисление адреса блока матрицы B
        BlockCAddr = ... // вычисление адреса блока матрицы C
        BlockMult(BlockAAddr, BlockBAddr, BlockCAddr);
}
```

Рис. 2. Основа алгоритма умножения матриц рекордной производительности

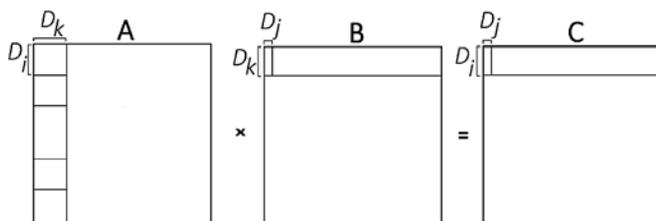


Рис 3. Схема разбиения матриц для высокоуровневой части алгоритма умножения матриц

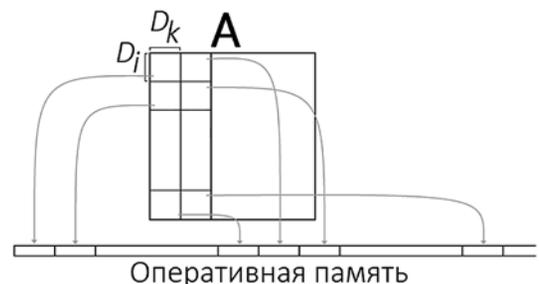


Рис 4. Размещение блоков матрицы A в оперативной памяти

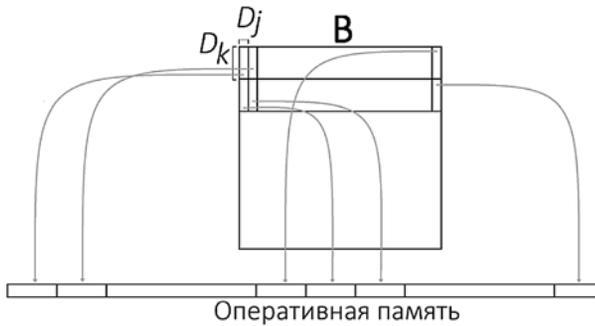


Рис 5. Размещение блоков матрицы В в оперативной памяти

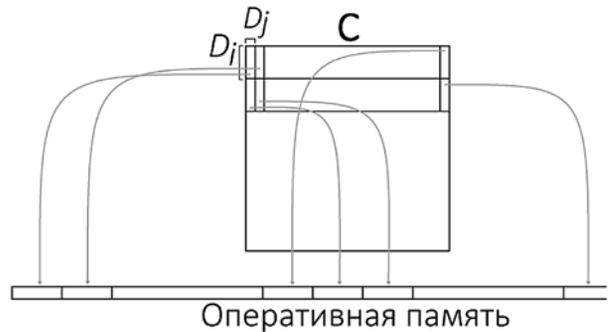


Рис 6. Размещение блоков матрицы С в оперативной памяти

```
void BlockMult(double* BlockA, double* BlockB, double* BlockC)
{
    for (dk=0; dk<Dk; dk+=L)
        for (di=0; di<Di; di+=4)
            for (sk=0; sk<L; ++sk)
                for (si=0; si<4; ++si)
                    for (sj=0; sj<8; ++sj)
                        BlockC[di*8+sj*4+si] +=
                            BlockA[dk*Di+di*L+sk*4+si]*BlockB[dk*8+sk*8+sj];
}
```

Рис. 7. Умножение блоков в рассматриваемом алгоритме умножения матриц

второго уровня матрицы **A** ( $Block_A$ ) и ширина блока матрицы **B** ( $Block_B$ ) выбирают исходя из числа и размеров векторных регистров процессора. Так, процессоры Intel Sandy Bridge/Ivy Bridge/Haswell включают 16 регистров размера 256 бит, каждый из которых может хранить четыре числа типа double. Для такой архитектуры матрицу **A** эффективно разбивать на блоки размера  $4 \times L$ , а матрицу **B** — на блоки размера  $L \times 8$ . При перемножении таких блоков восемь AVX-регистров резервируются для накопления результата, в то время как оставшиеся восемь регистров используются для загрузки данных из памяти. Подробнее использование данной методики описано в работе [2].

Параметр  $L$  выбирается исходя из размера кеш-памяти  $L1$ , а именно —  $L$  должно быть таким, чтобы перемножаемый блок второго уровня матрицы **A**, перемножаемый блок второго уровня матрицы **B**, а также блок матрицы **C** с результатами вычислений помещались в кеш-памяти  $L1$ :

$$(4 \cdot 8 + 4 \cdot L + 8 \cdot L) \text{sizeof}(\text{double}) \leq \text{sizeof}(L1\text{Cache}),$$

где  $\text{sizeof}(\text{double})$  — это размер типа  $\text{double}$  в байтах;  $\text{sizeof}(L1\text{Cache})$  — размер кеш-памяти  $L1$  в байтах.

В этом случае элементы блока матрицы **B** не будут вытесняться между итерациями.

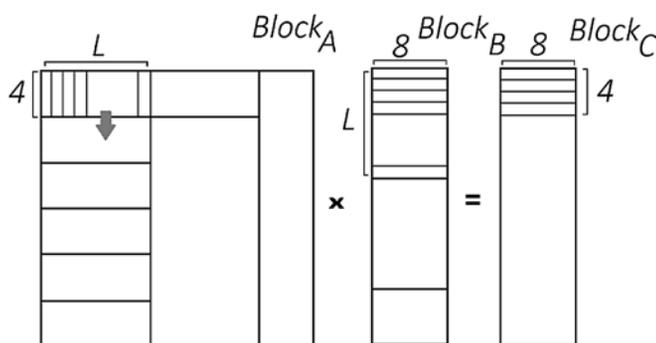


Рис 8. Разбиение блока  $Block_A$  матрицы А, блока  $Block_B$  матрицы В и блока-результата  $Block_C$  матрицы С для размещения их в оперативной памяти

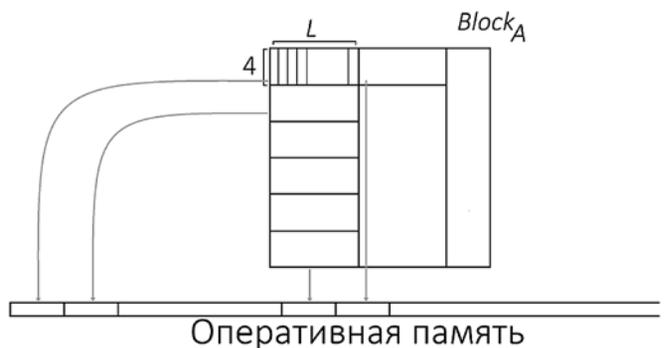


Рис 9. Размещение блоков матрицы А в оперативной памяти

Умножение блоков  $4 \times L$  и  $L \times 8$  реализовано в отдельной процедуре, которая для достижения максимальной производительности запрограммирована на языке ассемблера. Поэтому при переносе на новую архитектуру программную реализацию алгоритма умножения блоков потребуется каждый раз переписывать, что является непростой задачей. Так, для архитектуры Ivy Bridge размер программной реализации алгоритма умножения блоков у автора составил около 800 строк.

Для внутреннего цикла применяется преобразование раскрутки. Раскрутка цикла (*loop unrolling*) — преобразование циклов, заключающееся в многократном дублировании тела циклов с уменьшением числа итераций [16]. Также задействованы программная предвыборка данных (*software prefetching*), переупорядочивание инструкций, выравнивание данных.

Следует отметить, что для низкоуровневой оптимизации программной реализации умножения блоков не требуется напрямую писать код на языке ассемблера. В современных компиляторах языка C++ (например, ICC, GCC, Clang) существует возможность вставки в программную реализацию на языке C++ ассемблерных команд с использованием встроенных функций (*intrinsics*). Также с помощью специальных параметров компиляции (например, параметр `-march` компилятора GCC) имеется возможность указать компилятору генерировать векторные инструкции для конкретной архитектуры.

Автор провел эксперименты, в которых сравнивались программная реализация алгоритма умножения блоков, реализованная полностью на языке ассемблера, и программная реализация алгоритма умножения блоков, оптимизированная компилятором ICC с использованием встроенных функций.

Результаты экспериментов показали, что в случае использования ручной оптимизации достигается дополнительное ускорение в 6 %. Анализ сгенерированного компилятором кода показал, что компилятор неоптимально распределяет векторные регистры, что увеличивает число обращений к оперативной памяти. Также компилятор не самым оптимальным образом расставляет инструкции в раскрученном цикле, что приводит к замедлению работы конвейера. Подробнее о порядке следования инструкций в целях подстройки под конвейер описано в работе [17].

#### 4. Результаты численных экспериментов

В численных экспериментах сравнивалась производительность программных реализаций предлагаемого алгоритма, а также алгоритмов пакетов Intel MKL, PLASMA и OpenBLAS. Тестирование проводилось на вычислительной системе с процессором Intel Core i7-3820 (3600MHz, LGA2011, L3 10240Kb) с выключенной опцией Turbo Boost. Размер кеш-памяти L1 равен 64 Кбайт, размер кеш-памяти L2 равен 1 Мбайт, размер кеш-памяти L3 равен 10 Мбайт. В качестве компилятора использовался Intel C++ Composer XE.

Пиковая производительность вычислительной системы равна

$$PmHV = 4 \cdot 2 \cdot 3,6 \cdot 4 = 115,2 \text{ Гфлопс,}$$

где  $P = 4$  — число физических ядер процессора;  $m = 2$  — число команд, выполняемых параллельно;  $H = 3,6$  ГГц — базовая тактовая частота;  $V = 4$  — количество чисел типа `double`, уместяющихся в векторном регистре.

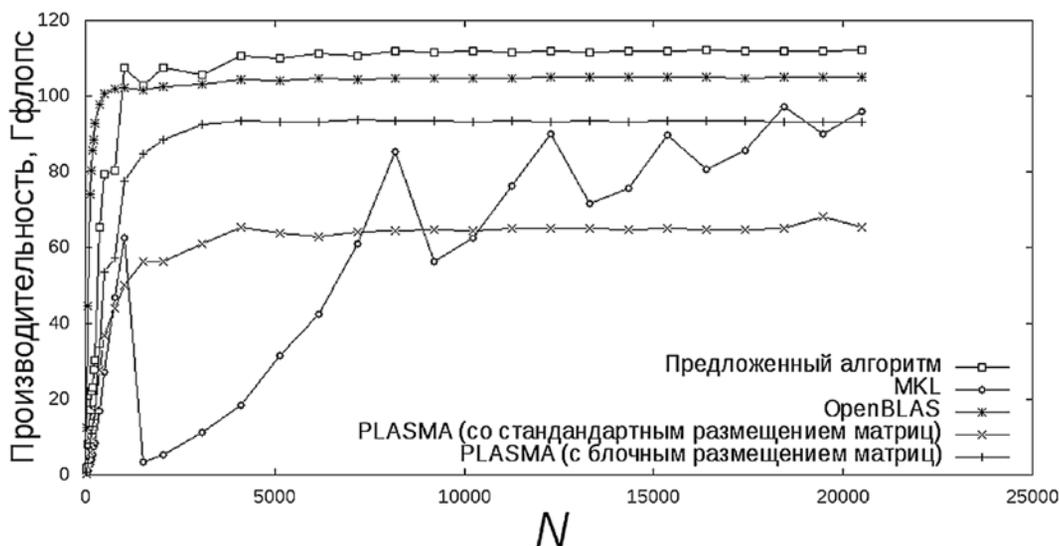


Рис. 10. Графики производительности программных реализаций блочного алгоритма, а также алгоритмов пакетов MKL, OpenBLAS и PLASMA. Размер матрицы A —  $N \times 512$ ; размер матрицы B —  $512 \times N$

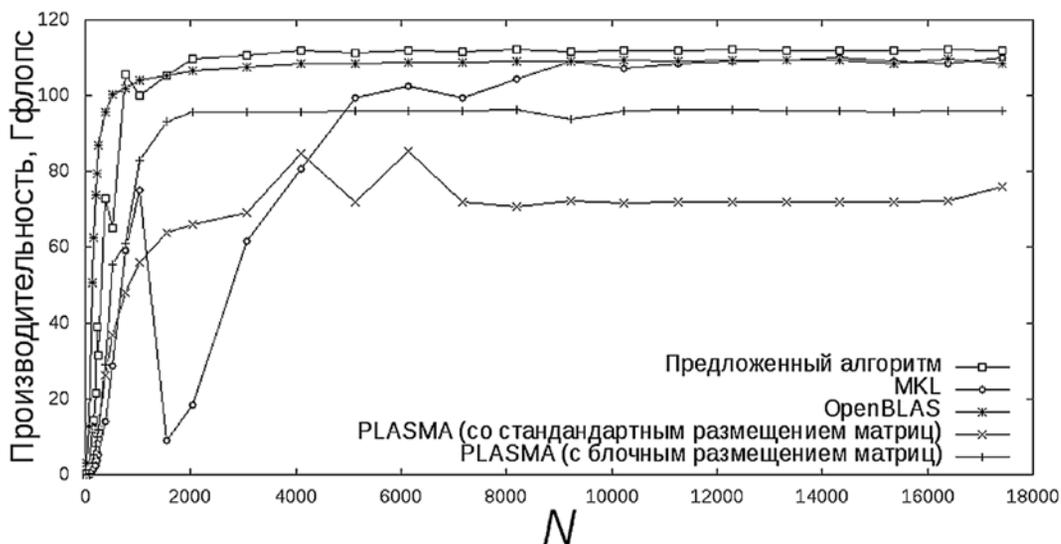


Рис 11. Графики производительности программных реализаций блочного алгоритма, а также алгоритмов из пакетов MKL, OpenBLAS и PLASMA. Матрица **A** и матрица **B** квадратные

Исходя из размера кеш-памяти L3,  $D_i = 512$ ,  $D_k = 256$ ,  $D_j = 8$ ,  $L = 256$ . Ниже приведены результаты проведения трех экспериментов. В каждом из экспериментов сравнивали производительность программной реализации предложенного автором блочного алгоритма с производительностью алгоритмов умножения матриц высокоуровневых библиотек OpenBLAS, MKL, PLASMA.

Библиотека PLASMA предусматривает возможность умножения матриц, размещенных блочно. В экспериментах присутствуют результаты измерений при использовании алгоритма, принимающего на вход стандартно размещенные матрицы, а также алгоритма, принимающего на вход матрицы, размещенные блочно.

Все используемые в экспериментах программные реализации для конкретных входных размеров матриц запускались по 5 раз, и считалось минимальное время работы. Замеры времени работы программных реализаций проводились с использованием процедуры clock. Производительность программной реализации считалась в гигафлопсах по следующей формуле:

$$Performance = \frac{2MKN}{10^9 T},$$

где  $T$  — время работы программы, с.

На рис. 10 представлены результаты сравнения производительности программных реализаций алгоритмов для случая, когда матрица **A** имеет размер  $512 \times N$ , а матрица **B** —  $N \times 512$ .

Полученные результаты свидетельствуют о том, что предложенный автором блочный алгоритм быстрее, чем алгоритмы из других библиотек. В част-

ности, данный алгоритм для  $N = 20480$  работает на 14 % быстрее алгоритма MKL, на 6 % быстрее алгоритма OpenBLAS, на 17 % быстрее алгоритма пакета PLASMA с блочным размещением входных матриц и на 41 % быстрее алгоритма пакета PLASMA со стандартным размещением входных матриц.

На рис. 11 представлены результаты сравнения производительности программных реализаций алгоритмов для случая, когда матрица **A** и матрица **B** имеют размер  $N \times N$ .

Полученные результаты свидетельствуют о том, что предложенный автором блочный алгоритм работает быстрее, чем алгоритмы из других библиотек. В частности, данный алгоритм для  $N = 17408$  работает на 2 % быстрее алгоритма MKL, на 3 % быстрее алгоритма OpenBLAS, на 14 % быстрее алгоритма пакета PLASMA с блочным размещением входных матриц и на 31 % быстрее алгоритма пакета PLASMA со стандартным размещением входных.

На рис. 12 представлены результаты сравнения производительности программных реализаций рассматриваемых алгоритмов для случая, когда размер матрицы **A** равен  $N \times 2048$  и размер матрицы **B** равен  $2048 \times N$ .

Полученные результаты свидетельствуют о том, что алгоритм блочного размещения работает быстрее, чем алгоритмы из других библиотек. В частности, данный алгоритм при  $N = 20480$  работает на 10 % быстрее алгоритма MKL, на 3 % быстрее алгоритма OpenBLAS, на 13 % быстрее алгоритма пакета PLASMA с блочным размещением входных матриц и на 66 % быстрее алгоритма пакета PLASMA со стандартным размещением входных.

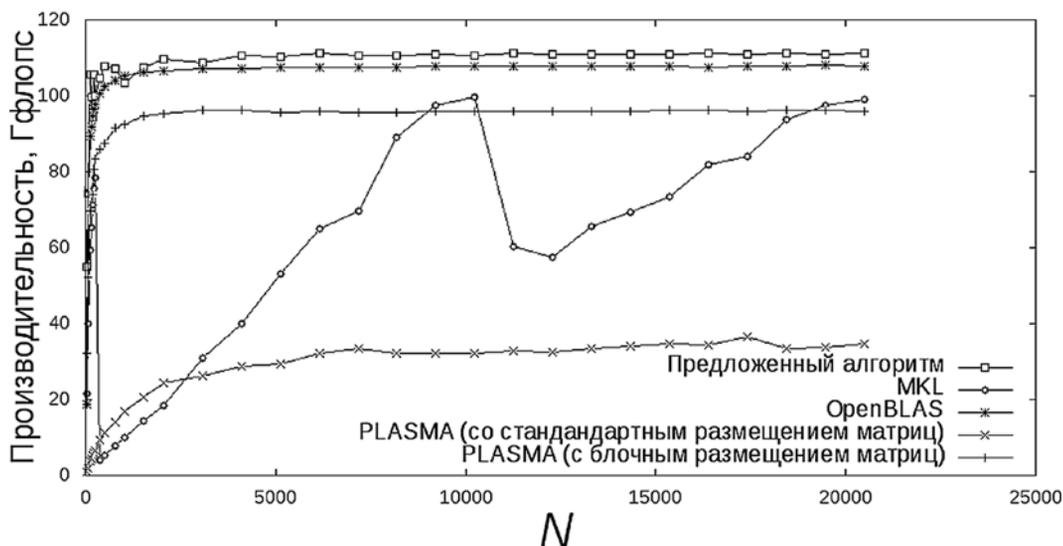


Рис 12. Графики производительности программных реализаций блочного алгоритма, а также алгоритмов из пакетов MKL, OpenBLAS и PLASMA. Размер матрицы  $A$  —  $N \times 2048$ , размер матрицы  $B$  —  $2048 \times N$

### Заключение

Представлен предлагаемый автором высокопроизводительный алгоритм умножения матриц. Начиная с размера матриц  $N > 2000$  как для прямоугольных, так и для квадратных матриц предлагаемый алгоритм опережает все остальные алгоритмы (MKL, PLASMA, OpenBLAS) по производительности. Ближе других по производительности к предлагаемому алгоритму является алгоритм пакета OpenBLAS, который имеет производительность на 3...6 % меньше, чем у предлагаемого алгоритма.

Новизна представленного автором алгоритма состоит в использовании нестандартного иерархического блочного размещения матриц. Такое размещение матриц позволяет уменьшить количество промахов к данным кеш-памяти, к данным кеш-памяти TLB, увеличить скорость подкачки данных в векторные регистры и тем самым увеличить общую производительность алгоритма.

### Список литературы

1. Zuckerman S., Perache M., Jalby W. Fine Tuning Matrix Multiplications on Multicore // High Performance Computing — HiPC 2008, 2008. Lecture Notes in Computer Science. Vol. 5374. P. 30–41.
2. Goto K., Van De Geijn R. A. Anatomy of High-Performance Matrix Multiplication // ACM Trans. Math. Softw. 2008. Vol. 34, N. 3. P. 1–25.
3. Ахо А., Хонкрофт Дж., Ульман Дж. Построение и анализ вычислительных алгоритмов. М.: Мир, 1979, 536 с.
4. Gustavson F. G., Wa'sniewski J. Rectangular full packed format for LAPACK algorithms timings on several computers // In Applied Parallel Computing, State of the Art in Scientific

Computing, PARA 2006, Springer-Verlag, Berlin Heidelberg 2007. Lecture Notes in Computer Science, Vol. 4699, P. 570–579.

5. Gunter B. C., Van De Geijn R. A. Parallel out-of-core computation and updating of the QR factorization // Journal ACM Transactions on Mathematical Software (TOMS). 2005. Vol. 31, Issue 1. P. 60–78.

6. Kurzak J., Luszczek P., Faverge M., Dongarra J. Programming the LU Factorization for a Multicore System with Accelerators // High Performance Computing for Computational Science — VECPAR 2012. 2013, Lecture Notes in Computer Science. Vol. 7851, P. 28–35.

7. Strassen V. Gaussian Elimination is not Optimal // Numerische Mathematik. 1969. Bd. 13. P. 354–356.

8. Coppersmith D., Winograd S. Matrix multiplication via arithmetic progressions // Journal of Symbolic Computation. 1990. N. 9. P. 251–280.

9. Штейнберг Б. Я. Оптимизация размещения данных в параллельной памяти. Ростов-на-Дону: Изд-во Южного фед. ун-та, 2010, 255 с.

10. Штейнберг Б. Я. Блочно-рекурсивное параллельное перемножение матриц // Известия вузов. Приборостроение. 2009. Т. 52, № 10. С. 33–41.

11. Гервич Л. Р., Штейнберг Б. Я., Юрушкин М. В. Разработка параллельных программ с оптимизацией использования структуры памяти. Ростов-на-Дону: Изд-во Южного федерального ун-та, 2014, 120 с.

12. Библиотека GotoBLAS. URL: <http://c2.com/cgi/wiki?GotoBlas>.

13. OPENBLAS. URL: <http://www.openblas.net/>

14. Intel MKL. URL: <http://software.intel.com/en-us/intel-mkl>

15. PLASMA Users' Guide, Parallel Linear Algebra Software for Multicore Architectures, Version 2.3, University of Tennessee, November 2010.

16. Касьянов В. Н. Оптимизирующие преобразования программ. М.: Наука, 1988. 336 с.

17. Fog A. Optimizing subroutines in assembly language: An optimization guide for x86 platforms. Technical University of Denmark. Last updated 2013-09-28.

---

---

# Double Block Data Layout in High Performance Matrix Multiplication Algorithm

M. V. Yurushkin, m.yurushkin@gmail.com, SFEDU, Rostov on Don, 344090, Russian Federation

Corresponding author:

Yurushkin Mikhail V., Engineer, SFEDU, Rostov on Don, 344090, Russian Federation,  
e-mail: m.yurushkin@gmail.com

Received on November 04, 2015

Accepted on December 21, 2015

In this paper matrix multiplication algorithm with double block data layout is suggested. This data layout method allows remarkably decrease amount of cache misses, TLB-cache misses and archive 97 % of peak performance. In the last section results of suggested algorithm with existing packages (MKL, PLASMA, OpenBLAS) comparison are reported. Author outlines that suggested algorithm supports only block matrices in contrast to MKL and OpenBLAS packages, which support matrices with standard data layout. As a consequence, suggested algorithm doesn't replace existing algorithms, but only supplements them.

**Keywords:** cache memory, block data layout, tiling, high performance computing

For citation:

Yurushkin M. V. Double Block Data Layout in High Performance Matrix Multiplication Algorithm, *Programmnyaya Ingeneria*, 2016, vol. 7, no. 3, pp. 132-139.

DOI: 10.17587/prin.7.132-139

## References

1. Zuckerman S., Perache M., Jalby W. Fine Tuning Matrix Multiplications on Multicore, *High Performance Computing — HiPC 2008*, 2008, Lecture Notes in Computer Science, vol. 5374, pp. 30—41.
2. Goto K., Van De Geijn R. A. Anatomy of High-Performance Matrix Multiplication, *ACM Trans. Math. Softw.*, 2008, vol. 34, no. 3, pp. 1—25.
3. Aho A. V., Hopcroft J. E., Ullman J. D. The Design and Analysis of Computer Algorithms. Addison-Wesley, 1976.
4. Gustavson, F. G., Wasniewski, J. Rectangular full packed format for LAPACK algorithms timings on several computers. In *Applied Parallel Computing, State of the Art in Scientific Computing, PARA 2006*, Springer-Verlag, Berlin Heidelberg, 2007. Lecture Notes, vol. 4699, pp. 570—579.
5. Gunter B. C., Van De Geijn R. A. Parallel out-of-core computation and updating of the QR factorization, *Journal ACM Transactions on Mathematical Software (TOMS)*, 2005, vol. 31, issue 1, pp. 60—78.
6. Kurzak J., Luszczek P., Faverge M., Dongarra J. Programming the LU Factorization for a Multicore System with Accelerators, *High Performance Computing for Computational Science — VECPAR 2012*, 2013, Lecture Notes in Computer Science, vol. 7851, pp. 28—35.
7. Strassen V. Gaussian Elimination is not Optimal, *Numerische Mathematik*, 1969, Bd. 13, pp. 354—356.
8. Coppersmith D., Winograd S. Matrix multiplication via arithmetic progressions. *Journal of Symbolic Computation*, 1990, no. 9, pp. 251—280.
9. Steinberg B. Ya. *Optimizacija razmeshhenija dannyh v parallel'noj pamjati* (Optimizing the placement of data in parallel memory). Rostov-na-Donu, Izd-vo Juzhnogo fed. un-ta, 2010, 255 p. (in Russian).
10. Steinberg B. Ya. Blochno-rekursivnoe parallel'noe peremnozhenie matric (Block-recursive parallel matrix multiplication), *Izvestija vuzov. Priborostroenie*, 2009, vol. 52, no10, pp. 33—41 (in Russian).
11. Gervich L. R., Steinberg B. Ya., Yurushkin M. V. *Razrabotka parallel'nyh programm s optimizaciej ispol'zovanija struktury pamjati* (The development of parallel programs by optimizing the use of memory structures), Rostov-na-Donu, Izd-vo Juzhnogo fed. un-ta, 2014, 120 p. (in Russian).
12. GotoBLAS, available at: <http://c2.com/cgi/wiki?GotoBlas>.
13. OPENBLAS, available at: <http://www.openblas.net/>
14. Intel MKL, available at: <http://software.intel.com/en-us/intel-mkl>
15. PLASMA Users' Guide, Parallel Linear Algebra Software for Multicore Architectures, Version 2.3, University of Tennessee, November 2010.
16. Kasyanov V. N. *Optimiziruyushie preobrazovania program* (An optimizing program transformation), Moscow, Nauka, 1988, 336 p. (in Russian).
17. Fog A. Optimizing subroutines in assembly language: An optimization guide for x86 platforms. Technical University of Denmark. Last updated 2013-09-28.

**С. З. Свердлов**, канд. техн. наук, проф., e-mail: c3c@mail.ru;  
Вологодский государственный университет

## Настройка насыщенности при обработке цифровых изображений

*Рассмотрен колориметрически-корректный метод изменения насыщенности, оставляющий неизменными светлоту и цветовой тон. Автором предложен алгоритм, максимизирующий насыщенность цифрового изображения без выхода за охват RGB-пространства.*

**Ключевые слова:** цифровая фотография, цветовая координатная система, обработка цветных цифровых изображений, насыщенность, цветность, программное обеспечение, цветокоррекция

### Введение

В практике обработки изображений используются стандартизованные RGB-пространства, такие как sRGB, Adobe RGB и др. Они являются нелинейными (используют гамма-коррекцию). Цветовые вычисления в таких пространствах некорректны с физической (колориметрической) точки зрения. Для настройки насыщенности предлагается использовать линейризованное RGB-пространство.

Способ вычисления насыщенности, принятый при использовании цветовых координатных систем (цветовых моделей) HSL и HSB, основан на простейших геометрических соображениях и не имеет строгого обоснования. Автором предлагается колориметрически-корректный метод изменения насыщенности, оставляющий неизменными светлоту и цветовой тон. Полученные формулы не предполагают ортогональности RGB-пространства. Также автором вводится понятие относительной насыщенности.

В статье рассматривается алгоритм, максимизирующий насыщенность цифрового изображения. При этом линейная насыщенность пикселей изображения увеличивается в одинаковое число раз без выхода за цветовой охват RGB-пространства. Алгоритм реализован в программе СЗС Color Wizard, предусматривающей автоматическую настройку насыщенности.

Автором предложен основанный на подходе П. Косенко алгоритм ограничения насыщенности с последующим увеличением контрастности, который позволяет в ряде случаев гармонизировать восприятие цифровой фотографии.

### Регулирование насыщенности

Насыщенность — один из атрибутов зрительного ощущения (два других — цветовой тон и светлота), который позволяет судить о степени отличия хро-

матических цветовых стимулов (цветов, излучений) от ахроматических [1, 2].

В практике цветовых расчетов используются две относящиеся к насыщенности величины: насыщенность (*saturation*) и цветность (*chroma*). Первая величина характеризует отличия хроматического стимула от ахроматического независимо от их яркости (светлоты). Вторая величина характеризует отличия стимулов одной яркости (светлоты) [1].

Рассмотрим линейное цветовое RGB-пространство. В качестве примера будем использовать линейризованное стандартное цветовое пространство sRGB [3]. Линейризация sRGB выполняется по формуле

$$C_{\text{лин}} = \begin{cases} \frac{C_{\text{srgb}}}{12,92}, & C_{\text{srgb}} \leq 0,04045 \\ \left(\frac{C_{\text{srgb}}}{1+a}\right)^{2,4}, & C_{\text{srgb}} > 0,04045 \end{cases},$$

где  $a = 0,055$ ;  $C_{\text{srgb}}$  — исходные (например, считанные из файла изображения) гамма-корректированные (нелинейные) sRGB-значения  $R$ ,  $G$  или  $B$  (формулы для преобразования  $R$ ,  $G$  и  $B$  одинаковы);  $C_{\text{лин}}$  — линейризованные значения  $R$ ,  $G$  или  $B$ . Здесь и далее предполагается, что цветовые координаты выражаются вещественными числами в диапазоне от 0,0 до 1,0.

Рассмотрим прямую в линейном RGB-пространстве, соответствующую ахроматическим цветам (черный, серые, белый) — "серую" ось (рис. 1). В сбалансированном RGB-пространстве ахроматические цвета имеют равные RGB-координаты:  $R = G = B$ .

Светлота  $L$  в линейном RGB-пространстве может быть вычислена как значение  $Y$ -координаты цветового пространства CIEXYZ (величина  $Y$  в CIEXYZ равна фотометрической яркости [1]). Например, в линейном пространстве sRGB  $L$  вычисляется с ис-

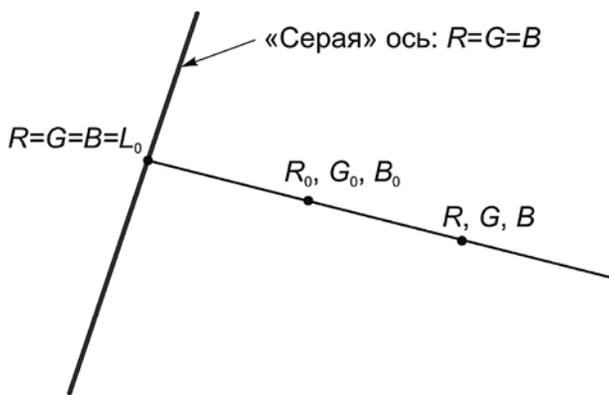


Рис. 1. Прямая изменения насыщенности

пользованием коэффициентов (их сумма равна 1) соответствующей строки матрицы преобразования sRGB → CIEXYZ:

$$L = Y = 0,2126R + 0,7152G + 0,0722B. \quad (1)$$

Здесь  $R$ ,  $G$  и  $B$  — линейные sRGB-координаты,  $R, G, B \in [0, 1]$ .

В соответствии с формулой (1) для ахроматических цветов ( $R = G = B$ ) значение светлоты совпадает со значением  $R$ ,  $G$  и  $B$ :  $L = R = G = B$ .

Рассмотрим точку RGB-пространства, значения цветовых координат которой равны  $R_0$ ,  $G_0$ ,  $B_0$  (рис. 1). Эта точка может соответствовать одному или нескольким пикселям цифрового изображения, цвета которых заданы значениями  $R_0$ ,  $G_0$ ,  $B_0$ . Светлота таких пикселей равна

$$L_0 = 0,2126R_0 + 0,7152G_0 + 0,0722B_0. \quad (2)$$

Отметим на "серой" оси точку с RGB-координатами  $R = G = B = L_0$ . Это точка ахроматического цвета, светлота которой равна светлоте, вычисленной в точке  $(R_0, G_0, B_0)$ . Рассмотрим прямую, проходящую через точки  $(L_0, L_0, L_0)$  и  $(R_0, G_0, B_0)$  (рис. 1). На ней расположены точки одинаковой светлоты и одинакового цветового тона (точки одинакового цветового тона располагаются в одной плоскости, проходящей через "серую" ось). Различные точки на рассматриваемой прямой отличаются только насыщенностью.

При регулировании насыщенности естественно менять ее так, чтобы светлота и цветовой тон точек изображения оставались неизменными. Для этого точка цветового пространства при изменении насыщенности должна двигаться вдоль рассматриваемой прямой.

Запишем уравнение прямой, проходящей через точки  $(L_0, L_0, L_0)$  и  $(R_0, G_0, B_0)$ , в параметрической форме:

$$\begin{aligned} R &= L_0 + (R_0 - L_0)t \\ G &= L_0 + (G_0 - L_0)t \\ B &= L_0 + (B_0 - L_0)t, \end{aligned} \quad (3)$$

где  $t$  — скалярный параметр;  $R_0$ ,  $G_0$ ,  $B_0$  — цветовые координаты точки до изменения насыщенности;  $L_0$  — светлота до, в процессе и после настройки насыщенности, вычисляется по значениям  $R_0$ ,  $G_0$ ,  $B_0$  по формуле (2);  $R$ ,  $G$ ,  $B$  — цветовые координаты точки при изменении насыщенности.

Имеет смысл рассматривать только неотрицательные значения параметра  $t$ . Значению  $t = 0$  соответствует ахроматический цвет (нулевая насыщенность). Значения  $0 < t < 1$  означают уменьшение насыщенности;  $t = 1$  — исходная насыщенность;  $t > 1$  — увеличение насыщенности.

Формулы (3) могут использоваться в программном обеспечении (ПО) обработки изображений для настройки насыщенности. Для увеличения насыщенности в  $k$  раз они должны быть применены к каждому пикселю изображения при одном и том же значении  $t = k$ .

Можно заметить, что в линейной ортогональной цветовой координатной системе (ЦКС), имеющей ось светлоты, например, в ЦКС Лру [2], прямая, вдоль которой изменяется только насыщенность, будет перпендикулярна оси светлоты. Описанный выше метод изменения насыщенности в  $k$  раз будет соответствовать увеличению расстояния точки от оси светлоты в  $k$  раз. Это расстояние, в свою очередь, равно значению цветности (*chroma*), а значение насыщенности (*saturation*) пропорционально цветности [2]. Таким образом, в  $k$  раз изменяются и цветность, и насыщенность.

### Максимальная и относительная насыщенность

Цветовой охват RGB-пространства представляет собой параллелепипед (не куб, поскольку оси RGB не ортогональны). Значения RGB-координат ограничены. В практике цифровой фотографии часто используется диапазон 0...255. При выполнении цветовых расчетов разумно представить цветовые координаты вещественными числами из диапазона 0,0...1,0. Упомянутый параллелепипед ограничен шестью плоскостями:  $R = 0$ ;  $R = 1$ ;  $G = 0$ ;  $G = 1$ ;  $B = 0$ ;  $B = 1$ .

При увеличении параметра  $t$  (при  $t \geq 1$ ) в формулах (3) точка цветового пространства, если она исходно не лежит на "серой" оси, выйдет за пределы охвата. Это произойдет, когда одна из координат достигнет значения 0 или 1, что будет соответствовать пересечению траектории движения точки одной из шести плоскостей, ограничивающих охват.

Обозначим  $C$  одно из значений  $R$ ,  $G$  или  $B$ ;  $C_0$  — одно из значений  $R_0$ ,  $G_0$  или  $B_0$ ;  $C_{\min} = \min(R_0, G_0, B_0)$ ;  $C_{\max} = \max(R_0, G_0, B_0)$ . Из формул (2) и (3) следует,

что минимальное  $t$  (для  $t > 1$ ), при котором значение  $C = L_0 + (C_0 - L_0)t$  будет равно 1, получится при  $C_0 = C_{\max}$ . Обозначим это значение  $t_1$ :

$$t_1 = \frac{1 - L_0}{C_{\max} - L_0}. \quad (4)$$

Минимальное  $t$ , при котором одна из цветовых координат станет равна 0 (обозначим это значение  $t_0$ ), получается при  $C_0 = C_{\min}$ :

$$t_0 = \frac{L_0}{L_0 - C_{\min}}. \quad (5)$$

Если  $t_1 < t_0$ , то выход из охвата происходит вследствие достижения одной из координат максимального значения 1. Если  $t_0 < t_1$ , то выход за охват происходит в силу достижения одной из координат нулевого значения. Значение

$$t_{01} = \min(t_0, t_1) \quad (6)$$

характеризует максимально возможное увеличение насыщенности для данной точки цветового пространства (для данного пикселя изображения), не приводящее к выходу за охват RGB. Обратную величину  $s_{\text{отн}} = \frac{1}{t_{01}}$  можно назвать относительной на-

сыщенностью. Для точек на границе охвата (на одной из плоскостей ограничивающего охват параллелепипеда)  $s_{\text{отн}} = 1$ ; для точек "серой" оси  $s_{\text{отн}} = 0$ ,  $t_{01} = \infty$ ; для других точек  $0 < s_{\text{отн}} < 1$ .

### Автоматическая настройка насыщенности

При обработке цветных цифровых фотографий с помощью специализированного ПО могут быть использованы разные стратегии и цели при регулировании насыщенности. Один из вариантов — получение максимально насыщенного изображения, что позволяет получить красочную и эффектную картинку. При этом, если речь идет о получении реалистичных изображений, желательно, чтобы не происходило выхода цветов пикселей изображения за охват цветового пространства.

Автором предложен представленный далее алгоритм автоматической настройки насыщенности, сохраняющий цветовой тон и светлоту пикселей изображения, выполняющий увеличение насыщенности всех пикселей в одинаковое число раз и максимизирующий насыщенность без выхода за цветовой охват.

**Шаг 1.** Для всех пикселей изображения по формулам (2), (4)—(6) определить значение  $t_{01}$ .

**Шаг 2.** Из найденных для всех пикселей значений  $t_{01}$  определить минимальное  $t_{\min}$  (максимальная относительная насыщенность).

**Шаг 3.** По формулам (3) при  $t = t_{\min}$  вычислить новые RGB-координаты пикселей.

Модификация этого алгоритма, позволяющая устранить влияние случайных и малозначащих обстоятельств, приводящих к появлению на изображении небольшого числа точек высокой насыщенности, может предусматривать построение гистограммы относительной насыщенности и выбор такого значения  $t_{\min}$ , использование которого в формулах (3) приведет к выходу за охват цвета относительно небольшого числа точек изображения, например, 0,05 % от общего числа пикселей изображения. За счет этого можно получить большее увеличение насыщенности изображения. Цвета малого числа точек, выходящие за охват, могут быть подвергнуты отсечению. Простейший вариант отсечения состоит в том, что цветовая координата, превышающая 1, принимается равной 1, координата, ставшая отрицательной, принимается равной 0. При этом искажаются цветовой тон и светлота пикселей. Однако, поскольку речь идет о небольшой доле точек, такое искажение обычно приемлемо.

### Реализация автоматической настройки насыщенности

Описанный выше алгоритм автоматической настройки насыщенности реализован в программе C3C Color Wizard (плагин для Adobe Photoshop, <http://www.uni-vologda.ac.ru/~c3c/plugin/colorwizardbasic.htm>) (рис. 2, см. третью сторону обложки).

На рис. 2 в окне просмотра показан снимок после автоматической настройки экспозиции и автоматической настройки насыщенности. Вначале выполнена настройка экспозиции (уровней белого и черного, параметры "Экспозиция" и "Вуаль"), затем выполнена автоматическая настройка насыщенности, которая привела к усилению насыщенности в 1,19 раза (0,25 на логарифмической шкале программы).

На рис. 3 (см. третью сторону обложки) показаны этапы автоматической обработки этой фотографии в программе C3C Color Wizard.

### Ограничение насыщенности

Другой стратегией настройки насыщенности является ее ограничение. Насыщенность снимка снижается по сравнению с максимально возможной. После этого может быть увеличена контрастность. Такой подход позволяет избежать появления "кричащих" цветов на фотографии, может гармонизировать ее восприятие зрителем [4].

Ограничение насыщенности (в том числе автоматическое) может быть реализовано следующим образом.

**Шаг 1.** Определить, как это описано выше, коэффициент усиления насыщенности  $t_{\min}$ , который позволяет получить максимальную насыщенность без выхода за цветовой охват (с выходом за охват небольшого числа точек).

**Шаг 2.** Для вычисления новых RGB-координат пикселей по формулам (3) использовать коэффициент  $t = dt_{\min}$ , где  $0 < d < 1$  — коэффициент ограничения насыщенности.

**Шаг 3.** Увеличивать контрастность изображения (светлые части становятся светлей, темные — темней). Предел усиления контрастности может выбираться, например, по условию невыхода цветов изображения за охват RGB.

На рис. 4 (см. четвертую сторону обложки) показано тестовое изображение — корзина с фруктами, к которому применено ограничение насыщенности. Коэффициент  $d$  равен 0,81. Контраст в средних тонах увеличен в 1,09 раза.

После обработки фотография выглядит не такой яркой, но при просмотре на мониторе лучше видна фактура поверхностей.

Ограничение насыщенности может уменьшить риск выхода цветов изображения за охват устройства отображения (например, монитора).

## Заключение

Корректность, эффективность и практическая значимость предложенных преобразований и алгоритмов настройки насыщенности подтверждены в процессе эксплуатации ряда программных решений, в которых эти алгоритмы реализованы. Примеры применения показывают, что выполняемые настройки соответствуют субъективному восприятию насыщенности человеком-наблюдателем, а также позволяют избавиться от артефактов, возникающих, в частности, при выполнении преобразований в нелинейных цветовых пространствах.

## Список литературы

1. **Wyszecki G., Stiles W. S.** Color Science. Concepts and Methods, Quantitative Data and Formulae. Second Edition, Wiley-Interscience Publication, 2000.
2. **Свердлов С. З.** Ортогональная цветовая координатная система // Вестник Вологодского государственного педагогического университета. 2013. № 5. С. 25–29. URL: [http://www.uni-vologda.ac.ru/~c3c/articles/Sverdlov\\_S\\_Z\\_Orthogonal\\_color\\_coordinate\\_system.pdf](http://www.uni-vologda.ac.ru/~c3c/articles/Sverdlov_S_Z_Orthogonal_color_coordinate_system.pdf)
3. **International Standard IEC 61966-2-1 ed1.0.** Multimedia systems and equipment — Colour measurement and management — Part 2-1: Colour management — Default RGB colour space — sRGB.
4. **Косенко П.** Живая цифра. М.: Тримедиа Контент, 2013. 283 с.

---

---

# Saturation Adjustment in the Digital Images Processing

**S. Z. Sverdlov**, c3c@uni-vologda.ac.ru, Vologda State University, Vologda, 160600, Russian Federation

*Corresponding author:*

**Sverdlov Sergey Z.**, Professor, Vologda State University, Vologda, 160600, Russian Federation, e-mail: c3c@uni-vologda.ac.ru

*Received on November 03, 2015*

*Accepted on December 22, 2015*

*Standardized RGB-spaces such as sRGB, Adobe RGB and etc. are common for image processing. Those spaces are non-linear (e.g. using gamma correction), so color calculations in such spaces are incorrect from a physical (colorimetric) point of view. Use of the linearized RGB space is proposed for adjusting the saturation.*

*A method for calculating the saturation employed in HSL and HSB color spaces (color models) is based solely on geometric considerations and does not provide a strict justification. The paper proposes a method for modifying the saturation leaving the lightness and color hue unchanged, which is correct from colorimetric perspective. Proposed formulas do not require orthogonal RGB spaces. A concept of relative saturation is introduced.*

*An algorithm for maximizing digital image saturation is proposed. The linear saturation of image pixels is increased by the same factor without moving out of RGB-space gamut. The algorithm is implemented in C3C Color Wizard software that is the only implementation currently known to the author.*

*An algorithm for saturation limitation followed by contrast increase is proposed on the basis of P. Kosenko's approach. The algorithm allows harmonizing visual perception of digital photography in some cases.*

**Keywords:** digital photography, color coordinate system, processing of digital color images, saturation, chroma, software, color adjustment

---

---

*For citation:*

**Sverdlov S. Z.** About Saturation Adjustment in the Digital Images Processing, *Programmная Ingeneria*, 2016, vol. 7, no. 3, pp. 140—144.

DOI: 10.17587/prin.7.140-144.

### References

1. **Wyszecki G., Stiles W. S.** *Color Science. Concepts and Methods, Quantitative Data and Formulae*. Second Edition, Wiley-Interscience Publication, 2000.

2. **Sverdlov S. Z.** Orthogonalnaj zvetovaja koordinatnaja sistema (Orthogonal Color Coordinate System), *Vestnik Vologodskogo*

*gosudarstvennogo pedagogicheskogo universiteta*, 2013, no. 5, pp. 25—29 (in Russian).

3. **International Standard IEC 61966-2-1 ed1.0.** Multimedia systems and equipment — Colour measurement and management — Part 2-1: Colour management — Default RGB colour space — sRGB.

4. **Kosenko P.** *Zhivaja zifra* (Live Digit). Moscow, Trimedia-Content, 2013, 283 p. (in Russian).

---

---

## ИНФОРМАЦИЯ

### *Продолжается подписка на журнал "Программная инженерия" на первое полугодие 2016 г.*

Оформить подписку можно через подписные агентства  
или непосредственно в редакции журнала.

Подписные индексы по каталогам:

Роспечать — 22765; Пресса России — 39795

*Адрес редакции:* 107076, Москва, Стромьинский пер., д. 4,  
Издательство "Новые технологии",  
редакция журнала "Программная инженерия"

Тел.: (499) 269-53-97. Факс: (499) 269-55-10. E-mail: prin@novtex.ru

---

---

**ООО "Издательство "Новые технологии".** 107076, Москва, Стромьинский пер., 4  
Технический редактор *Е. М. Патрушева*. Корректор *З. В. Наумаова*

Сдано в набор 28.12.2015 г. Подписано в печать 17.02.2016 г. Формат 60×88 1/8. Заказ РІ316  
Цена свободная.

Оригинал-макет ООО "Авансед солишнз". Отпечатано в ООО "Авансед солишнз".  
119071, г. Москва, Ленинский пр-т, д. 19, стр. 1. Сайт: [www.aov.ru](http://www.aov.ru)