

ВЕСТНИК НОВОСИБИРСКОГО ГОСУДАРСТВЕННОГО УНИВЕРСИТЕТА

Научный журнал
Основан в ноябре 1999 года

Серия: Информационные технологии

2019. Том 17, № 2

СОДЕРЖАНИЕ

<i>Даниловский К. Н., Дудаев А. Р., Глинских В. Н., Никитенко М. Н., Москаев И. А.</i> Программное обеспечение на основе веб-технологий для геонавигации нефтяных скважин	5
<i>Дрозд О. В., Ченцов С. В., Капулин Д. В.</i> Моделирование структурной организации единого информационного пространства предприятия – разработчика микроэлектронных систем	18
<i>Ерёменко Ю. И., Олюнина Ю. С.</i> Применение метода k -средних для идентификации пользователя по клавиатурному почерку	30
<i>Ерёменко В. С., Наумова В. В.</i> Система каталогизации и мониторинга территориально распределенных вычислительных узлов в среде WPS сервисов для решения геологических задач	39
<i>Иванов Р. А., Валеев Т. Ф.</i> Автоматический рефакторинг Java-кода с использованием Stream API	49
<i>Иртегов Д. В., Нестеренко Т. В., Чурина Т. Г.</i> Системы автоматизированной оценки заданий по программированию: разработка, использование и перспективы	61
<i>Кадирова Л. А.</i> Методологические аспекты применения инструментария ГИС для управления трудоустройством выпускников региональных вузов	74
<i>Куликов И. М., Протасов В. А.</i> Код «Виртуальный планетарий» для моделирования астрофизических объектов: математическая модель, методология и первые результаты	81
<i>Лебедев Р. К., Корякин И. А.</i> Разработка среды программирования для мобильных устройств под управлением операционной системы Android	91
<i>Литвинов В. С., Власов А. А., Тейтельбаум Д. В.</i> Разработка программных средств прозрачного удаленного доступа к технологическим параметрам бурения и данным скважинной телеметрической системы в условиях нестабильной связи	101

<i>Нечахин В. А., Пищик Б. Н.</i> Применение методов глубинного обучения для обнаружения вторжений	114
<i>Петров С. А.</i> Опыт разработки и внедрения информационной системы для поддержки процесса направления студентов на практику в НИУ МЭИ	122
<i>Постаногов И. С., Турова И. А.</i> Платформа комплексирования и тестирования средств трансформации естественно-языковых запросов в SPARQL-запросы	138
Информация для авторов	153

VESTNIK

NOVOSIBIRSK STATE UNIVERSITY

Scientific Journal
Since 1999, November
In Russian

Series: Information Technologies

2019. Volume 17, № 2

CONTENTS

<i>Danilovskiy K. N., Dudaev A. R., Glinskikh V. N., Nikitenko M. N., Moskaev I. A.</i> Web-Technologies Based Software for Oil and Gas Wells Geosteering	5
<i>Drozd O. V., Chentsov S. V., Kapulin D. V.</i> Modeling the Structural Organization of Unified Information Space for the Microelectronics Design Center	18
<i>Eremenko Yu. I., Olyunina Yu. S.</i> Using the <i>K</i> -means Method to Identify a User by Keyboard Handwriting	30
<i>Eremenko V. S., Naumova V. V.</i> The System of Cataloguing and Monitoring of Geographically Distributed Computing Nodes in WPS Services Environment for Geological Problems Solving	39
<i>Ivanov R. A., Valeev T. F.</i> Automatic Refactoring of Java Code Using the Stream API	49
<i>Irtegov D. V., Nesterenko T. V., Churina T. G.</i> Systems for Automated Evaluation of Programming Tasks: Development, Use and Perspectives	61
<i>Kadirova L. A.</i> Methodological Aspects for the Use of GIS Tools for Managing the Employment of Graduates	74
<i>Kulikov I. M., Protasov V. A.</i> “Virtual Planetarium” Code for Astrophysical Objects Modeling: Mathematical Model, Methodology and First Results	81
<i>Lebedev R. K., Koryakin I. A.</i> Development of IDE for Android Devices	91
<i>Litvinov V. S., Vlasov A. A., Teytelbaum D. V.</i> Development of Software for Transparent Remote Access to Technological Parameters of Drilling and Data From a Downhole Telemetry System in Unstable Communication Conditions	101
<i>Nechakhin V. A., Pishchik B. N.</i> Using Deep Learning Methods for Intrusion Detection	114
<i>Petrov S. A.</i> Experience in Development and Implementation of the Information System, Supporting Students Assignment for Practice to NRU MPEI	122
<i>Postanogov I. S., Turova I. A.</i> Platform for Integrating and Testing Tools which Transform Natural Language Queries into SPARQL Queries	138
Instructions to Contributors	153

Editor in Chief Anatolij M. Fedotov

Vice-Editor A. V. Avdeev

Executive Secretary N. N. Pestereva

Editorial Board of the Series

- I. V. Bychkov*, professor, academician (Irkutsk), *B. M. Glinsky*, professor (Novosibirsk)
A. N. Gorban', professor (Lester, GB), *E. P. Gordov*, professor (Tomsk)
B. S. Dobronets, professor (Krasnoyarsk), *A. M. Elizarov*, professor (Kazan)
G. N. Erokhin, professor (Kaliningrad), *A. I. Kamyshnikov*, professor (Khanty-Mansijsk)
G. P. Karev, professor (Maryland, USA), *N. A. Kolchanov*, professor, academician (Novosibirsk)
M. M. Lavrentjev, professor (Novosibirsk), *V. E. Malyshkin*, professor (Novosibirsk)
N. N. Mirenkov, professor (Aizu, Japan), *N. M. Oskorbin*, professor (Barnaul)
D. E. Palchunov, professor (Novosibirsk), *T. Pizansky*, professor (Ljubljana, Slovenia)
V. P. Potapov, professor (Kemerovo), *O. I. Potaturkin*, professor (Novosibirsk)
V. A. Serebryakov, professor (Moscow), *A. V. Starchenko*, professor (Tomsk)
S. I. Smagin, professor, corresponding member of RAS (Khabarovsk)
D. A. Tusupov, professor (Astana, Kazakhstan)
V. V. Shajdurov, professor, corresponding member of RAS (Krasnoyarsk)
Yu. I. Shokin, professor, academician (Novosibirsk)

*The journal is published quarterly in Russian since 1999
by Novosibirsk State University Press*

The address for correspondence

Faculty of Information Technologies, Novosibirsk State University

1 Pirogov Street, Novosibirsk, 630090, Russia

Tel. +7 (383) 363 42 46

E-mail address: inftech@vestnik.nsu.ru

On-line version: <http://elibrary.ru>

Программное обеспечение на основе веб-технологий для геонавигации нефтегазовых скважин

К. Н. Даниловский¹, А. Р. Дудаев^{1,2}, В. Н. Глинских¹⁻³
М. Н. Никитенко¹, И. А. Москаев^{1,2}

¹ Институт нефтегазовой геологии и геофизики им. А. А. Трофимука СО РАН

² Новосибирский государственный университет

³ Новосибирский государственный технический университет
Новосибирск, Россия

Аннотация

Для эффективного освоения залежей углеводородов чрезвычайно важной становится высокая точность проводки горизонтальных скважин в целевых пластах-коллекторах. В процессе бурения геонавигация скважины со сложной траекторией выполняется по геофизическим данным в реальном масштабе времени. Представленная работа посвящена разработке нового программного обеспечения для геонавигации нефтегазовых скважин с горизонтальным завершением. Для решения задач геонавигации используются алгоритмы, основанные на методах корреляции каротажных данных и численной инверсии измерений электромагнитного каротажа. Разработанное новое программное приложение основано на веб-технологиях и имеет клиент-серверную архитектуру. Для оптимизации времени выполнения ресурсоемких расчетов используются высокопроизводительные облачные вычисления.

Ключевые слова

программное обеспечение, веб-технологии, геонавигация, каротаж в процессе бурения, горизонтальная скважина, численная инверсия, высокопроизводительные вычисления, клиент-серверная архитектура, облачные вычисления

Благодарности

Авторы благодарят Научно-производственное предприятие геофизической аппаратуры «Луч» за предоставленные практические материалы.

Развитие программно-алгоритмической базы математического моделирования и инверсии электромагнитных полей в пространственных моделях сред выполнено в рамках ФНИ № 0331-2019-0014 «Скважинная геофизика в электропроводящих анизотропных диспергирующих средах на основе высокопроизводительных решений трехмерных задач, высокоточных данных каротажа и лабораторных исследований керна».

Для цитирования

Даниловский К. Н., Дудаев А. Р., Глинских В. Н., Никитенко М. Н., Москаев И. А. Программное обеспечение на основе веб-технологий для геонавигации нефтегазовых скважин // Вестник НГУ. Серия: Информационные технологии. 2019. Т. 17, № 2. С. 5–17. DOI 10.25205/1818-7900-2019-17-2-5-17

Web-Technologies Based Software for Oil and Gas Wells Geosteering

K. N. Danilovskiy¹, A. R. Dudaev^{1,2}, V. N. Glinskikh¹⁻³
M. N. Nikitenko¹, I. A. Moskaev^{1,2}

¹ Trofimuk Institute of Petroleum Geology and Geophysics SB RAS

² Novosibirsk State University

³ Novosibirsk State Technical University
Novosibirsk, Russian Federation

Abstract

Accuracy of the horizontal well placement in the target reservoir becomes essential for efficient oilfield development. Geosteering of a well with a complex trajectory is performed using real-time geophysical data obtained while drilling.

The presented work is devoted to the development of a new software for horizontal oil and gas wells geosteering. Algorithms based on logging data correlation and electromagnetic logging data numerical inversion methods are used for well placement. The developed application is based on web-technologies and has a client-server architecture. To optimize the resource-intensive calculations execution time, high-performance cloud computing is used.

Keywords

software, web-technologies, geosteering, logging while drilling, horizontal well, numerical inversion, high-performance computations, client-server architecture, cloud computations

Acknowledgements

The authors acknowledge Scientific Production Enterprise of geophysical equipment “Looch” (SPE “Looch”) for the provided practical materials.

The development of the software-algorithmic base of electromagnetic fields mathematical modeling and inversion in spatial models of media was carried out within the FSR No. 0331-2019-0014 “Borehole geophysics in electrically conductive anisotropic dispersing media based on three-dimensional problems high-performance solutions, high-precision logging data and laboratory core studies”.

For citation

Danilovskiy K. N., Dudaev A. R., Glinskikh V. N., Nikitenko M. N., Moskaev I. A. Web-Technologies Based Software for Oil and Gas Wells Geosteering. *Vestnik NSU. Series: Information Technologies*, 2019, vol. 17, no. 2, p. 5–17. (in Russ.) DOI 10.25205/1818-7900-2019-17-2-5-17

Введение

В связи с повсеместным переходом в бурении от вертикальных скважин к скважинам с горизонтальным завершением существенно усложняется и расширяется круг задач промышленной геофизики. При бурении горизонтального участка скважины применяют геонавигацию – оперативную корректировку траектории бурения с целью ее удержания в целевом нефтеносном пласте. Следует отметить, что геонавигация – это сложный процесс, требующий использования дорогостоящего оборудования и программного обеспечения (ПО), а также работы квалифицированных геофизиков-геонавигаторов. Решение о корректировке траектории принимают непосредственно в процессе бурения на основе каротажных данных с использованием приборов, входящих в состав буровой колонны. Стандартные программные комплексы, предназначенные для интерпретации каротажных данных из вертикальных скважин, не подходят для проведения геонавигации, что приводит к необходимости разработки специализированного ПО.

Каротаж в процессе бурения начал активно развиваться еще в 80-х гг. прошлого столетия [1]. Несмотря на это, в России для каротажа в процессе бурения до сих пор используют приборы в основном только крупных иностранных нефтесервисных компаний: *Baker Hughes*, *Halliburton*, *Schlumberger*, *Weatherford*. В рамках реализации программ импортозамещения *Научно-производственным предприятием геофизической аппаратуры «Луч»* (НПП ГА «Луч», Новосибирск) и *Институтом нефтегазовой геологии и геофизики им. А. А. Трофимука СО РАН* (ИНГГ СО РАН, Новосибирск) разработан и опробован на месторождениях Западной Сибири аппаратный комплекс «ЛУЧ». Комплекс включает в себя ряд приборов для проведения каротажа в процессе бурения, при этом набор геофизических измерений достаточен для проведения как геонавигации, так и петрофизической интерпретации.

Оперативная интерпретация данных каротажа в процессе бурения может проводиться либо самим заказчиком – нефтедобывающей компанией, либо теми же нефтесервисными компаниями. Для нефтедобывающих компаний на российском рынке в настоящее время представлен ряд программных решений для геонавигации: «Геонафт» от компании *Геонавигационные технологии*, «Geosteering office» от *Геонавигация*, «StarSteer» от *ROGII*. Однако всё перечисленное ПО обладает одним недостатком: не использует наиболее продвинутые подходы для задач геонавигации, основанные на алгоритмах численных решений прямых и обратных задач электродинамики в классе реалистичных интерпретационных моделей. Сервисные компании используют ПО, узко ориентированное на «свои» приборы. В связи с этим в ИНГГ СО РАН разработано ПО, позволяющее проводить геонавигацию скважин

с использованием комплекса «ЛУЧ» двумя методами: корреляции каротажных данных и численной инверсии измерений электромагнитного каротажа (ЭМК).

Аппаратурный комплекс «ЛУЧ» для каротажа в процессе бурения

Приборы, входящие в состав комплекса каротажа в процессе бурения (рис. 1), позволяют проводить геофизические исследования в скважинах следующими методами:

- многозондовый высокочастотный индукционный каротаж (ВИКПБ);
- боковой сканирующий каротаж (БКС);
- гамма-каротаж (ГК);
- нейтрон-нейтронный каротаж по тепловым нейтронам (ННК-Т);
- гамма-гамма плотностной каротаж (ГГК-П);
- другие измерения (инклинометрия, положения угла отклонителя, температура, ударные нагрузки).



Рис. 1. Приборы каротажа в процессе бурения, входящие в состав комплекса «ЛУЧ»:
а – ВИКПБ; б – БКС

Fig. 1. Logging while Drilling Tools Included in the Complex “LOOCH”:
a – VIKPB; b – BK/BKS

Для решения ключевых задач геологического сопровождения бурения используют данные ЭМК из-за его наибольшей радиальной глубинности. Входящий в аппаратный комплекс «ЛУЧ» прибор ВИКПБ, основанный на методе высокочастотного индукционного каротажа ВИКИЗ [2; 3], выполняет измерения относительных амплитудно-фазовых характеристик эдс, наведенной в приемных катушках. Измерения эдс проводятся в двух основных и четырех дополнительных трехкатушечных зондах с длинами от 0,7 до 1,4 м на двух частотах 0,88 и 3,5 МГц. Зонды одинаковой длины различаются базой, т. е. расстоянием между дальней и ближней приемными катушками. В парах приемных катушек регистрируются разности фаз наведенных эдс $\Delta\varphi$ и отношения амплитуд ΔA , а также выполняется их трансформация в кажущееся удельное электрическое сопротивление (УЭС).

Проведенные опытно-промышленные испытания приборов каротажа в процессе бурения «ЛУЧ» на месторождениях АО «НК «Роснефть» и АО «Сургутнефтегаз» показывают, что приборы соответствуют техническим и эксплуатационным требованиям для бурения наклонно направленных и горизонтальных скважин [4; 5]. Однако для успешной проводки скважины недостаточно одних приборов каротажа в процессе бурения, необходимо также и специализированное ПО для оперативной интерпретации геофизических данных.

Архитектура и ИТ-технологии

Разработанное ПО для геонавигации представляет собой веб-приложение, состоящее из трех основных структурных блоков: клиентского, серверного и вычислительного (рис. 2).

Клиент отвечает только за визуализацию данных, что сводит к минимуму требования, предъявляемые к оборудованию. Использовать такое ПО можно на любых устройствах: от персональных компьютеров до смартфонов. Требуется лишь наличие веб-браузера и доступ к серверу по сети. При этом обновление приложения выполняется для всех клиентов одновременно благодаря использованию единого ресурса. Серверная часть обрабатывает пользовательские запросы, управляет очередностью выполнения ресурсоемких задач и возвращает результаты расчетов клиентам. В вычислительной части непосредственно производятся расчеты на серверах (кластерах) с использованием распределенных масштабируемых и высокопроизводительных параллельных вычислений.

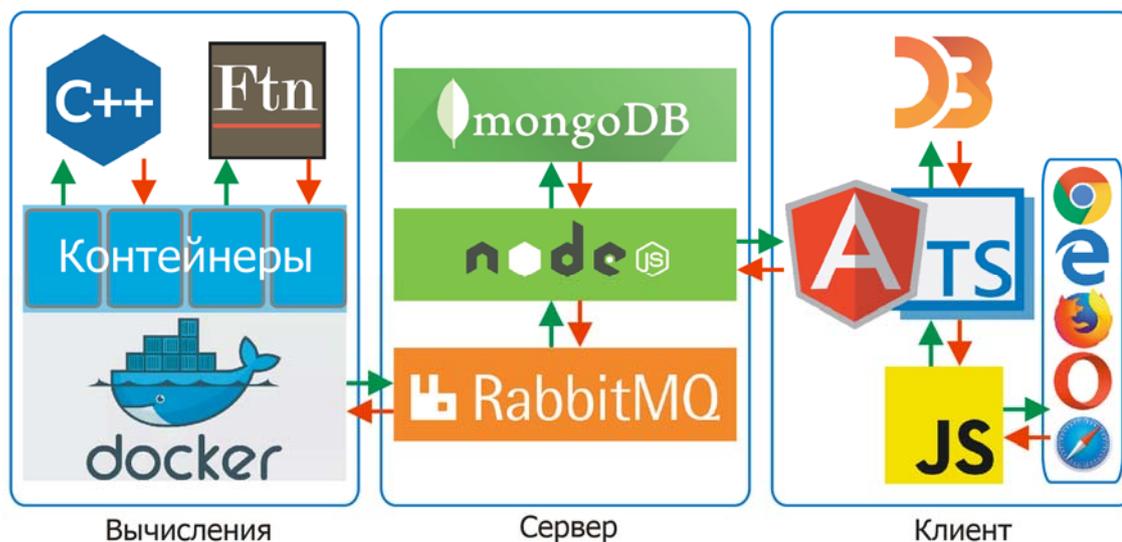


Рис. 2. Схема архитектуры разработанного приложения для геонавигации

Fig. 2. Architecture Scheme of the Developed Geosteering Application

Ниже приведен полный стек технологий, использованный при разработке ПО:

- TypeScript¹ и JavaScript² – языки программирования (ЯП), используемые для создания графического интерфейса, отображаемого в браузерах;
- Angular³ – платформа для сборки веб-приложений из созданных компонентов и используемых библиотек;
- D3⁴ – библиотека для визуализации в формате SVG сложных элементов: траектории скважины, моделей геологических пластов, каротажных данных;
- NodeJS⁵ – платформа, позволяющая разрабатывать серверную часть на языке JavaScript;
- MongoDB⁶ – нереляционная база данных для хранения учетных записей и пользовательских данных;
- RabbitMQ⁷ – брокер сообщений для взаимодействия сервера с вычислительными модулями;

¹ TypeScript documentation / Microsoft. URL: <https://www.typescriptlang.org/docs>

² JavaScript documentation / Mozilla foundation. URL: <https://developer.mozilla.org/bm/docs/Web/JavaScript>

³ Angular documentation / Google. URL: <https://angular.io/docs>

⁴ D3.js documentation / Bostock M., Davies J., Heer J., Ogievetsky V. URL: <https://github.com/d3/d3/wiki>

⁵ Node.js documentation / Various. URL: <https://nodejs.org/en/docs>

⁶ MongoDB documentation / MongoDB Inc. URL: <https://docs.mongodb.com/manual>

- Docker⁸ – платформа для создания кроссплатформенных вычислений;
- C++⁹, Fortran¹⁰ – ЯП, позволяющие разрабатывать высокопроизводительные вычислительные алгоритмы.

Клиент

Интерфейс, с которым работает пользователь, отображается в веб-браузере. За счет этого достигается кроссплатформенность клиентской части, поскольку веб-браузеры есть практически во всех операционных системах. Движок JavaScript V8¹¹, считывающий и исполняющий исходный код на JavaScript, а также другие аналоги встроены в большинство браузеров. Однако не все браузеры в одинаковой степени хорошо поддерживают новые стандарты ЯП. Поэтому для полной совместимости необходимо транслировать код в более старые версии JavaScript. Для этого в проекте используется язык TypeScript. Кроме того, статическая типизация, которая вводится TypeScript, фиксирует несоответствия типов на этапе трансляции в необходимый стандарт JavaScript до непосредственного выполнения программы. Это облегчает читаемость кода, а также помогает осуществлять поиск ошибок на этапе разработки и компиляции.

Для упрощения и ускорения разработки существуют различные фреймворки, определяющие структуру веб-приложения. Одним из наиболее известных является активно развивающийся Angular. Его отличительная особенность состоит в том, что он предназначен для разработки приложений именно на TypeScript. Для создания интуитивно понятного интерфейса используется библиотека Angular Material¹², являющаяся адаптацией Google Material Design¹³. Этот пакет упрощает создание таких компонентов взаимодействия с интерфейсом, как кнопки, переключатели, вкладки и пр. В вышеописанной библиотеке реализованы только простые компоненты, поэтому не все элементы интерфейса могут быть отображены с ее помощью. Более сложные элементы, такие как каротажные диаграммы и графические изображения моделей, визуализируются с помощью библиотеки D3. Она отличается высокой производительностью, гибкостью и возможностью работы с масштабируемой векторной графикой.

Сервер

Серверная часть принимает запросы пользователей, выполняет их обработку, взаимодействует с вычислительной частью и отправляет результаты обратно клиенту. Для обеспечения доступности (возможности всегда отвечать на запросы пользователей) сам сервер не выполняет сложных вычислений. При получении запроса на проведение расчетов происходит генерация входных данных для вычислительных алгоритмов. Они добавляются в очередь сообщений с заданной меткой, затем выполняются расчеты, и результаты возвращаются обратно в очередь сообщений. Последние изымаются и обрабатываются, а пользователю отправляются данные для отображения.

Для разработки серверной части выбрана программная платформа NodeJS, позволяющая разрабатывать серверную и клиентскую части на одном ЯП. NodeJS основана на движке V8, транслирующем JavaScript в машинный код. Поскольку для введения типизации используется TypeScript, перед запуском код транслируется в JavaScript, используя компилятор TypeScript. Для хранения пользовательских учетных записей и геофизических данных используется NoSQL база данных MongoDB. MongoDB предназначена для хранения данных без задания схем. Это позволяет проще менять формат хранимой информации, в сравнении

⁷ RabbitMQ documentation / Pivotal Software. URL: <https://www.rabbitmq.com/documentation.html>

⁸ Docker documentation / Docker Inc. URL: <https://docs.docker.com>

⁹ C++ documentation / Cppreference.com. URL: <https://cppreference.com>

¹⁰ Fortran documentation / O. Certic. URL: <https://www.fortran90.org>

¹¹ V8 documentation / V8 project. URL: <https://v8.dev/docs>

¹² Angular Material documentation / Google. URL: <https://material.angular.io>

¹³ Material Design documentation / Google. URL: <https://material.io>

с SQL решениями. Для взаимодействия с вычислительной серверная часть использует очередь сообщений RabbitMQ. Это делает их слабо зависимыми друг от друга, а также решает задачу сохранения еще не выполненных задач.

Вычислительные модули и организация взаимодействия с ними

Алгоритмы численной инверсии, лежащие в основе одного из методов геонавигации, требуют значительных вычислительных мощностей, особенно для трехмерных моделей среды. Для повышения быстродействия используются распределенные масштабируемые и высокопроизводительные параллельные вычисления. Численная инверсия данных ЭМК выполняется последовательно в заданных интервалах скважины (инверсионных окнах). Это делает возможным организацию параллельных вычислений на графических ускорителях и сопроцессорах [6].

Высокопроизводительные вычисления требуют распараллеливания кода на низкоуровневых ЯП (C++ / Fortran). Использование таких языков требует предварительной компиляции кода со всеми необходимыми зависимостями под каждый вычислительный ресурс. Для проверки корректности работы алгоритма его необходимо тестировать на каждой машине в отдельности. Альтернативным вариантом является запуск программы с помощью платформы Docker. Достаточно поместить вычислительную программу и все ее зависимости в Docker-контейнер вместе со скриптом, передающим данные из очереди сообщений в программу и возвращающим результаты. Такой контейнер работает быстро и надежно вне зависимости от вычислительного окружения, как на Linux, так и на Windows.

Методы геонавигации

Разработанное ПО позволяет проводить оперативную интерпретацию данных каротажа в процессе бурения, полученных комплексом «ЛУЧ». За проведение геонавигации отвечают модули «Корреляция» и «Инверсия», использующие методы корреляции каротажных данных и численной инверсии соответственно.

Корреляция каротажных данных

Большая часть существующих программных решений для геонавигации основана на идейно близких методах, имеющих в литературе различные названия: «модельно-стратиграфический», «сравнения каротажных данных», «матчинг», «model-compare-update» [7; 8]. Их суть состоит в сопоставлении данных, измеряемых приборами в процессе бурения, и синтетических данных, полученных в моделях среды, построенных по опорным скважинам. По этим априорным данным создается двумерная модель части разреза, в которой предполагается проводить горизонтальный участок скважины. Синтетические диаграммы каротажа вычисляются в такой модели на проекции запланированной траектории горизонтальной скважины. Начальная модель корректируется так, чтобы добиться совпадения между синтетическими и практическими данными.

В разработанном ПО каротажные данные пилотной субвертикальной скважины, приведенные к истинной вертикальной глубине, распространяются по латерали, тем самым формируя стартовую двумерную модель геологической среды. Синтетические каротажные диаграммы в горизонтальной скважине, как правило, получают путем линейной интерполяции данных модели среды в точках с соответствующими вертикальными глубинами. Однако линейная интерполяция не учитывает влияния прилегающих пластов (вмещающей среды) на сигналы ЭМК, которое обусловлено большой глубиной метода. В связи с этим синтетические сигналы ВИКПБ рассчитываются с помощью специально разработанных алгоритмов численного моделирования [9].

Геонавигатор работает с данными, привязанными к глубине по стволу скважины, а также с их проекциями на вертикальную и горизонтальную оси. Стартовая модель разбивается на блоки по горизонтали, при этом для каждого из блоков задается свой угол наклона

(рис. 3). При наличии предпосылок в модель могут вводиться разломы, т. е. линии, на которых происходит скачкообразное изменение параметров модели. После каждого изменения параметров модели или траектории скважины синтетические каротажные диаграммы рассчитываются заново, и процесс подбора повторяется. Достоверность модели, полученной в результате таких операций, достигается за счет наилучшего совпадения синтетических и практических данных каротажа в процессе бурения. Совпадение признается достаточным, если различие между практическими и синтетическими данными в среднем не превышает погрешности измерений.

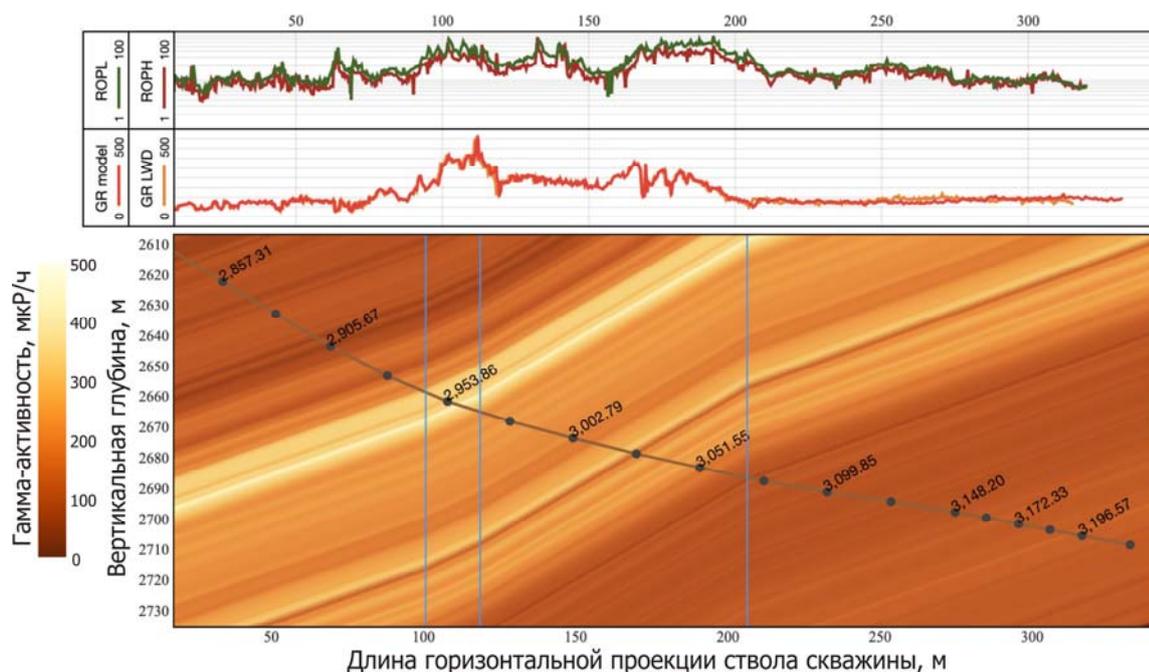


Рис. 3. Интерфейс модуля для проведения геонавигации методом корреляции каротажных данных

Fig. 3. Interface of the Module for Geosteering Using Logging Data Correlation Method

Однако метод геонавигации на основе корреляции каротажных данных обладает существенным недостатком. Поскольку датчики каротажных приборов располагаются на некотором удалении от бурового долота, пересечение геологической границы определяется лишь постфактум. Траекторию скважины корректируют, чтобы вернуться в пласт, а не просто удержаться в нем. Такие способы геонавигации принято называть «реактивными». Незапланированные пересечения геологических границ зачастую приводят к нежелательным последствиям. В частности, пересечение водонефтяного контакта может привести к поступлению воды в скважину, а вскрытие газовой шапки – и вовсе к аварийной ситуации.

Численная инверсия данных ЭМК

В противоположность реактивным выделяют «проактивные» методы геонавигации, когда траекторию скважины корректируют до того, как произойдет нежелательное пересечение геологической границы буровым долотом [10–13]. Радиальная глубинность исследования ВИКПБ может достигать 3 м [14; 15]. Это позволяет проводить проактивную геонавигацию и картировать геоэлектрические границы коллектора в процессе бурения, используя алгоритмы численной инверсии сигналов прибора.

Численная инверсия данных ЭМК представляет собой целенаправленный подбор параметров геоэлектрической модели путем минимизации функционала невязки измеренных и синтетических данных [16–18]. В разработанном ПО моделирование синтетических данных проводится путем численного решения прямой задачи ЭМК в классе слоисто-однородных моделей методом разделения переменных. Минимизация функционала невязки выполняется с использованием алгоритма на основе метода деформируемых многогранников.

Разработанное ПО предоставляет пользователю гибкий интерфейс для выполнения по-онокной инверсии сигналов ВИКПБ в автоматическом и ручном режимах (рис. 4). Геонавигатор загружает входные данные, выбирает интервал для инверсии, устанавливает ширину и перекрытие окон. Стартовая модель для инверсии подбирается, исходя из априорных данных, либо импортируется из модуля геонавигации методом корреляции каротажных данных. Алгоритм численной инверсии позволяет использовать произвольные комбинации сигналов из полного набора измерений ВИКПБ. Благодаря этому при ограниченной скорости передачи данных в реальном времени для инверсии можно задействовать только часть измеряемых сигналов. Уточнить параметры модели можно, получив всю информацию из памяти прибора, после его подъема из скважины на поверхность.

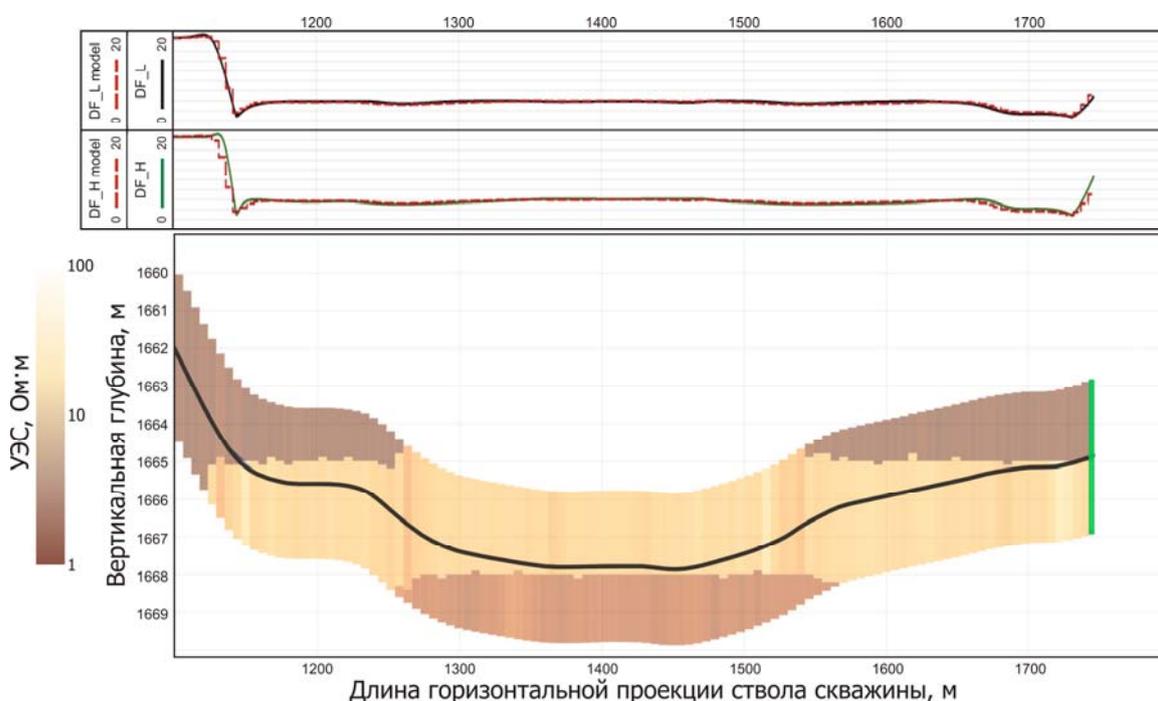


Рис. 4. Интерфейс модуля для проведения геонавигации методом инверсии данных ЭМК

Fig. 4. Interface of the Module for Geosteering Using Electromagnetic Logging Data Inversion Method

Применение метода геонавигации с использованием численной инверсии данных ВИКПБ позволяет картировать границы пласта-коллектора и заблаговременно корректировать траекторию скважины в процессе бурения, не пересекая сами границы. Такой подход значительно повышает эффективную длину ствола скважины, т. е. расстояние, пройденное скважиной внутри продуктивного пласта. Помимо этого, инверсия данных ЭМК позволяет изучать электрофизические характеристики отложений, не вскрытых скважиной и располагающихся ниже по разрезу.

Методы геонавигации скважин развиваются в связи с возникновением практической потребности нефтегазовой отрасли в высокоточной проводке скважин с горизонтальным завершением во все более сложных геологических условиях. При этом ключевую роль играет развитие аппаратной базы наклонно направленного бурения и каротажа в процессе бурения. Дальнейшее развитие методического аспекта ПО связано с внедрением модуля интерпретации скважинных имиджей УЭС, предоставляемых азимутальным прибором БКС [19–21].

Заключение

Разработано ПО для геологического сопровождения бурения с использованием данных аппаратного комплекса для каротажа в процессе бурения «ЛУЧ». Алгоритмическая составляющая ПО реализует два метода геонавигации. Первый использует традиционный подход корреляции каротажных данных, второй – численную инверсию данных ЭМК. Использование этих методов в комплексе позволяет повысить достоверность результатов интерпретации данных каротажа в процессе бурения, что приводит к более эффективной проводке скважины. Предложен облачный сервис для обработки и интерпретации геофизических данных с использованием кроссплатформенных распределенных масштабируемых вычислений. Использование современных вычислительных технологий позволяет значительно ускорить решение задачи численной инверсии, что необходимо для оперативной геонавигации.

Список литературы / References

1. **Аксельрод С. М.** Методы опережающей навигации при бурении горизонтальных скважин (по материалам зарубежной литературы) // Каротажник. 2012. № 219. С. 87–122.
Akselrod S. M. Metody operezhajushhej navigacii pri burenii gorizontol'nyh skvazhin (po materialam za-rubezhnoj literatury) [Advance navigation methods for drilling horizontal wells (based on materials from foreign literature)]. *Karotazhnik*, 2012, no. 219, p. 87–122. (in Russ.)
2. **Глинских В. Н., Никитенко М. Н., Даниловский К. Н., Еремин В. Н., Москаев И. А.** Телеметрические системы каротажа: программно-методическое обеспечение в процессе бурения наклонно-горизонтальных скважин // Neftegaz.RU. 2017. № 10. С. 42–49.
Glinskikh V. N., Nikitenko M. N., Danilovskiy K. N., Eremin V. N., Moskaev I. A. Telemetricheskie sistemy karotazha: programmno-metodicheskoe obespechenie v processe burenija naklonnogo-gorizontol'nyh skvazhin [Telemetry logging systems: software and methodological support in the process of drilling inclined-horizontal wells]. *Neftegaz.RU*, 2017, no. 10, p. 42–49. (in Russ.)
3. **Глинских В. Н., Никитенко М. Н., Эпов М. И., Ерёмин В. Н.** Алгоритмы и программное обеспечение для геонавигации по данным электромагнитного каротажа в процессе бурения // II Науч.-практ. конф. «Горизонтальные скважины 2017: Проблемы и перспективы»: расширенные тезисы докладов, 2017. 3G05. DOI 10.3997/2214-4609.201700431
Glinskikh V. N., Nikitenko M. N., Epov M. I., Eremin V. N. Algorithms and Software for Geosteering According to LWD EM Data. In: Horizontal Wells 2017 Challenges and Opportunities: extended abstracts, 2017, pap. 3G05. (in Russ.) DOI 10.3997/2214-4609.201700431
4. **Ерёмин В. Н., Волканин Ю. М., Тарасов А. В.** Аппаратурно-методическое обеспечение электромагнитного каротажа в процессе бурения // Каротажник. 2013. № 226. С. 62–69.
Eremin V. N., Volkanin Yu. M., Tarasov A. V. Apparaturno-metodicheskoe obespechenie elektromagnitnogo karotazha v protsesse bureniya [Instrumental and methodological support of electromagnetic logging while drilling]. *Karotazhnik*, 2013, no. 226, p. 62–69 (in Russ.)

5. **Каюров К. Н., Ерёмин В. Н., Эпов М. И., Глинских В. Н., Сухорукова, К. В., Никитенко, М. Н.** Аппаратура и интерпретационная база электромагнитного каротажа в процессе бурения // Нефтяное хозяйство. 2014. № 12. С. 112–115.
Kayurov K. N., Eremin V. N., Eпов M. I., Glinskikh V. N., Sukhorukova K. V., Nikitenko M. N. Electromagnetic-logging-while-drilling equipment and numerical inversion software. *Oil Industry*, 2014, no. 12, p. 112–115. (in Russ.)
6. **Дудаев А. Р., Соболев А. Ю., Глинских В. Н.** Программное обеспечение с облачными вычислениями для электромагнитного каротажа в процессе бурения нефтегазовых скважин // XIX Науч.-практ. конф. «Геомодель 2017 – Вопросы геологоразведки и разработки месторождений нефти и газа»: расширенные тезисы докладов, 2017. CD-ROM 43814.
Dudaev A. R., Sobolev A. Yu., Glinskikh V. N. Development of cloud computing software for LWD measurements and geosteering. In: Extended Abstracts of the 19th Science and Applied Research Conference on Oil and Gas Geological Exploration and Development “Geomodel 2017”, 2017, CD-ROM 43814 (in Russ.)
7. **Salehi M., Abadie J., Asif S., Ito K., Maggs D., Morriss C., Ortenzi L., Rasmus J., Griffiths R.** Rapid and practical characterization of near-wellbore layer structure and properties in high-angle and horizontal wells. In: Transactions of the SPWLA 56th Annual Logging Symposium, 2015, pap. DDDD.
8. **Polyakov V., Omeragic D., Shetty S., Brot B., Habashy T., Mahesh A., Friedel T., Vik T., Flugsrud T.** 3D Reservoir Characterization Workflow Integrating High Angle and Horizontal Well Log Interpretation with Geological Models. In: Extended abstracts of the International Petroleum Technology Conference, 2013, pap. 16828. DOI 10.2523/iptc-16828-abstract
9. **Nikitenko M. N., Eпов M. I., Glinskikh V. N., Suhorukova K. V., Eremin V. N.** Development of LWD High-frequency Resistivity Tool. In: Extended abstracts of the 6th Saint Petersburg International Conference & Exhibition «Geosciences – Investing in the Future», 2014, pap. ThD05. DOI 10.3997/2214-4609.20140177
10. **Rabinovich M., D’Arcy B., Leslie R.** Quantifying VOI in geosteering: a North Sea case study. In: Transactions of the SPWLA 59th Annual Logging Symposium, 2018, pap. P.
11. **Larsen D., Antonov Y., Luxey P., Skillings J., Skaug M., Wagner V.** Navigating the horizontal section in a heterogeneous formation while using Extra Deep Azimuthal Resistivity for optimizing the wellbore placement within a narrow TVD window. In: Transactions of the SPWLA 57th Annual Logging Symposium, 2016, pap. III.
12. **Sviridov M., Mosin A., Antonov Y., Nikitenko M., Martakov S., Rabinovich M.** New software for processing of LWD extra-deep resistivity and azimuthal resistivity data. *SPE Reservoir Evaluation and Engineering*, 2014, vol. 17, no. 02, p. 109–127. DOI 10.2118/160257-pa
13. **Tilsley-Baker R., Antonov Y., Martakov S., Maurer H., Mosin A., Sviridov M., Klein K., Iversen M., Barbosa J., Carneiro G.** Extra-Deep Resistivity Experience in Brazil Geosteering Operations. In: Extended abstracts of the SPE Annual Technical Conference and Exhibition, 2013, pap. 166309. DOI 10.2118/166309-ms
14. **Эпов М. И., Никитенко М. Н., Глинских В. Н., Сухорукова К. В.** Численное моделирование и анализ сигналов электромагнитного каротажа в процессе бурения // Каротажник. 2014. № 245. С. 29–42.
Eпов M. I., Nikitenko M. N., Glinskikh V. N., Sukhorukova K. V. Chislennoe modelirovanie i analiz signalov elektromagnitnogo karotazha v protsesse bureniya [Numerical modeling and analysis of electromagnetic logging while drilling signals]. *Karotazhnik*, 2014, no. 245, p. 29–42. (in Russ.)
15. **Эпов М. И., Никитенко М. Н., Сухорукова К. В., Глинских В. Н.** Исследование возможностей электрического и электромагнитного каротажа в электрически макроанизотропных породах

тропных пластах, вскрытых наклонно-горизонтальными скважинами // Каротажник. 2016. № 260. С. 64–79.

Епов М. И., Nikitenko M. N., Sukhorukova K. V., Glinskikh V. N. Issledovaniye vozmozhnostey elektricheskogo i elektromagnitnogo karotazha v elektricheski makroanizotropnykh plastakh, vskrytykh naklonno-gorizont'al'nymi skvazhinami [Investigation of the possibilities of electrical and electromagnetic logging in electrically macroanisotropic formations intersected by inclined-horizontal wells]. *Karotazhnik*, 2016, no. 260, p. 64–79. (in Russ.)

16. **Никитенко М. Н., Сухорукова К. В., Глинских В. Н., Ерёмин В. Н.** Оперативная обработка и интерпретация данных электромагнитного каротажа на разных этапах строительства нефтегазовой скважины // IV Междунар. науч.-практ. конф. «Тюмень-2015: Глубокие горизонты науки и недр»: расширенные тезисы докладов. 2015. P06. DOI 10.3997/2214-4609.201412045

Nikitenko M. N., Sukhorukova K. V., Glinskikh V. N., Eremin V. N. Operativnaya obrabotka i interpretatsiya dannykh elektromagnitnogo karotazha na raznykh etapakh stroitel'stva neftegazovoy skvazhiny [Fast processing and interpretation of electromagnetic logging data at different oil and gas well construction stages]. In: Tyumen 2015 – Deep Subsoil and Science Horizons, 2015, pap. P06 (in Russ.) DOI 10.3997/2214-4609.201412045

17. **Эпов М. И., Глинских В. Н., Сухорукова К. В., Никитенко М. Н., Ерёмин В. Н.** Численное моделирование и инверсия данных электромагнитного каротажа в процессе бурения и шаблонирования нефтегазовых скважин // Геология и геофизика. 2015. № 56 (8). С. 1520–1529. DOI 10.15372/gig20150809

Епов М. И., Glinskikh V. N., Sukhorukova K. V., Nikitenko M. N., Eremin V. N. Forward modeling and inversion of LWD induction data. *Russian Geology and Geophysics*, 2015, vol. 56, no. 8, p. 1194–1200. (in Russ.) DOI 10.15372/gig20150809

18. **Эпов М. И., Никитенко М. Н., Сухорукова К. В., Глинских В. Н., Ерёмин В. Н., Горбатенко А. А., Павлова М. А.** Сигналы электромагнитного каротажа в процессе бурения и их численная инверсия // Геология и минерально-сырьевые ресурсы Сибири. 2014. № 3 (19). С. 49–55.

Епов М. И., Nikitenko M. N., Sukhorukova K. V., Glinskikh V. N., Eremin V. N., Gorbatenko A. A., Pavlova M. A. Electromagnetic LWD signals and their numerical inversion. *Geology and mineral resources of Siberia*, 2014, no. 3 (19), p. 49–55. (in Russ.)

19. **Danilovskiy K., Glinskikh V., Nechaev O.** 3D modelling of the new resistivity microimaging tool signals for logging while drilling. In: Extended Abstracts of the 80th EAGE Conference & Exhibition, 2018, pap. Tu SP2 01. DOI 10.3997/2214-4609.201801679

20. **Глинских В. Н., Даниловский К. Н., Нечаев О. В.** Трехмерное численное моделирование сигналов азимутального микробокового каротажа в процессе бурения // Геология, геофизика и разработка нефтяных и газовых месторождений. 2018. № 10. С. 32–39. DOI 10.30713/2413-5011-2018-10-32-39

Glinskikh V. N., Danilovsky K. N., Nechaev O. V. 3D numerical simulation of the azimuthal microlateral LWD signals. *Geology, Geophysics and Development of Oil and Gas Fields*, 2018, no. 10, p. 32–39. (in Russ.) DOI 10.30713/2413-5011-2018-10-32-39

21. **Даниловский К. Н., Нечаев О. В.** Моделирование сигналов электрического LWD-микроимиджера применительно к задаче геонавигации горизонтальных скважин // Материалы V Всерос. молодежной науч.-практ. конф. «Науки о Земле. Современное состояние», 2018. С. 23–25.

Danilovskiy K. N., Nechaev O. V. Modelirovanie signalov elektricheskogo LWD-mikroimidzhera primenitel'no k zadache geonavigatsii gorizont'al'nykh skvazhin [Simulation of LWD resistivity microimaging tool signals applied to the horizontal wells geosteering]. In: Proceed-

ings of the 5th All-Russian Youth Scientific and Practical Conference "Earth Sciences. Modern state", 2018, p. 23–25. (in Russ.)

Материал поступил в редколлегию
Received
21.03.2019

Сведения об авторах / Information about the Authors

Даниловский Кирилл Николаевич, аспирант, младший научный сотрудник, Институт нефтегазовой геологии и геофизики им. А. А. Трофимука СО РАН (пр. Академика Коптюга, 3, Новосибирск, 630090, Россия)

Kirill N. Danilovskiy, PhD Student, Junior Researcher, Trofimuk Institute of Petroleum Geology and Geophysics SB RAS (3 Koptuyug Ave., Novosibirsk, 630090, Russian Federation)

DanilovskiiKN@ipgg.sbras.ru
ResearcherID: O-1475-2017

Дудаев Александр Русланович, лаборант, Институт нефтегазовой геологии и геофизики им. А. А. Трофимука СО РАН (пр. Академика Коптюга, 3, Новосибирск, 630090, Россия), магистрант факультета информационных технологий, Новосибирский государственный университет (ул. Пирогова, 1, Новосибирск, 630090, Россия)

Alexander R. Dudaev, Laboratory Assistant, Trofimuk Institute of Petroleum Geology and Geophysics SB RAS (3 Koptuyug Ave., Novosibirsk, 630090, Russian Federation), Master's Student, Faculty of Information Technologies, Novosibirsk State University (1 Pirogov Str., Novosibirsk, 630090, Russian Federation)

DudaevAR@ipgg.sbras.ru
ResearcherID: D-4181-2019

Глинских Вячеслав Николаевич, доктор физико-математических наук, профессор РАН, заведующий лабораторией, Институт нефтегазовой геологии и геофизики им. А. А. Трофимука СО РАН (пр. Академика Коптюга, 3, Новосибирск, 630090, Россия), доцент кафедры геофизики геолого-геофизического факультета, Новосибирский государственный университет (ул. Пирогова, 1, Новосибирск, 630090, Россия), профессор кафедры интеллектуальных геофизических систем ФТФ, Новосибирский государственный технический университет (пр. К. Маркса, 20, Новосибирск, 630073, Россия)

Viacheslav N. Glinskikh, Doctor in Physics and Mathematics, Professor of the Academy of Sciences, Head of Laboratory, Trofimuk Institute of Petroleum Geology and Geophysics SB RAS (3 Koptuyug Ave., Novosibirsk, 630090, Russian Federation), Associate Professor of Chair of Geophysics at the Department of Geology and Geophysics, Novosibirsk State University (1 Pirogov Str., Novosibirsk, 630090, Russian Federation), Professor, Novosibirsk State Technical University (20 Karl Marx Ave., Novosibirsk, 630073, Russian Federation)

GlinskikhVN@ipgg.sbras.ru
ResearcherID: C-6551-2018

Никитенко Марина Николаевна, кандидат технических наук, старший научный сотрудник, Институт нефтегазовой геологии и геофизики им. А. А. Трофимука СО РАН (пр. Академика Коптюга, 3, Новосибирск, 630090, Россия)

Marina N. Nikitenko, PhD, Senior Researcher, Trofimuk Institute of Petroleum Geology and Geophysics SB RAS (3 Koptug Ave., Novosibirsk, 630090, Russian Federation)

NikitenkoMN@ipgg.sbras.ru
ResearcherID: J-6876-2018

Москаев Илья Алексеевич, инженер, Институт нефтегазовой геологии и геофизики им. А. А. Трофимука СО РАН (пр. Академика Коптюга, 3, Новосибирск, 630090, Россия), магистрант геолого-геофизического факультета, Новосибирский государственный университет (ул. Пирогова, 1, Новосибирск, 630090, Россия)

Илья А. Moskaev, Engineer, Trofimuk Institute of Petroleum Geology and Geophysics SB RAS (3 Koptug Ave., Novosibirsk, 630090, Russian Federation), Master's Student, Department of Geology and Geophysics, Novosibirsk State University (1 Pirogov Str., Novosibirsk, 630090, Russian Federation)

MoskaevIA@ipgg.sbras.ru
ResearcherID: D-6392-2019

Моделирование структурной организации единого информационного пространства предприятия – разработчика микроэлектронных систем

О. В. Дрозд, С. В. Ченцов, Д. В. Капулин

*Сибирский федеральный университет,
Институт космических и информационных технологий
Красноярск, Россия*

Аннотация

При формировании единого информационного пространства (ЕИП) широко используется моделирование как объектов информационного обмена, так и инфраструктуры ЕИП. Несмотря на большое разнообразие подходов к решению данной задачи, в частности с использованием метода матрицы структуры проекта, предлагаемые решения не обеспечивают непосредственную связь между элементами описываемой структуры и используемыми аппаратными и программными средствами. Также существующие подходы к моделированию информационной структуры не учитывают возможные внешние и внутренние ограничения, накладываемые на размещение компонентов ЕИП в пространстве предприятия. В работе предлагается подход к формализации структуры ЕИП предприятия – разработчика микроэлектронных систем, призванный устранить данные недостатки.

В статье рассмотрена обобщенная модель структурной организации ЕИП по параллельному методу интеграции средств поддержки жизненного цикла изделия. Модель включает в себя описание элементов структуры информационного пространства, каналов связи, их статических и динамических характеристик. Предложенная модель также учитывает ограничения, накладываемые на формирование структуры ЕИП.

Имитационное моделирование структурной организации ЕИП, представленной в виде совокупности многоканальных систем массового обслуживания, позволило сделать выводы о пропускной способности и надежности рассматриваемых вариантов организации ЕИП с учетом предлагаемых к использованию аппаратных (серверное решение) и программных (система управления базами данных) средств.

Ключевые слова

конструкторское проектирование изделия, единое информационное пространство, моделирование структурной организации, средства информационной поддержки

Благодарности

Работа выполнена при финансовой поддержке Министерства науки и высшего образования Российской Федерации (договор № 03.G25.31.0279).

Для цитирования

Дрозд О. В., Ченцов С. В., Капулин Д. В. Моделирование структурной организации единого информационного пространства предприятия – разработчика микроэлектронных систем // Вестник НГУ. Серия: Информационные технологии. 2019. Т. 17, № 2. С. 18–29. DOI 10.25205/1818-7900-2019-17-2-18-29

Modeling the Structural Organization of Unified Information Space for the Microelectronics Design Center

O. V. Drozd, S. V. Chentsov, D. V. Kapulin

*Siberian Federal University,
School of Space and Information Technologies
Krasnoyarsk, Russian Federation*

Abstract

During the process of forming a unified information space (UIS) is widely used modeling of information exchange facilities and UIS infrastructure. Despite the large variety of approaches to modeling a UIS, in particular using the de-

sign structure matrix method, the proposed solutions do not provide a direct link between elements of the described structure and hardware and software equipment. Also, existing approaches for modeling the UIS do not take into account possible external and internal constraints imposed on the placement of UIS components in the enterprise space.

The paper proposes an approach to formalizing the UIS structure of the microelectronics design center, which is intended to eliminate these shortcomings. The structure of the UIS according to a parallel organization model includes following main components: a product data management system, participants of the microelectronic system design process, data-processing hardware and software. The structure of the UIS as a dynamic system can be reduced to a generalized model that includes the set of subscribers – participants of the design process, primary and secondary data-processing tools and objects of information exchange. The mathematical description is made using elements of theory of sets, matrices and queuing systems. The proposed model takes into account the constraints that affect a form of implementation of the UIS structure, as well as many characteristics of software and hardware.

Simulation modeling of the UIS structural organization, presented as a set of multichannel queuing systems, helped to draw conclusions about bandwidth and reliability of considered variants for the UIS organization based on proposed hardware (the server solution) and software (the database management system) tools that are being used.

Keywords

product design, unified information space, structural organization modeling, information support tools

Acknowledgements

This work was supported by the Ministry of Education and Science of the Russian Federation (agreement no. 03.G25.31.0279).

For citation

Drozd O. V., Chentsov S. V., Kapulin D. V. Modeling the Structural Organization of Unified Information Space for the Microelectronics Design Center. *Vestnik NSU. Series: Information Technologies*, 2019, vol. 17, no. 2, p. 18–29. (in Russ.) DOI 10.25205/1818-7900-2019-17-2-18-29

Введение

Реализация конструкторского проектирования (КП) микроэлектронных систем (МЭС) предполагает активное использование программных и аппаратных средств поддержки проектирования с последующей их интеграцией в единую среду информационной поддержки КП МЭС.

Существующие решения по информационной поддержке этапов жизненного цикла (ЖЦ) МЭС можно подразделить на средства интеграции по последовательной и параллельной модели. Решения первого типа предоставляются компаниями – разработчиками элементной базы и систем автоматизированного проектирования, ориентированных на приборостроительную отрасль [1], также они включают средства преобразования описания процесса функционирования МЭС на языке высокого уровня в описание на уровне регистровых передач [2–4]. Решения второго типа предполагают обмен информацией посредством системы управления данными об изделии, при этом формируется единое информационное пространство (ЕИП) предприятия, позволяющее централизованно управлять информацией об изделии [5, 6]. Подобные системы поставляются в составе средств автоматизации проектирования общего назначения, или как часть комплекта бизнес-приложений.

ЕИП, как и большинство типов информационных систем, создается прежде всего для удовлетворения потребностей специалистов – участников ЖЦ изделия в автоматизации их профессиональной деятельности и обеспечении информационного обмена. При этом необходимо отметить, что участники ЖЦ изделия могут быть распределены по территориально несвязанным друг с другом площадкам с формированием «виртуального предприятия» [7]. Вопрос информационной поддержки изделия связан не только со способами представления и управления конструктивными решениями, но и с формализацией и анализом структурной организации ЕИП с учетом ограничений на размещение ее компонентов.

При формировании ЕИП как распределенной информационной системы широко используется моделирование и объектов информационного обмена, и инфраструктуры ЕИП. Моделирование позволяет структурировать разрабатываемую систему, выделить основные объекты и субъекты, обнаружить закономерности и связи между элементами системы, а также сформулировать критерии оптимизации и существующие при этом ограничения [8–10].

Вне зависимости от способа интеграции отдельных компонентов моделирование структуры ЕИП предполагает:

- описание элементов структуры системы;
- описание статических и динамических характеристик элементов структуры и каналов связи;
- выделение множества свойств и характеристик как системы в целом, так и отдельных компонентов.

Несмотря на большое разнообразие подходов к решению данной задачи, в частности с использованием метода матрицы структуры проекта, предлагаемые решения не обеспечивают непосредственную связь между элементами описываемой структуры и используемыми аппаратными и программными средствами. Также существующие подходы к моделированию информационной структуры не учитывают возможные внешние и внутренние ограничения, накладываемые на размещение компонентов ЕИП в пространстве предприятия [11–13]. Предлагаемый ниже подход к формализации структуры ЕИП предприятия – разработчика МЭС призван устранить данные недостатки.

Структура ЕИП по параллельной модели организации

Рассмотрим структуру ЕИП организации – разработчика МЭС по параллельной модели интеграции (рис. 1). Объектом информационного обмена в ЕИП является электронный конструкторский документ (ЭКД).



Рис. 1. Структура ЕИП организации – разработчика МЭС по параллельной модели интеграции

Fig. 1. The Structure of the UIS of the Microelectronics Design Center for the Parallel Integration Model

ЕИП (1) включает следующие основные составляющие:

- *систему управления данными (СУД) (2)* – информационная система, представляющая унифицированный обменный формат ЭКД, реализующая модель представления конструкторских данных;

торских данных об изделии, а также функций обработки данных и обеспечения взаимодействия между участниками ЖЦ;

- *участников* процесса конструкторского проектирования – поставщики и потребители информации;
- *аппаратные средства обработки данных* – стационарные и портативные рабочие станции, измерительное оборудование и т. д., способные взаимодействовать с ЕИП посредством сетевых интерфейсов;
- *программные средства обработки данных* – развернутые на базе аппаратных средств.

Основными источниками конструкторских данных в ЕИП являются системы автоматизированного проектирования и моделирования, а также интегрированные среды разработки встроенного программного обеспечения МЭС. Приведение разнородных документов от программных и аппаратных систем сторонних разработчиков к унифицированному формату осуществляется посредством программы-посредника.

Обобщенная модель структурной организации ЕИП

Процесс конструкторского проектирования изделия, равно как и любая другая стадия ЖЦ МЭС, может быть представлен в виде совокупности этапов, задач и процедур. Тип задачи или процедуры зависит от осуществляемой формы преобразования информации (создание, редактирование, актуализация, поиск и т. д.).

Для любого подмножества процедур и задач могут существовать соответствующие *средства поддержки*, выполняющие преобразование информации, тип которого зависит от поддерживаемой задачи или процедуры. Совокупность средств поддержки конструкторского проектирования, связанных множеством каналов связи, а также СУД образуют структуру ЕИП предприятия, в которой каждое из множества средств поддержки может быть представлено в виде набора программных и аппаратных средств обработки информации и абонентов (участников КП МЭС), обслуживающих эти средства.

Переход изделия на следующую задачу, этап КП или ЖЦ в целом приводит к изменению структуры ЕИП с появлением:

- новой формы реализации структуры ЕИП;
- новой структуры связей между элементами ЕИП;
- новых свойств структуры ЕИП в целом.

Таким образом, ЕИП это динамическая система, структурная организация которой может быть сведена к обобщенной модели, включающей в себя следующие элементы (рис. 2):

1) абоненты – участники конструкторского проектирования ($P = \{p_i\}$, $i = \overline{1, I}$ – множество номеров абонентов);

2) первичные средства обработки информации – аппаратное обеспечение ($B = \{b_j\}$, $j = \overline{1, J}$ – множество номеров первичных источников);

3) вторичные средства обработки информации – программное обеспечение ($D = \{d_b\}$, $b = \overline{1, B}$ – множество номеров вторичных источников);

4) первичные объекты информационного обмена, используемые первичными средствами обработки ($N_S = \{ns_a\}$, $a = \overline{1, A}$ – множество номеров объектов);

5) вторичные объекты информационного обмена, используемые вторичными средствами обработки ($A_S = \{as_q\}$, $q = \overline{1, Q}$ – множество номеров объектов).

Реализацию структуры информационного пространства определяет множество схмотехнических решений Δ_a :

$$\Delta_a = \left\{ a = \langle mm, mo, al, ap \rangle_{mm \in MM, mo \in MO, al \in AL, ap \in AP} \right\},$$

которое включает в себя: модель представления конструкторских данных (MM), функции обработки данных (MO), функции обеспечения информационного взаимодействия абонентов

(AL), программно-аппаратную реализацию серверной составляющей СУД (AP), включающую в себя сервер баз данных (СБД) и файловый сервер (ФС).

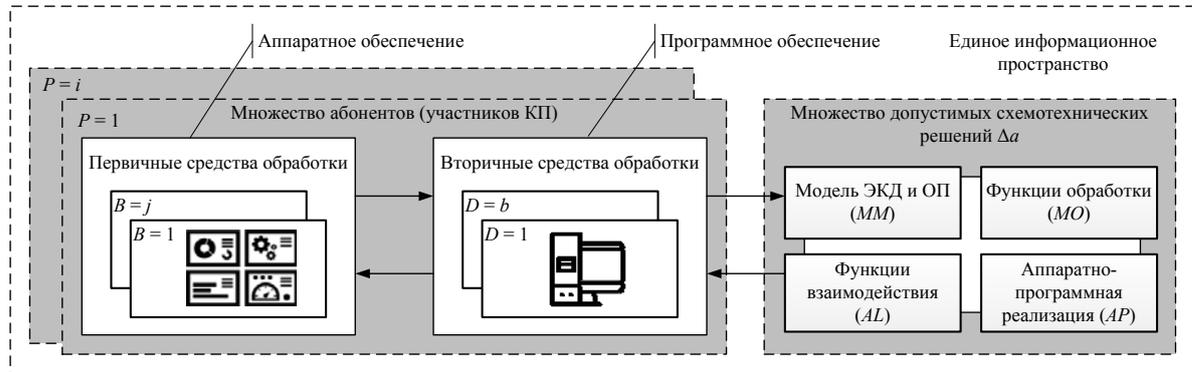


Рис. 2. Обобщенная модель структурной организации информационного пространства

Fig. 2. The Generalized Structural Model of the Information Space Organization

Для обработки информации в режиме мягкого реального времени (T) также вводится множество $W^{(a)}: N_S \times A_S \times T \rightarrow A_S^{(a)} \times N_S^{(a)}$, ограничивающее множество преобразований W на множестве решений Δ_a . Средняя допустимая длительность обработки входного запроса принята равной трем секундам, что соответствует установленным требованиям к информационным системам электронного документооборота¹.

На форму реализации структуры ЕИП и его характеристики влияют ограничения, накладываемые со стороны пользователя (V_C), размещением компонентов ЕИП и абонентов на предприятии (E_C), характеристиками аппаратного и программного обеспечения (D_C) и формой представления объекта информационного обмена (S_C).

Множество составляющих ЕИП образует пространство альтернативных вариантов (a) схмотехнического решения при заданных ограничениях (множество декартовых произведений) $F_{V_C, E_C, D_C, S_C}^{(a)}$. Цель формирования структуры ЕИП состоит в обеспечении информационного взаимодействия между участниками КП изделия. Задача заключается в поиске оптимального схмотехнического решения $\Delta_a^{\text{опт}}$ организации взаимодействия компонентов ЕИП с учетом заданных ограничений V_C, E_C, D_C, S_C при ограничениях на множество преобразований $W^{(a)}$:

$$F_{V_C, E_C, D_C, S_C}^{(a)} \subseteq \Delta_a^{\text{опт}} \left\{ \begin{array}{l} \langle p_i^{(a)}, b_j^{(a)}, d_r^{(a)}, ns_a^{(a)}, as_q^{(a)} \rangle \\ \Phi(a): F_{V_C}^{(a)} \cap F_{E_C}^{(a)} \cap F_{D_C}^{(a)} \cap F_{S_C}^{(a)} \\ W^{(a)}: N_S \times A_S \times T \rightarrow A_S^{(a)} N_S^{(a)} \end{array} \right.$$

Результатом поиска является конкретное схмотехническое решение, определяющее структуру и конфигурацию аппаратного и программного обеспечения ЕИП, необходимого для обеспечения информационной поддержки сопровождаемой стадии ЖЦ изделия.

Структурная организация ЕИП описывается множеством характеристик аппаратных и программных средств (табл. 1).

¹ Приказ Министерства связи и массовых коммуникаций РФ от 02.09.2011 № 221.

Множество характеристик структурной организации ЕИП

Таблица 1

Characteristics of the UIS Structural Organization

Table 1

Характеристика	Описание
Характеристики структуры ЕИП	
1. $A = \{a_n\}, n = \overline{1, N}$	множество абонентов – участников жизненного цикла (N)
2. $P = \{p_m\}, m = \overline{1, M}$	множество серверов баз данных (M)
3. $B = \{b_k\}, k = \overline{1, K}$	множество файловых серверов (K)
Характеристики участников информационного взаимодействия	
1. $R = \{r_d\}, d = \overline{1, D}$	множество приложений – источников запросов (D)
2. $S = \{s_q\}, q = \overline{1, Q}$	множество типов запросов
3. $T = \{T_1, T_2\}$	множество способов ввода-вывода информации (T_1 : клиент \rightarrow сервер, T_2 : сервер \rightarrow клиент)
Статические характеристики ЕИП	
1. C_v	скорость передачи данных по каналу связи, Мбит/с
2. C_d	постоянная задержка передачи данных по каналу связи, мс
3. θ_{sb}	постоянная задержка обработки данных в СБД, мс
4. θ_{sf}	постоянная задержка обработки данных в ФС, мс
Динамические характеристики ЕИП	
1. $\Lambda = \{\lambda_n\}$	множество интенсивностей запросов (n) на считывание данных, запрос/с
2. $M = \{\mu_n\}$	множество интенсивностей запросов (n) на запись данных, запрос/с

Статические характеристики ЕИП определяются параметрами используемых аппаратных и программных средств. Оценки параметров СП и СБД значения характеристик определяются исходя из параметров центрального процессора (число вычислительных ядер, максимальная тактовая частота и максимальный объем сегмента) и дисковой подсистемы (среднее время доступа к данным и потоковая скорость передачи данных). Характеристики канала связи зависят от полосы пропускания канала связи, мощности сигнала и определяются по теореме Шеннона – Хартли.

Множество характеристик структурной организации ЕИП можно представить в виде двух групп матриц распределения, описывающих межсоединения элементов пространства (Ξ, H, Γ) и информационное взаимодействие между ними (O, Y, Φ), также выделяется матрица связей между группами (Ψ):

- средств обработки информации верхнего уровня по серверам баз данных: $\Xi = \|\xi_{a_n p_m}\| = \|\xi_{nm}\|, n = \overline{1, N}, m = \overline{1, M}$;

- средств обработки информации верхнего уровня по файловым серверам: $H = \|\eta_{a_n b_k}\| = \|\eta_{nk}\|, n = \overline{1, N}, k = \overline{1, K}$;

- серверов баз данных по файловым серверам: $\Gamma = \|\gamma_{p_m b_k}\| = \|\gamma_{mk}\|, m = \overline{1, M}, k = \overline{1, K}$;

- средств обработки информации верхнего уровня по способам ввода-вывода запросов: $O = \left\| o_{r_d t, 2} \right\| = \left\| o_{d 1, 2} \right\|, d = \overline{1, D}, t = 1, 2;$
- типов запросов по способам ввода-вывода запросов: $Y = \left\| v_{s_q t, 2} \right\| = \left\| v_{q 1, 2} \right\|, q = \overline{1, Q}, t = 1, 2;$
- источников информационных сообщений по типам запросов: $\Phi = \left\| \phi_{r_d s_q} \right\| = \left\| \phi_{d q} \right\|, d = \overline{1, D}, q = \overline{1, Q};$
- матрица связей между группами (приложениями – источниками запросов и абонентами): $\Psi = \left\| \psi_{a_r r_d} \right\| = \left\| \psi_{d q} \right\|, a = \overline{1, A}, r = \overline{1, R}.$

Элементы матриц Ξ, H, Γ могут принимать значение 1 или 0: наличие или отсутствие связи между элементами (a, b, p) . Любой узел ЕИП характеризуется тройкой значений (ξ, η, γ) описывающей все множество его связей с другими компонентами.

Элементы матриц O, Y, Φ могут принимать непрерывные значения от 0 до 1, соответствующие вероятности того, что запросы, генерируемые данным источником, характеризуются определенным способом ввода-вывода и типом запроса. Источник запросов в системе характеризуется тройкой значений (o, v, ϕ) , которая описывает формат создаваемых им запросов в большинстве возможных ситуаций.

Графически взаимодействие между элементами матриц $(\Gamma, H, \Xi, O, Y, \Phi, \Psi)$ показано на рис. 3. Куб со сторонами (Γ, H, Ξ) представляет собой модель соединений компонентов ЕИП и соответствует сетевому уровню модели *OSI (L3)*. Куб со сторонами (O, Y, Φ) представляет собой модель информационного взаимодействия компонентов ЕИП и соответствует представительскому и прикладному уровням модели *OSI (L6 и L7)* соответственно).

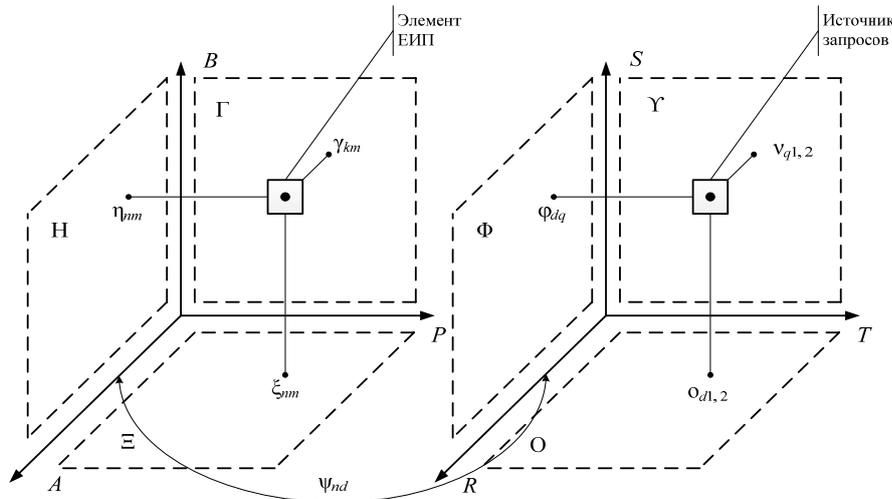


Рис. 3. Графическое представление матриц $(\Gamma, H, \Xi, O, Y, \Phi, \Psi)$

Fig. 3. Graphical representation of matrices $(\Gamma, H, \Xi, O, Y, \Phi, \Psi)$

Представленные матрицы распределения соответствуют матричному описанию структуры системы массового обслуживания (СМО). Так, для рассматриваемой клиент-серверной архитектуры модель структурной организации ЕИП (рис. 4) можно представить в виде совокупности СМО₁ (взаимодействие клиентской и серверной составляющих системы) и СМО₂ (взаимодействие серверов баз данных и файловых серверов).

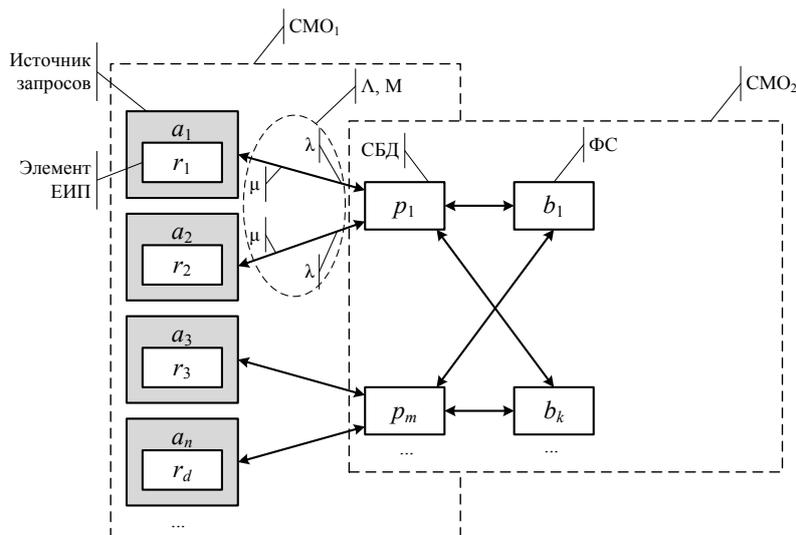


Рис. 4. Клиент-серверная организация ЕИП как совокупность многоканальных СМО

Fig. 4. Client-Server Organization of the UIS as a Combination of Multichannel QS

В рассматриваемой модели приняты следующие ограничения для представления СБД и ФС:

- СБД и ФС считаются независимыми;
- СБД и ФС обладают накопителями ограниченной емкости;
- входной поток запросов соответствует распределению Пуассона;
- поведение СМО во времени описывается марковским процессом с непрерывным временем и дискретным множеством состояний.

Математическое описание структурной организации ЕИП в виде СМО было выполнено в виде системы дифференциальных уравнений Колмогорова – Чепмена для вероятностей состояний [14]. Решение системы осуществлялось методом Рунге – Кутты с переменным шагом [15].

Многоканальная модель СМО позволяет учесть использование многопроцессорных и распределенных серверных решений и программного обеспечения с различными процедурами обработки информационных сообщений.

Имитационное моделирование структурной организации ЕИП

Кратко рассмотрим начальные условия и результаты моделирования структурной организации единого информационного пространства КП МЭС. Значения начальных условий при проведении моделирования зависят от существующих ограничений на пропускную способность оборудования и каналов связи (статические характеристики ЕИП) и состава участников ЕИП (характеристики структуры ЕИП). Для автоматизации процесса формирования номенклатуры средств реализации инфраструктуры ЕИП целесообразно использовать программные средства параллельного моделирования и параметрической оптимизации, такие как *MATLAB Parallel Computing Toolbox* и *Optimization Toolbox*.

Аппаратная реализация моделируемой структуры ЕИП включает в себя следующие ключевые компоненты:

- серверное решение (СБД и ФС): *DELL R630*;
- система управления базами данных: *Microsoft SQL Server 2012*;
- центральный процессор: *Intel Xeon E5-2699* (2.3 ГГц, 18 ядер, 36 потоков, количество – 2 шт.);

- оперативная память: *DDR4 RDIMM PC4-21300* (8 Гб, 2133 МГц, количество – 12 шт.);
- внешняя память: *Seagate ST1200MM0129* (1,2 Тб, количество – 1 шт.)
- линия связи: экранированная витая пара *Hyperline LSZH CMR C5e* (1943 м).

При проведении моделирования СМО были приняты следующие начальные условия:

- интенсивность обслуживания: 0,7;
- интенсивность входного потока заявок (λ): от 1,0 до 5,5 с шагом 0,5;
- обслуживание заявок в режиме разделения времени;
- интенсивность ухода заявок из очереди: 1,5;
- допустимое число запросов в системе: 10;
- число активных абонентов в единицу времени: от 2 до 5.

Данные начальные условия соответствуют организации – разработчику МЭС (дизайн-центру) с тремя основными структурными подразделениями: лабораториями системного проектирования, топологического проектирования и подготовки к производству с числом активных пользователей 60 человек и 45 обращениями к СУД от одного пользователя в час. Примерами подобной специализированной проектной организации являются:

- *Microelectronics Design Center* (Высшая техническая школа, Цюрих, Швейцария);
- *Yonga Technology Microelectronics R&D* (Технопарк Стамбула, Турция);
- *ITB Microelectronics Center* (Технологический институт, Бандунг, Индонезия);
- Дизайн-центр микроэлектроники (Институт точной механики и вычислительной техники РАН, Москва).

В ходе моделирования регистрировались следующие параметры СМО:

- вероятность отказа в обслуживании;
- относительная пропускная способность;
- абсолютная пропускная способность;
- вероятность наличия очереди;
- вероятность загрузки всех каналов обслуживания;
- среднее количество заявок;
- среднее время пребывания заявки;
- средняя длина очереди;
- среднее время пребывания заявки в очереди.

Пример результатов моделирования СМО, характеризующих общую пропускную способность и надежность рассматриваемой структуры (вероятность отказа в обслуживании (P_n), относительная пропускная способность (Q), абсолютная пропускная способность (A)), представлены в табл. 2.

Таблица 2

Параметры СМО, характеризующие структурную организацию ЕИП

Table 2

The Parameters of the QS, Describing the Structural Organization of the UIS

Эксперимент	СМО типа <i>M/M/2/10/RR</i>			СМО типа <i>M/M/2/10/RR</i>		
	P_n	P_n	P_n	P_n	Q	A
1	0,000	0,000	0,000	0,000	1,000	1,000
2	0,000	0,000	0,000	0,000	1,000	1,500
3	0,000	0,000	0,000	0,000	1,000	2,000
4	0,000	0,000	0,000	0,001	0,999	2,499
5	0,000	0,000	0,000	0,002	0,998	2,995
6	0,001	0,001	0,001	0,004	0,996	3,486
7	0,001	0,001	0,001	0,008	0,992	3,967
8	0,003	0,003	0,003	0,015	0,985	4,435
9	0,005	0,005	0,005	0,023	0,977	4,883
10	0,009	0,009	0,009	0,035	0,965	5,310

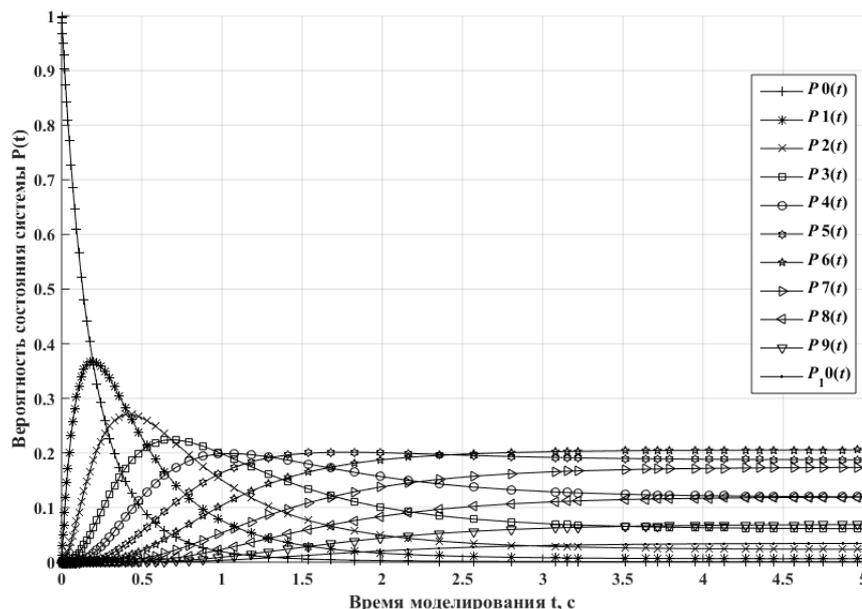


Рис. 5. Вероятности состояний ЕИП, представленной в виде системы массового обслуживания типа $M/M/2/10/RR$ при значении $\lambda = 5,5$

Fig. 5. The Probabilities of the UIS States Presented in the form of a $M/M/2/10/RR$ Queuing System with the Value $\lambda = 5.5$

Пример графика распределения вероятностей состояния СМО при заданном значении λ представлен на рис. 5. Моделирование показало, что введение дополнительного СБД или ФС дает снижение вероятности отказа в обслуживании на 0,86 %, увеличение относительной и абсолютной пропускной способности на 0,86 и 4,66 %. При использовании только комбинации «сервер баз данных – файловый сервер» значения данных параметров составляют $P_n = 0,012$, $Q = 0,965$ и $A = 5,310$, что соответствует требованиям к обеспечению информационной поддержки КП изделия.

Таким образом, введение дополнительных серверов обработки данных при достаточно низкой частоте поступления запросов на получение доступа к информации относительно производительности используемых средств обработки данных, дает положительный эффект с точки зрения надежности информационной структуры, но он незначителен и не окупает дополнительных финансовых затрат.

Заключение

В результате работы получена обобщенная модель структурной организации единого информационного пространства по параллельному методу интеграции средств поддержки жизненного цикла изделия. Данная модель включает в себя описание элементов структуры информационного пространства, каналов связи, их статических и динамических характеристик. Предложенная модель также учитывает ограничения, накладываемые на формирование структуры ЕИП. Выполнено имитационное моделирование структурной организации ЕИП, представленной в виде совокупности многоканальных систем массового обслуживания. По результатам моделирования сделаны выводы о пропускной способности и надежности рассматриваемого варианта организации ЕИП с учетом предлагаемых к использованию аппаратных (серверное решение) и программных (система управления базами данных) средств.

Список литературы / References

1. **Yang H., Zhang J., Sun J. and Yu L.** Review of advanced FPGA architectures and technologies. *Journal of Electronics (China)*, 2014, vol. 31, no. 5, p. 371–393. DOI 10.1007/s11767-014-4090-x
2. **Nane R., Sima V.-M., Pilato C., Choi J., Fort B., Canis A., Chen Y. T., Hsiao H., Brown S., Ferrandi F., Anderson J., Bertels K.** A Survey and Evaluation of FPGA High-Level Synthesis Tools. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2016, vol. 35, no. 10, p. 1591–1604. DOI 10.1109/TCAD.2015.2513673
3. **Meeus W., Van Beeck K., Goedeme T., Meel J., Stroobandt D.** An overview of today's high-level synthesis tools. *Design Automation for Embedded Systems*, 2012, vol. 16, no. 3, p. 31–51. DOI 10.1007/s10617-012-9096-8
4. **Cong J., Bin L., Neuendorffer S., Noguera J., Vissers K., Zhiru Z.** High-Level Synthesis for FPGAs: From Prototyping to Deployment. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2011, vol. 30, no. 4, p. 473–491. DOI 10.1109/TCAD.2011.2110592
5. **Chang K. H.** Product Design Modeling Using CAD/CAE. Elsevier, 2014.
6. **Watts F. B.** Engineering Documentation Control Handbook. Elsevier, 2012.
7. **Davidow W. H., Malone M. S.** The Virtual Corporation: Structuring and Revitalizing the Corporation for the 21st Century. New York: HarperCollins, 1992.
8. **Воронин А. А., Губко М. В., Мишин С. П., Новиков Д. А.** Математические модели организаций. М.: URSS, 2008.
Voronin A. A., Gubko M. V., Mishin S. P., Novikov D. A. Matematicheskie modeli organizatsiy [Mathematical models of organizations]. Moscow, URSS Publ., 2008. (in Russ.)
9. **Кульга К. С., Кривошеев И. А.** Модели и методы создания интегрированной системы для автоматизации технической подготовки и управления авиационным и машиностроительным производством: Монография. М.: Машиностроение, 2011.
Kulga K. S., Krivosheev I. A. Modeli i metody sozdaniya integrirovannoy sistemy dlya avtomatizatsii tekhnicheskoy podgotovki i upravleniya aviatsionnym i mashinostroitelnym proizvodstvom: monografiya [Models and methods for creating an integrated system for the automation of technical training and management of aviation and engineering production: a monograph]. Moscow, Mashinostroenie Publ., 2011. (in Russ.)
10. **Краснянский М. Н.** Проектирование информационных систем управления документооборотом научно-образовательных учреждений. Тамбов: ТГТУ, 2015.
Krasnyanskiy M. N. Proektirovanie informatsionnykh sistem upravleniya dokumentooborotom nauchno-obrazovatelnykh uchrezhdeniy [Designing information systems for document management of scientific and educational organizations]. Tambov, Tambov State Technical University Publ., 2015. (in Russ.)
11. **Ghosh J.** Optimal Information Production in Product Development Environments: A Model With Simulations. *IEEE Transactions on Engineering Management*, 2013, vol. 60, no. 3, p. 604–616. DOI 10.1109/TEM.2012.2233743
12. **Yang Q., Yao T., Lu T., Zhang B.** An Overlapping-Based Design Structure Matrix for Measuring Interaction Strength and Clustering Analysis in Product Development Project. *IEEE Transactions on Engineering Management*, 2014, vol. 61, no. 1, p. 159–170. DOI 10.1109/TEM.2013.2267779
13. **Browning T. R.** Design Structure Matrix Extensions and Innovations: A Survey and New Opportunities. *IEEE Transactions on Engineering Management*, 2016, vol. 63, no. 1, p. 27–52. DOI 10.1109/TEM.2015.2491283

14. **Матвеев В. Ф., Ушаков В. Г.** Системы массового обслуживания. М.: МГУ, 1984.
Matveev V. F., Ushakov V. G. Sistemy massovogo obsluzhivaniya [Queuing systems]. Moscow, Moscow State University Publ., 1984. (in Russ.)
15. **Shampine L. F. and Reichelt M. W.** The MATLAB ODE Suite. *SIAM Journal on Scientific Computing*, 1997, vol. 18, no. 1, p. 1–22. DOI 10.1137/S1064827594276424

*Материал поступил в редколлегию
 Received
 09.04.2019*

Сведения об авторах / Information about the Authors

Дрозд Олег Владимирович, аспирант кафедры «Информационные технологии на радиоэлектронном производстве», Институт космических и информационных технологий, Сибирский федеральный университет (ул. Киренского, 26Б, Красноярск, 660074, Россия)

Drozd Oleg Vladimirovich, Graduate Student of the Department of Information Technologies on Radio-Electronic Production, School of Space and Information Technology, Siberian Federal University (26B Kirensky Str., Krasnoyarsk, 660074, Russian Federation)

odrozd@sfu-kras.ru

Ченцов Сергей Васильевич, доктор технических наук, профессор, заведующий кафедрой «Систем автоматики, автоматизированное управление и проектирование», Институт космических и информационных технологий, Сибирский федеральный университет (ул. Киренского, 26Б, Красноярск, 660074, Россия)

Chentsov Sergey Vasilevich, Doctor of Technical Science, professor, head of the Department of Systems of Automatic Equipment, Automated Management and Design, School of Space and Information Technology, Siberian Federal University (26B Kirensky Str., Krasnoyarsk, 660074, Russian Federation)

schentsov@sfu-kras.ru

Капулин Денис Владимирович, кандидат технических наук, доцент, заведующий кафедрой «Информационные технологии на радиоэлектронном производстве», Институт космических и информационных технологий, Сибирский федеральный университет (ул. Киренского 26 Б, Красноярск, 660074, Россия)

Kapulin Denis Vladimirovich, Candidate of Technical Sciences, Associate Professor, head of the Department of Information Technologies in Radioelectronic Production, School of Space and Information Technology, Siberian Federal University (26B Kirensky Str., Krasnoyarsk, 660074, Russian Federation)

dkapulin@ sfu-kras.ru

Применение метода k -средних для идентификации пользователя по клавиатурному почерку

Ю. И. Еременко, Ю. С. Олюнина

*Старооскольский технологический институт (филиал) ННТУ «МИСиС»
Старый Оскол, Россия*

Аннотация

Рассматривается проблема защиты от несанкционированного доступа к данным путем идентификации пользователей по биометрическим характеристикам, а именно по клавиатурному почерку. Для решения этой задачи авторами была проведена серия опытов для получения статистической выборки образцов клавиатурного почерка, которые используются в качестве биометрической характеристики для установления личности ее владельца. Для анализа данных и определения автора конкретного образца почерка в работе рассматривается метод k -средних, являющийся одним из наиболее простых и эффективных статистических методов классификации в случае, когда число кластеров заранее известно. Оценка эффективности предложенного метода для решения поставленной задачи оценивается с помощью коэффициентов ложного доступа и ложного отказа в доступе, которые являются основными характеристиками биометрических систем аутентификации. Полученные результаты позволяют сделать вывод о наличии ряда ограничений при использовании данного метода, возникающих в силу того, что данная задача является слабоформализуемой, зависящей от множества факторов, в том числе и не поддающихся математическому описанию, таких как нестабильность клавиатурного почерка, объясняемая изменением психофизиологического состояния пользователя, эргономичностью клавиатуры и т. д. Учитывая упомянутые особенности, дальнейшее решение задачи предлагается рассматривать в перспективе использования методов, основанных на интеллектуальной обработке данных, которые позволяют обнаруживать в потоке данных скрытые закономерности и зависимости.

Ключевые слова

идентификация пользователя, клавиатурный почерк, биометрические характеристики, метод k -средних

Для цитирования

Ерёменко Ю. И., Олюнина Ю. С. Применение метода k -средних для идентификации пользователя по клавиатурному почерку // Вестник ННТУ. Серия: Информационные технологии. 2019. Т. 17, № 2. С. 30–38. DOI 10.25205/1818-7900-2019-17-2-30-38

Using the K-means Method to Identify a User by Keyboard Handwriting

Yu. I. Eremenko, Yu. S. Olyunina

*Stary Oskol Technological Institute (branch) NUST "MISiS"
Stary Oskol, Russian Federation*

Abstract

This article discusses the problem of protection against unauthorized access to data by identifying users by biometric characteristics – keyboard handwriting. To do this, the authors conducted a series of experiments to obtain a set of samples of keyboard handwriting, which are used as a biometric characteristic to identify the owner. To analyze the data and determine the author of a particular handwriting sample, the authors consider the k -means method. This method is one of the simplest and most effective statistical classification methods when the number of clusters is known in advance. To evaluate the effectiveness of this method for solving this problem, the authors propose using the coefficients of false access and false access denial, which are the main characteristics of biometric authentication systems. The results suggest that there are some limitations when using this method. They are related to the fact that this problem is poorly formalized and depends on many factors that can not be mathematically described. For

example, the instability of the keyboard handwriting, which is due to changes in the psycho-physiological state of the user, the ergonomics of the keyboard and others. Given these features, the authors propose to solve the problem using methods based on intelligent data processing, which allow to detect hidden patterns and dependencies in the data flow.

Keywords

user identification, keyboard handwriting, biometric characteristics, the method k -means

For citation

Eremenko Yu. I., Olyunina Yu. S. Using the K -means Method to Identify a User by Keyboard Handwriting. *Vestnik NSU. Series: Information Technologies*, 2019, vol. 17, no. 2, p. 30–38. (in Russ.) DOI 10.25205/1818-7900-2019-17-2-30-38

Введение

Одной из наиболее активно возрастающих угроз безопасности информационных систем является проблема утечки данных. Для современных организаций она становится все более актуальной вместе с ростом интенсивности использования информационных технологий. Значительную опасность представляют угрозы, возникающие при обработке конфиденциальной информации в организациях, и в первую очередь это угроза несанкционированного доступа к данным¹.

Аналитики центра InfoWatch, занимающегося изучением вопросов защиты от утечек информации, отмечают, что лишь в 22 % случаев утечка информации происходила в результате хакерской активности, в большинстве случаев (75 %) информация утекала по вине внутреннего нарушителя². Отследить факты появления внутренних угроз гораздо сложнее, поскольку в отличие от внешних угроз их нельзя достаточно точно контролировать с помощью аппаратно-технических мер. На сегодняшний день одним из наиболее простых и распространенных средств защиты данных является парольная идентификация.

В работе [1] отмечаются основные недостатки данного метода: неоднозначность идентификации оператора ключевой системы, возможность обмана системы защиты, например путем кражи или взлома пароля, «невозможность обнаружения подмены законного авторизованного пользователя. В данном случае злоумышленник может нанести вред обрабатываемой КС информации, когда оператор оставляет без присмотра КС с пройденной процедурой авторизации» [1].

В качестве перспективного варианта решения проблемы возможно применение биометрических систем идентификации пользователя. Актуальность выбора данного метода описана в работах [2; 3]. Такой подход имеет ряд преимуществ по сравнению с другими средствами идентификации, описанными в работах [1; 3–5] и др. На сегодняшний день биометрическая идентификация личности является одним из перспективных направлений в сфере информационной безопасности и характеризуется такими преимуществами, как неотделимость биометрической характеристики от владельца и крайняя сложность подделки [2]. Кроме того, в отличие от парольных средств защиты, которые человек может непреднамеренно или по своему умыслу сообщить злоумышленнику, или от различных карт и ключей, которые могут быть утеряны или украдены, биометрические характеристики позволяют однозначно идентифицировать человека [6].

В рамках данной работы предложено использование клавиатурного почерка, поскольку из всех биометрических методов защиты информации данный метод идентификации является самым простым для внедрения и наименее затратным [4], не требует установки специальных аппаратных средств, не нуждается в сопровождении, является прозрачным для конечного пользователя [5], т. е. не причиняет ему неудобств и позволяет проводить скрытую аутентификацию.

Анализ работ по данной теме показал, что для классификации образцов клавиатурного почерка с целью установления личности его владельца и защиты данных компьютерной сис-

¹ Аналитический центр InfoWatch. URL: www.infowatch.ru/analytics/ (дата обращения 21.11.2018).

² Там же.

темы от несанкционированного доступа наибольшее распространение получила обработка данных с использованием методов математической статистики и теории вероятностей либо методы, используемые в совокупности с дополнительными механизмами защиты (парольной защитой либо использованием дополнительных аппаратных средств). Необходимость двухфакторной идентификации обусловлена тем, что значения коэффициентов ложного доступа и ложного отказа в доступе, являющихся основными критериями надежности биометрических систем, достаточно высоки. Использование методов математической статистики «основано на анализе усредненных значений временных параметров, характеризующих манеру работы пользователя с клавиатурой, поэтому результаты, полученные при их применении, не позволяют произвести процедуру идентификации достаточно эффективно» [6].

В рамках данной работы рассмотрено использование альтернативного метода обработки данных, такого как метод k -средних, который является одним из наиболее простых и эффективных методов машинного обучения для решения задач классификации, характеризуется простотой реализации и возможностью распространения полученных решений на новые наблюдения [7]. Кроме того, данный метод не требует вычисления и хранения матрицы расстояний и позволяет распараллеливать процесс вычислений.

Постановка задачи

Задачей данного исследования является необходимость оценить эффективность использования метода k -средних для решения задачи идентификации пользователя по клавиатурному почерку и сделать дальнейшие выводы о возможности его применения или необходимости модификации с целью получения максимального значения целевой функции, характеризующей определение вероятности принадлежности образца почерка конкретному пользователю [8].

Описание предмета исследования

Клавиатурный почерк представляет собой набор характеристик работы пользователя на клавиатуре и характеризуется множеством параметров, описанных в работах [3; 5].

На основании описанных характеристик была произведена серия опытов по их регистрации. Для их проведения были использованы фразы-панграммы, включающие в себя все буквы алфавита, в частности, такие как: «съешь ещё этих мягких французских булок, да выпей же чаю»; «в чашах юга жил-был цитрус... – да, но фальшивый экземпляр» и т. п.

Результаты проведенных экспериментов представлены в виде графиков. Для более наглядного представления результатов на графиках отображена только часть набранных фраз, так, например, на рис. 1 представлена динамика написания слова «каждый», которая является схожей во всех проведенных сериях опытов. Это утверждение верно и при наборе каждой фразы целиком. На данном графике по оси ординат обозначено время удержания нажатой клавиши, на оси абсцисс – соответствующая буква. Опыт проводился одним и тем же пользователем 10 раз.

Кроме того, на рис. 1 видно, что числовые характеристики почерка каждого человека остаются схожими вне зависимости от количества опытов, но в то же время почерк отдельного человека очевидно отличим от других.

На основании полученных результатов можно сделать вывод о том, что клавиатурный почерк является достаточно информативной биометрической характеристикой, и его использование возможно для определения личности владельца почерка.

Однако клавиатурный почерк как динамическая (поведенческая) характеристика пользователя подвержен влиянию множества факторов, что делает его весьма нестабильным. Основные сложности при работе с клавиатурным почерком связаны с большим разнообразием навыков набора текста у пользователей. Процедура идентификации может быть неэффективной или невозможной для лиц, не имеющих стойких навыков работы с клавиатурой. Из тео-

рии машинописи и делопроизводства установлено, что срок, необходимый для формирования устоявшегося клавиатурного почерка, составляет не менее 6 месяцев. Кроме того, на характеристики набора текста влияют факторы, обусловленные психологическим состоянием человека: сонливость, тревога, усталость. Не менее значимыми являются и другие объективные причины, например травма кисти или пальца, использование устройств ввода нестандартного размера, обладающих другой эргономичностью [1]. Все эти факторы снижают достоверность идентификации. В связи с этим выбор метода обработки параметров клавиатурного почерка является сложной задачей, требующей тщательного анализа используемых методов обработки данных.

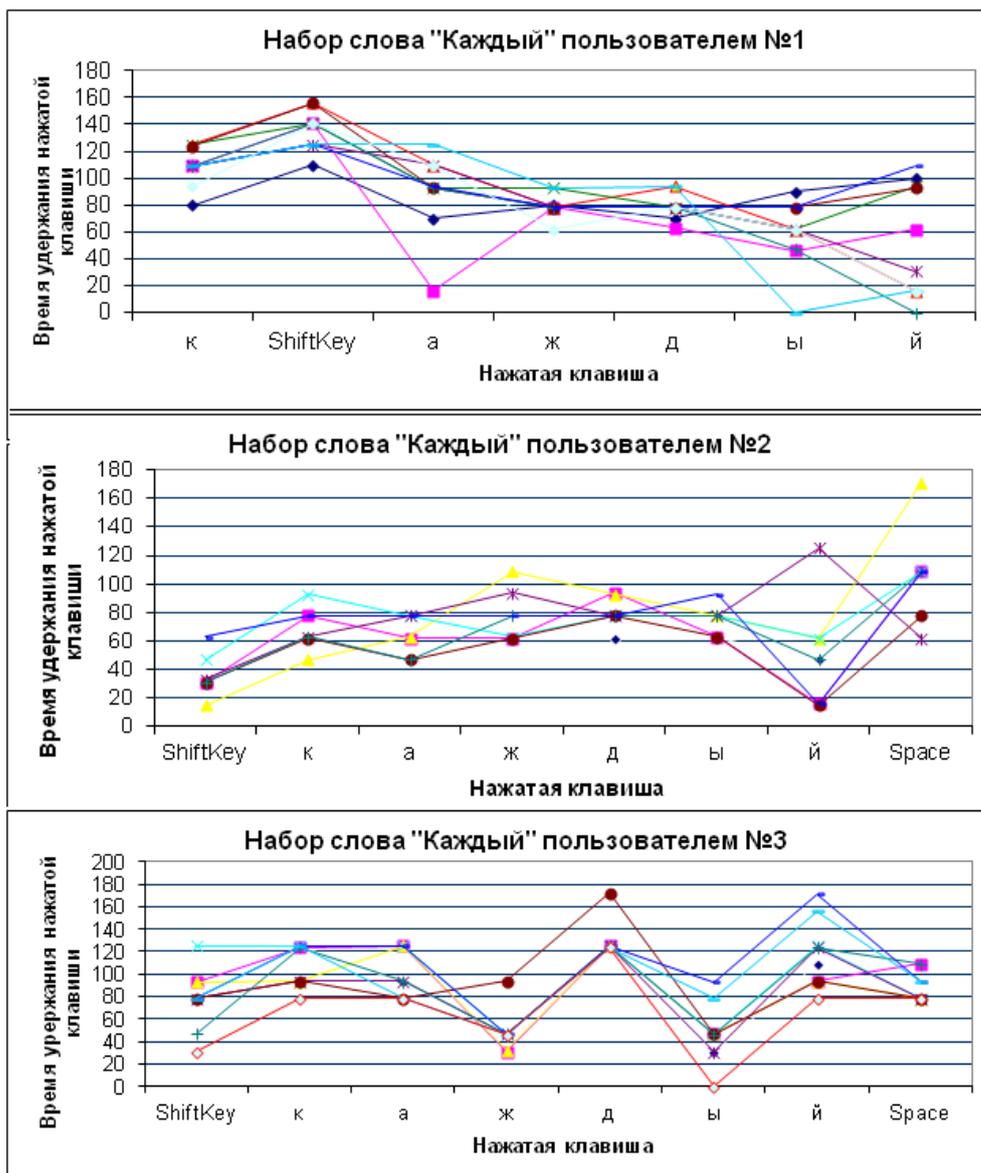


Рис. 1. Образцы клавиатурного почерка пользователей № 1–3

Fig. 1. Samples of Users' Keyboard Handwriting № 1–3

Поставленная задача по своей сути является задачей классификации, так как образцы клавиатурного почерка представляют собой конечное множество объектов, которое нужно соотнести с некоторым множеством классов. Для ее решения возможно использование таких методов, как регрессионный анализ, метод k -средних, метод главных компонент и т. д., которые позволят отсеять наименее информативные признаки клавиатурного почерка и, как следствие, предварительно уменьшить размерность обучающей выборки. При использовании регрессионного анализа для решения поставленной задачи авторами был выявлен ряд ограничений, влияющих на достоверность идентификации, что подробно рассмотрено в [8].

В данной работе предложено использование метода k -средних, поскольку он эффективен в случае, когда число кластеров заранее известно, достаточно прост в реализации, поэтому его можно использовать для анализа больших объемов данных. Кроме того, данный метод может учитывать весовые коэффициенты, что является несомненным преимуществом, поскольку одной из задач исследовательской работы авторов является оценка значимости параметров клавиатурного почерка.

Метод k -средних основан на вычислении расстояния от объекта до k ближайших соседей, причем число k задается индивидуально для каждого случая. Под «соседями» в данном случае будем понимать объекты, близкие к исследуемому. Определение принадлежности объекта к тому или иному классу (кластеру) производится на основе вывода о том, к какому классу относятся его ближайшие соседи [9]. На практике вопрос выбора оптимального значения параметра k является достаточно сложным, поскольку от правильности его выбора зависит правильность решения задачи о принадлежности объекта к некоторому классу. Так, если взять k слишком малым, возможна ситуация, при которой единственным ближайшим соседом окажется объект с неверно определенным классом, что приведет к принятию неверного решения о принадлежности данного объекта. Если же взять k , напротив, слишком большим, например равным количеству объектов, то расстояние до центра не будет иметь вообще никакого значения.

В рамках поставленной задачи классификация данных производилась путем разбиения множества объектов $X = 100$ на $k = 10$ кластеров, поскольку требуется идентификация 10 человек, предоставивших образцы почерка. В качестве меры близости объекта к центру кластера можно использовать расстояние Евклида, манхэттенское расстояние, формулу Минковского и др. В данном случае используется Евклидово расстояние, поскольку реализация метода k -средних осуществлялась с использованием программной среды Statistica, в вычислительный алгоритм которой заложена именно эта мера:

$$\rho(x, y) = |x - y| = \sqrt{\sum_{p=1}^n (x_p - y_p)^2},$$

где R^n – пространство множества объектов; x, y – координаты объекта в пространстве.

Применение данного метода было реализовано в программной среде Statistica. Для получения статистических данных в эксперименте приняли участие 10 пользователей, каждый из которых предоставил по 10 образцов клавиатурного почерка. Таким образом, в целом была получена выборка данных объемом 100 образцов клавиатурного почерка. В качестве текста для набора были использованы фразы-панграммы, примеры которых были приведены ранее.

Выбор такого варианта текста для набора основан на том, что данные фразы позволяют максимально эффективно оценить все параметры клавиатурного почерка. В работах [2; 3] отмечалось, что каждый образец почерка характеризуется 17 параметрами. Таким образом, вся статистическая выборка представляет собой матрицу X размером 100×17 .

Классификация данных производилась путем разбиения множества объектов $X = 100$ на $k = 10$ кластеров, поскольку поставленная задача требует идентификации 10 человек, предоставивших образцы почерка.

Достигнутые результаты

На рис. 2 представлено окно описания параметров в программе Statistica.

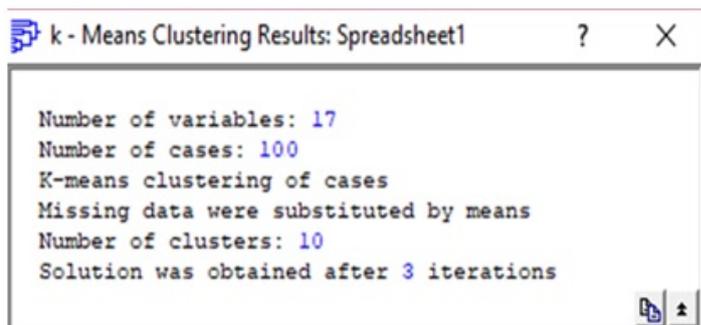


Рис. 2. Окно описания параметров анализа данных в программе Statistica
 Fig. 2. A Window Describing the Parameters of Data Analysis in the Program Statistica

Одним из важнейших показателей эффективности применения метода идентификации является достоверность аутентификации пользователей [4]. При принятии решения о личности владельца почерка система идентификации может выдать как неправильное решение, приняв нелегального пользователя за легального, так и противоположное решение, отказав авторизованному пользователю в доступе. На основании этих факторов оценка эффективности применения данного метода производилась путем вычисления значений коэффициента ложного доступа FAR и коэффициента ложного отказа в доступе FRR. Коэффициент FAR (False Acceptance Rate) возникает в случае, когда зарегистрированный пользователь аутентифицируется как не имеющий доступа к системе, а коэффициент FRR (False Rejection Rate) характеризует ситуацию, при которой пользователь, не имеющий прав доступа, аутентифицируется как зарегистрированный [4].

При определении принадлежности образца почерка (Case_No) определенному кластеру (Cluster) в программе Statistica учитывалось его расстояние от центра данного кластера, вычисленного на основании данных о координатах элементов, ближайших к рассматриваемому. Окно результатов оценивания представлено в формате таблицы Word.

Результаты классификации

Classification Result

№ образца почерка (Case_No)	№ кластера (Cluster)	Расстояние до центра кластера (Distance)
4	4	0,06
5	4	0,06
21	4	0,06
22	4	0,06
23	4	0,06
24	4	0,05
25	4	0,06
26	4	0,06
27	4	0,07
28	4	0,07
30	4	0,06

Таблицы, аналогичные представленной, получены для каждого из 10 выделенных кластеров. На основании полученных результатов вычислялось среднее значение коэффициентов FAR и FRR, которые являются основными характеристиками биометрических систем аутентификации. Их значения составили 28 и 30 % соответственно. Полученные результаты являются неудовлетворительными, поскольку стандартным порогом доступа биометрических систем является значение 90 % [1].

Оценивая результаты, следует отметить, что использование данного метода для решения поставленной задачи имеет ряд ограничений, к которым можно отнести:

- необходимость хранения всей выборки данных для проведения классификации, что требует выделения большого объема памяти;
- использование данных без возможности дообучения, что критично для решения поставленной задачи, поскольку клавиатурный почерк представляет собой динамическую биометрическую характеристику, т. е. может изменяться с течением времени;
- близкую расположенность центров кластеров друг к другу в силу небольшого размаха значений выборки (в некоторых случаях разница в тысячные доли), что приводит к увеличению значений FRR и FAR. На рис. 3 видно, что центры кластеров, характеризующих пользователей 2, 5, 6, 9 практически совпадают, и результаты классификации данных, соответствующих этим пользователям, показали самые высокие значения ошибки.

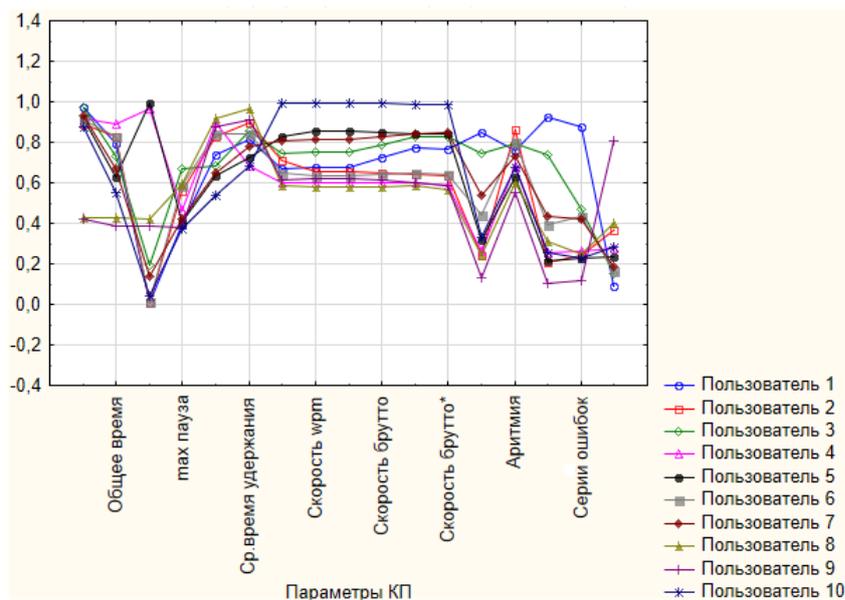


Рис. 3. Распределение центров кластеров

Fig. 3. Distribution of Cluster Centers

На основании описанных выше недостатков данного метода для повышения эффективности идентификации пользователя по клавиатурному почерку предложено использовать интеллектуальные методы, в частности искусственные нейронные сети. Применение метода k -средних показало, что центры кластеров находятся слишком близко друг к другу, поэтому плохо различимы. Такая ситуация может привести к тому, что значения, соответствующие разным пользователям, могут быть идентифицированы как принадлежащие одному человеку, и наоборот, что может привести к увеличению значений FRR и FAR. Точности метода не хватает для учета таких ограничений, в то же время нейронные сети обладают более высокой способностью к разграничению данных, что позволяет выдвинуть предположение о том, что их применение позволит получить более высокий результат. Кроме того, использование дан-

ного метода подразумевает использование статистики без возможности дообучения, что не позволяет учитывать факт изменения клавиатурного почерка пользователя со временем или под влиянием каких-либо внешних факторов. В то же время одним из важнейших преимуществ нейронных сетей является способность обучаться на основе данных окружающей среды и в результате обучения повышать свою производительность посредством интерактивного процесса корректировки синаптических весов и порогов. Кроме того, ИНС обладают такими преимуществами, как возможность выявлять скрытые закономерности и зависимости в потоке данных, а также обеспечивать более высокое быстродействие по сравнению с другими методами за счет распараллеливания процесса обработки данных.

Заключение

В работе рассматривается задача идентификации пользователя по клавиатурному почерку. После анализа существующих методов решения данной задачи авторами предложено использование метода k -средних, являющегося одним из наиболее простых в реализации методов машинного обучения для решения задач классификации. В ходе проведенных экспериментов получены результаты, представленные в виде значения ошибок первого и второго рода, которые являются стандартными критериями оценок надежности всех биометрических систем. Анализ полученных результатов показал, что пороговые значения доступа ниже, чем стандартные пороговые значения для биометрических систем. В связи с этим авторами предложено использование альтернативных методов для обработки статистических данных.

Список литературы / References

1. **Савинов А. Н.** Методы, модели и алгоритмы распознавания клавиатурного почерка в ключевых системах: Дис. ... канд. техн. наук. СПб., 2013. 128 с.
Savinov A. N. Methods, models and algorithms for recognition of keyboard handwriting in key systems. Dis. ... Cand. Tech. Sci. St. Petersburg, 2013, 128 p. (in Russ.)
2. **Еременко Ю. И., Олюнина Ю. С.** Об идентификации клавиатурного почерка пользователей // Сборник материалов XXVIII Междунар. науч.-практ. конф. «Перспективы развития информационных технологий». 2016. С. 145–151.
Eremenko Yu. I., Olyunina Yu. S. About the identification of keyboard handwriting users. In: Collection of materials of the XXVIII International Scientific and Practical Conference “Prospects for the development of information technology”, 2016, p. 145–151. (in Russ.)
3. **Еременко Ю. И., Олюнина Ю. С.** Об обработке потока данных с целью выявления скрытых характеристик клавиатурного почерка // Современные сложные системы управления. HTCS’2017: Материалы XII Междунар. науч.-практ. конф.: В 2 ч. Липецк: Изд-во Липецк. гос. техн. ун-та, 2017. Ч. 2. С. 31–36.
Eremenko Yu. I., Olyunina Yu. S. About the processing of the data stream to identify latent characteristics of the keyboard handwriting. In: *Sovremennyye slozhnyye sistemy upravleniya*. Lipetsk, Lipetsk State Technical University Publ., 2017, p. 31–36. (in Russ.)
4. **Шарипов Р. Р.** Разработка полигауссова алгоритма аутентификации пользователей в телекоммуникационных системах и сетях по клавиатурному почерку: Дис. ... канд. техн. наук: 05.12.13. Казань, 2006. 135 с.
Sharipov R. R. Development of poly-Gaussian algorithm for user authentication in telecommunication systems and networks using keyboard handwriting. Dis. ... Cand. Tech. Sci. Kazan, 2006, 135 p. (in Russ.)
5. **Скуратов С. В.** Использование клавиатурного почерка для аутентификации в компьютерных информационных системах // Безопасность информационных технологий. 2010. № 2. С. 35–38.

- Skuratov S. V.** Using keyboard handwriting for authentication in computer information systems. *Besopasnost informacionnykh technologij* 2010, no. 2, p. 35–38. (in Russ.)
6. **Епифанцев Б. Н., Ложников П. С., Сулавко А. Е.** Альтернативные сценарии авторизации при идентификации пользователей по динамике подсознательных движений // Вопросы защиты информации. 2013. № 2. С. 28–35.
Epifantsev B. N., Lozhnikov P. S., Sulavko A. E. Alternative authorization scenarios for identifying users by the dynamics of subconscious movements. *Voprosi sashiti informatsii*, 2013, no. 2, p. 28–35. (in Russ.)
7. **Агеев М. С., Добров Б. В.** Метод эффективного расчета матрицы ближайших соседей для полнотекстовых документов // Вестник Санкт-Петербургского университета. Серия 10: Прикладная математика, информатика, вопросы управления. 2011. Вып. 3. С. 72–84.
Ageev M. S., Doprov B. V. The method of effective calculation of the matrix of nearest neighbors for full-text documents. *Vestnik St. Petersburg University. Series 10*, 2011, no. 3, p. 72–84. (in Russ.)
8. **Еременко Ю. И., Олюнина Ю. С.** Об определении наиболее значимых параметров клавиатурного почерка с помощью регрессионного анализа // Системы управления и информационные технологии. 2018. № 2 (72). С. 28–31.
Eremenko Yu. I., Olyunina Yu. S. About determination of the most significant parameters of keyboard handwriting using regression analysis. *Sistemy upravleniya i informatsionnie tehnologii*, 2018, no. 2 (72), p. 28–31. (in Russ.)
9. **Стрюков Р. К., Шашкин А. И.** О модификации метода ближайших соседей // Вестник ВГУ. Серия: Системный анализ и информационные технологии. 2015. № 1. С. 114–120.
Strukov R. K., Shashkin A. I. On the modification of the nearest neighbors method. *Vestnik VSU. Seriya: Sistemy analiz i informatsionnye tehnologii*, 2015, no. 1, p. 114–120. (in Russ.)

*Материал поступил в редколлегию
Received
20.03.2019*

Сведения об авторах / Information about the Authors

Еременко Юрий Иванович, доктор технических наук, профессор, заведующий кафедрой АИСУ, Старооскольский технологический институт (филиал) НИТУ «МИСиС» (мкр. Макаренко, 42, Старый Оскол, 309516, Белгородская область, Россия)

Yuriy I. Eremenko, Doctor of Technical Sciences, Professor, Head of the Department of Stary Oskol Technological Institute branch of the National Research Technological Institute “Moscow Institute of Steel and Alloys” (42 Makarenko md., Stary Oskol, 309516, Belgorod Region, Russian Federation)
erem49@mail.ru

Олюнина Юлия Сергеевна, аспирант, ассистент кафедры АИСУ, Старооскольский технологический институт (филиал) НИТУ «МИСиС» (мкр. Макаренко, 42, Старый Оскол, Белгородская область, 309516, Россия)

Yuliya S. Olyunina, graduate student, assistant at the department of AISU, Stary Oskol Technological Institute branch of the National Research Technological Institute “Moscow Institute of Steel and Alloys” (42 Makarenko md., Stary Oskol, 309516, Belgorod Region, Russian Federation)
julijasergeevna@mail.ru

УДК 004.550
DOI 10.25205/1818-7900-2019-17-2-39-48

Система каталогизации и мониторинга территориально распределенных вычислительных узлов в среде WPS сервисов для решения геологических задач

В. С. Ерёмченко, В. В. Наумова

*Государственный геологический музей им. В. И. Вернадского РАН
Москва, Россия*

Аннотация

Данная работа посвящена разработке подхода к каталогизации разнородных веб-сервисов территориально распределенных вычислительных узлов в вычислительно-аналитической геологической среде, а также принципов по созданию системы мониторинга веб-сервисов, присутствующих в каталоге. Для этого в статье рассматриваются используемые в мире подходы по организации каталогов веб-сервисов на основе SOAP (Simple Object Access Protocol) протокола, с использованием UDDI (Universal Description Discovery and Integration) реестров. На основе исследованного материала предлагается схема для описания разнородных веб-сервисов с возможностью дальнейшего создания каталога веб-сервисов, включающего функции поиска и получения метаданных о каждом сервисе в каталоге. Также в статье кратко описываются принципы, на основе которых функционирует вычислительно-аналитическая геологическая среда, в рамках которой разрабатывается система каталогизации и мониторинга, и приводится список территориально распределенных веб-сервисов по обработке разнородных данных, используемых в данной среде.

Ключевые слова

вычислительно-аналитическая среда, веб-сервисы, каталог веб-сервисов, Web Processing Service, мониторинг сервисов

Благодарности

Работы выполняются в рамках государственного задания ГГМ РАН по теме № 0140-2019-0005 «Разработка информационной среды интеграции данных естественнонаучных музеев и сервисов их обработки для наук о Земле».

Для цитирования

Ерёмченко В. С., Наумова В. В. Система каталогизации и мониторинга территориально распределенных вычислительных узлов в среде WPS сервисов для решения геологических задач // Вестник НГУ. Серия: Информационные технологии. 2019. Т. 17, № 2. С. 39–48. DOI 10.25205/1818-7900-2019-17-2-39-48

The System of Cataloguing and Monitoring of Geographically Distributed Computing Nodes in WPS Services Environment for Geological Problems Solving

V. S. Eremenko, V. V. Naumova

*V. I. Vernadsky State Geological Museum of the Russian Academy of Sciences
Moscow, Russian Federation*

Abstract

The study describes an approach for heterogeneous web services cataloguing of geographically distributed computing nodes in the computational and analytical geological environment. The approach of creation of web services monitoring system is proposed. In this study the approach for web services catalogue organization based on SOAP (Simple Object Access Protocol) protocol, with use of UDDI (Universal Description Discovery and Integration) registries is reviewed. The scheme for heterogeneous web services description with an ability of further creation of catalogue is

© В. С. Ерёмченко, В. В. Наумова, 2019

proposed. The catalogue includes functions for searching and receiving of meta-information about every web service in it. The study also describes an approach for computational and analytical geological environment creation. A number of web services for heterogeneous data processing used in geological research is described. The approaches for organization of cataloguing and web services monitoring provide a higher level of reliability of computational and analytical geological environment. They provide an information about web service reliability within environment.

Keywords

computational analytical environment, web services, web services catalogue, Web Processing Services, service monitoring

Acknowledgements

The study is supported by the Government contract no. 0140-2019-0005 with SGM RAS “Development of an information environment for integrating data from natural science museums and their processing services for Earth sciences”.

For citation

Eremenko V. S., Naumova V. V. The System of Cataloguing and Monitoring of Geographically Distributed Computing Nodes in WPS Services Environment for Geological Problems Solving. *Vestnik NSU. Series: Information Technologies*, 2019, vol. 17, no. 2, p. 39–48. (in Russ.) DOI 10.25205/1818-7900-2019-17-2-39-48

Введение

С развитием информационных технологий всё больше программных продуктов, направленных на решение различных научных задач, предоставляются пользователям в виде облачных сервисов. При использовании данного подхода проблемы, связанные с обеспечением необходимых вычислительных мощностей, подготовкой специализированного программного окружения и поддержкой актуальности программного продукта, решаются на стороне сервиса. Существует ряд сервисов, предоставляющих возможность обработки и анализа научных данных в режиме онлайн. Такие сервисы расположены на территориально распределенных вычислительных узлах, которые разрабатываются в научных институтах, университетах и коммерческих компаниях, специализирующихся на обработке и анализе различных типов данных [1].

Построение вычислительно-аналитических систем по обработке данных, основанных на взаимодействии с пользователем через веб-интерфейс, активно ведется в различных предметных областях. Так, в области обработки спутниковых данных к подобным системам можно отнести проект NASA-Giovanni, систему Google Earth Engine, платформы ESA G-POD, а также системы See the Sea, Vega-Science и др., созданные в Институте космических технологий РАН на основе авторской технологии GEOSMIS [2]. В области обработки и анализа пространственных данных примером такой информационной системы является ArcGIS Online.

В настоящее время актуальной является разработка тематических вычислительно-аналитических сред, имеющих единые точки доступа к территориально распределенным вычислительно-аналитическим ресурсам, позволяющие в рамках единой системы решать различные задачи по обработке и анализу в заданной предметной области.

А. М. Федотовым, В. Б. Барахнинным и др. [3] предложена концепция распределенной информационно-аналитической среды для исследований экологических систем. В статье описывается модель виртуальной среды, определяются категории данных, объекты среды и приводится пример схемы среды с описанием используемых технологий.

Е. П. Гордовым, В. Н. Крупчатниковым и др. [4] представлен проект по созданию тематической виртуальной исследовательской среды для анализа, оценки и прогнозирования воздействия глобального изменения климата. Проект среды разрабатывается с целью обеспечения свободного доступа к различным ресурсам данных и службам обработки через веб-браузер.

В работе L. Candela и др. [5] приводится общий обзор существующих виртуальных исследовательских сред, выделяются общие и отличительные особенности различных подходов

к построению таких сред, разбираются проблемы, которые необходимо решать в данной области.

И. В. Бычковым, Г. М. Ружниковым и др. [6; 7] разработана и успешно функционирует среда сервисов обработки геоданных WPS. Эта среда поддерживает вызов сервисов обработки, построенных с использованием интерфейса OGC WPS (Web Processing Service). В среде реализована возможность построения цепочек обработки с использованием языка javascript для формирования сценария обработки.

Создание подобных сред подразумевает необходимость разработки методов и технологий каталогизации и мониторинга состояния сервисов.

Решение задачи каталогизации сервисов представлено в 2000 г. в виде стандарта для размещения веб-сервисов UDDI. Основная идея данного подхода – создание каталогов (реестров) веб-сервисов на основе языка описания веб-сервисов WSDL (Web Services Description Language), которые бы позволяли осуществлять быстрый поиск и получение информации о сервисах, а также интегрировать различные сервисы в сторонние системы. Однако стандарт UDDI не получил дальнейшего развития в силу ограничений, не учитывающих специфику сервисов в различных предметных областях.

Стандарт WSDL разработан для веб-сервисов на основе протокола SOAP (Simple Object Access Protocol), предназначенного для обмена XML (eXtensible Markup Language) сообщениями. Однако в настоящее время множество веб-сервисов по обработке и анализу научных данных используют другие технологии реализации интерфейсов, в частности интерфейсы на основе REST (Representational State Transfer) архитектуры.

В публикации И. В. Бычкова, Г. М. Ружникова и др. [6] предлагается подход к созданию каталога WPS-сервисов, обеспечивающего функции поиска сервисов и возможность предварительной верификации входных данных перед использованием сервиса. Каталог WPS-сервисов реализован в виде программного модуля для системы управления контентом, используемой в среде. Данный подход предлагает каталогизацию локальных и внешних сервисов на основе OGC WPS интерфейса. Использование сервисов на основе других интерфейсов взаимодействия в каталоге не предполагается.

Целью данной работы является создание подхода к организации системы каталогизации и мониторинга территориально распределенных вычислительно-аналитических узлов в рамках разрабатываемой вычислительно-аналитической среды по обработке и анализу геологических данных.

Общее описание вычислительно-аналитической среды для геологических исследований

В научных геологических исследованиях используются разнотипные данные: количественные, спутниковые, пространственные, текстовые, медиаданные и т. д. Для обработки каждого из перечисленных типов данных используются различные методы, реализованные в специализированных программных системах. Наличие необходимого набора внешних сервисов позволяет создать специализированную вычислительно-аналитическую среду для решения научных геологических задач, в том числе и крупных комплексных задач.

Нами разработана и реализуется вычислительно-аналитическая геологическая среда [1]. Среда построена с использованием принципов слабосвязанной архитектуры, где каждый элемент функционирует независимо друг от друга. Основные элементы среды (рис. 1):

- платформа обработки, обеспечивающая пользовательский интерфейс;
- система управления обработкой, обеспечивающая разграничение доступа и контролирующая очередь выполнения процессов обработки;
- система хранения данных, обеспечивающая временное хранение пользовательских данных и результатов обработки;
- территориально распределенные узлы обработки – поставщики сервисов по обработке и анализу данных.

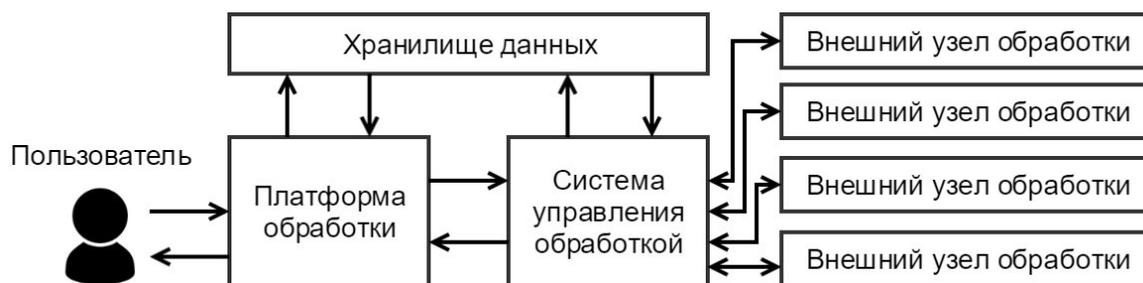


Рис. 1. Схема вычислительно-аналитической среды

Fig. 1. Scheme of the Computational and Analytical Environment

Взаимодействие пользователя и внешних сервисов обработки происходит через единую точку входа, предоставляемую платформой обработки. Существует возможность взаимодействия с платформой обработки в режиме «система – система» через программный интерфейс приложения. Платформа позволяет загружать данные для обработки непосредственно с компьютера пользователя либо по ссылке с внешнего ресурса.

В настоящее время вычислительно-аналитическая геологическая среда включает в себя следующие узлы обработки.

- **Многомерные методы анализа данных.** Включает в себя набор методов для многомерного анализа количественных данных, таких как факторный анализ, кластерный анализ, регрессионный анализ и др. В качестве компонента для реализации модуля статистического анализа количественных данных был выбран язык программирования R [8]. Интерфейс взаимодействия с сервисами построен с использованием модуля Rserve. Узел разработан и поддерживается в Государственном геологическом музее им. В. И. Вернадского РАН.

- **Обработка спутниковых данных.** Включает в себя методы первичной обработки спутниковых данных, такие как калибровка и пространственная привязка данных некоторых спутников [9]. Узел разработан и поддерживается в Институте автоматизации и процессов управления ДВО РАН.

- **Обработка петролого-геохимических данных.** В Институте физики Земли РАН разработана интерактивная база методов обработки петролого-геохимических данных [10]. Эта система предоставляет сервисы построения спайдерграмм, гистограмм и классификационных диаграмм; сервис идентификации минералов по их химическому составу; сервис интерпретации состава минерала и разложение на минералы и т. д. Интерфейс взаимодействия с сервисами построен на основе REST архитектуры.

- **Структурный анализ публикаций.** В междисциплинарном центре математического и вычислительного моделирования (Университет Варшавы, Польша) разработан сервис для извлечения метаданных из научных публикаций [11]. Метаданные включают в себя авторов, аффилиацию, аннотацию, ключевые слова, название журнала, объем, год выпуска, разобранные библиографические ссылки, структуру разделов документа, заголовки разделов и абзацы. Интерфейс взаимодействия с сервисами построен на основе REST архитектуры.

- **Обработка естественного языка.** В Университете Шеффилда в рамках проекта GATE (General Architecture for Text Engineering) разработан ряд сервисов по обработке текстовых данных для различных языков [12]. Для обработки текстовых данных на русском языке предоставляются сервисы по определению частей речи слов, а также выделению именованных существностей, таких как имена и фамилии, названия организаций, географические названия, даты, денежные единицы и т. д. Интерфейс взаимодействия с сервисами построен на основе REST архитектуры.

Предполагается, что создаваемая вычислительно-аналитическая геологическая среда должна обеспечить пользователям доступ к хранилищам современных наукоемких алгоритмов и вычислительным ресурсам, необходимым для оперативной обработки больших массивов разнородных данных (Big Data).

Используя внешние сервисы обработки, необходимо учитывать возможность временного ограничения доступа к определенным сервисам из-за различных технических проблем со стороны владельцев сервисов или линий передачи данных, а также возможность изменения интерфейса доступа к сервисам со стороны поставщиков сервисов. Для этого необходимо организовать систему мониторинга работы территориально распределенных сервисов, позволяющую оперативно реагировать на возможные технические проблемы в автоматическом или ручном режиме.

Организация вызова внешних сервисов обработки и анализа

В рассматриваемой вычислительно-аналитической среде для обеспечения единого подхода к вызову территориально распределенных сервисов обработки и анализа данных используется стандарт OGC WPS в качестве промежуточного интерфейса. Данный интерфейс разработан в качестве стандарта для веб-сервиса по выполнению процедур обработки данных. В первую очередь этот стандарт разрабатывался для процедур обработки пространственных данных, однако он имеет универсальный интерфейс для использования процедур обработки других видов данных. Использование стандарта OGC WPS для вызова сервисов обработки и анализа данных позволяет организовать работу разнородных сервисов в рамках единой вычислительно-аналитической среды.

Каждый метод для обработки и анализа данных в среде представлен в виде отдельного процесса WPS. Такой подход позволяет организовать единый интерфейс вызова внешних разнородных сервисов. Система управления обработкой позволяет контролировать распределение ресурсов между процессами, а также ограничивать доступ к различным процессам на основе прав пользователя. Обработка некоторых типов геологических данных может занимать продолжительное время, поэтому такие методы обработки вызываются в асинхронном режиме с возможностью отслеживания пользователем статуса выполнения процесса на всём протяжении обработки. При решении комплексных задач существует возможность передачи результата одного выполняемого процесса в качестве входных данных для другого процесса. Поддерживается возможность параллельного выполнения нескольких процессов, результаты которых передаются на вход одному процессу, как только все требуемые для его запуска процессы успешно завершатся.

Используя интерфейс OGC WPS для вызова внешних сервисов обработки, можно также отслеживать статус выполнения WPS процесса, соответствующего определенному внешнему сервису обработки. После завершения выполнения каждого процесса в базу данных поступает отчет о выполнении процесса, включающий параметры запуска сервиса, время выполнения, статус и результат. Данная информация также может быть использована в разрабатываемой системе мониторинга для более детального отслеживания состояния работы конкретного сервиса.

Для добавления внешнего вычислительно-аналитического сервиса в среду достаточно реализовать возможность вызова данного сервиса в виде WPS процесса.

Каталогизация веб-сервисов

Для решения задачи каталогизации сервисов разработан единый формат для описания отдельного сервиса на каждом вычислительном узле. Описание сервиса содержит набор информации, позволяющий осуществлять мониторинг любого сервиса среды с учетом используемых протоколов взаимодействия и интерфейсов доступа.

На основе компонентов для описания веб-сервисов в UDDI реестрах нами предложены параметры описания веб-сервисов в создаваемой среде:

- Основная информация
 - Идентификатор
 - Название
 - Описание
 - Область применения
 - Ключевые слова
- Поставщик сервиса
 - Название организации
 - Контактное лицо
 - Контактный адрес
 - Контактный телефон
 - Веб-сайт
- Служебная информация
 - IP адрес (доменное имя)
 - Порт
 - Протокол
 - Версия протокола
 - Описание интерфейса доступа
 - Авторизация
 - Логин
 - Пароль
 - Ключ доступа
 - Точка доступа

Структура описания сервиса состоит из трех основных частей. В первой части содержится основная информация о сервисе, позволяющая осуществлять поиск сервисов по названию, описанию, ключевым словам и т. д. Вторая часть содержит описание поставщика сервиса, включая информацию для связи с поставщиком. В третьей части располагается служебная информация, которая необходима для непосредственного взаимодействия с сервисом.

Предложенную структуру описания предлагается использовать в качестве метаданных для веб-сервисов. На основе разработанной структуры метаданных предлагается создание каталога веб-сервисов, предоставляющего пользователям и сторонним информационным системам информацию о сервисах в стандартном виде (рис. 2).



Рис. 2. Информация о веб-сервисе со статистикой его доступности

Fig. 2. Web-Service Meta Information with Availability Statistics

Информация о сервисе также формируется в форматах XML и JSON (рис. 3).

```

<ns:information>
  <ns:identifier>a3b556ca-bc46-4766-977d-cc0775e2ad40</ns:identifier>
  <ns:title>Метод главных компонент</ns:title>
  <ns:abstract>Метод главных компонент – это один из способов понижения размерности, состоящий ...
</ns:abstract>
  <ns:subject>Professional, Scientific, and Technical Services</ns:subject>
  <ns:keywords>
    <xsd:string>Principal component analysis</xsd:string>
    <xsd:string>R language</xsd:string>
  </ns:keywords>
</ns:information>
<ns:serviceProvider>
  <ns:organization>Государственный Геологический Музей им. Вернадского РАН</ns:organization>
  <ns:contact>Платонов К. А., м.н.с ГИМ РАН</ns:contact>
  <ns:email>k.platonov@sgm.ru</ns:email>
  <ns:phoneNumber />
  <ns:website>sgm.ru</ns:website>
</ns:serviceProvider>
<ns:serviceInformation>
  <ns:address>212.92.98.119</ns:address>
  <ns:port>6311</ns:port>
  <ns:protocol>rserve</ns:protocol>
  <ns:protocolVersion>1.8</ns:protocolVersion>
  <ns:protocolDescription>https://www.rforge.net/Rserve/dev.html</ns:protocolDescription>
  <ns:endpoint />
  <ns:authentication />
</ns:serviceInformation>

```

Рис. 3. Информация о веб-сервисе в формате XML
 Fir. 3. Web-Service Meta Information in XML Format

Основные параметры мониторинга сервисов в рамках вычислительно-аналитической среды

Для обеспечения высокого уровня надежности работы сервисов в рамках вычислительно-аналитической среды разработана система мониторинга, позволяющая оперативно реагировать на изменения в работе сервисов. Использование разнородных сервисов, взаимодействие с которыми осуществляется с помощью различных протоколов и по различным интерфейсам, подразумевает ряд ограничений на предмет мониторинга. Таким образом, имея общую техническую информацию о каждом сервисе (веб-адрес сервиса, протокол, версия протокола и т. д.), можно реализовать следующие общие виды проверок:

- а) проверка доступности удаленного узла;
- б) проверка работоспособности сервиса на удаленном узле по требуемому протоколу взаимодействия;
- в) проверка наличия изменений в работе сервиса на основе тестовых запросов к WPS процессам.

Более сложные виды проверок состояния сервисов требуют знания детального описания интерфейса взаимодействия, что делает такой вид проверок зависимым от конкретной реализации сервиса.

Используя описанные методы проверки состояния сервисов, можно формировать статистику доступности отдельных сервисов. В случае проблем с доступом к определенному сервису можно предлагать пользователям альтернативные реализации подобного сервиса при наличии их в среде.

Описание технологии мониторинга

Мониторинг территориально распределенных узлов вычислительно-аналитической среды осуществляется с использованием вышеописанных параметров мониторинга.

Проверка доступности узла (а) производится раз в минуту путем отправки запроса по указанному адресу узла по протоколу ICMP (Internet Control Message Protocol) или TCP (Transmission Control Protocol) по определенному порту в случае блокировки ICMP запросов со стороны удаленного узла.

Проверка доступности сервиса (б) производится раз в минуту путем отправки запроса, подтверждающего возможность установления соединения с сервисом, с использованием указанных адреса, порта и протокола сервиса.

Проверка сервиса на изменения (в) производится каждые 30 минут путем отправки одного или нескольких тестовых запросов к сервису и сравнения результатов работы сервиса с имеющимися в системе мониторинга. Она позволяет выявить изменения в параметрах вызова сервиса и в результатах работы сервиса.

В случае отсутствия доступа к узлу или сервису система уведомляет администратора вычислительно-аналитической среды и помечает сервис недоступным для использования.

Периодичность проверки каждого параметра мониторинга выбрана исходя из количества времени и вычислительных мощностей, необходимых для соответствующего типа проверки; достаточной оперативности для принятия соответствующего решения со стороны системы мониторинга; результатов работы системы мониторинга в тестовый период.

Количественные оценки для предложенного подхода мониторинга

Результаты работы системы мониторинга в период с 5 апреля 2019 г. по 6 мая 2019 г.:

Вычислительный узел и сервисы	Владелец узла	Доступность, %		Изменение сервисов (в), %
		узла (а)	сервисов (б)	
Многомерные методы анализа данных: 1) описательная статистика 2) регрессионный анализ 3) метод главных компонент 4) тройная диаграмма 5) иерархический кластерный анализ 6) метод k -средних 7) факторный анализ 8) линейный дискриминантный анализ 9) анализ канонических переменных	ГГМ РАН	100	98,873	0
Обработка петролого-геохимических данных: 1) построение классификационных диаграмм 2) система идентификации минералов 3) минералогические пересчеты 4) интерпретация состава минерала и разложение на миналы 5) оценка РТ условий с помощью геосенсоров	ИФЗ РАН	99,952	99,952	0
Обработка спутниковых данных (узел в разработке): 1) калибровка 2) пространственная привязка 3) атмосферная коррекция	ИАПУ ДВО РАН	0	0	0
Обработка естественного языка: 1) распознавание понятий в тексте 2) распознавание частей речи	Университет Шеффилда	99,963	99,963	0
Структурный анализ публикаций: 1) извлечение метаданных из научных статей	Варшавский Университет	100	100	0

Используя полученные результаты, можно сделать вывод, что все территориально распределенные узлы, за исключением узла обработки спутниковых данных, находящего в раз-

работке на текущий момент, показывают высокий уровень надежности для дальнейшего их использования в качестве источника сервисов обработки и анализа данных.

Вычислительно-аналитическая среда по обработке и анализу данных для научных исследований в геологии (<http://service.geologyscience.ru>) разрабатывается в рамках работ Государственного геологического музея им. В. И. Вернадского РАН (ГГМ РАН) по созданию информационно-аналитической геологической среды (<http://geologyscience.ru>).

Заключение

Авторами предложен подход к организации системы каталогизации и мониторинга территориально распределенных вычислительных узлов для вычислительно-аналитической среды обработки и анализа геологических данных. Предложенный подход реализован в виде программного модуля, встроеного в вычислительно-аналитическую среду.

Предложенные принципы организации каталогизации и мониторинга веб-сервисов позволяют обеспечить более высокий уровень надежности работы вычислительно-аналитической геологической среды, предоставляя пользователю информацию о надежности работы конкретного веб-сервиса в рамках среды. На основе данных подходов разработана программная реализация каталога веб-сервисов и системы мониторинга в виде работающего прототипа на основе существующей вычислительно-аналитической геологической среды.

Список литературы / References

1. **Eremenko V. S., Naumova V. V., Platonov K. A., Dyakov S. E., Eremenko A. S.** The main components of a distributed computational and analytical environment for the scientific study of geological systems. *Russian Journal of Earth Sciences*, 2018, vol. 18, iss. 6. DOI 10.2205/2018ES000636.
2. **Толпин В. А., Балашов И. В., Ефремов В. Ю., Лупян Е. А., Прошин А. А., Уваров И. А., Флитман Е. В.** Создание интерфейсов для работы с данными современных систем дистанционного мониторинга (система GEOSMIS) // Современные проблемы дистанционного зондирования Земли из космоса. 2011. Т. 8, № 3. С. 93–108.
Tolpin V. A., Balashov I. V., Efremov V. Yu., Lupyan E. A., Proshin A. A., Uvarov I. A., Flitman E. V. Sozdanie interfejsov dlya raboty s dannymi sovremennykh sistem distantsionnogo monitoringa (sistema GEOSMIS) [Creation of interfaces for working with data of modern remote monitoring systems (GEOSMIS system)]. *Modern problems of Earth remote sensing from space*, 2011, vol. 8, no. 3, p. 93–108. (in Russ.)
3. **Федотов А. М., Баракнин В. Б., Гуськов А. Е., Молородов Ю. И.** Распределенная информационно-аналитическая среда для исследований экологических систем // Вычислительные технологии. 2006. Т. 11. Спец. вып. С. 113–125.
Fedotov A. M., Barakhnin V. B., Guskov A. E., Molorodov Yu. I. Raspredeleonnaya informatsionno-analiticheskaya sreda dlya issledovaniy ekologicheskikh sistem [Distributed information and analytical environment for environmental systems research]. *Computational Technologies*, 2006, vol. 11, special issue, p. 113–125. (in Russ.)
4. **Gordov E. P., Krupchatnikov V. N., Okladnikov I. G., Fazliev A. Z.** Thematic virtual research environment for analysis, evaluation and prediction of global climate change impacts on the regional environment. In: Proc. SPIE 10035, 22nd International Symposium on Atmospheric and Ocean Optics: Atmospheric Physics, 100356J (29 November 2016). DOI 10.1117/12.2249118.
5. **Candela L., Castelli D., Pagano P.** Virtual Research Environments: An Overview and a Research Agenda. *Data Science Journal*, 2013, vol. 12, p. GRDI75–GRDI81. DOI 10.2481/dsj.GRDI-013

6. **Бычков И. В., Ружников Г. М., Фёдоров Р. К., Шумилов А. С.** Компоненты среды WPS-сервисов обработки геоданных // Вестник НГУ. Серия: Информационные технологии. 2014. Т. 12, № 3. С. 16–24.
Buchkov I. V., Ruzhnikov G. M., Fedorov R. K., Shumilov A. S. Components of WPS Environment for Geoprocessing. *Vestnik NSU. Series: Information Technologies*, 2014, vol. 12, no. 3, p. 16–24. (in Russ.)
7. **Бычков И. В., Ружников Г. М., Фёдоров Р. К., Шумилов А. С.** Система планирования и выполнения композиций веб-сервисов в гетерогенной динамической среде // Вычислительные технологии. 2016. Т. 21, № 6. С. 18–35.
Buchkov I. V., Ruzhnikov G. M., Fedorov R. K., Shumilov A. S. System for dynamic execution of composition services in the heterogeneous environment. *Computational Technologies*, 2016, vol. 21, no. 6, p. 18–35. (in Russ.)
8. **Platonov K. A.** Methods and Technologies for Integration and Processing of Territorially Distributed Quantitative Geological Information. In: Proceedings of the XX International Conference “Data Analytics and Management in Data Intensive Domains” (DAMDID/RCDL’2018), Moscow, 2018, p. 348–353.
9. **Dyakov S. E.** Cross-calibration channels of ir-radiometers and sea surface temperature retrieval. In: Proceedings of the IV International Conference “Modern Information Technologies in Earth Sciences”. Yuzhno-Sakhalinsk, 2016, p. 52.
10. **Иванов С. Д.** Интерактивный реестр геосенсоров на основе веб-приложения // Компьютерные исследования и моделирование, 2016, Т. 8, № 4, С. 621–632.
Ivanov S. D. Interaktivnyj reestr geosensorov na osnove veb-prilozheniya [Interactive Web Application Based Geosensors Registry]. *Computer Research and Modeling*, 2016, vol. 8, no. 4, p. 621–632. (in Russ.)
11. **Tkaczyk D., Szostek P., Fedoryszak M., Dendek P., Bolikowski L.** CERMINE: automatic extraction of structured metadata from scientific literature. *International Journal on Document Analysis and Recognition*, 2015, vol. 18, no. 4, p. 317–335. DOI 10.1007/s10032-015-0249-8.
12. **Maynard D., Bontcheva K., Augenstein I.** Synthesis Lectures on the Semantic Web: Theory and Technology. In: Natural Language Processing for the Semantic Web, December 2016, vol. 6, no. 2, p. 1–194.

*Материал поступил в редколлегию
Received
29.03.2019*

Сведения об авторах / Information about the Authors

Ерёменко Виталий Сергеевич, младший научный сотрудник, Государственный геологический музей им. В. И. Вернадского РАН (ул. Моховая, д. 11, стр. 11, Москва, 125009, Россия)

Vitaliy S. Eremenko, Junior Researcher, V. I. Vernadsky State Geological Museum of the Russian Academy of Sciences (11 Mokhovaya Str., building 11, Moscow, 125009, Russian Federation)
vitaer@gmail.com

Наумова Вера Викторовна, доктор геолого-минералогических наук, главный научный сотрудник, Государственный геологический музей им. В. И. Вернадского РАН (ул. Моховая, д. 11, стр. 11, Москва, 125009, Россия)

Vera V. Naumova, Doctor of Science (Geology and Mineralogy), Senior Scientist, V. I. Vernadsky State Geological Museum of the Russian Academy of Sciences (11 Mokhovaya Str., building 11, Moscow, 125009, Russian Federation)
v.naumova@sgm.ru

Автоматический рефакторинг Java-кода с использованием Stream API

Р. А. Иванов¹, Т. Ф. Валеев²

¹ *Новосибирский государственный университет
Новосибирск, Россия*

² *Институт систем информатики им. А. П. Ершова СО РАН
Новосибирск, Россия*

Аннотация

Долгое время функциональное программирование на Java было невозможно. Однако в 8-й версии Java появились лямбда-выражения. Благодаря поддержке стандартных библиотечных классов (Stream, Optional и т. д.) на Java стало возможно описывать преобразования над данными в функциональном стиле.

Java – достаточно старый язык, на нем написано большое количество императивного кода. Для того чтобы воспользоваться преимуществами нового подхода, требуется выполнить нетривиальный рефакторинг, что в случае осуществления человеком может быть весьма утомительным, легко совершить ошибку. К счастью, для достаточно большого количества ситуаций данный рефакторинг можно безопасно осуществить автоматически.

На основе IntelliJ Idea был разработан программный инструмент, который позволяет обнаружить места, где возможно автоматическое преобразование императивного кода в эквивалентный с использованием Stream API, а также автоматическое исправление, которое позволяет произвести замену. Рефакторинг пользуется средствами IntelliJ Idea для анализа Java-кода, а также интегрируется в саму IDE. Одним из основных критериев корректности работы алгоритма является безопасность данного преобразования. Пользователь не может доверять инструменту, если преобразование может изменять семантику кода. В данной статье рассматриваются различные ограничения, которые накладываются на шаблоны кода для того, чтобы преобразование без искажения семантики было возможно.

Данный рефакторинг был протестирован на различных библиотеках для проверки сохранения семантики путем проверки результатов тестирования до и после применения рефакторинга.

В статье не будет обсуждаться влияние использования Stream API на производительность приложения.

Ключевые слова

Stream API, refactoring, рефакторинг, преобразование кода, IntelliJ, IDE

Для цитирования

Иванов Р. А., Валеев Т. Ф. Автоматический рефакторинг Java-кода с использованием Stream API // Вестник НГУ. Серия: Информационные технологии. 2019. Т. 17, № 2. С. 49–60. DOI 10.25205/1818-7900-2019-17-2-49-60

Automatic Refactoring of Java Code Using the Stream API

R. A. Ivanov¹, T. F. Valeev²

¹ *Novosibirsk State University
Novosibirsk, Russian Federation*

² *A. P. Ershov Institute of Informatics Systems SB RAS
Novosibirsk, Russian Federation*

Abstract

For a long time, functional Java programming was not possible. However, lambda expressions appeared in version 8 of the Java language. Due to the support of standard library classes (Stream, Optional, etc.) in Java, it became possible to describe transformations over data in a functional style.

Java is a rather old language with a large imperative code base written in it. In order to take advantage of the new approach, it is necessary to perform a non-trivial refactoring, which can be very tedious and error prone when applied manually.

Fortunately, for a sufficiently large number of situations, this refactoring can be safely performed automatically.

Using IntelliJ Idea as a platform software tool was developed that allows you to find places where it is possible to automatically convert an imperative code to an equivalent code that uses Stream API, as well as an automatic fix that allows you to make a replacement. Refactoring uses IntelliJ Idea framework to analyze Java code, and integrates into the IDE itself.

One of the main criteria for the correct operation of the algorithm is the security of this transformation. The user cannot trust the tool if the transformation can change the semantics of the code.

This article discusses various constraints that are imposed on code patterns so that transformation without distortion of semantics is possible.

Refactoring has been tested in various libraries to verify the semantics are preserved by checking the test results before and after applying refactoring.

This article will not describe the impact of using the Stream API on the performance of the application.

Keywords

refactoring, IDE, code transformation, IntelliJ, Stream API

For citation

Ivanov R. A., Valeev T. F. Automatic Refactoring of Java Code Using the Stream API. *Vestnik NSU. Series: Information Technologies*, 2019, vol. 17, no. 2, p. 49–60. (in Russ.) DOI 10.25205/1818-7900-2019-17-2-49-60

Введение

Целью данной работы было создание инструмента для автоматического преобразования кода на Java, написанного в императивном стиле, в код на функциональном стиле с использованием цепочки Stream API, а также интеграция этого инструмента в среду разработки IntelliJ Idea.

Использование Stream API вместо традиционных циклов имеет весьма значительные преимущества: возможна автоматическая параллелизация, которая может дать прирост производительности, часто такой код намного более короткий и лаконичный, а также более четко выражает намерение автора¹.

Для среды разработки крайне важно предлагать различные варианты преобразования кода, так как каждый из них может обладать своими преимуществами и недостатками, и программист может сам решать, какой из них лучше в данной ситуации.

Благодаря инфраструктуре IDE данные автоматические преобразования могут производиться в одно нажатие кнопки.

Несмотря на то, что в IDE NetBeans данный рефакторинг уже присутствует, в нем было обнаружено значительное количество недостатков и возможностей улучшения, которые учтены при разработке. Данный рефакторинг было решено разрабатывать для IDE IntelliJ Idea, поскольку она является самой распространенной IDE для Java. Использовать разработки других IDE не представлялось возможным из-за больших различий в семантической модели Java, а также из-за того, что модель рефакторинга там значительно более проста.

Операции Stream API

Stream API² предоставляет возможность обработки совокупности однотипных данных. Сценарий обработки записывается декларативно посредством цепочки вызовов. Типичный пример использования Stream API выглядит следующим образом:

¹ Rahad K., Cao Z., Cheon Y. A Thought on Refactoring Java Loops Using Java 8 Streams, 2017.

² <https://docs.oracle.com/javase/8/docs/api/java/util/stream/package-summary.html>

```
List<String> result = listOfStrings.stream()
    .map(s -> s.trim())
    .filter(s -> !s.isEmpty())
    .limit(10)
    .collect(Collectors.toList());
```

Здесь у каждой строки из списка `listOfStrings` удаляются начальные и конечные пробелы с помощью операции `map`; затем пустые строки отфильтровываются с помощью `filter`; после этого результат ограничивается десятью строками с помощью `limit`, и, наконец, выполняется сборка в результирующий список с помощью `collect`. Здесь имеются операции различных типов.

- Источник потока – метод `stream` (входит в интерфейс `List`), создает поток из списка. Имеются другие виды источников, которые, например, создают поток из массива, из строк файла, из целых чисел в заданном диапазоне и т. д. Источник возвращает объект типа `Stream`.

- Промежуточные (intermediate) операции – `map`, `filter` и `limit`. Эти операции также возвращают объект типа `Stream`, позволяя объединять в цепочку несколько промежуточных операций. Важная особенность заключается в том, что промежуточные операции всегда «ленивы»: они не производят вычислений, а только запоминают действия, которые необходимо выполнить. Имеются и другие промежуточные операции: `distinct` (удалить повторяющиеся элементы), `skip` (пропустить заданное количество первых элементов) и т. д.

- Концевая (terminal) операция – `collect`. Именно при выполнении концевой операции производится вся цепочка вычислений и сбор результатов. Операция `collect` принимает так называемый коллектор – объект, специфицирующий способ объединения результатов. В данном случае используется коллектор `toList`, который собирает результаты в список. Другие коллекторы могут собирать в множество, ассоциативный массив, в строку с заданным разделителем и т. д. Помимо `collect` имеются и другие концевые операции. Например, `count` просто подсчитывает количество результатов, а `anyMatch` проверяет, имеется ли хоть один результат, удовлетворяющий заданному условию.

Промежуточные операции делятся на операции без состояния (`map` и `filter`) и операции с состоянием (`limit` или `distinct`). Операции без состояния значительно проще по своей сути: они единообразно обрабатывают каждый элемент, при этом порядок их обработки неважен. Обычно такие операции легко выражаются в традиционном цикле. Например, операции `map` соответствует присваивание или объявление переменной, а операции `filter` – условный оператор `if`. Операции с состоянием сложнее: состояние необходимо хранить в дополнительной переменной либо получать неявно. В традиционных циклах соответствующая семантика выражается разными способами, и их поддержка в процессе преобразования цикла в цепочку `Stream API` представляет определенную трудность. Ниже приводится два способа выразить тот же алгоритм с помощью традиционного цикла:

```
List<String> result = new ArrayList<>();
int count = 0;
for (String s : listOfStrings) {
    s = s.trim();
    if (!s.isEmpty()) {
        result.add(s);
        if (++count == 10) break;
    }
}

List<String> result = new ArrayList<>();
for (String s : listOfStrings) {
    s = s.trim();
    if (!s.isEmpty()) {
        result.add(s);
        if (result.size() == 10) break;
    }
}
```

В первом варианте для подсчета количества добавленных элементов используется дополнительная переменная-счетчик `count`. Во втором способе мы пользуемся размером результирующего списка, который изначально пуст, и на каждом шаге туда добавляется ровно одна запись. Второй способ короче, но менее универсален. К примеру, он не сработает, если результаты требуется собрать не в список, а в множество или в строку с разделителем. Оба приведенных варианта IntelliJ Idea поддерживает, но возможны и другие варианты: некоторые предпочитают проверять счетчик перед добавлением элемента, считать в обратном порядке (от 10 до 0) и т. д.

У операции `limit` есть другая важная особенность: это так называемая короткозамкнутая операция. Она может завершить весь процесс до того, как закончатся входные данные. Короткозамкнутыми могут быть как промежуточные, так и конечные операции. Пример конечной короткозамкнутой операции – `anyMatch`. Эта операция возвращает булеву «истину», если хоть один элемент потока удовлетворяет условию, переданному в виде лямбда-выражения параметром `anyMatch`. При этом, естественно, обработка прекращается.

Поток элементов объектного типа T имеет тип `java.util.stream.Stream<T>`. Так как в языке Java примитивный тип не может являться аргументом обобщенного типа, невозможно создать, к примеру, поток элементов типа `float`. Однако для удобства и повышенного быстродействия три примитивных типа – `int`, `long` и `double` – поддерживаются специально: для них имеются потоковые типы `IntStream`, `LongStream` и `DoubleStream`. Далее под «поддерживаемым типом» мы будем иметь в виду любой тип, который может быть элементом потока: либо любой объектный тип, либо `int`, `long` или `double`. Другие примитивные типы (`boolean`, `short` и т. д.) назовем неподдерживаемыми типами. Также для простоты под `Stream<T>` мы будем иметь в виду поток элементов любого поддерживаемого типа (если T соответствует типу `int`, то вместо `Stream<int>` подразумевается `IntStream`).

Преобразование фрагментов кода в лямбда-выражение

Будем называть фрагментом Java-выражение (`expression` глава 15 JLS), Java-утверждение (`statement`, JLS §14.5) либо совокупность из нескольких подряд идущих Java-утверждений в пределах одного блока (JLS §14.2). Возможен также пустой фрагмент, не содержащий никаких утверждений и выражений. Фрагмент можно преобразовать в тело лямбда-выражения для потоковой операции при соблюдении трех условий.

У1. Фрагмент не должен ссылаться на локальные переменные и параметры, которые определены за пределами этого фрагмента и не являются финальными и эффективно-финальными.

У2. Фрагмент не должен бросать проверяемых исключений.

У3. Фрагмент не должен содержать операторов, передающих управление за пределы данного фрагмента за исключением оператора `throw`.

Условие У1 необходимо для любого лямбда-выражения Java в соответствии с § 15.27.2 спецификации языка Java 8 (далее JLS)³. Понятие финальной и эффективно-финальной переменной приводится в JLS § 4.12.4. Основное требование заключается в том, чтобы значение присваивалось переменной ровно один раз на каждом пути управления.

Условие У2 не требуется для любых лямбда-выражений. Лямбда-выражение может бросать проверяемые исключения, если соответствующий абстрактный метод функционального интерфейса (JLS § 9.8) их объявляет. Однако ни один из функциональных интерфейсов, используемых в Stream API (к примеру, `java.util.function.Predicate`), не объявляет проверяемых исключений.

Условие У3 также требуется для любых лямбда-выражений в соответствии с их семантикой в языке Java в отличие, например, от языка Kotlin, где в ряде случаев возможна нелокальная передача управления из лямбда-выражения. Оператор `throw`, бросающий исключе-

³ <https://docs.oracle.com/javase/specs/>

ние, допустим, так как, по спецификации Stream API, любое исключение, выброшенное из лямбда-выражения, используемого в цепочке операций, будет выброшено наружу. Разумеется, в соответствии с У2, допустимы только непроверяемые исключения.

Условие У2 можно проверить для всего цикла: если в цикле есть хоть одна операция, бросающая проверяемое исключение, то преобразование цикла в Stream API невозможно. С другой стороны, условие У1 для всего цикла проверять неоправданно. Во-первых, изменение переменной может не попасть в лямбда-выражения, а стать результатом цепочки вызовов Stream API, как, к примеру, в следующем листинге:

```
static long countNonEmptyStrings(List<String> strings) {
    long count = 0;
    for(String s : strings) {
        if(!s.isEmpty()) {
            count++;
        }
    }
    return count;
}
```

Здесь переменная count не является эффективно-финальной (модифицируется в цикле) и объявлена за пределами цикла. Однако этот код легко трансформировать с использованием Stream API:

```
static long countNonEmptyStrings(List<String> strings) {
    long count = strings.stream().filter(s -> !s.isEmpty()).count();
    return count;
}
```

Поэтому каждый фрагмент, который необходимо превратить в лямбда-выражение, следует проверять отдельно. В данном примере лямбда-выражением становится только условие !s.isEmpty(), для которого У1 выполняется.

Также возможна ситуация, когда переменная, не являющаяся эффективно-финальной в исходном цикле, станет таковой после преобразования. Тогда преобразование все еще возможно. К примеру, рассмотрим такой цикл:

```
for(int i=0; i<10; i++) {
    System.out.println(i);
}
```

Здесь требуется преобразовать в лямбда-выражение вызов метода System.out.println(i), который ссылается на изменяемую переменную i, определенную за пределами выражения. Однако после преобразования цикла i становится неизменяемой:

```
IntStream.range(0, 10).forEach(i -> System.out.println(i));
```

Условие У3 также не стоит проверять для всего цикла сразу. Действительно, лямбда-выражение не может содержать операторов типа break (если только само не содержит цикл), а оператор return завершит только само лямбда-выражение, но не окружающий его метод. Однако многие циклы, содержащие операторы управления потоком, все-таки можно преобразовать в цепочку Stream API. Операторы break и return иногда можно трансформировать в короткозамкнутую операцию. Оператор continue, переходящий на следующую итерацию цикла, допустим в качестве тела условного оператора if. К примеру, следующие два фрагмента семантически эквивалентны:

```

List<String> result = new ArrayList<>();
for (String s : listOfStrings) {
    if (!s.isEmpty()) {
        result.add(s);
    }
}

List<String> result = new ArrayList<>();
for (String s : listOfStrings) {
    if (s.isEmpty()) continue;
    result.add(s);
}

```

Модель Stream API

Для обсуждения алгоритма преобразования циклов в вызовы Stream API нам потребуется несколько определений.

Для заданной переменной Java-кода V будем обозначать через T_V ее тип.

Подфрагментом F' фрагмента F назовем фрагмент, целиком содержащийся во фрагменте F (являющийся его частью) и не совпадающий с F . Обозначим $F' \subset F$.

Генератор $G(V, S, F)$ – это совокупность выходной переменной V поддерживаемого типа, Java-выражения S , имеющего тип $\text{Stream}<T_V>$, и фрагмента F .

Представление генератора $G(V, S, F)$ есть следующий Java-цикл:

```

for(Iterator<T> it = S.iterator(); it.hasNext(); ) {
    TV V = it.next();
    F
}

```

Имеются простые и составные генераторы. Простой генератор создается непосредственно по Java-циклу L таким образом, что представление этого генератора семантически эквивалентно исходному циклу L . Определим несколько видов простых генераторов, каждый из них соответствует определенному виду циклов (табл. 1).

Таблица 1

Определения генераторов

Table 1

Generator Definitions

Название	Вид цикла	Операция
Обход коллекции	for($T_V V : \text{EXPR}$) F EXPR имеет тип <code>java.util.Collection</code>	$G_{\text{coll}} = (V, \text{"(EXPR).stream()", } F)$
Обход массива	for($T_V V : \text{EXPR}$) F EXPR имеет тип массива	$G_{\text{arr}} = (V, \text{"Arrays.stream(EXPR)", } F)$
Диапазон чисел	for($T_V V = \text{START}; V < \text{BOUND}; V++$) F T_V – либо <code>int</code> , либо <code>long</code>	$G_{\text{range}} = (V, \text{"IntStream / LongStream.range(START, BOUND)", } F)$
Диапазон чисел (закрытый)	for($T_V V = \text{START}; V <= \text{BOUND}; V++$) F T_V – либо <code>int</code> , либо <code>long</code>	$G_{\text{closed}} = (V, \text{"IntStream / LongStream.rangeClosed(START, BOUND)", } F)$
Строки BufferedReader	<code>String V;</code> while($(V = \text{BR.readLine()}) \neq \text{null}$) F BR имеет тип <code>java.io.BufferedReader</code>	$G_{\text{br}} = (V, \text{"BR.lines()", } F)$

Преобразование $X(E)$ – функция от Java-выражения типа Stream, которая возвращает новое Java-выражение типа Stream (возможно, с другим типом элементов). В большинстве случаев преобразование добавляет новый вызов к существующей цепочке вызовов. К примеру, фильтрующее преобразование может выглядеть так:

$$\text{Filt}_{V, P}(E) := E.\text{filter}(V \rightarrow P),$$

где V – переменная, E – Java-выражение, имеющее тип $\text{Stream}\langle T_V \rangle$, а P – Java-выражение, представляющее собой предикат от переменной V (условие фильтрации) и удовлетворяющее условиям У1–У3. Результирующее выражение также имеет тип $\text{Stream}\langle T_V \rangle$.

Операция $O(V_{in}, F_{in})$ – это функция от входной переменной V_{in} и фрагмента, которая возвращает тройку (V_{out}, X, F_{out}) , состоящую из выходной переменной V_{out} , преобразования X и выходного фрагмента F_{out} , который является подфрагментом F_{in} . Операции определены только для некоторых фрагментов. К примеру, операция фильтрации определена для фрагментов вида $\text{if}(P) F$ и возвращает тройку $(V_{in}, \text{Filt}_{V_{in}, P}, F)$.

Всякая операция O обладает следующим свойством: для любого генератора $G(V, S, F)$, для которого $O(V, F)$ определена и возвращает тройку (V_{out}, X, F_{out}) , представление генератора G семантически эквивалентно представлению генератора $G'(V_{out}, X(S), F_{out})$. Генератор G' назовем составным генератором.

Мы определили следующие операции:

- *операция фильтрации* определена для фрагментов вида $\text{if}(P) F$. Возвращает тройку $(V_{in}, E \rightarrow E.\text{filter}(V_{in} \rightarrow P), F)$;
- *операция обратной фильтрации* определена для фрагментов вида $\text{if}(P) \text{continue}; F$. Возвращает тройку $(V_{in}, E \rightarrow E.\text{filter}(V_{in} \rightarrow !(P)), F)$;
- *операция объявления переменной* определена для фрагментов вида $T_V V = R; F$, где V – объявленная переменная поддерживаемого типа T_V , причем V_{in} используется только в выражении R и не используется в F . Возвращает тройку $(V, E \rightarrow E.\text{map}(V_{in} \rightarrow R), F)$. Если тип входной или выходной переменной примитивный, вместо map может использоваться другой метод Stream API (например, mapToInt , boxed , asLongStream и т. д.);
- *операция переписывания переменной* определена для фрагментов вида $V_{in} = R; F$. Возвращает тройку $(V_{in}, E \rightarrow E.\text{map}(V_{in} \rightarrow R), F)$;
- *операция вложенного цикла* определена для фрагментов, являющихся циклами, для которых существует простой генератор $G(V_{out}, S, F)$, причем V_{in} не используется в F . Возвращает тройку $(V_{out}, E \rightarrow E.\text{flatMap}(V_{in} \rightarrow S), F)$. Если тип входной или выходной переменной примитивный, могут использоваться конструкции вида $E.\text{flatMapToInt}(V_{in} \rightarrow S)$ или $E.\text{mapToObj}(V_{in} \rightarrow S).\text{flatMap}(\text{Function.identity}())$.

Расширенное преобразование назовем функцией $\text{EX}(F_b; E; F_a)$, которая возвращает новое Java выражение.

В отличие от преобразования, помимо непосредственно выражения типа Stream, принимает еще и 2 фрагмента, которые расположены непосредственно до внешнего цикла и после внешнего цикла.

Назовем терминалом функцию $T(G, F_b, F_l, F_a)$, где $G(V, S, F)$, которая для S возвращает расширенное преобразование X . Так же, как и операция, может быть определена не для всех фрагментов.

Определим следующие терминалы (табл. 2). В табл. 2 для краткости префикс $F_b; E; F_a \rightarrow$ был опущен. Данный список не является полным, рефакторинг содержит значительно больше вариантов преобразований, в табл. 2 приведены примеры из каждого класса терминалов.

Таблица 2

Определения терминалов

Table 2

Terminal Definitions

Название терминала	Требования к фрагментам	Расширенное преобразование
Нахождение первого элемента	Fb вида <code>Optional<T> x = Optional.empty();</code> F1 вида <code>if (P) result = R; break;</code>	<code>Optional<T> x = E.findFirst(V -> P);</code> Fa;
Редукция элементов	Fb вида <code>T a = I;</code> F1 вида <code>for (T' x: c) a # x,</code> где # – одна из операций {+, *, , &&, , &, ^}	<code>T a = I; A # = E.reduce(I, (a, b) -> a # b);</code> Fa, где I – значение идентичности (identity value) для данной операции, а переменные a и b – уникальные в данной области видимости
Нахождение минимального / максимального элемента	Fb вида <code>T a = I;</code> F1 вида <code>for (T' x: c) if (x # a) a = x,</code> где # обозначает > или <	<code>T a = op(I, E.max());</code> Fa, где op – max или min, в зависимости от #
Сохранение в список	Fb вида <code>List<T> l = new ArrayList<>()</code> F1 вида <code>for (T' x : c) l.add(x)</code>	<code>E.collect(Collectors.toList());</code> Fa
Склейка строк	Fb вида: <code>StringBuilder sb = new StringBuilder(prefix);</code> F1 вида: <code>boolean isTail = false;</code> <code>for (T e: collection)</code> <code>if (isTail) sb.append(delimiter);</code> <code>else isTail = true;</code> <code>sb.append(F);</code> Fa вида: <code>sb.append(suffix).toString();</code> где F содержит только переменную e или другие финальные переменные	<code>E.map(F).collect(Collectors.joining(delimiter, prefix, suffix))</code>
Применение операции для каждого элемента	Отсутствуют	Fb; <code>E.forEach(V -> F1);</code> Fa

Рассмотрим подробнее алгоритм преобразования императивного кода в функциональный. Далее приводится псевдокод данного алгоритма:

```
extractStream(Fb, Fl, Fa) {
    G = findGeneratorFor(Fl)
    if (G == null) return null
    return extractTail(G, Fb, Fa)
}

extractTail(G, Fb, Fa) {
    O = findOperationFor(G.F, G.V)
    (Vout, X, Fout) = O(G.v, G.F)
    G' = G(Vout, X(G.s), Fout)
    if (G' != null) {
        return extractTail(G', Fb, Fa)
    } else {
        return extractTerminal(G, Fc)(G, Fb, G.F, Fa)
    }
}
```

Полагаем, что Fb – фрагмент контекста до цикла, Fl – фрагмент цикла, Fa – фрагмента контекста после цикла, а G – базовый генератор для текущего шага рекурсии.

```
extractTerminal(G, Fb, Fa) {
    for (matcher in terminalMatchers) {
        T = matcher.match(G.F, G.V, Fb, Fa)
        if (O != null) return T
    }
    return null
}

findOperationFor(F, V) {
    for (matcher in operationMatchers) {
        O = matcher.match(F, V)
        if (O != null) return O
    }
    return null
}
```

Следует пояснить, что `operationMatchers` и `terminalMatchers` соответствуют описанию операций и терминалов, которые были рассмотрены ранее.

Нетрудно заметить, что подход основан на преобразовании некоторых шаблонов кода, из которых и строится цикл. В статьях [1–3] были исследованы различные шаблоны кода в открытых кодовых базах, большая их часть была поддержана в данном рефакторинге.

Сравнение с конкурирующими реализациями

Данная реализация рефакторинга не была первой, аналоги есть в других IDE для Java: Eclipse и NetBeans. Далее мы приведем их сравнение с рефакторингом в IntelliJ Idea.

LambdiFicator

Аналогичная возможность реализована в рамках проекта `LambdiFicator` [4; 5] для IDE NetBeans 8.1. По сравнению с разработкой, представленной в данной статье, эта реализация обладает рядом недостатков.

1. Из промежуточных операций поддерживаются только операции `filter` и `map`. Операции `flatMap`, `distinct`, `limit`, `sorted` не поддерживаются ни в каком виде.

2. Из конечных операций поддерживаются только `reduce`, `forEach`, `anyMatch` и `noneMatch`. Не поддерживаются такие важные способы завершения потока, как `collect(toList()); collect(joining()); toArray(); findFirst()`.

3. Примитивные типы отождествляются с соответствующими им объектными типами (к примеру, тип `int` с типом `java.lang.Integer`), в то время как Stream API обеспечивает специальную поддержку примитивных типов `int`, `long`, `double` с помощью интерфейсов `IntStream`, `LongStream` и `DoubleStream` соответственно. Такое отождествление не только ухудшает производительность результирующего кода, но и в некоторых случаях может опасным образом изменить семантику кода. Рассмотрим, к примеру, следующий листинг:

```
import java.util.*;

public class Test {
    public static void main(String[] args) {
        List<Integer> list = new ArrayList<>(Arrays.asList(1, 2, 3));
        List<Integer> input = Arrays.asList(0, 1);
        for (int idx : input) {
            if (idx > 0) {
                list.remove(idx);
            }
        }
        System.out.println(list);
    }
}
```

Данная Java-программа при выполнении выдает `[1, 3]`, так как в цикле вызывается метод удаления элемента списка по индексу: `remove(int)`. IDE NetBeans заменяет цикл следующим образом:

```
input.stream().filter((idx) -> (idx > 0)).forEachOrdered((idx) -> {
    list.remove(idx);
});
```

При этом тип переменной `idx` меняется на `java.lang.Integer`, и уже вызывается метод удаления элемента списка, эквивалентного заданному: `remove(Integer)`. В результате изменяется семантика программы: при выполнении она выдает `[2, 3]`. Для сравнения, IntelliJ Idea предлагает заменить тот же самый цикл следующим образом:

```
input.stream().mapToInt(idx -> idx)
    .filter(idx -> idx > 0).forEach(list::remove);
```

В данном случае благодаря применению операции `mapToInt Stream<Integer>` превращается в `IntStream`, и изменения семантики не происходит.

Eclipse plugin

В IDE Eclipse данная функциональность планировалась к реализации с использованием плагина `Convert-For-Each-Loop-to-Lambda-Expression-Eclipse-Plugin` [6]. Плагин не был доведен до рабочего состояния (на момент 19 декабря 2018 г.), даже базовая функциональность не поддерживается.

Доступность разработки

Код проекта может быть найден на GitHub в репозитории IntelliJ Idea ⁴.

Точка входа – класс `StreamApiMigrationInspection`, он предназначен для обнаружения шаблонов кода, подходящих для преобразования. После того как подходящий кусок кода найден, будет предложен вариант миграции – объект, класс которого `BaseStreamApiMigration`.

Непосредственно преобразованием кода занимается класс `MigrateToStreamFix`. Здесь же находятся классы `StreamSource` и `Operation`, наследники которых соответствуют определению простого генератора и операции соответственно из статьи.

Классу `TerminalBlock` напрямую не соответствует терминал из модели, он обозначает фрагмент, который еще не был обработан и, возможно, содержит терминал.

Данный автоматический рефакторинг активирован только для версий Java больше 7.

Заключение

В данной статье была выстроена модель Stream API, на основании нее реализован рефакторинг для преобразования циклов в цепочки вызовов Stream API и внедрен в платформу IntelliJ Idea. Сейчас это преобразование доступно к использованию у миллионов пользователей IDE. Данный рефакторинг успешно распознает большую часть распространенных способов преобразования данных. Корректность обеспечивается сотнями интеграционных тестов и проверками на реальном программном обеспечении, в частности самом коде IntelliJ Idea.

Список литературы / References

1. **Allamanis M., Barr E. T., Bird C., Devanbu P., Marron M., Sutton C.** Mining Semantic Loop Idioms. *IEEE Transactions on Software Engineering*, Jul. 2018, vol. 44, no. 7, p. 651–668. DOI 10.1109/TSE.2018.2832048
2. **Barua A., Cheon Y.** A Systematic Derivation of Loop Specifications Using Patterns, Technical Report 15-90, Department of Computer Science, University of Texas at El Paso, El Paso, TX, December 2015.
3. **Barua A., Cheon Y.** Finding Specifications of While Statements Using Patterns. *New Trends in Networking, Computing, E-learning, Systems Sciences, and Engineering*, Nov. 2014, p. 581–588. DOI 10.1109/FOSE.2007.27
4. **Franklin L., Gyori A., Lahoda J., Dig D.** LambdaFicator: From imperative to functional programming through automated refactoring. In: 35th International Conference on Software Engineering (ICSE), May 2013. DOI 10.1109/ICSE.2013.6606699
5. **Gyori A., Franklin L., Dig D., Lahoda J.** Crossing the gap from imperative to functional programming through refactoring. In: Proceedings of the 2013 9th Joint Meeting on Foundations of Software Engineering – ESEC/FSE, 2013. DOI 10.1145/2491411.2491461
6. **Arefin M., Khatchadourian R.** Porting the NetBeans Java 8 enhanced for loop lambda expression refactoring to eclipse. In: Companion Proceedings of the 2015 ACM SIGPLAN International Conference on Systems, Programming, Languages and Applications: Software for Humanity – SPLASH Companion, 2015. DOI 10.1145/2814189.2817277.

*Материал поступил в редколлегию
Received
04.03.2019*

⁴ <https://github.com/JetBrains/intellij-community/tree/master/java/java-impl/src/com/intellij/codeInspection/streamMigration>

Сведения об авторах / Information about the Authors

Иванов Роман Андреевич, магистрант факультета информационных технологий Новосибирского государственного университета (ул. Пирогова, 1, Новосибирск, 630090, Россия)

Roman A. Ivanov, Master's Student, Faculty of Information Technologies, Novosibirsk State University (1 Pirogov Str., Novosibirsk, 630090, Russian Federation)

roman.ivanov@jetbrains.com

Валеев Тагир Фаридович, кандидат физико-математических наук, старший научный сотрудник, Институт систем информатики имени А. П. Ершова СО РАН (пр. Академика Лаврентьева, 6, Новосибирск, 630090, Россия)

Tagir F. Valeev, Candidate of Physical and Mathematical Sciences, senior research associate of the A. P. Ershov Institute of Informatics Systems of the Siberian Branch of the Russian Academy of Sciences (6 Academician Lavrentiev Ave., Novosibirsk, 630090, Russian Federation)

amaembo@gmail.com

УДК 004.4, 37.018

DOI 10.25205/1818-7900-2019-17-2-61-73

Системы автоматизированной оценки заданий по программированию: разработка, использование и перспективы

Д. В. Иртегов¹, Т. В. Нестеренко^{1,2}, Т. Г. Чурина^{1,2}

¹ *Новосибирский государственный университет
Новосибирск, Россия*

² *Институт систем информатики им. А.П. Ершова СО РАН
Новосибирск, Россия*

Аннотация

Приводится обзор существующих систем проверки заданий по программированию, кратко описано современное состояние архитектуры системы NSUts, созданной в Новосибирском государственном университете, и практика ее использования в учебном процессе и в олимпиадах по программированию различных уровней. Приведен анализ опыта использования таких систем и обозначены перспективы их развития.

Ключевые слова

тестирование знаний, обучение программированию, олимпиады по программированию, системы тестирования, NSUts

Для цитирования

Иртегов Д. В., Нестеренко Т. В., Чурина Т. Г. Системы автоматизированной оценки заданий по программированию: разработка, использование и перспективы // Вестник НГУ. Серия: Информационные технологии. 2019. Т. 17, № 2. С. 61–73. DOI 10.25205/1818-7900-2019-17-2-61-73

Systems for Automated Evaluation of Programming Tasks: Development, Use and Perspectives

D. V. Irtegov¹, T. V. Nesterenko^{1,2}, T. G. Churina^{1,2}

¹ *Novosibirsk State University
Novosibirsk, Russian Federation*

² *A. P. Ershov Institute of Informatics Systems SB RAS
Novosibirsk, Russian Federation*

Abstract

The paper is devoted to the issues related to the development and use of automated assessment systems for programming tasks. The work on automated testing of programming tasks had been held in Novosibirsk State University since 1998. It was created several generations of systems of this type. The first program of about 1,000 lines was written in Perl. Currently, the automated system, named NSUts, is used in the programming contests and in the educational process. As a result of work in it, as well as acquaintance with other similar systems, both Russian and foreign, it was possible to formulate requirements for software development of this kind, to analyze the difference between the systems for testing competitions and training support systems.

The article also provides an overview of the existing systems for checking tasks on programming, an analysis of the advantages and disadvantages of each of these systems, and a detailed comparison of the NSUts system with the Yandex Contest system. The current state of the NSUts system architecture is described. Much attention is paid to the practice of using the system in the educational process, statistics and graphs of students' progress of students studying with the help of this system are given. The article describes the analysis of the experience of using the NSUts system and outlines the prospects for its development.

© Д. В. Иртегов, Т. В. Нестеренко, Т. Г. Чурина, 2019

Keywords

knowledge testing, programming training, programming contests, testing systems, NSUts

For citation

Irtegov D. V., Nesterenko T. V., Churina T. G. Systems for Automated Evaluation of Programming Tasks: Development, Use and Perspectives. *Vestnik NSU. Series: Information Technologies*, 2019, vol. 17, no. 2, p. 61–73. (in Russ.) DOI 10.25205/1818-7900-2019-17-2-61-73

Введение

Идея создания автоматизированной тестовой системы для проверки заданий по программированию не нова. Попытки создания таких систем предпринимались еще в 80-е гг. XX в. [1]. Можно выделить две основные проблемы, которые затруднили обретение самостоятельности этим направлениям исследований.

Во-первых, многие из разработчиков систем такого рода используют слова «автоматизированное тестирование» в описании задачи, поэтому соответствующие исследования относят либо к автоматизированному контролю качества программного обеспечения, либо к педагогическому тестированию [2–4]. Задача автоматизированного контроля навыков программирования хотя и находится в родстве с обоими названными направлениями, но требует иных подходов и иных инструментов.

Во-вторых, многие разработчики таких систем ставят задачу слишком узко. Так, разработчики системы ejudge¹ описывают свой программный продукт как «систему для проведения различных мероприятий, в которых необходима автоматическая проверка программ», а краткое описание системы на английском языке звучит как “contest management system”.

В англоязычной литературе также нет единого термина для обозначения программных продуктов такого типа. Так, организаторы международных командных соревнований по программированию ACM ICPC (ACM International Collegiate Programming Contest)² называют систему, используемую для проведения соревнований, “Programming Contest Control System” (система управления программными соревнованиями)³. В настоящее время в качестве такой системы используется система Kattis⁴, авторы которой называют ее “Automated assessment system” (автоматизированная система оценки) [5]. Еще один термин, часто встречающийся в англоязычных названиях систем такого рода, – “Online judge”, вне контекста этот термин допускает множество неверных толкований.

С 1998 г. в НГУ ведутся работы по автоматизированной проверке заданий по программированию [6]. Было создано несколько поколений систем такого типа. Первая программа объемом около 1 000 строк была написана на языке Perl. В настоящее время в олимпиадном программировании и в учебном процессе используется система NSUts. В результате работы в ней, а также знакомства с другими подобными системами – как российскими, так и зарубежными, удалось сформулировать требования к программным разработкам такого рода [7], провести анализ отличия олимпиадных систем от систем поддержки обучения [8].

Обзор существующих систем проверки знаний

Одной из первых российских систем автоматизированной проверки была система T/Run для MS-DOS, которая состояла из двух программ: run и t. Первая из них осуществляла запуск решений с ограничением по времени работы, а вторая проверяла решения на заданных тестах. Это простое решение во многих случаях является и достаточным. Будучи дополнена

¹ Система ejudge. URL: <https://ejudge.ru>

² Сайт международной олимпиады по программированию ACM-ICPC. URL: <http://icpc.baylor.edu>

³ Система CCS. URL: https://clics.ecs.baylor.edu/index.php/Contest_Control_System

⁴ Система Kattis. URL: <https://open.kattis.com/>.

набором командных файлов MS-DOS, T/Run успешно обеспечивала тестирование решений даже на Всероссийских олимпиадах школьников по информатике. Сейчас T/Run представляет ограниченный интерес из-за малой применимости к решениям, написанным под операционную систему Windows.

После системы T/Run на Всероссийских олимпиадах использовалась система Cyber Judge, созданная М. А. Бабенко. Это сложное и расширяемое решение основано на платформе .NET. Дописывание к ней небольшого модуля позволило, например, демонстрировать фотографии участников, вызываемых на личное тестирование, во время проведения XIII Всероссийской олимпиады. Но использование этой системы затруднительно, поскольку ее можно применять только в присутствии автора. Это связано с тем, что, во-первых, настройка не ориентирована на пользователя и предполагает редактирование файлов конфигурации. Во-вторых, по системе нет документации. В-третьих, часто настройка системы предполагает дописывание к ней небольших модулей для конкретной олимпиады, что при отсутствии документации является большим препятствием даже для профессиональных программистов.

В следующей системе olympiads.ru⁵ содержится, кроме программы запуска решений с контролем времени работы и использования памяти, еще база данных задач. Эта система уже имеет простой графический интерфейс пользователя, с помощью которого осуществляется редактирование базы данных и запуск решений на проверку. Появление в системе самотестирования базы данных и интерфейса отражает потребность участников в упрощении настройки систем тестирования. Между тем к появлению базы данных ведет и другое требование – автоматизация деятельности жюри по подсчету баллов и подведению итогов.

Примером системы самотестирования является tchoose, которая использовалась на Всероссийской олимпиаде по информатике для отображения результатов работы правильных и неправильных решений на подготовленных тестах. Отображение результатов тестирования в графическом виде кажется перспективной возможностью, удобной не только при подготовке олимпиад, но и во время тренировок.

Объединение двух систем Automated Programming Problem Evaluation System (APPES) на Java и Programming Contest Management System (PCMS) на Delphi, созданных в Санкт-Петербургском государственном институте точной механики и оптики, и их последующее развитие превратилось в проект PCMS2⁶. Гибкость этой системы достигается низкой связностью модулей и качественным выделением высокоуровневых концепций. В плане возможностей PCMS2 значительно превосходила на тот момент все современные требования [9].

Однако такая гибкость имеет высокую цену. Даже использование настроенного сервера PCMS2 требует предварительного обучения и понимания синтезированных концепций, не относящихся к предметной области. Кроме того, «настроенным» PCMS2 никогда не бывает, поскольку большинство действий в нем осуществляется написанием файлов конфигурации, а пользовательский интерфейс имеет преимущественно информационные функции. Файлы конфигурации в формате XML составляют значительную часть системы как по важности, так и по объему, связывая ее модули вместе. Одни XML-файлы описывают формат других XML-файлов, внутри текстовых строк применяется особо разработанный синтаксис подстановок, некоторые XML-файлы представляют собой программы на специально созданном языке сценариев. Отдельные части системы еще не до конца разработаны, поэтому часть настроек выполняется путем написания и подключения специализированных модулей, что также могут выполнить только авторы системы Г. А. Корнеев и А. С. Станкевич. Отметим, что сервер PCMS2 поддерживает только ОС Windows. Связано это с различиями в способах запуска решений и замера потребления ресурсов под разными операционными системами.

В Московском государственном университете была создана система ejudge, которая поддерживает только операционную систему Linux. Она написана на языке Си и предоставляет

⁵ Система olympiads.ru. URL: <http://www.olympiads.ru/>

⁶ Система PCMS2. URL: <http://neerc.ifmo.ru/trains/information/software.html>

определенный набор функций, удовлетворяющих современным требованиям. В системе ejudge имеются все те же конфигурационные файлы, описанные в документации. Многие действия требуется выполнять из командной строки запуском соответствующих утилит. Базовые возможности администрирования доступны через веб-интерфейс.

В системе ejudge можно отметить три недостатка, которые следуют из того, что тестирование проводится на той же машине, что и серверная часть системы. Первый из них – это возможная конкуренция за ресурсы между тестируемым решением и самой системой. В результате система может выдавать ложный вердикт «превышен лимит времени» для решений, которые на выделенной машине вполне уложились бы в этот лимит. Для обхода этой проблемы ejudge использует многократную прогонку решений с вердиктом «превышен лимит времени». Но очевидно, что это только снижает вероятность возникновения проблемы, а не решает ее. Ко второму недостатку можно отнести ограниченные возможности горизонтального масштабирования. Это означает, что подключить дополнительные машины, кроме той, на которой проходит тестирование, физически невозможно. Единственное, что можно сделать, это запускать несколько процессов проверки, по числу процессорных ядер на сервере. Но это может привести к конкуренции решений за оперативную память, т. е. к росту числа ложных вердиктов «превышен лимит времени». При проверке решений на олимпиадах могут случаться значительные задержки между отправкой решения и получением вердикта. Третий недостаток состоит в том, что тестирование возможно только под той же ОС, под которой работает сама система, т. е. под Linux. Это ограничивает применение системы, особенно для школьных соревнований, где правила требуют ОС Windows и компиляторы, доступные только для Windows или для учебных курсов, где ОС и язык программирования заданы программой курса.

Система автоматической проверки с архивом задач в испанском городе Вальядолиде⁷ была создана гораздо раньше аналогичных российских разработок и получила большую популярность среди студентов. Для ее работы используется операционная система Linux, поддерживаются языки программирования: GNU C, GNU C++, Free Pascal. Архив постоянно пополняется новыми задачами и тестами. Для каждой задачи приводится статистика: отношение числа правильных решений задачи к общему числу попыток сдать данную задачу. Недостатком системы является тот факт, что вся коммуникация с участниками ведется через электронную почту. Также для многих неудобным является использование системы Linux. Система развернута на серверах университета Вальядолид. Доступ к этим серверам бесплатен, но скачать программное обеспечение и развернуть его на своем оборудовании невозможно. Также все изменения в системе: добавление языков и версий компиляторов, изменение правил подсчета рейтинга – возможны только с разрешения и при активном участии ее разработчиков.

Главным же недостатком данной системы и ряда рассматриваемых далее систем является то, что они используют модель Software as Service (SaS). По такой модели программное обеспечение не устанавливается на компьютеры пользователей, а развертывается на серверах поставщика услуг. Фактически, пользуясь услугой, клиент получает и программу, и вычислительные ресурсы для ее работы. Часто программное обеспечение вообще не доступно конечному пользователю. Данная модель используется как для коммерческих, так и для бесплатных или условно-бесплатных услуг. Например, такие популярные сервисы, как Gmail или Yandex mail, работают именно по этой модели. Для многих сценариев использования эта модель привлекательна, поэтому она и получает широкое распространение.

Но эта модель имеет и ряд недостатков, часть из которых существенна для учебного процесса и «долгоиграющих» мероприятий, таких как проводимая на протяжении десятилетий ежегодная олимпиада. Наиболее очевидный недостаток – это полная зависимость от поставщика услуги. Если поставщик резко изменит условия или вообще откажется от данного сер-

⁷ Архив задач с автоматической системой проверки в испанском городе Вальядолиде. URL: <http://acm.uva.es>

виса, пользователь, в данном случае организатор олимпиады, ничего сделать не сможет. Он может даже потерять доступ к данным старых мероприятий. Также данная модель затрудняет доработку системы под нужды конкретного пользователя или конкретного мероприятия. Если организаторы олимпиады захотят поменять правила, они вынуждены обращаться к поставщику SaS, который может и не выполнить этот запрос.

Известна система автоматической проверки *timus*⁸ Уральского государственного университета с архивом задач. Система используется для проверки олимпиад, при этом участникам предоставляется набор языков программирования: C, C++ и Pascal (Delphi). Эта система предоставляется по модели SaS и поэтому обладает недостатками, характерными для этой модели и изложенными ранее.

В последние годы широкую известность в России получила система Yandex Contest компании Яндекс⁹. Она также предоставляется на условиях SaS. С 2013 г. она используется для проведения ряда соревнований, а также в учебном процессе, например в курсах Школы анализа данных Яндекс. Тестирование проводится под ОС Linux, и ей присущи недостатки, изложенные при описании системы *ejudge*. Кроме того, тестирование проводится на виртуальных машинах, а не на выделенных физических машинах. Это удешевляет развертывание системы, но создает опасность ложных вердиктов «превышено время тестирования» из-за конкуренции процессорных ядер за доступ к ОЗУ и накладных расходов виртуализации. Система Yandex Contest имеет ограниченные возможности увеличения емкости, что наблюдалось на региональных этапах Всероссийской олимпиады школьников по информатике 2017–2018 г. в тех регионах, где соревнования проводились на Yandex Contest. Время ожидания вердикта системы было довольно продолжительным. Возможно, это связано с архитектурными ограничениями на количество или размещение тестирующих машин.

Система NSUts

С помощью автоматизированной системы NSUts¹⁰ проводятся соревнования по правилам ACM-ICPC и по правилам Всероссийской олимпиады школьников. Система поддерживает тестирование программ на языках C, C++, Java, Pascal, Python, Kotlin, C# для Win32. В рамках одного тура можно проводить проверку программ на нескольких диалектах одного языка, например Visual C или Borland C, так как компилятор языка программирования явно указывается участником при отправке задания. Для отправки задач участники используют веб-интерфейс системы, в котором можно отслеживать состояние тестирования, просматривать рейтинг, а также «общаться» с жюри. Для хранения задач, очереди тестирования и рейтинга используется реляционная СУБД. Заметим, что сохраняются все попытки отправки решений, все исходные тексты и точные значения времени их отправки. Это важно для проведения апелляций и для выявления попыток мошенничества. Время исполнения задачи в многозадачной среде определяется сложными сценариями конкуренции за разные ресурсы, такие как оперативная память, кэш процессора, время переключения контекста процессора и т. д. Поэтому проверка решений проводится на выделенных компьютерах, что сводит влияние такой конкуренции к минимуму.

К автоматизированным системам тестирования знаний предъявляются следующие требования:

- обеспечение глубокого и адекватного тестирования знаний и навыков;
- эффективная защита от мошенничества [10];
- простота в эксплуатации для специалистов средней квалификации.

В системе NSUts [6] можно выделить три основные составляющие:

- сервер олимпиад;

⁸ Архив задач с автоматической системой проверки в Екатеринбурге. URL: <http://acm.timus.ru>

⁹ Система Яндекс Контест. URL: <https://contest.yandex.ru/>

¹⁰ Система NSUts. URL: <https://olympic.nsu.ru/nsuts-new/login.cgi>

- клиентское программное обеспечение;
- тестирующий клиент.

Главной частью системы является сервер. Тестирующих клиентов в системе может быть несколько. К клиентскому программному обеспечению относятся браузер и среда разработки для создания и отладки решений. Оно должно быть установлено на компьютере пользователя, и не предоставляется системой, но принимает активное участие в работе.

Олимпиада – это основной объект, над которым производятся действия сервера. Она может включать несколько туров. Каждый тур – это «олимпиада» с точки зрения участников, которая состоит из набора задач, представляющихся им для решения. Сервер тестирования осуществляет проведение олимпиады (тура). Он предоставляет участникам и членам жюри веб-интерфейс для взаимодействия с системой тестирования, автоматизирует управление олимпиадой (туром).

Под управлением олимпиадой понимается:

- осуществление хранения в базе данных условий задач, тестов и решений;
- проверка решений участников и визуализация ее результатов;
- создание и отображение рейтинга мест, которые заняли участники соревнования;
- генерация отчетов о проведении олимпиады (тура);
- предоставление средств для перетестирования решений участников;
- поддержка системы регистрации участников, обеспечение обратной связи с жюри и решение других организационных задач;
- администрирование олимпиад и туров.

Тестирующий клиент забирает решения участников, полученные сервером, из очереди решений, и выполняет их проверку на наборе тестов. В процессе обработки решения он получает исходный код, информацию о требуемом компиляторе и тестовые данные. Результат проверки решения передается обратно серверу, который производит его обработку, включающую отправку его участнику и создание рейтинга. Взаимодействие с системой участников олимпиад и членов жюри осуществляется посредством веб-интерфейса, который исключает обращение к тестирующему клиенту напрямую.

Тестирующие клиенты взаимодействуют с сервером через протокол HTTP, поэтому они могут быть размещены как на той же физической машине, что и сервер, так и на выделенных машинах, физических или виртуальных. В НГУ для тестирования олимпиад используются тестирующие клиенты на выделенных физических машинах.

Проведем сравнение системы NSUts, например, с системой Yandex Contest.

1. Система NSUts развернута на инфраструктуре разработчиков, и ее исходный код доступен НГУ. Это позволяет планировать мероприятия, в том числе статусные или просто ответственные, без согласований с внешними для НГУ организациями. При проведении ответственного мероприятия на системе Yandex Contest необходимо это согласовывать с компанией Yandex, так как в ней на это время могут быть запланированы профилактические или другие работы, закрывающие или ограничивающие доступ. Изменять правила соревнований, добавлять или исключать языки программирования невозможно без согласования с внешними организациями.

На Открытой Всесибирской олимпиаде им. И. В. Поттосина традиционно проводится первый тур по нестандартным правилам. «Стандартные» правила ACM хоть и медленно, но также изменяются. Языки и версии компиляторов требуется добавлять каждый год. Нужно иметь возможность дорабатывать и адаптировать систему в любом желательном для НГУ направлении, например добавить MathLab для приема задач у физиков.

2. В системе NSUts тестирование происходит на выделенных компьютерах под операционной системой Windows. Это решение обеспечивает адекватное судейство, связанное, во-первых, с точным подсчетом процессорного времени, поскольку правила большинства мероприятий предусматривают ограничение времени на исполнение программы. Во-вторых, обеспечивается возможность проводить школьные олимпиады и использовать компиляторы, доступные только под Windows (Visual Studio, Delphi).

Yandex Contest использует виртуальные машины с операционной системой Linux. У используемых компиляторов есть аналоги под систему Linux (Mono для C#, Free Pascal), но все-таки сами компиляторы и среды исполнения не тождественным тем, что указаны в правилах соревнований, а это дает основания для апелляций и иногда приводит к реальным проблемам.

3. В системе NSUts хранится история олимпиад, проводившихся в НГУ с 2010 г.

История знает немало случаев, когда публичные сервисы внезапно закрывались или изменяли политику до неприемлемой, самый известный пример – Sourceforge, который после смены собственника начал подкладывать программные модули, показывающие рекламу (adware), в размещенные на нем программные проекты. Для многолетней деятельности этот риск вполне реалистичен и неприемлем.

4. Yandex Contest заявляют одновременную поддержку 1 000 пользователей, сервер NSUts тоже на такую нагрузку регулярно тестируется.

Опыт использования системы NSUts

Как было отмечено, система NSUts создана для обеспечения проверки решений участников олимпиад по программированию, поэтому главной частью ее использования является проведение олимпиад по программированию всех уровней, в том числе открытых интернет-олимпиад. Она используется при проведении Открытой Всесибирской олимпиады по программированию им. И. В. Поттосина [11], районных и региональных школьных олимпиад по программированию в Новосибирской области¹¹. В первых этапах Всероссийской олимпиады школьников по информатике, проводимых в Новосибирской области с помощью системы NSUts, одновременно участвовало несколько сотен школьников, было проверено свыше тысячи решений. Некоторые студенческие олимпиады собирают около 1 000 участников, при этом проверяется в онлайн-режиме более 10 000 решений в течение одного тура.

Приведем некоторые цифры по Открытой Всесибирской олимпиаде по программированию им. И. В. Поттосина. Олимпиада является командной, каждая команда состоит из трех участников. Имея один компьютер, команда стремится сдать как можно больше задач. Олимпиада проводится с 2000 г. Первые годы эта олимпиада проводилась в три тура. Первые два тура были отборочными с помощью Интернета, третий – очный. В последние годы олимпиада проводится в два тура: первый, отборочный, интернет-тур проходит заочно по традиционным правилам международного студенческого чемпионата ACM-ICPC. Второй тур – очный, состоит из двух номинаций. Первая номинация проводится по правилам, напоминающим Marathon Match на TopCoder¹². В нем предлагается одна игровая или исследовательская задача на 5 часов работы. Как правило, для него жюри разрабатывает оригинальный проект, а также интерфейс и модули, которые позволяют встраивать решения участников в интерфейс, визуализировать и проигрывать их на сайте олимпиады. Для проведения этой номинации выполняется специальная настройка автоматизированной системы тестирования NSUts. Вторая номинация проводится по правилам международного чемпионата ICPC-ACM, а результат определяется по сумме двух номинаций.

30 сентября 2018 г. прошел интернет-тур XIX Олимпиады¹³. В нем участвовало 411 команд более чем из 100 университетов и 100 школ 21 страны. Результаты тура приведены на сайте олимпиады.

По правилам ACM-ICPC задача считается непринятой до тех пор, пока не пройдет все тесты жюри. Поэтому многие команды в течение тура на одну задачу отправляют на проверку несколько решений. Всего во время тура было отправлено около 4 000 решений.

¹¹ Сайт олимпиад Новосибирского государственного университета. URL: <https://olympic.nsu.ru/>

¹² URL: <https://www.topcoder.com/community/data-science/how-to-compete>

¹³ XIX Открытая Всесибирская олимпиада по программированию им. И. В. Поттосина. URL: <https://olympic.nsu.ru/widesiberia/2018/news>

Очный тур прошел с 17 по 19 ноября 2018 г. В нем приняло участие 40 команд из Абакана, Барнаула, Екатеринбург, Ижевска, Иркутска, Кирова, Красноярска, Москвы, Новосибирска, Омска, Перми, Рубцовска, Санкт-Петербурга, Томска, Тюмени, Якутска.

Номинация была проведена 17 ноября 2018 г. Команды получили игровую задачу. В данной задаче требовалось написать программу «Умный Пилот» для управления космической ракетой. Цель полета – взлететь с поверхности Земли и выйти на эллиптическую орбиту с перигейным расстоянием не меньше заданного. Перигей – точка орбиты, ближайшая к центру Земли, а перигейное расстояние – расстояние от перигея до центра Земли. В условии задачи описано строение ракеты, ее управление и физика задачи, представлен протокол взаимодействия, описан статус ракеты и программа-интерактор, которая после статуса выдает текущее состояние ракеты. В архиве с игрой представлены все материалы, необходимые для работы с данной задачей, а также приведены примеры решений. Помимо описания задачи и протокола работы для нее представлены требования к программе, система оценки, генератор тестов.

Для тестирования первой номинации использовались средства NSUts. Участники сдавали свои решения через нее, результаты промежуточной проверки тоже отображались в системе. Однако проверяющая программа была реализована специально для данной задачи отдельно от системы. Вторая номинация была проведена 18 ноября 2018 г. Результаты обеих номинаций и общие результаты тура представлены на сайте олимпиады. Статистика за последние одиннадцать лет – в табл. 1.

Таблица 1

Статистика Открытой Всесибирской олимпиады им. И. В. Поттосина
(2008–2018 гг.)

Table 1

Pottosin Open Widesiberia Olympiad Statistics
(2008–2018)

Год	Количество команд интернет-тура	Количество участников (округленно)	Количество команд – участниц очного тура	Количество вузов + школ	Количество стран
2008	274	820	46	67 + 14	7
2009	272	820	48	98 + 30	7
2010	259	760	49	96 + 23	7
2011	301	870	49	95 + 29	7
2012	253	850	48	78 + 21	6
2013	228	680	51	82 + 12	8
2014	341	1000	48	98 + 74	7
2015	372	Более 1 000	50	96 + 39	12
2016	393	Более 1 000	47	96 + 24	12
2017	379	Более 1 000	46	101 + 27	31
2018	411	Более 1 000	42	106 + 11	22

В течение последних семи лет система NSUts активно используется преподавателями НГУ для промежуточного контроля заданий студентов по дисциплине «Программирование».

Отметим ряд свойств системы NSUts, которые дали возможность использовать ее в учебном процессе.

1. В NSUts реализована двухуровневая система организации данных. На первом уровне выделены в качестве основных объектов олимпиады, а на втором, внутри олимпиад, – туры. Каждый тур содержит некоторое количество задач. Применительно к учебному процессу под

олимпиадами можно понимать годовой или семестровый практический курс, а каждый тур рассматривать как недельное задание, состоящее из задач, объединенных темой, которая разбирается на семинарских занятиях. Регистрация на такую «олимпиаду» ограничена студентами одной группы, таким образом обеспечивается некоторая конфиденциальность информации.

2. Студенты могут сдавать задачи в систему в любое время суток, что поддерживает их самостоятельную работу.

3. Набор задач каждого тура вместе с тестами подготавливается заранее и может использоваться в течение многих лет. Решения студентов проверяются на этих тестах, что значительно экономит время преподавателя при проверке заданий. На занятиях остается больше времени для обсуждения методов кодирования алгоритмов или правил оформления текстов программ. Интерфейс системы NSUts предоставляет преподавателю средства, позволяющие проверять задания студентов удаленно и во внеурочное время. Заметим, что автоматизированная проверка программ не может исключить личного общения преподавателя со студентом. Преподаватель, например, может не зачесть решение студента, которое прошло все тесты и получило вердикт «принято», если посчитает, что студент не до конца понимает суть решаемой задачи, не может ее объяснить или решил задачу самостоятельно.

4. Еженедельную работу студентов можно стимулировать системой баллов, которые непосредственно влияют на общую оценку. При этом важно учитывать время сдачи задания, оно хранится в системе NSUts для каждой посылки. Объявленные заранее правила получения оценок порождают стремление студентов сдать все задачи в кратчайшие сроки, что, несомненно, идет на пользу обучению.

5. Использование автоматизированной проверки задач способствует формализации отношений между преподавателем и студентом, практически исключает необъективность и предвзятость.

Рассмотрим статистику использования NSUts на первом курсе ФИТ НГУ за шесть лет (табл. 2). В первом семестре требовалось выполнять по одному заданию в неделю, всего 15 заданий. В каждое задание входило от трех до восьми задач. Всего за семестр нужно было сдать 60 задач различного уровня сложности. При успешной сдаче решения задачи в систему в течение первой недели со дня выдачи задания за нее начисляется 10 баллов, а далее баллы понижаются. При сдаче программы в первый день после соответствующего семинара за нее начисляется еще один бонусный балл.

Таблица 2

Результаты работы студентов за семестр

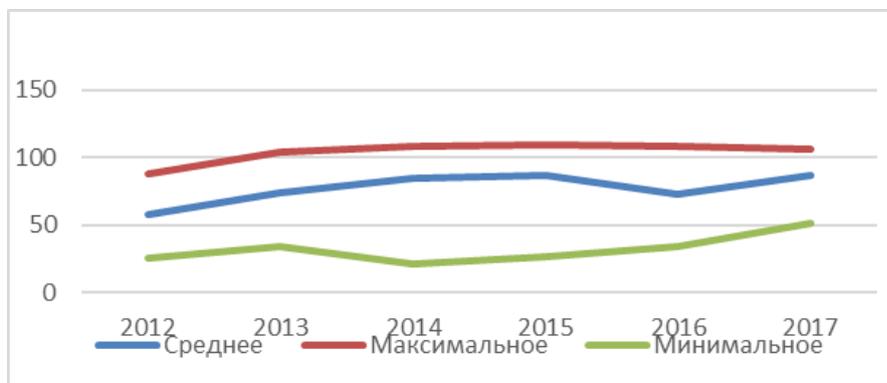
Table 2

Students Work Statistics for Semester

	Количество баллов, набранное студентами за семестр						Количество задач, решенных студентами за семестр, из 60 возможных					
	2012	2013	2014	2015	2016	2017	2012	2013	2014	2015	2016	2017
Среднее	350	442	507	519	440	520	39	49	54	53	48	53
Максимальное	537	621	648	653	647	637	52	59	60	60	60	60
Минимальное	157	205	128	159	202	304	24	28	31	35	24	34

Студенту выставляется отличная оценка, если он набирает не менее 75 % от базового количества баллов, что составляет не менее 450 баллов. Из приведенной в табл. 2 статистики видно, что среднее количество баллов, которое набирали студенты, очень близко к этой циф-

ре, а в половине случаев даже ее превосходит. Максимальное же количество баллов значительно превосходит 600. Это говорит о том, что в каждой группе были студенты, которые стремились все сдать в первый день и получить за каждую задачу по 11 баллов. Процентное соотношение набранных баллов по годам показано на рисунке.



Процент от возможных 600 баллов, набранный студентами за семестр
The Percentage of Possible 600 Points Scored by Students per Semester

Поскольку система NSUts изначально создавалась не для использования ее в учебных занятиях, в ней не была предусмотрена соответствующая функциональность. Отметим некоторые ее особенности, которые хотелось бы модифицировать и развивать.

1. Большой проект, состоящий из нескольких модулей, в данной системе проверить нельзя, так как решение участника для проверки записывается в одном файле.

2. Для усвоения некоторых структур данных, таких как списки, стеки, очереди, деревья, необходимо научиться эти структуры обрабатывать. Для этого часто требуется уметь писать функции, которые для построенных структур должны выдавать результат преобразования. В системе NSUts, как и в любой другой системе проверки решений, этого сделать нельзя, поскольку входные и выходные данные для решения должны быть записаны в файлы. Следовательно, студенту дополнительно нужно строить саму структуру, что является отдельной задачей.

3. Никакая система не может заменить преподавателя, который должен проверять код студента. Иначе может оказаться, что система при проверке решения выдала вердикт «принято», но задача решена не та, которая была поставлена.

4. В системе рейтинг участников строится в туре. Построение общего рейтинга по всем турам осуществимо только внешними средствами. Автоматизация процесса построения такого рейтинга была бы желательна.

5. В олимпиадах по программированию доступ к набору тестов имеет только жюри, в учебном процессе в этой роли выступает преподаватель. Было бы полезно давать возможность студентам видеть тесты, это часто помогает отладить программу.

Заключение

Многолетний опыт использования нескольких поколений систем автоматической оценки заданий по программированию [6] позволил разработать мощное и гибкое решение, применимое для широкого спектра мероприятий: проведения олимпиад, практикумов по программированию, тренировок олимпиадных команд и др. Но в процессе работы определились направления развития системы. Одним из них является адаптация системы к требованиям учебного процесса. Хотелось бы, например, реализовать в системе поддержку для различных стратегий обучения. Также имеется большая потребность создать и поддерживать банк задач

без привязки к конкретной олимпиаде или туру. Это связано с тем, что накопилось большое количество задач разных уровней сложности и тематики, все они хранятся в своих турах, что усложняет их поиск.

В настоящее время ведутся работы по переходу сервера на клиент-серверную архитектуру AJAX [12]. Большая часть системы реализована по технологии динамического HTML, когда сервер полностью генерирует код страницы и передает ее клиентскому браузеру на каждый запрос. При характерной для NSUts нагрузке, когда разные пользователи в большинстве случаев видят разные данные, это приводит к тому, что сервер делает много лишней работы, и невозможно использовать кэширование. AJAX – подход к разработке современных веб-сайтов, используемый ведущими интернет-сервисами Gmail, Facebook, V Kontakte и др., состоящий в том, что сервер генерирует только данные в компактном формате, например в формате JSON, а генерация и отрисовка всего пользовательского интерфейса производится на клиентском браузере при помощи JavaScript. Это позволяет решить проблемы с «лишней работой» для сервера и кэшированием, задействовать процессоры клиентских компьютеров (когда сервер один, а клиентских компьютеров сотни, преимущества кажутся очевидными), а также ввести ряд полезных функций, уже реализованных в других системах: автообновление очереди и рейтинга, оповещение об изменениях рейтинга.

Существующая система уже начинает испытывать нехватку ресурсов, если в ней оказывается больше 1 000 пользователей. Созданы прототипы новой архитектуры, которые обеспечивают приемлемую производительность вплоть до 2–3 тысяч пользователей. В результате модернизации системы ожидается повышение ее производительности под высокими нагрузками, кратное повышение емкости, а также снижение потребности сервера в ОЗУ в десятки или сотни раз, по сравнению с текущими 16 Gb.

Система NSUts постоянно развивается, предполагаются дополнительные направления ее доработки, такие как выкладывание исходного кода системы в открытый доступ, улучшение дизайна системы и ее эргономики.

Список литературы / References

1. **Reek K. A.** The TRY System – or – How to Avoid Testing Student Programs. *Proceedings SIGCSE Bulletin*, February 1989, vol. 21, no. 1, p. 112–116.
 2. **Jackson D., Usher M.** Grading Student Programs Using ASSYST. In: *Proceedings SIGCSE '97*, 1997, p. 335–339.
 3. **Jones E. L.** The SPRAE Framework for Teaching Software Testing in the Undergraduate Curriculum. In: *Proceedings ADMI 2000*, June 1–4. Hampton, VA USA, 2000.
 4. **Edwards S.** Using Software Testing to Move Students from Trial-and-Error to Reflection-in-Action. AMC Press, 2004, p. 26–30.
 5. **Enstrom E., Kreitz G., Niemela F., Soderman P., Kann V.** Five years with kattis – Using an automated assessment system in teaching. In: *Proceedings of the 2011 Frontiers in Education Conference (FIE '11)*. IEEE Computer Society. Washington, DC, USA, p. T3J-1–T3J-6. DOI 10.1109/FIE.2011.6142931
 6. **Иртегов Д. В., Нестеренко Т. В., Чурина Т. Г.** Разработка систем автоматизированной оценки заданий по программированию // *Системная информатика*. 2017. № 11. С. 91–116. DOI 10.31144/si.2307-6410.2017.n11
- Irtegov D. V., Nesterenko T. V., Churina T. G.** Development of Automated Evaluation Systems for Programming Tasks. *System Informatics Journal*, 2017, no. 11, p. 91–116. (in Russ.) DOI 10.31144/ si.2307-6410.2017.n11

7. **Боженкова Е. Н., Иртегов Д. В., Киров А. В., Нестеренко Т. В., Чурина Т. Г.** Автоматизированная система тестирования NSUts: требования и разработка прототипа // Вестник НГУ. Серия: Информационные технологии. 2010. Т. 8, № 4. С. 46–53.
Bozhenkova E. N., Irtegov D. V., Kirov A. V., Nesterenko T. V., Churina T. G. NSUts Automated Testing System: Prototype Requirements and Development. *Vestnik NSU. Series: Information Technologies*, 2010, vol. 8, no. 4, p. 46–53. (in Russ.)
8. **Чурина Т. Г., Иртегов Д. В.** Требования к автоматической системе тестирования знаний // Тр. VI Междунар. конф. «Интеллектуальные технологии в образовании, экономике и управлении». Воронеж, 2009, С. 309–317.
Churina T. G., Irtegov D. V. Requirements for an automated knowledge testing system. In: Proceedings of the VI International Conference “Intellectual Technologies in Education, Economics and Management”. Voronezh, 2009, p. 309–317. (in Russ.)
9. **Корнеев Г. А., Елизаров Р. А.** Автоматическое тестирование решений на соревнованиях по программированию // Телекоммуникации и информатизация образования. 2003. № 1. С. 61–73.
Korneev G. A., Elizarov R. A. Automatic testing of solutions in programming competitions. *Telecommunications and education informatization*, 2003, no. 1, p. 61–73. (in Russ.)
10. **Боженкова Е. Н., Воронков А. Д., Иртегов Д. В., Коньшева Е. Н., Черненко С. А., Чурина Т. Г.** Модель разграничения прав доступа в системе автоматизированной проверки корректности программных приложений // Вестник НГУ. Серия: Информационные технологии. 2011. Т. 9, № 4. С. 79–85.
Bozhenkova E. N., Voronkov A. D., Irtegov D. V., Konysheva E. N., Chernenok S. A., Churina T. G. Model of access rights delimitation in the system of automated verification of software applications correctness. *Vestnik NSU. Series: Information Technologies*, 2011, vol. 9, no. 4, p. 79–85. (in Russ.)
11. **Чурина Т. Г., Боженкова Е. Н., Нестеренко Т. В.** Задачи Открытой Всесибирской олимпиады по программированию имени И. В. Поттосина: от теории к практике // Вестник НГУ. Серия: Информационные технологии. 2007. Т. 5, № 1. С. 40–46.
Churina T. G., Bozhenkova E. N., Nesterenko T. V. Problems of the Pottosin Open Widesiberia Olympiad: from theory to practice. *Vestnik NSU. Series: Information Technologies*, 2007, vol. 5, no. 1, p. 40–46. (in Russ.)
12. **Боженкова Е. Н., Иртегов Д. В., Колбин Я. С.** Оптимизация производительности веб-интерфейса приложения NSUts средствами динамического HTML // Вестник НГУ. Серия: Информационные технологии. 2015. Т. 13, № 2. С. 13–21.
Bozhenkova E. N., Irtegov D. V., Kolbin Ya. S. Optimizing the performance of the NSUts web interface using dynamic HTML. *Vestnik NSU. Series: Information Technologies*, 2015, vol. 13, no. 2, p. 13–21. (in Russ.)

*Материал поступил в редколлегию
Received
15.01.2019*

Сведения об авторах / Information about the Authors

Иртегов Дмитрий Валентинович, доцент, заведующий лабораторией, факультет информационных технологий, Новосибирский государственный университет (ул. Пирогова, 1, Новосибирск, 630090, Россия)

Dmitry V. Irtegov, Associate Professor, Novosibirsk State University (1 Pirogov Str., Novosibirsk, 630090, Russian Federation)

fat@nsu.ru

Нестеренко Татьяна Викторовна, научный сотрудник, Институт систем информатики им. А. П. Ершова СО РАН (пр. Академика Лаврентьева, 6, Новосибирск, 630090, Россия), доцент, факультет информационных технологий, Новосибирский государственный университет (ул. Пирогова, 1, Новосибирск, 630090, Россия)

Tatyana V. Nesterenko, Researcher, A. P. Ershov Institute of Informatics Systems SB RAS (6 Academician Lavrentiev Ave., Novosibirsk, 630090, Russian Federation), Associate Professor, Novosibirsk State University (1 Pirogov Str., Novosibirsk, 630090, Russian Federation)

nest@iis.nsk.su

Чурина Татьяна Геннадьевна, кандидат физико-математических наук, доцент, старший научный сотрудник, Институт систем информатики им. А. П. Ершова СО РАН (пр. Академика Лаврентьева, 6, Новосибирск, 630090, Россия), доцент, факультет информационных технологий, Новосибирский государственный университет (ул. Пирогова, 1, Новосибирск, 630090, Россия)

Tatyana G. Churina, Senior Researcher, A. P. Ershov Institute of Informatics Systems SB RAS (6 Academician Lavrentiev Ave., Novosibirsk, 630090, Russian Federation), Associate Professor, Novosibirsk State University (1 Pirogov Str., Novosibirsk, 630090, Russian Federation)

tanch@iis.nsk.su

УДК 004. 8
DOI 10.25205/1818-7900-2019-17-2-74-80

Методологические аспекты применения инструментария ГИС для управления трудоустройством выпускников региональных вузов

Л. А. Кадирова

*Андижанский государственный университет
Андижан, Республика Узбекистан*

Аннотация

Целью данного исследования является совершенствование механизма трудоустройства выпускников вузов по специальностям с учетом нужд региона с использованием современных компьютерных технологий. В статье рассматриваются преимущества и методологические аспекты применения инструментария географических информационных систем, которые являются симбиозом возможностей управления большими базами данных и пространственной визуализации, характерной для топографических карт с использованием QGIS. Каждый информационный слой представлен таблицей реляционной базы данных или электронной таблицей, в которых имеется поле привязки к географическому положению, что позволяет разместить данные на карте. Научная значимость статьи обусловлена тем, что данная технология способствует созданию единого информационного пространства рынка работодателей по востребованным специальностям, а также активному налаживанию сотрудничества между вузами и работодателями с целью совершенствования механизма трудоустройства выпускников вузов на основе компетентностного подхода. Результатом исследования являются методологические аспекты применения открытой геоинформационной системы QGIS для управления трудоустройством молодых специалистов. Кроме того, поставлены задачи для перспективного развития механизма применения геоинформационных технологий в трудоустройстве выпускников вузов.

Ключевые слова

трудоустройство, географические информационные системы, рынок образования, рынок труда, методология, слои, топографическая карта, базы данных

Для цитирования

Кадирова Л. А. Методологические аспекты применения инструментария ГИС для управления трудоустройством выпускников региональных вузов // Вестник НГУ. Серия: Информационные технологии. 2019. Т. 17, № 2. С. 74–80. DOI 10.25205/1818-7900-2019-17-2-74-80

Methodological Aspects for the Use of GIS Tools for Managing the Employment of Graduates

L. A. Kadirova

*Andijan State University
Andijan, Republic of Uzbekistan*

Abstract

The purpose of this study is to improve the mechanism of employment of graduates of universities in the field in the region with the use of modern computer technology. Namely, the article discusses the advantages and methodological aspects of using the tools of geographic information systems, which are a symbiosis of the management capabilities of large databases and spatial visualization characteristic of topographic maps by using QGIS. A relational database table or a spreadsheet with a geolocation field, which allows data to be placed on a map, represents each information layer.

© Л. А. Кадирова, 2019

The scientific significance of the article is due to the fact that this technology contributes to the creation of a single information space of the employers market in the context of popular specialties, as well as to the active establishment of cooperation between universities and employers in order to improve the mechanism of employment of graduates on the bases of competence-based approach.

The result of the research is the methodological aspects of the use of open geographic information system QGIS for managing the employment of young professionals. In addition, tasks have been set for the mechanism for applying geo-information technologies in the employment of university graduates.

Keywords

employment, geographic information systems, education market, labor market, methodology, layers, topographic maps, databases

For citation

Kadirova L. A. Methodological Aspects for the Use of GIS Tools for Managing the Employment of Graduates. *Vestnik NSU. Series: Information Technologies*, 2019, vol. 17, no. 2, p. 74–80. (in Russ.) DOI 10.25205/1818-7900-2019-17-2-74-80

Введение

Применение географических информационных систем (ГИС) в управлении трудоустройством выпускников вузов – это внедрение современных компьютерных технологий в процесс управления трудоустройством кадров, являющих собой карты или типографические ресурсы нового поколения, позволяющие собирать, хранить и отображать данные на экране в электронном виде. Применительно к рынку работодателей ГИС могут содержать помимо географии местности и информацию по статистике, демографии, которая позволяет делать анализ. В решении проблемы управления трудоустройством выпускников вузов ГИС – это симбиоз возможностей управления большими базами данных (БД) и пространственной визуализации, характерной для топографических карт. Многослойная тематическая информация, в частности по востребованным специальностям на рынке работодателей, хранящаяся в базах данных системы, имеет привязку к их географическому местоположению.

Использование ГИС с целью трудоустройства выпускников вузов имеет преимущества: пространственное, 3D-представление параметров, что облегчает восприятие; предоставляет возможности интеграции данных из различных источников в единый массив для общего пользования; обеспечивает автоматический анализ пространственной географической информации и отчетность; дает возможность расшифровки данных спутниковой съемки, полученных ранее схем и планов местности. Функции, выполняемые ГИС представлены на схеме (рис. 1).



Рис. 1. Функции ГИС Авторская разработка

Fig. 1. Functions of GIS. Author's Approach

Согласно данным Статистического управления Республики Узбекистан¹, по состоянию на 01.01.2017 функционирующих в стране юридических лиц по регионам визуально можно представить на карте (рис. 2). Распределение функционирующих в республике предприятий по видам экономической деятельности представлено на рис. 3.



Рис. 2. Функционирующие в республике юридические лица (по данным Госкомстата Республики Узбекистан по состоянию на 01.01.2017)

Fig. 2. Legal Entities Operating in Uzbekistan (Data from the State Statistics Committee of the Republic of Uzbekistan for 01.01.2017)

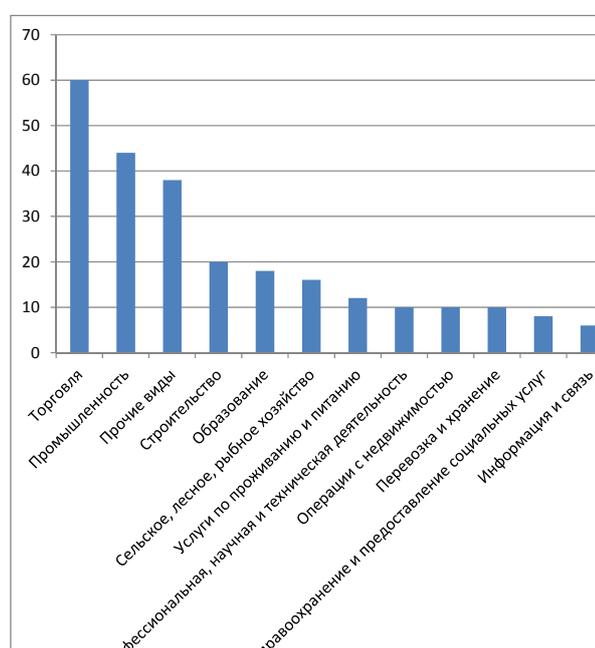


Рис. 3. Количество действующих по республике предприятий по видам экономической деятельности (данные Статистического управления Республики Узбекистан. Авторская разработка)

Fig. 3. The Number of Enterprises Operating in the Republic of Uzbekistan by Economic Activity (Data from the State Statistics Committee of the Republic of Uzbekistan for 01.01.2017. Author's Approach)

¹ Государственный комитет Республики Узбекистан по статистике. URL: <https://stat.uz/uz>

Наша задача – заложить методологические аспекты представления данных такого типа в разрезе востребованных специальностей каждого вида деятельности в регионах и визуализация их посредством ГИС технологий. Маркетинговые исследования зарубежного опыта трудоустройства выпускников вузов [1] показывают, насколько актуально решение этой проблемы в Республике Узбекистан.

Модернизация государства, либерализация экономики на пути поступательного реформирования обеспечили стабильный рост количества зарегистрированных субъектов малого предпринимательства. Для сравнения: по состоянию на 01.01.2017 их число составило около 233 тыс. ед., что по сравнению с 01.01.2013 выросло на 27 тыс. ед. Если в 2012 г. соотношение вновь созданных субъектов малого предпринимательства к общему количеству действующих субъектов малого предпринимательства составляло 11 %, то в 2016 г. данный показатель превысил 14 %.

С целью визуализации данных по рынку труда посредством современного технологического аппарата, а именно ГИС карты, предварительно необходимо создать единое информационное пространство всех специальностей, востребованных работодателями по видам экономической деятельности юридических лиц как государственного, так и частного сектора.

Методология применения инструментария ГИС для управления трудоустройством выпускников вузов

Предлагаемая нами методология предполагает использование открытой геоинформационной системы QGIS, которая дает нам возможность, в частности, применять ее в управлении трудоустройством выпускников вузов, изучать принципы ее работы и модифицировать, распространять копии, совершенствовать и опубликовывать разработанные нами продукты для массового доступа.

Мы намерены рассмотреть объекты ГИС, которые представлены в виде слоев. Каждый слой состоит из однотипных данных, соответствующих группе объектов на карте, и хранится в таблице QGIS. Помимо данных, отражающих, например, владельца помещения, юридический адрес, площадь зданий и конструкций, вид деятельности юридического лица, наименование и количество вакансий на предприятии, в таблице QGIS отводится место для невидимого столбца, в котором содержатся пространственные данные, дающие возможность отобразить на топографической карте каждый объект, описанный в соответствующей строке этой таблицы.

Для создания ГИС карты возможно использование данных из Интернета, находящихся в открытом доступе: векторные слои (границы административно-территориального округа, пункты проживания населения, дорожные развязки, маршруты транспорта, интересующие объекты и т. п.). Возможны несколько вариантов просмотра спутникового изображения региона.

Самый популярный – картографический ресурс «Google-maps». Однако он не является бесплатными и не работает в режиме реального времени. Второй способ просмотра карты региона – это карты Yandex. Опять же мы увидим изображение со спутника, сохраненное некоторое время назад. Третий вариант спутникового просмотра вашего региона – это приложение «Google планета Земля». Для его установки уходит около 5 минут. Наименование приложения «Google-earth», с его помощью можно побывать в любой точке земного шара. Кроме прочего, здесь имеется огромное количество приложений, обеспечивающих дополнительной информацией. Также есть специальная картографическая и навигационная программа SASPlanet – полностью бесплатная и позволяющая работать с большим количеством онлайн карт: находим населенный пункт, выбираем из списка доступных карт карты от Bing – бесплатный аналог «Google-maps» в Яндекс. Выбираем масштаб 20 – самый крупный из возможных, нажимаем «Начать», чем запускаем операцию «приклеивания» объектов на карту.

Для разработки технологии создания ГИС карты трудоустройства выпускников вузов необходимо освоить навыки по установке и поддержанию QGIS в рабочем состоянии; по на-

стройке интерфейса и расширению возможностей системы дополнительными модулями; по управлению данными, находящимся в произвольных системах координат и ортогоналей; по осуществлению привязки сканированных карт и изображений; по созданию векторных изображений, контролю их топологии и заполнению атрибутов; по визуализации векторных слоев; по обработке и визуализации растровых изображений (оцифрованного рельефа поверхности, спутниковой съемки); по созданию и подготовке к печати карты; по работе с дополнительными источниками данных (БД, услугами WMS и WFS); использованию дополнительно других полезных приложений [2–4].

Известно, что изображения в компьютерной графике либо растровые, либо векторные. Это относится и к объектам визуализации, наносимым на карты. В качестве векторных объектов используются точки, прямые и полигоны (контуры территорий). Создается проект, в котором создаются новые слои. Можно также занести в проект созданные ранее либо находящиеся в открытом доступе таблицы слоев, используя различные форматы файлов, в том числе *.shp, включая продукты QGIS. Возможны варианты с использованием БД, слоев из различных сервисов Интернета. Тогда при передаче картографической информации с целью ее дальнейшего применения отправляется либо отдельный файл «*.shp», либо архив папки с файлами всего проекта.

Как было отмечено, геометрия хранится в отдельном поле таблицы. Если такого поля не предусмотрено, то его создают самостоятельно. К примеру, вполне возможно присоединение к проекту электронной таблицы приложения Excel, в которой хранится информация различного характера и в любом количестве, например, о работодателях в формате «*.csv», с последующим созданием в нем поля пространственных координат или конвертацией в стандарт «*.shp» с целью визуализации данных в ГИС. Но возможно и присоединение к проекту иных форматов файлов: *.csv, MapInfo с последующей конвертацией их в формат *.shp с целью произведения над ними дополнительных действий, например коррекции стиля.

Часто возникает проблема, когда код текста присоединенного слоя ошибочен. Тогда из свойств слоя подбирается подходящий вариант кодирования, и проблема будет решена. После интеграции слоев в проекте все изменения сохраняются в исходные файлы, поэтому их можно будет увидеть во всех приложениях, которые используют данные из этой таблицы.

Присоединенные к проекту слои не подвержены редакции, поэтому, если появится необходимость нанести новые слои, внести изменения в данные полей, добавлять новые поля невозможно. Чтобы открыть доступ к этим действиям надо выделить слой и нажать кнопку редактирования. Только после этих действий станет возможным его редактирование. Все текущие правки могут быть произведены только в выделенном слое. Если произошло переключение с текущего слоя на другой, то невозможно отметить на карте новый объект из прежней таблицы слоя вплоть до активации исходного слоя. Изменения, производимые в проекте, периодически необходимо сохранять.

Итак, будем исходить из того, что одна таблица – это один слой с однотипными данными. QGIS разделяет действия, связанные с хранением таблицы и управлением ее стилем. Стиль настраивается по усмотрению пользователя и, как правило, определяет такие характеристики, как цвет, тип обозначений и растров; положение и добавление надписей к соответствующим объектам и полей, коммутирующих с ними; масштаб надписей, обеспечение связи оформления слоя на карте и полей текущей или коммутирующей таблицы. Для трудоустройства выпускников вуза посредством ГИС на региональной карте разными знаками оформляются рынок работодателей, предлагаемые ими вакансии по каждой специальности. С другой стороны, можно визуализировать рынок образовательных услуг, по цветовой гамме соответствующих требованиям работодателей. Дополнительно настраиваются действия, производимые посредством щелчка мыши по топографическому обозначению с целью визуализации дополнительной информации.

Для отправки топографической информации, как правило, пользуются интернет-сервисами WMS или WFS. Интернет-услуга WMS обеспечивает отправку графической информации

о топографическом ресурсе как растры с привязкой к системе координат. Сервис WFS обеспечивает возможность запроса и редакции векторных данных, таких, например, как дорожные развязки, очертания берегов и территорий.

Из модуля QGIS «QuickMapServices», кроме услуг WMS, можно воспользоваться множеством других сервисов массовой доступности с целью визуализации топографического ресурса, для чего активизируется элемент управления «получить источники данных» из настроек – «загрузить сервисы». Тогда появится возможность воспользоваться кадастровой картой массового пользования, фото местностей Google, Yandex, станет доступной карта из «OpenStreetMap» (OSM), а также другие информационные таблицы, которые могут быть размещены в генерируемой ГИС.

Существует много дополнительных интернет-сервисов, обеспечивающих полезной информацией и дающих возможность анализа топографических элементов. В частности, в OSM можно извлечь карту с цифровыми идентификаторами всех дорог области и их параметрами, что может быть использовано для создания слоя с местонахождением работодателей и другой связанной с ними дополнительной информацией.

Применив функцию «геокодинг» к информации, сохраненной в Excel с атрибутами работодателей и предлагаемых ими вакансий, ее можно проанализировать на карте: в QGIS таблица из csv файла конвертируется в слой *.shp (модуль «RuGeocoder»). Тогда в таблице будет получено скрытое пустое поле, содержащее геометрию. Этот же продукт позволяет воспользоваться процедурой «геокодинг» с указанием готовой таблицы слоя и ее поля с адресами. В этом случае выбирается поставщик услуги – например, для работы с почтовыми адресами на русском языке выбирается Yandex. Запускается процедура геокодинга, в результате которой все обрабатываемые объекты расставляются на карте.

Местоположение объектов в ГИС определяется в заданном варианте системы координат. Например, если это широты и долготы точек на плоской модели земной поверхности – карте, то их измеряют градусами и десятичными дробями градусов. Локально же, на маленькой территории от фиксированной нулевой точки, возможно применение Декартовой системы координат, в связи с чем она и называется локальной системой координат, в которой полагается, что Земля на этой территории плоская. Поскольку такая система через несколько сот километров из-за неучтенного искривления поверхности планеты будет давать большую погрешность, то в QGIS предусмотрена возможность применения для разных слоев разных систем координат с последующей конвертацией их из одного типа в другой. Для этого слой сохраняется в «shape» файл или в БД, и в качестве параметра выбирается новая система. Но при выводе проекта на экран все слои приводят в единую координатную плоскость [5–7].

Если информация о типе системы координат отсутствует, тогда будет необходимо открыть таблицу объектов этого слоя, выделить любую строку и нажать кнопку перехода к объекту. В случае отображения на экране неправильной карты делаем вывод, что QGIS не распознал систему координат. Для этого необходимо уточнить информацию о системе координат, в которых хранятся данные, и установить ее для этого слоя в QGIS. Порой приходится самим создавать нужную систему или так называемую «пользовательскую систему координат», если таковая отсутствует. С этой целью используется запрос с наименованием искомой системы плюс, например, «система координат QGIS пользователя» в Google.

Если слои реестра землепользователей сдвигаются относительно снимков, снятых со спутника, то это озадачивает и не позволяет оценить контуры территории. Подобный казус наблюдается в случае добавления слоя реестра землепользователей массовой доступности в QGIS совместно со снимками Yandex, Google. Для исправления нюанса рекомендуется подобрать параметры для слоев карты экспериментально и создать собственную систему координат. Таким образом проблема будет решена.

Заключение

Резюмируя изложенное, можно предположить следующие перспективы использования формата ГИС карты при анализе рынка труда и трудоустройства выпускников вузов:

- ГИС карта рынка работодателей является удобным инструментом содействия трудоустройству выпускников вузов;
- для повышения эффективности ГИС карты возможна разработка клиентского приложения для мобильных устройств;
- рекомендуется создание единого информационного пространства для рынка работодателей;
- рекомендуется обеспечить автоматизацию пополнения БД появляющихся рабочих мест из доступных источников;
- ГИС технологии рекомендованы для использования при трудоустройстве выпускников вузов на всех закрепленных территориях региона.

Список литературы / References

1. **Кадирова Л. А.** Изучение зарубежного опыта маркетинговых исследований трудоустройства выпускников вузов // *International scientific review of the technical sciences, mathematics and computer sciences*. Boston, USA, 2019. P. 40–50.
Kadirova L. A. Studying of foreign experience of marketing research of graduates employment. In: *International scientific review of the technical sciences, mathematics and computer sciences*. Boston, USA, 2019, p. 40–50. (in Russ.)
2. **Некрылова Н.** Геоинформационные системы (ГИС) – это что такое? 1 апреля, 2017. URL: <http://syl.ru/article/305035>
Nekrylova N. Geographic information systems (GIS) – what is it? April, 1. 2017. URL: <http://syl.ru/article/305035> (in Russ.)
3. **Graham S. R., Carlton Ch., Jamison B.** The Benefits of Using Geographic Information Systems as a Community Assessment Tool. *Public Health Rep.*, 2011, no. 126 (2), p. 298–303. DOI 10.1177/003335491112600224
4. **Clarke K. C.** Advances in geographic information systems. *Computers, Environment and Urban Systems*, 1986, vol. 10, iss. 3–4, p. 175–184. DOI 10.1016/0198-9715(86)90006-2
5. **Nedovic-Budic Z.** The impact of GIS technology. *Environment and Planning. B: Planning and Design*, 1998, September, no. 25 (5), p. 681–692. DOI 10.1068/b250681
6. **Quy Touya Thy Truong, Cyril de Runz, Guillaume Touya.** Analysis of collaboration networks in OpenStreetMap through weighted social multigraph mining. *International Journal of Geographical International Science*, 2018. DOI 10.1080/13658816.2018.1556395
7. **Asiam Nilufer Sari, Cheng T., Cheshire J.** A high – precision heuristic model to detect home and work locations from smart card data. *Geo-spatial Information Science*, 2019, vol. 22, no. 1, p. 1–11. DOI 10.1080/10095020.2018.1545884

*Материал поступил в редколлегию
Received
21.03.2019*

Сведения об авторе / Information about the Author

Кадирова Лола Алимджановна, старший преподаватель кафедры информационных технологий Андижанского государственного университета (ул. Университетская, 129, Андижан, 170100, Республика Узбекистан)

Lola A. Kadirova, Senior Lecturer, Department of Information Technology, Andijan State University (129 Universitetskaya Str., Andijan, 170100, Republic of Uzbekistan)

Lola_temur@mail.ru

Researcher ID: F-5632-2019

**Код «Виртуальный планетарий»
для моделирования астрофизических объектов:
математическая модель, методология и первые результаты**

И. М. Куликов, В. А. Протасов

*Институт вычислительной математики и математической геофизики СО РАН
Новосибирск, Россия*

Аннотация

Сложность астрофизических процессов заключается в совместном рассмотрении компонент различной природы. Так, например, в задаче столкновения галактик рассматривается трехмерная динамика межзвездного газа и звездной компоненты. В основе моделирования этих компонент могут лежать совершенно разные классы численных методов. Одним из возможных решений этой проблемы является использование эйлерово-лагранжева подхода, в котором физические величины сосредоточены в материальных точках, что характерно для метода SPH (Smoothed Particle Hydrodynamics), а расчет сил производится на адаптивной сетке, привязанной к системе материальных точек. Такой подход единообразно учитывает динамику как сплошной среды, так и дискретных частиц, а также позволяет устранить ряд недостатков, присущих оригинальному методу. Расчет гравитационного взаимодействия осуществляется путем решения уравнения Пуассона для гравитационного потенциала. При этом все частицы проецируются на расчетную сетку, и уже по ней вычисляются значения потенциала в каждой ячейке. Решение уравнения Пуассона для гравитационного потенциала выполняется с использованием быстрого преобразования Фурье. В статье описан новый код «Виртуальный планетарий» для моделирования астрофизических объектов на основе метода SPH, дополненного методом Годунова для вычисления потоков давления и импульса между частицами и методом быстрого преобразования Фурье для решения уравнения Пуассона для гравитационного потенциала. В работе описано обоснование для перехода к такой вычислительной модели, детально описаны кинетические и гидродинамические подходы. Выполнено моделирование коллапса изотермического газового облака, показана возможность метода воспроизводить развитие неустойчивостей в виде образования двух рукавов плотности.

Ключевые слова

вычислительная астрофизика, задача многих тел, уравнение Пуассона

Благодарности

Работа поддержана грантом Российского фонда фундаментальных исследований (проект № 18-41-543012) и бюджетной тематикой ИВМиМГ СО РАН (проект № 0315-2016-0009)

Для цитирования

Куликов И. М., Протасов В. А. Код «Виртуальный планетарий» для моделирования астрофизических объектов: математическая модель, методология и первые результаты // Вестник НГУ. Серия: Информационные технологии. 2019. Т. 17, № 2. С. 81–90. DOI 10.25205/1818-7900-2019-17-2-81-90

**“Virtual Planetarium” Code for Astrophysical Objects Modeling:
Mathematical Model, Methodology and First Results**

I. M. Kulikov, V. A. Protasov

*Institute of Computational Mathematics and Mathematical Geophysics SB RAS,
Novosibirsk, Russian Federation*

Abstract

The complexity of astrophysical processes lies in simultaneous consideration of components with different nature. So, for example, in the problem of collision of galaxies the three-dimensional dynamics of interstellar gas and stellar component is considered. The modeling of these components could be based on completely different classes of numerical methods. One possible solution to this problem is the use of the Eulerian-Lagrangian approach, in which the

physical quantities are concentrated in material points, which is typical for the SPH method, and the calculation of forces is made on an adaptive grid bound to the system of material points. This approach uniformly takes into account both the dynamics of the continuous medium and discrete particles, and also eliminates a number of disadvantages typical for the original method. The calculation of the gravitational interaction is carried out by solving the Poisson equation for the gravitational potential. Herewith, all the particles are projected onto the computational grid and the potential values in each cell are calculated using this grid. The solution of the Poisson equation for the gravitational potential is performed using Fast Fourier Transform. The new “Virtual Planetarium” code for astrophysical objects modeling based on SPH method, supplemented by Godunov method for calculating pressure and momentum flows between particles, and Fast Fourier Transform method to solve the Poisson equation for the gravitational potential, is described in the paper. Rationale for the transition to such numerical model is given in the paper. Kinetic and hydrodynamic approaches are described in detail. The modeling of collapse of an isothermal gas cloud is performed, the ability of the method to reproduce the development of instabilities in the form of the formation of two density sleeves is shown.

Keywords

numerical astrophysics, N-body problem, Poisson equation

Acknowledgment

The work is supported by the grant of RFBR (project 18-41-543012) and budget thematic of ICMMG SB RAS (project 0315-2016-0009)

For citation

Kulikov I. M., Protasov V. A. “Virtual Planetarium” Code for Astrophysical Objects Modeling: Mathematical Model, Methodology and First Results. *Vestnik NSU. Series: Information Technologies*, 2019, vol. 17, no. 2, p. 81–90. (in Russ.) DOI 10.25205/1818-7900-2019-17-2-81-90

Введение

Современная теоретическая астрофизика практически исчерпала возможности аналитических и полуаналитических решений, которые можно получить при допущениях пространственной симметрии или рассматриваемых процессов. В настоящее время основным инструментом теоретической астрофизики стал вычислительный эксперимент. Сложность астрофизических процессов заключается в совместном рассмотрении компонент различной природы. Так, например, в задаче столкновения галактик рассматривается трехмерная динамика межзвездного газа и звездной компоненты [1]. Для описания такого процесса требуется разрешение уравнений гидродинамики для описания движения газа и решение кинетического уравнения для записи движения звезд. В основе методов решения лежат совершенно разные классы численных методов. Один из трендов для разрешения подобных проблем заключается в использовании лагранжево-эйлеровых методов (Arbitrary Lagrangian-Eulerian – ALE в зарубежной литературе), которые одновременно подходят как для решения уравнений гидродинамики, так и для описания движения частиц. Хотя следует отметить, что существуют работы по модификации математических моделей для описания различных компонент единым классом гиперболических уравнений [2]. Мы остановимся на развитии лагранжево-эйлерового подхода.

В последнее десятилетие в области вычислительной астрофизики было создано несколько кодов, успешно и в то же время различным образом реализующих лагранжево-эйлеровый подход. Среди них коды AREPO [3], VETHE-HYDRO [4], GIZMO [5]. Так или иначе, все коды основаны на представлении решения в виде набора материальных точек с размещенными в них физическими характеристиками, что характерно для метода сглаженных частиц [6] (Smooth Particles Hydrodynamics – SPH в зарубежной литературе). Далее используются различные механизмы способа взаимодействия с соседними точками, в частности способ построения сеток. Остановимся на описании каждого из кодов чуть подробнее.

В основе кода AREPO лежит технология подвижных сеток на основе триангуляции Вороного и Делоне с регуляризацией Ллойда. Такой подход позволяет адаптировать сетку под решение. В качестве основного метода решения уравнений гидродинамики используется классический метод Годунова. Это связано с тем, что достаточно тяжело построить схему

более высокого порядка на подвижной сетке. Для решения уравнения Пуассона используется подход на основе записи уравнения для полной механической энергии в уравнение для суммы всех видов энергии (внутренняя, кинетическая и потенциальная). Такое уравнение для полной энергии имеет в правой части производную по времени от потенциала и его градиента, что вычисляется с помощью интеграла Пуассона методами типа «частица – сетка». Для интегрирования по времени используется индивидуальный шаг по времени для различных ячеек. При всех достоинствах такого подхода он достаточно тяжелый в плане вычислительных затрат, а также остается открытым вопрос о качестве решения в областях, описываемых менее подробными сеточными ячейками. Однако код AREPO является одним из наиболее используемых в мире в данный момент.

В основе кода VETHE-HYDRO лежит ALE-подход, сочетающий достоинства как эйлерова так и лагранжева подходов. Уравнения гидродинамики формулируются в лагранжевой неконсервативной форме и решаются на неструктурированной сетке. В основе численного метода – операторный подход, который позволяет построить согласованные схемы для аппроксимации операторов «градиент» и «дивергенция». Для решения уравнения Пуассона в одномерной постановке используется метод прогонки (или метод Томаса в зарубежной литературе). В двумерной постановке уравнение Пуассона решается с помощью метода сопряженных градиентов. Далее происходит коррекция потенциала для сохранения полной энергии (сумма кинетической, внутренней и потенциальной энергий) системы. Отметим, что сохранить полную энергию системы все равно не удастся, но ошибка на задаче коллапса составляет порядка 10^{-2} процента, что очень незначительно. К сожалению, код не был развит для трехмерного случая.

В программном коде GIZMO разработан и реализован новый бессеточный подход к решению уравнений гравитационной газовой динамики. Подход основан на комбинации классических сеточных методов и метода SPH. Метод состоит в использовании уравнений газовой динамики в эйлеровых координатах, которые с использованием вариационного принципа Галеркина умножаются на пробные функции. Особенностью этих функций является то, что они привязаны не к расчетной сетке, а к отдельным частицам, аналогичным по своей природе SPH частицам. Для определения значений на границах области используется решение задачи Римана с использованием метода Годунова.

Анализ этих кодов показывает, что основной вычислительной проблемой является построение сетки (или ее аналога) для описания взаимодействия между частицами. При этом во всех методах сохранена идеология кинетического подхода на основе описания движения материальных точек. В этой концепции разработана вычислительная модель, описанная и в данной статье. Для организации вычислений мы будем использовать равномерную кубическую сетку, в которой будет помещаться облако частиц. Такой подход позволяет достаточно просто организовать эйлеров этап вычислений.

Вычислительная модель

Описание вычислительной модели мы разобьем на две части. Сначала опишем динамику частиц на основе сил гравитации. Эта модель соответствует газу с нулевым давлением (pressureless hydrodynamics в зарубежной литературе). Затем добавим силы давления при использовании адиабатического уравнения состояния.

Кинетическая модель

Запишем уравнения гравитационной гидродинамики с нулевым давлением в виде законов сохранения массы и момента импульса в эйлеровых координатах:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{u}) = 0, \quad (1)$$

$$\frac{\partial \rho \vec{u}}{\partial t} + \nabla \cdot (\rho \vec{u} \times \vec{u}) = -\rho \nabla \Phi, \quad (2)$$

дополненные уравнением Пуассона для гравитационного потенциала:

$$\Delta \Phi = 4\pi G \rho,$$

где ρ – плотность, \vec{u} – вектор скорости, Φ – гравитационный потенциал, G – гравитационная постоянная.

Будем рассматривать динамику материальной точки (частицы) с произвольным номером n . Закон сохранения массы частицы получается естественным образом при интегрировании уравнения (1) в малой окрестности точки, в которую не попадают другие частицы. Второй закон Ньютона для частицы с номером n запишем, используя уравнение (2) в лагранжевой постановке:

$$\rho \frac{d\vec{u}}{dt} = -\rho \nabla \Phi. \quad (3)$$

В уравнении (3) необходимо сократить функцию плотности, которая в малой окрестности точки отлична от нуля. В результате получим уравнения для движения частиц:

$$\frac{d\vec{u}}{dt} = -\nabla \Phi, \quad \frac{d\vec{x}}{dt} = \vec{u}. \quad (4)$$

Для решения уравнений (4) нам необходимо вычислить градиент потенциала. Для этого введем равномерную кубическую эйлерову сетку с шагом h по каждому направлению, и каждую частицу спроецируем на соответствующую ячейку. Другими словами, каждая частица n будет давать вклад в массу соответствующей ячейки (i, j, k) . Таким образом, в каждой ячейке мы можем вычислить плотность по формуле

$$\rho_{ijk} = \frac{\sum_l M_l}{h^3}, \quad (5)$$

где M_l – это масса частицы, попадающей в ячейку (i, j, k) . Зная плотность в каждой ячейке расчетной области, с помощью метода, основанного на использовании быстрого преобразования Фурье, мы можем найти гравитационный потенциал. Для этого необходимо с помощью прямого преобразования Фурье перейти от функции плотности к ее амплитудам в пространстве гармоник σ_{pqr} , затем перейти к амплитудам гармоник потенциала ϕ_{pqr} с помощью формулы

$$\phi_{pqr} = h^2 \frac{\frac{2}{3} \pi G}{1 - \left(1 - \frac{2 \sin^2 \frac{\pi p}{I}}{3}\right) \left(1 - \frac{2 \sin^2 \frac{\pi q}{K}}{3}\right) \left(1 - \frac{2 \sin^2 \frac{\pi r}{L}}{3}\right)} \sigma_{pqr}, \quad (6)$$

где I, K, L – размер расчетной сетки по каждому из направлений (в настоящей статье эти величины принимаются равными, так как используется равномерная кубическая сетка). Затем с помощью обратного преобразования Фурье из амплитуд гармоник потенциала восстанавливаются значения функции потенциала в соответствующих ячейках Φ_{ijk} . Исследование и описание метода решения уравнения Пуассона подробно описано в работах [7; 8]. Так как

функция гравитационного потенциала суть гладкая функция, то для нахождения градиента потенциала мы будем использовать центральные разности:

$$\left. \frac{\partial \Phi}{\partial x} \right|_{ijk} = \frac{\Phi_{i+1,j,k} - \Phi_{i-1,j,k}}{2h} \quad \left. \frac{\partial \Phi}{\partial y} \right|_{ijk} = \frac{\Phi_{i,j+1,k} - \Phi_{i,j-1,k}}{2h},$$

$$\left. \frac{\partial \Phi}{\partial z} \right|_{ijk} = \frac{\Phi_{i,j,k+1} - \Phi_{i,j,k-1}}{2h}.$$

Для разрешения уравнений (4) для частицы n на временном шаге m мы будем использовать двухслойную схему с перешагиванием, где скорости вычисляются по формуле

$$u_x^{m+1/2} = u_x^m - \frac{\tau}{2h} (\Phi_{i+1,j,k} - \Phi_{i-1,j,k}),$$

$$u_y^{m+1/2} = u_y^m - \frac{\tau}{2h} (\Phi_{i,j+1,k} - \Phi_{i,j-1,k}), \quad (7)$$

$$u_z^{m+1/2} = u_z^m - \frac{\tau}{2h} (\Phi_{i,j,k+1} - \Phi_{i,j,k-1}),$$

а координаты – с использованием формулы

$$x^{m+1} = x^m + \tau u_x^{m+1/2}, \quad y^{m+1} = y^m + \tau u_y^{m+1/2}, \quad z^{m+1} = z^m + \tau u_z^{m+1/2}. \quad (8)$$

Такая схема соответствует эйлерово-лагранжеву подходу, использованному коллективом авторов ранее [2; 7; 8].

Важным условием устойчивости численной схемы является выбор шага по времени. Для этого будем использовать условие Куранта для введенной сетки:

$$\tau = \frac{CFL}{\max_n |\vec{u}_n|} h, \quad (9)$$

где $CFL < 1$ – число Куранта – Фридрихса – Леви.

Гидродинамическая модель

При использовании гидродинамической модели вычислительная процедура осложняется введением новой силы – давления. Основная сложность заключается в том, что функция давления в общем случае – разрывная функция, и для определения его градиента необходимо использовать решение задачи о распаде разрыва. Последовательно опишем изменения в вычислительной модели.

Уравнения гравитационной гидродинамики в виде законов сохранения массы и момента импульса в эйлеровых координатах записываются в виде

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{u}) = 0, \quad (10)$$

$$\frac{\partial \rho \vec{u}}{\partial t} + \nabla \cdot (\rho \vec{u} \times \vec{u}) = -\nabla p - \rho \nabla \Phi. \quad (11)$$

Уравнения (10), (11) дополнены уравнением Пуассона для гравитационного потенциала (3):

$$\Delta \Phi = 4\pi G \rho,$$

где ρ – плотность, \vec{u} – вектор скорости, p – давление, Φ – гравитационный потенциал, G – гравитационная постоянная. Уравнения замкнуты уравнением состояния для адиабатического газа:

$$p = S\rho^\gamma,$$

где γ – показатель адиабаты, S – значение энтропии.

Выше было описано, как получить плотность газа в ячейке и восстановить гравитационный потенциал. Опустив очевидные выкладки, уравнения (4) перепишем в виде

$$\frac{d\vec{u}}{dt} = -\frac{\nabla p}{\rho} - \nabla\Phi, \quad \frac{d\vec{x}}{dt} = \vec{u}. \quad (12)$$

Для разрешения уравнений (12) для частицы n на временном шаге m мы будем использовать двухслойную схему с перешагиванием, где, сохранив способ пересчета координат (8), изменим формулу для определения скорости (7), которая записывается в виде

$$\begin{aligned} u_x^{m+1/2} &= u_x^m - \frac{\tau}{2h} (\Phi_{i+1,j,k} - \Phi_{i-1,j,k}) - \frac{\tau}{h} (P_{i+1/2,j,k} - P_{i-1/2,j,k}), \\ u_y^{m+1/2} &= u_y^m - \frac{\tau}{2h} (\Phi_{i,j+1,k} - \Phi_{i,j-1,k}) - \frac{\tau}{h} (P_{i,j+1/2,k} - P_{i,j-1/2,k}), \\ u_z^{m+1/2} &= u_z^m - \frac{\tau}{2h} (\Phi_{i,j,k+1} - \Phi_{i,j,k-1}) - \frac{\tau}{h} (P_{i,j,k+1/2} - P_{i,j,k-1/2}), \end{aligned} \quad (13)$$

где величины $P_{i\pm 1/2, j\pm 1/2, k\pm 1/2}$ – поток давления, получаемый в результате решения задачи о распаде разрыва для эйлерова этапа численного метода. Для аппроксимации потока рассмотрим комбинацию правой (R – *right*) и левой (L – *left*) расчетных ячеек, получим [7]

$$P = \frac{p_L + p_R}{2} + \frac{u_L - u_R}{2} \sqrt{\gamma \frac{p_L + p_R}{\rho_L + \rho_R}}. \quad (14)$$

В случае адиабатического уравнения состояния (в этом случае величина энтропии суть постоянная величина для всех материальных точек), уравнение (14) записывается в виде

$$P = S \frac{\rho_L^\gamma + \rho_R^\gamma}{2} + \frac{u_L - u_R}{2} \sqrt{\gamma S \frac{\rho_L^\gamma + \rho_R^\gamma}{\rho_L + \rho_R}}.$$

В случае изотермического уравнения состояния (суть энтропии S – температура T , показатель адиабаты $\gamma = 1$) уравнение (14) записывается в еще более простой форме:

$$P = T \frac{\rho_L + \rho_R}{2} + \frac{u_L - u_R}{2} \sqrt{T}.$$

Перепишем уравнения (13) для изотермического газа:

$$\begin{aligned} u_x^{m+1/2} &= u_x^m - \frac{\tau}{2h} ([\Phi_{i+1,j,k} - \Phi_{i-1,j,k}] - T[\rho_{i+1,j,k} - \rho_{i-1,j,k}]) + \sqrt{T} [u_{x,i+1,j,k} - 2u_{x,ijk} + u_{x,i-1,j,k}], \\ u_y^{m+1/2} &= u_y^m - \frac{\tau}{2h} ([\Phi_{i,j+1,k} - \Phi_{i,j-1,k}] - T[\rho_{i,j+1,k} - \rho_{i,j-1,k}]) + \sqrt{T} [u_{y,i,j+1,k} - 2u_{y,ijk} + u_{y,i,j-1,k}], \\ u_z^{m+1/2} &= u_z^m - \frac{\tau}{2h} ([\Phi_{i,j,k+1} - \Phi_{i,j,k-1}] - T[\rho_{i,j,k+1} - \rho_{i,j,k-1}]) + \sqrt{T} [u_{z,i,j,k+1} - 2u_{z,ijk} + u_{z,i,j,k-1}]. \end{aligned} \quad (15)$$

Как уже говорилось ранее, формула пересчета координат (8) осталась без изменений. Важным моментом в формуле (15) является проекция скоростей материальных точек в ячейку. Для этого мы будем использовать весовую функцию по массе точек:

$$u_{x,ijk} = \frac{\sum_l u_{x,l} M_l}{\sum_l M_l}, \quad u_{y,ijk} = \frac{\sum_l u_{y,l} M_l}{\sum_l M_l}, \quad u_{z,ijk} = \frac{\sum_l u_{z,l} M_l}{\sum_l M_l}, \quad (16)$$

где M_l и u_l – это масса и соответствующие компоненты скорости всех частиц, попадающих в ячейку (i, j, k) . Далее перейдем к основным этапам реализации программного кода.

Структура кода

В этом разделе приведем описание алгоритма, приведя все расчетные формулы для каждого этапа. Не снижая общности, мы остановимся на случае периодических граничных условий и изотермическом процессе. Также будем считать, что трехмерное преобразование Фурье уже реализовано в библиотеке (например, в работе [9]) в виде процедуры FFT3D.

1. Определим значения: L – размер расчетной области, N – размер расчетной сетки, D – количество материальных точек, T – температура газа, $h = L / N$ – шаг по пространству, определим CFL – число Куранта, M – масса каждой частицы. Моделировать динамику газа будем до момента времени $TimeMax$.

2. Для всех частиц определим координаты \vec{x} и вектор скорости \vec{u} .

3. Обнулим время динамики системы $Time = 0$.

4. Если $Time < TimeMax$ переходим к шагу 5, в противном случае процесс моделирования заканчиваем.

5. Для каждой ячейки (i, j, k) определяем ее плотность по формуле (5).

6. В случае использования гидродинамической модели нам необходима сеточная функция скорости. Для этого в каждой ячейке (i, j, k) определяем компоненты скорости по формулам (16).

7. Определяем шаг по времени из условия (9).

$$\tau = \frac{CFL}{\max_n |\vec{u}_n| + \sqrt{T}} h.$$

8. Прибавляем к текущему времени временной шаг $Time = Time + \tau$.

9. Делаем прямое преобразование Фурье для функции плотности:

$$\sigma_{pqr} = FFT3D(\rho_{ijk}).$$

10. Решаем уравнение Пуассона в пространстве гармоник по формуле (6).

11. Делаем обратное преобразование Фурье для гармоник потенциала:

$$\Phi_{ijk} = FFT3D^{-1}(\phi_{pqr}).$$

12. Для каждой частицы пересчитываем ее скорость по формуле (15).

13. Для каждой частицы пересчитываем ее координату с использованием формулы (8) и полученных скоростей.

14. Переходим на шаг 4.

Как уже было сказано, при использовании кинетической модели шаг 6 пропускается. Далее мы обсудим дискуссионные вопросы по вычислительной модели и перспективы развития модели для более сложных случаев.

Дискуссия

Опишем дискуссионные моменты, связанные с построенной вычислительной моделью и программной реализацией. Эти моменты пока не включены в код, однако могут быть использованы в развитии кода.

1. Во введении было сказано об использовании только равномерной кубической сетки. Это сделано только для упрощения описания вычислительной модели и ее использования для нужд новосибирского планетария. Конечно, использование материальных точек позволяет нам достаточно простым и естественным образом вводить адаптивные сетки, используя соответствующие численные методы для восстановления гравитационного потенциала. В дальнейшем мы планируем развить наш подход в этом направлении.

2. Мы использовали адиабатическое и изотермическое уравнения состояния, которые зависят только от функции плотности. Для проекции частиц на эйлерову сетку для определения плотности достаточно использовать закон сохранения масс. Однако для описания идеального газа необходимо использовать термодинамическую переменную – энтропию, и способ проекции, аналогичный проекции скоростей материальных точек на ячейку.

3. Для нахождения потока давления через границу использовалась схема первого порядка точности из работы [7]. Использование равномерной сетки позволяет достаточно просто повысить порядок точности схемы, например, как это было сделано в работе [8].

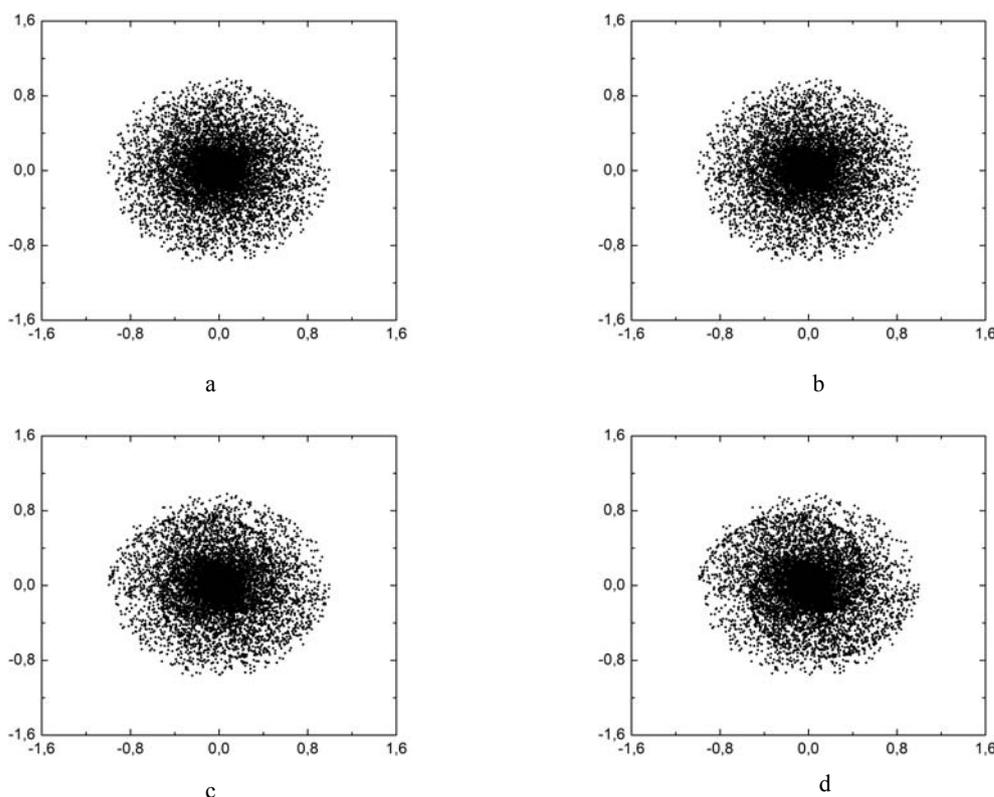
4. Также в статье была использована простейшая проекция материальных точек на расчетную сетку. Для устранения вычислительных шумов необходимо использовать ядра сглаживания, аналогичные методам SPH и методам типа «частиц-в-ячейках». В дальнейшем при необходимости вопрос выбора ядра будет рассмотрен отдельно.

5. Формулы пересчета скоростей и координат материальных точек (7), (8) и (15) явно записываются в векторной форме, что делает логичным использование векторных SSE расширений, встроенных в процессоры Intel и AMD современных персональных компьютеров, для которых и рассчитан программный код из этой статьи. Также большинство персональных компьютеров оснащено графическими ускорителями NVIDIA, для которых в рамках технологии CUDA созданы специальные библиотеки для расчета быстрого преобразования Фурье (например, cuFFT).

Результаты моделирования

В качестве модельной задачи будем рассматривать коллапс изотермического облака в безразмерных переменных. Для этого в кубической области размером $L = 3,2$ введем вспомогательную расчетную сетку 128^3 . В начальный момент времени в сфере единичного радиуса равномерно распределим частицы единичной суммарной массы с модельной температурой $T = 0,1$ и кеплеровской скоростью вращения. В нашей модельной задаче мы будем исследовать поведение ансамбля частиц газа при различных значениях дисперсии скоростей δ . Проблема исследования фрагментации вращающегося холодного облака особенно интересна в контексте эволюции эллиптических галактик [10].

Анализ результатов моделирования при различных δ показывает (см. рисунок), что при увеличении дисперсии скоростей до $\delta = 0,5$ для одинакового начального распределения плотности имеет место развитие неустойчивостей в виде двух рукавов плотности. В рамках настоящей статьи мы не планируем проводить анализ развития неустойчивостей, подобных описанным в работе [8]. Основной результат вычислительных экспериментов – это возможность воспроизведения подобных возмущений разработанной вычислительной моделью. В дальнейшем мы более подробно рассмотрим развитие неустойчивостей в системе многих тел в контексте моделирования планетных систем и галактик [11].



Эволюция вращающегося холодного облака при различных значениях дисперсии скоростей:
 $\delta = 0,1$ (a), $\delta = 0,3$ (b), $\delta = 0,5$ (c), $\delta = 0,8$ (d)
Evolution of the rotating cold cloud with different dispersion of velocities:
 $\delta = 0.1$ (a), $\delta = 0.3$ (b), $\delta = 0.5$ (c), $\delta = 0.8$ (d)

Заключение

В статье приведено описание нового кода «Виртуальный планетарий» для моделирования астрофизических объектов. Код основан на эйлерово-лагранжевой комбинации метода сглаженных частиц и метода Годунова. В работе описано обоснование для перехода к такой вычислительной модели. С помощью разработанного кода смоделирован коллапс вращающегося газового облака.

Список литературы / References

1. **Тутуков А. В., Лазарева Г. Г., Куликов И. М.** Газодинамика центрального столкновения двух галактик: слияние, разрушение, пролет, образование новой галактики // *Астрономический журнал*. 2011. Т. 88, № 9. С. 837–851.
Tutukov A. V., Lazareva G. G., Kulikov I. M. Gas dynamics of a central collision of two galaxies: Merger, disruption, passage, and the formation of a new galaxy. *Astronomy Reports*, 2011, vol. 55, no. 9, p. 770–783.
2. **Kulikov I.** GPUPEGAS: A New GPU-accelerated Hydrodynamic Code for Numerical Simulations of Interacting Galaxies. *The Astrophysical Journal. Supplements Series*, 2014, vol. 214, I.1, article number 12.

3. **Springel V.** E pur si muove: Galilean-invariant cosmological hydrodynamical simulations on a moving mesh. *Monthly Notices of the Royal Astronomical Society*, 2010, vol. 401, p. 791–851.
4. **Murphy J., Burrows A.** BETHE-Hydro: An Arbitrary Lagrangian-Eulerian Multidimensional Hydrodynamics Code for Astrophysical Simulations. *The Astrophysical Journal. Supplement Series*, 2008, vol. 179, p. 209–241.
5. **Hopkins P.** A new class of accurate, mesh-free hydrodynamic simulation methods. *Monthly Notices of the Royal Astronomical Society*, 2015, vol. 450, I.1, p. 53–110.
6. **Monaghan J.** Smoothed Particle Hydrodynamics. *Rep. Prog. Phys.*, 2005, vol. 68, p. 1703–1759.
7. **Vshivkov V., Lazareva G., Snytnikov A., Kulikov I., Tutukov A.** Hydrodynamical code for numerical simulation of the gas components of colliding galaxies. *The Astrophysical Journal. Supplement Series*, 2011, vol. 194, I.2, article number 47.
8. **Kulikov I., Vorobyov E.** Using the PPML approach for constructing a low-dissipation, operator-splitting scheme for numerical simulations of hydrodynamic flows. *Journal of Computational Physics*, 2016, vol. 317, p. 318–346.
9. **Frigo M., Johnson S.** The Design and Implementation of FFTW3. In: Proceedings of the IEEE, 2005, vol. 93, I.2, p. 216–231.
10. **Worakitpoonpon T.** Spherical symmetry breaking in cold gravitational collapse of isolated systems. *Monthly Notices of the Royal Astronomical Society*, 2015, vol. 446, p. 1335–1346.
11. **Benhaïem D., Sylos Labini F., Joyce M.** Long-lived transient structure in collisionless self-gravitating systems. *Physics Review E*, 2019, vol. 99, article number 022125.

Материал поступил в редколлегию
Received
18.02.2019

Информация об авторах / Information about the Authors

Куликов Игорь Михайлович, доктор физико-математических наук, старший научный сотрудник, Институт вычислительной математики и математической геофизики СО РАН (пр. Академика Лаврентьева, 6, Новосибирск, 630090, Россия)

Igor M. Kulikov, Doctor of Science in Physics and Mathematics, Senior Research Scientist, Institute of Computational Mathematics and Mathematical Geophysics SB RAS (6 Academician Lavrentiev Ave., Novosibirsk, 630090, Russian Federation)

kulikov@ssd.sccc.ru
ORCID 0000-0002-1959-780X

Протасов Виктор Александрович, инженер, Институт вычислительной математики и математической геофизики СО РАН (пр. Академика Лаврентьева, 6, Новосибирск, 630090, Россия)

Viktor A. Protasov, Engineer, Institute of Computational Mathematics and Mathematical Geophysics SB RAS (6 Academician Lavrentiev Ave., Novosibirsk, 630090, Russian Federation)

inc_13@mail.ru
ORCID 0000-0003-4221-4480

УДК 004.9
DOI 10.25205/1818-7900-2019-17-2-91-100

Разработка среды программирования для мобильных устройств под управлением операционной системы Android

Р. К. Лебедев, И. А. Корякин

*Новосибирский государственный университет
Новосибирск, Россия*

Аннотация

Мобильные устройства часто используются для образовательных целей в связи с их высокой доступностью и портативностью. Однако образовательные возможности мобильных устройств ограничены из-за отсутствия качественных IDE. Большинство сред разработки языка C++ для Android не имеют автодополнения и анализа кода, а остальные предоставляют эти возможности с крайне низкой производительностью. В этой работе мы описываем технологию переноса программного обеспечения, примененную к компилятору Clang (и другим необходимым инструментам), и предлагаем метод ускорения компиляции, автодополнения и анализа кода в нашей IDE. Наряду с этим мы предлагаем методы оптимизации пользовательского интерфейса. В работе описан основанный на регулярных выражениях эвристический метод для генерации предкомпилированных заголовков, который имеет низкие требования к процессору и ОЗУ. Эффективность метода проверена на различных образовательных примерах.

Ключевые слова

портирование ПО, Android, оптимизация, мобильные устройства, образование, IDE

Для цитирования

Лебедев Р. К., Корякин И. А. Разработка среды программирования для мобильных устройств под управлением операционной системы Android // Вестник НГУ. Серия: Информационные технологии. 2019. Т. 17, № 2. С. 91–100. DOI 10.25205/1818-7900-2019-17-2-91-100

Development of IDE for Android Devices

R. K. Lebedev, I. A. Koryakin

*Novosibirsk State University
Novosibirsk, Russian Federation*

Abstract

Mobile devices are frequently used for educational purposes considering their high availability and portability. However, educational abilities of mobile devices are quite limited due to the lack of proper IDE applications. Most Android integrated development environments for C++ are lacking code completion and real-time code analysis, while others provide these features at unacceptable speeds.

Therefore, we decided to develop a next-generation mobile C++ IDE that will not only have these features, but will also provide them at real-time speeds. This will make mobile devices more suitable for learning C++ and will provide students with an opportunity to leave their laptops at home and to learn C++ in different situations, where usage of laptop is not convenient.

In this paper, we describe the software porting technology applied to Clang compiler and other required tools and propose the method to speed up compilation, code completion and code analysis in our IDE. We also propose UI optimization methods. We also describe a regular expression-based heuristic algorithm for precompiled header generation that has small processor time and RAM requirements. The efficiency of the algorithm is tested on different educational code examples.

Our experiments show that our IDE is the only one at the moment, that can keep code completion real-time in all usage scenarios. The performance gain is 300 % average, when STL is used, reaching 3350 % when Boost library is involved. There is a small performance loss when C is used, but it is not significant, as the performance is still acceptable for real-time usage.

© Р. К. Лебедев, И. А. Корякин, 2019

Keywords

software porting, Android, optimization, mobile, education, IDE

*For citation*Lebedev R. K., Koryakin I. A. Development of IDE for Android Devices. *Vestnik NSU. Series: Information Technologies*, 2019, vol. 17, no. 2, p. 91–100. (in Russ.) DOI 10.25205/1818-7900-2019-17-2-91-100**Введение**

Развитие мобильных устройств приводит к значительному расширению числа областей, в которых они могут использоваться. Одной из таких новых областей является мобильное образование. В магазинах приложений популярных мобильных операционных систем появились приложения, обучающие языкам, точным и гуманитарным наукам [1]. Присутствуют в магазинах и приложения, обучающие различным языкам программирования.

Растущая производительность мобильных устройств позволила перенести на них и среды разработки языков программирования. В частности, существуют такие приложения, как Ruidroid, интегрированная среда разработки языка Python, или C4droid, среда разработки языка C++. Данные приложения позволяют обучающимся избежать необходимости брать с собой на место учебы ноутбук или же и вовсе заменяют персональный компьютер, что наиболее важно в странах, где учащиеся зачастую могут позволить себе смартфон, но не могут приобрести ноутбук. Существуют и случаи замены персональных компьютеров в терминальных классах на планшеты под управлением операционной системы Android с установленной мобильной средой разработки¹.

Как правило, под интегрированной средой разработки понимают программный продукт, обладающий следующими возможностями:

- редактирование текста;
- компиляция или интерпретация исходного кода;
- подсветка синтаксиса;
- автоматическое дополнение кода;
- проверка ошибок во время редактирования;
- установка и подключение сторонних программных библиотек.

Некоторые из данных возможностей требовательны к ресурсам процессора. В случае языка C++ для автоматического дополнения кода и проверки ошибок в реальном времени требуется произвести большую часть шагов, обычно выполняемых при компиляции кода, за исключением связывания и трансляции программы в язык ассемблера. Подходы, используемые в средах программирования, предназначенных для рабочих станций, зачастую неприменимы на мобильных устройствах, поскольку в силу меньшей производительности последних не обеспечивают комфортной работы в режиме реального времени.

Существуют и особенности, связанные с размером экранов мобильных устройств: многим интегрированным средам разработки для обычных компьютеров свойственно наличие большого числа пунктов меню и кнопок, которые невозможно разместить в мобильном приложении [2].

Операционная система Android использует свою стандартную библиотеку языка Си, Bionic. По этой причине пользователи не могут работать со статическими или динамическими библиотеками, загруженными с их официальных сайтов: количество библиотек, собранных для Bionic, очень мало, большинство библиотек собраны для библиотек языка Си популярных дистрибутивов Linux или операционной системы Windows. Единственным оставшимся путем является самостоятельная сборка библиотек из исходного кода, что затруднительно для целевой аудитории программ такого рода: студентов и школьников.

Целью данной работы является разработка мобильной среды программирования языка C++, обладающей средствами автодополнения кода и проверки ошибок в режиме реального

¹ BVBCET (Индия) в 2012 г. лицензировал 900 копий приложения C4droid для использования на планшетных ПК.

времени, а также удобным для мобильного устройства интерфейсом и возможностью легкой установки популярных библиотек.

В работе описывается процесс переноса необходимых программных библиотек и программ на платформу Android, описан процесс сборки библиотек для программ конечного пользователя, а также рассмотрены методы оптимизации пользовательского интерфейса. Наряду с этим анализируются существующие подходы для ускорения автодополнения и анализа кода, а также описывается новый алгоритм, разработанный в рамках данной работы. Также приводится сравнение производительности разработанной среды программирования со всеми популярными конкурентами.

Выбор библиотек и программ для среды разработки языка C++

Типичным для программирования является следующий набор программных инструментов:

- командный интерпретатор;
- компилятор;
- набор утилит для работы с исполняемыми файлами;
- система автоматизации сборки программ из исходного кода.

В качестве командного интерпретатора был выбран проект BusyBox, поскольку он обладает небольшими требованиями к производительности, а также позволяет встроить базовый набор утилит командной строки, которые могут присутствовать не на всех версиях операционной системы Android, решая проблему совместимости.

В качестве компилятора был выбран проект Clang/LLVM, поскольку он предоставляет наиболее широкие возможности по анализу и автодополнению кода, а также обладает удобной лицензией.

Для работы с исполняемыми файлами был выбран проект GNU Binutils, поскольку он является самым популярным и стабильным проектом такого рода. В первую очередь данный проект обеспечивает связывание объектных файлов в исполняемый файл при помощи линкера.

Системой сборки, используемой для проектов, состоящих из нескольких файлов, была выбрана пара CMake и GNU Make. CMake легче в освоении для начинающих пользователей и позиционируется как более дружелюбный к кросс-компиляции, что обеспечивает возможность работы с одним и тем же проектом и на ПК, и на мобильном устройстве. Однако CMake не осуществляет непосредственную сборку, а только генерирует промежуточные файлы для других систем, поэтому возникла необходимость перенести и программу GNU Make. Последний, в свою очередь, может пригодиться и более опытным пользователям программы, предпочитающим эту систему сборки.

Перенос необходимых программ и библиотек на платформу Android

Кросс-компиляция – процесс сборки исходного кода на одной платформе для исполнения на другой. Для операционной системы Android официально рекомендуется использование программного инструментария Android NDK (*Android Native Development Kit*), позволяющего собирать код для мобильных устройств с использованием персонального компьютера [3]. На практике список поддерживаемых Android NDK по умолчанию систем сборки невелик: поддерживаются Soong, Android.mk и CMake. Первые две системы сборки эксклюзивны для операционной системы Android, поэтому мало полезны для переноса программного обеспечения, написанного для других платформ. CMake – более применимая в этом отношении система сборки, однако в случае использования ее с Android NDK возникают некоторые нюансы, например необходимость ручного редактирования файлов конфигурации.

Поскольку необходимое для проекта программное обеспечение использует различные системы сборки, не поддерживаемые Android NDK, было решено собрать более традиционный набор инструментов для кросс-компиляции (*cross-compiler toolchain*), совместимый с системами сборки GNU Make и GNU Autotools [4]. Для этого был использован сценарий

командной строки `make-standalone-toolchain.sh`, доступный в комплекте поставки Android NDK.

При сборке инструментов были выбраны библиотека языка C++ LLVM `libc++` и компилятор Clang. Данный выбор параметров обусловлен тем, что позволяет переиспользовать собранные на рабочей станции библиотеки непосредственно в мобильной среде разработки. Поскольку библиотека языка Си, библиотека языка C++ и компилятор совпадают, была обеспечена полная двоичная совместимость между библиотеками и заголовками кросс-компилятора и целевой системы. Таким образом, при помощи этой же системы сборки можно собирать и образовательные библиотеки, предназначенные для использования непосредственно в пользовательских программах. Полный список необходимых для воспроизведения результатов параметров различных систем сборки приведен в табл. 1, значением переменной окружения `PREFIX` является полное имя целевой платформы (например, `arm-linux-androideabi` для устройства с ARM процессором и ОС Android).

Таблица 1

Параметры, использованные для различных систем сборки

Table 1

Parameters Used for Different Build Systems

Система сборки	Имя переменной / параметра	Значение
GNU Make	<code>CC</code>	<code>\$PREFIX-clang</code>
	<code>CXX</code>	<code>\$PREFIX-clang++</code>
GNU Autotools	<code>--host</code>	<code>\$PREFIX</code>
	<code>--target</code>	<code>\$PREFIX</code>
CMake	<code>CMAKE_SYSTEM_NAME</code>	Android
	<code>CMAKE_C_COMPILER</code>	<code>\$PREFIX-clang</code>
	<code>CMAKE_CXX_COMPILER</code>	<code>\$PREFIX-clang++</code>

Слой совместимости с другими средами разработки

К сожалению, на практике стандарты языков программирования не всегда соответствуют тому, что ожидают увидеть пользователи. В разных средах программирования на различных операционных системах могут присутствовать некоторые уникальные функции, не описанные в стандартах. Хорошими примерами таких функций являются Windows API, доступный на операционной системе Windows, или `iostream.h` – заголовочный файл Borland C++, использовавшийся во времена, когда язык программирования C++ еще не был стандартизован.

Был проведен анализ отрицательных отзывов у приложений-конкурентов, доступных в Google Play, в результате чего был выделен следующий список возможностей, критичных для достаточно большого числа пользователей.

- Команды `system("cls")` и `system("pause")`. Данные команды операционной системы Windows используются для того, чтобы очистить экран или ожидать нажатия клавиши. Несмотря на то, что существуют более удачные способы осуществления данных операций, было решено написать свою реализацию этих команд, исполняемые файлы `cls` и `pause` были собраны при помощи кросс-компилятора, пути к ним добавлены в переменную окружения `PATH`.

- Заголовочный файл `conio.h`. Данный заголовочный файл использовался в операционной системе MS-DOS для создания текстового интерфейса пользователя. Было решено реализовать наиболее популярные функции этого заголовочного файла с использованием управляющих последовательностей ANSI – традиционного в Linux способа организации текстового интерфейса.

- Команда `fflush(stdin)`. Данная команда, по мнению пользователей, должна очищать буфер стандартного потока ввода, однако стандарт не гарантирует данное поведение: функция `fflush` определена только для потоков вывода. Изучение исходного кода библиотеки языка Си Android показало, что на этой операционной системе `fflush` потоков ввода ничего не делает. Более ожидаемое поведение было реализовано при помощи вызова `fpurge`, реализация была добавлена в статическую библиотеку `libandroid_support`, включаемую в каждую программу пользователя. Функция названа `fflush_compat`, а в заголовочный файл `stdio.h` добавлено переопределение всех вызовов `fflush` на `fflush_compat`. Это решение позволяет сохранить полную совместимость с библиотеками, которые ожидали вызов оригинальной функции `fflush`.

Разработка и оптимизация интерфейса для мобильных устройств

В ходе разработки было уделено отдельное внимание удобству интерфейса. Под удобством понималось соответствие ожидаемого поведения программы действительному, а также общая скорость выполнения различных операций пользователем.

Исследование проводилось в два этапа. На первом этапе была выбрана небольшая группа пользователей для качественного исследования наиболее часто используемых функций путем опроса, согласно [5].

Выделен следующий список наиболее часто используемых функций:

- ввод особых символов, специфичных для программирования, однако недоступных без дополнительных действий на большинстве программных клавиатур общего назначения;
- автоматическая компиляция и запуск программы по нажатию на одну кнопку;
- автодополнение кода;
- проверка кода в реальном времени;
- установка библиотек;
- форматирование кода.

Далее был спроектирован интерфейс программы: наиболее часто используемые функции были размещены в зоне быстрого доступа в соответствии с исследованием [6], а функции, к которым обращаются реже, перенесены в зоны с меньшей доступностью. Прототип был представлен крупной группе потенциальных пользователей программы для количественной оценки качества спроектированного прототипа. В ходе оценки качества выявлено, что часть функций не пользуется ожидаемой популярностью, поэтому эти функции были перенесены в зону меньшей доступности, а на их место перенесены функции, оказавшиеся более популярными. Кроме того, было подтверждено соответствие функций элементов интерфейса их иконкам: по результатам измерений, пользователи без труда находили запрашиваемое действие среди предложенных иконок.

Метод дальнейшей оптимизации интерфейса

После выпуска первой версии продукта для принятия решений стал использоваться способ А/Б тестирования. Части случайно выбранных пользователей (в количестве 30 %), установивших приложение из Google Play Store, предоставлялся доступ к А-версии продукта (версии с некоторыми изменениями, которые нужно было проверить), а другой части – доступ к Б-версии (первоначальной версии, выступающей в качестве эталона).

По прошествии двух недель измерялись ключевые показатели качества продукта:

- вовлеченность пользователей – частота совершения определенных действий в приложении;
- среднее время нахождения в приложении;
- количество пользователей, запустивших приложение хотя бы раз на второй и седьмой дни эксперимента.

При помощи данного тестирования удалось определить оптимальный порядок и состав пунктов меню приложения: в частности, расположение кнопки запуска в строке заголовка приводило к снижению вовлеченности пользователей, в то время как расположение кнопки

ближе к правому нижнему углу оказалось предпочтительным, хотя на первый взгляд приводит к уменьшению рабочего пространства редактора.

Снимки экрана двух вариантов, использованных в эксперименте с расположением кнопки запуска, представлены на рис. 1.

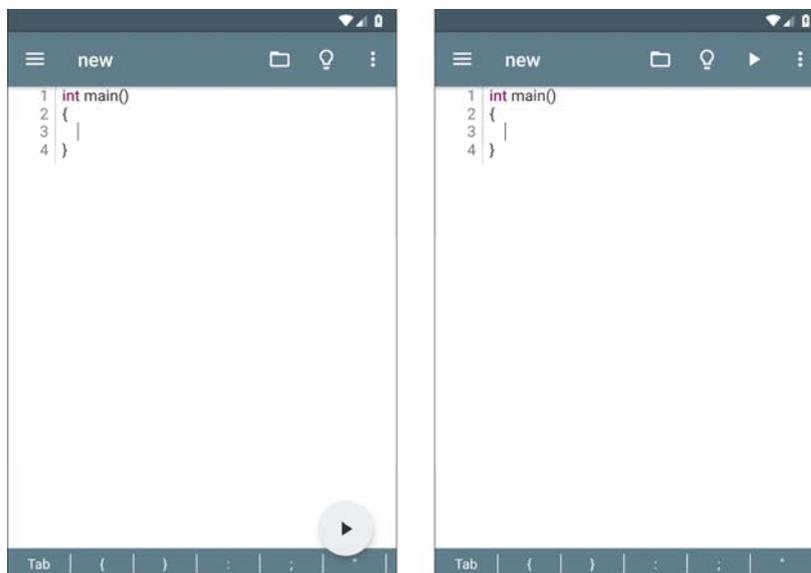


Рис. 1. А и Б варианты продукта с различным расположением кнопки запуска
Fig. 1. A and B Variants of Product with Different Locations of Launch Button

Автодополнение и проверка ошибок в реальном времени

Традиционный процесс автодополнения и проверки кода требует полной обработки исходного текста с самого начала каждый раз, когда вызывается автодополнение или проверка. Так работают некоторые настольные среды разработки, поскольку даже средний персональный компьютер имеет достаточную производительность, для того чтобы выполнять эти задачи в режиме реального времени.

Большинство программ на языке C++ используют различные заголовочные файлы, которые включаются в программу с использованием специальной директивы компилятора, #include. Эти заголовочные файлы часто достаточно велики, чтобы занимать большую часть времени компиляции, более того, существуют библиотеки, состоящие только из заголовочных файлов, которые полностью компилируются каждый раз, когда пользователь компилирует основную программу. Этот процесс может быть оптимизирован с использованием предкомпилированных заголовков, специальных файлов, которые содержат закешированное состояние компилятора после обработки некоторого списка заголовочных файлов [7]. Большинство сред разработки, однако, требует ручного создания данных файлов (значимым примером является файл stdafx.h, используемый Microsoft Visual Studio), что может быть затруднительно для пользователя мобильного устройства, поскольку ему придется редактировать несколько файлов и переключаться между ними каждый раз, когда он захочет добавить в программу очередной заголовочный файл. Поэтому возникла необходимость реализации алгоритма, который сочетал бы в себе удобство традиционного медленного подхода и скорость подхода, использующего предкомпилированные заголовки.

После автоматического анализа исходного кода многих популярных C++ проектов с открытым исходным кодом и ручной проверки выводов на простых учебных программах мы пришли к следующим выводам:

- программы могут иметь информацию о лицензии и авторе, расположенную в комментарии в самом начале исходного текста;
- директивы `#define` используются для констант и макросов (реже), они также часто появляются в начале программы;
- директивы `#include` обычно располагаются до объявленных пользователем функций, неподалеку от макросов и констант (но не всегда).

На основании анализа истории изменений проектов с открытым исходным кодом было установлено, что упомянутые части меняются достаточно редко в сравнении с остальным кодом. Из этого можно сделать вывод, что данные части программы являются хорошими кандидатами на включение в кэш компилятора.

Алгоритм ускорения автодополнения и проверки ошибок

Было решено встроить технологию оптимизации в процесс проверки исходного кода: мы можем безопасно генерировать предкомпилированные заголовки, только если текущая программа является корректным исходным кодом C++. Если проверка исходного кода не находит ошибок, мы создаем новый текст предкомпилированного заголовка с использованием регулярных выражений, и тогда, только если заэкшированного заголовка с таким текстом не существует, мы собираем его в фоновом потоке.

Если сгенерированный текст предкомпилированного заголовка совпадает с одним из существующих файлов кэша, автодополнение кода и проверка ошибок выполняются в ускоренном режиме с заменой текста на включение единственного заэкшированного заголовочного файла.

Схема алгоритма представлена на рис. 2.



Рис. 2. Схема алгоритма генерации предкомпилированных заголовков

Fig. 2. Diagram of Precompiled Header Generation Algorithm

Для генерации кэшируемого текста было решено использовать наиболее простое возможное регулярное выражение, поскольку это часто выполняемая операция [8]. В результате в кэшируемый текст были включены следующие части:

- однострочный и многострочный комментарий;
- любая директива компилятора, начинающаяся с символа `#`;
- пустые строки.

Существует, однако, важное замечание: алгоритм также проверяет, что количество включенных в заголовки директив `#if` и `#ifdef` равно количеству `#endif`. Хотя данные конструкции сравнительно нечасто используются в коде образовательных программ (особенно в опасной для алгоритма форме, по разные стороны от пользовательских объявлений переменных и функций), их игнорирование приводит к ошибке компиляции, что недопустимо для надежного алгоритма.

Оценка производительности полученного алгоритма

Чтобы оценить производительность алгоритма, было решено сравнить производительность автодополнения кода с наиболее популярными приложениями-конкурентами, доступными в магазине приложений Google Play: `Cppdroid` и `C4droid`. Производительность проверки кода не замерялась, поскольку реализована только в приложении `Cppdroid`, в котором, в свою очередь, был дополнительно доступен статический анализатор, что могло повлиять на достоверность результатов.

Для замеров использован набор исходных текстов образовательных программ, разбитый на несколько категорий:

- 1) примеры языка Си;
- 2) примеры языка C++, базовое использование стандартной библиотеки шаблонов (STL), разрешен только заголовок `iostream`;
- 3) примеры языка C++, обычное использование STL, разрешены заголовок `iostream` и контейнеры (например, `vector` и `map`);
- 4) примеры языка C++, расширенное использование STL, разрешено всё, включая последние стандарты;
- 5) примеры языка C++ с использованием библиотеки Boost.

Для замера времени выполнения операции автодополнения был выбран метод записи экрана и последующего перевода числа кадров во время, поскольку этот метод позволяет непосредственно оценить отзывчивость программы, а также избегает необходимости в обратной разработке программ, что может быть запрещено их лицензионным соглашением.

Устройством для произведения замеров был выбран планшетный компьютер Google Nexus 7 (2013), поскольку его невысокая производительность позволяет оценить снизу скорость работы программы на реальных устройствах, а также понижает относительную погрешность измерений за счет больших временных интервалов.

В результате проведения серий экспериментов, были получены результаты, представленные в табл. 2.

Таблица 2

Сравнительная таблица производительности автодополнения кода

Table 2

Comparison of Code Completion Performance

Программный продукт	Среднее время автодополнения кода в данной категории примеров, с				
	1	2	3	4	5
C4droid	0,35	2,62	3,05	3,86	27,29
Cppdroid	0,23	3,08	3,79	Н/Д ²	Н/Д ³
Sxxdroid (данная работа)	0,49	0,63	0,97	0,67	0,79
Отношение производительности работы к усредненному конкуренту	0,59	4,52	3,53	5,76	34,54

² Используемая в `Cppdroid` стандартная библиотека шаблонов устарела и не поддерживает стандарт C++11, требуемый примерами данной категории.

³ `Cppdroid` не поддерживает библиотеку Boost, и ее не удалось установить.

Заключение

В результате выполнения работы на платформу Android были успешно перенесены инструменты Clang, GNU Binutils, BusyBox, их зависимости, а также набор популярных библиотек, таких как Boost, GMP, libcurl. Был разработан оптимизированный для мобильных устройств интерфейс. Возможности по анализу и автодополнению кода были реализованы при помощи Clang. Создан алгоритм ускорения упомянутых возможностей, проверена его производительность и корректность. Все данные компоненты объединены в мобильную среду разработки языка C++.

Эксперименты показали, что разработанная среда программирования является единственной, обеспечивающей работу автодополнения и анализа кода в режиме реального времени в любых сценариях использования. Прирост производительности в среднем составил 300 %, когда используется стандартная библиотека шаблонов языка C++, достигая 3350 %, когда используется библиотека Boost. Присутствует небольшая потеря производительности при использовании языка Си, однако она незначительна в абсолютной величине и не влияет на возможность использования продукта в реальном времени.

Список литературы / References

1. **Prextová T.** Mobile technology in education. In: International Conference of Computational Methods In Sciences and Engineering 2018 (ICCMSE 2018), 2018. DOI 10.1063/1.5079081.
2. **Brewster S.** Overcoming the Lack of Screen Space on Mobile Computers. *Personal and Ubiquitous Computing*, May 2002, vol. 6, no. 3, p. 188–205. DOI 10.1007/s007790200019.
3. **Son K.-C., Lee J.-Y.** The method of android application speed up by using NDK. In: 3rd International Conference on Awareness Science and Technology (iCAST), 2011. DOI 10.1109/icawst.2011.6163104
4. **Schiele R.** Building and Using a Cross Development Tool Chain. In: GCC Developers Summit, May 2003, p. 213.
5. **Vermeeren A. P. O. S., Law E. L.-C., Roto V., Obrist M., Hoonhout J., Väänänen-Vainio-Mattila K.** User experience evaluation methods. In: Proceedings of the 6th Nordic Conference on Human-Computer Interaction Extending Boundaries – NordiCHI'10, 2010. DOI 10.1145/1868914.1868973
6. **Yoo H., Yoon J., Ji H.** Index Finger Zone. In: Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct – MobileHCI'15, 2015. DOI 10.1145/2786567.2793704
7. **Krishnaswamy T.** Automatic precompiled headers: Speeding up c++ application build times. In: Proc. of the 1st Conf. on Industrial Experiences with Systems Softw. (WIESS), 2000. Berkeley, CA, USENIX Association, vol. 1, p. 57–66.
8. **Cody-Kenny B., Fenton M., Ronayne A., Considine E., McGuire T., O'Neill M.** A search for improved performance in regular expressions. In: Proceedings of the Genetic and Evolutionary Computation Conference on – GECCO'17, 2017. DOI 10.1145/3071178.3071196

Материал поступил в редакцию

*Received
05.04.2019*

Сведения об авторах / Information about the Authors

Лебедев Роман Константинович, магистрант факультета информационных технологий, Новосибирский государственный университет (ул. Пирогова, 1, Новосибирск, 630090, Россия)

Roman K. Lebedev, Master's Student, Novosibirsk State University (1 Pirogov Str., Novosibirsk, 630090, Russian Federation)

n0n3m4@gmail.com

Корякин Илья Алексеевич, магистрант факультета информационных технологий, Новосибирский государственный университет (ул. Пирогова, 1, Новосибирск, 630090, Россия)

Ilya A. Koryakin, Master's Student, Novosibirsk State University (1 Pirogov Str., Novosibirsk, 630090, Russian Federation)

ed4140@gmail.com

Разработка программных средств прозрачного удаленного доступа к технологическим параметрам бурения и данным скважинной телеметрической системы в условиях нестабильной связи

В. С. Литвинов¹, А. А. Власов², Д. В. Тейтельбаум³

¹ *Новосибирский государственный университет
Новосибирск, Россия*

² *Институт нефтегазовой геологии и геофизики им. А. А. Трофимука СО РАН
Новосибирск, Россия*

³ *ООО НПП ГА «Луч»
Новосибирск, Россия*

Аннотация

В процессе бурения нефтегазовых скважин требуется оказание удаленной поддержки с использованием существующих каналов связи с офисом, заключающейся в контроле параметров и принятии неотложных решений. Наиболее распространена спутниковая и мобильная связь, для которой характерна низкая скорость приема и передачи данных, частые разрывы, использование NAT. В рамках данной работы разработано программное средство прозрачного доступа к параметрам бурения и данным скважинной телеметрической системы с целью организации удаленного сопровождения по существующим каналам связи с учетом их ограничений.

В основу программного средства легла реализация протокола SSH, что обеспечило сжатие и шифрование передаваемых данных и прозрачный доступ к TCP-сервисам на буровой. Благодаря этому возможно использование существующего ПО для доступа к данным телеметрической системы, что позволяет снизить объем передаваемых данных в сравнении с ПО удаленных рабочих столов.

Программное средство прошло апробацию в условиях, близких к полевым, и показало свою пригодность к решению поставленных задач.

Ключевые слова

наклонно направленное бурение, сопровождение бурения, SSH, port forwarding

Благодарности

Выражаем благодарности ведущему инженеру-программисту ИТС СУБР-1 ПАО «Сургутнефтегаз» Рустему Рафаилевичу Гильфанову, начальнику отдела по телеметрическому сопровождению скважин управления по бурению ПАО «Сургутнефтегаз» Андрею Владимировичу Овчаренко за помощь во внедрении разработанного средства.

Работа выполнена при поддержке проекта ФНИ № 0331-2019-0015 «Реалистичные теоретические модели и программно-методическое обеспечение геоэлектрики гетерогенных геологических сред»

Для цитирования

Литвинов В. С., Власов А. А., Тейтельбаум Д. В. Разработка программных средств прозрачного удаленного доступа к технологическим параметрам бурения и данным скважинной телеметрической системы в условиях нестабильной связи // Вестник НГУ. Серия: Информационные технологии. 2019. Т. 17, № 2. С. 101–113. DOI 10.25205/1818-7900-2019-17-2-101-113

Development of Software for Transparent Remote Access to Technological Parameters of Drilling and Data from a Downhole Telemetry System in Unstable Communication Conditions

V. S. Litvinov¹, A. A. Vlasov², D. V. Teytelbaum³

¹ Novosibirsk State University
Novosibirsk, Russian Federation

² Trofimuk Institute of Petroleum-Gas Geology and Geophysics SB RAS
Novosibirsk, Russian Federation

³ SPE «Looch»
Novosibirsk, Russian Federation

Abstract

In some cases, the drilling of oil and gas wells requires support for the drilling process, which consists in monitoring the parameters and making urgent decisions. Accompanying drilling takes place using the existing communication channels between the office and drilling. Satellite and mobile communications are most common. They are characterized by a low speed of receiving and transmitting data, frequent disconnects, the use of NAT. As part of this work, a software tool has been developed for transparent access to drilling parameters and measurements while drilling from a telemetry system in order to organize remote tracking via existing communication channels, taking their limitations into account.

The software was based on the implementation of the SSH protocol, which provided compression and encryption of the transmitted data and transparent access to TCP services at the rig. This makes it possible to use existing software to access drilling measurements, which makes it possible to reduce the amount of data transferred in comparison with remote desktops.

The software has been tested in conditions close to the field, and has shown its suitability for solving the tasks it was designed for.

Keywords

directional drilling, drilling tracking, SSH, port forwarding

Acknowledgements

We are grateful to Rustem R. Gilfanov, Chief Engineer of ITT SUBR-1 of PJSC “Surgutneftegaz”, and Head of the Department for Telemetry Support of the Well of the Drilling Department of PJSC “Surgutneftegaz” Andrei V. Ovcharenko for assistance in implementing the developed tool.

This work was supported by the FNI project no. 0331-2019-0015 “Realistic theoretical models and software and methodological support for geoelectrics of heterogeneous geological environments”

For citation

Litvinov V. S., Vlasov A. A., Teytelbaum D. V. Development of Software for Transparent Remote Access to Technological Parameters of Drilling and Data From a Downhole Telemetry System in Unstable Communication Conditions. *Vestnik NSU. Series: Information Technologies*, 2019, vol. 17, no. 2, p. 101–113. (in Russ.) DOI 10.25205/1818-7900-2019-17-2-101-113

Введение

При бурении нефтегазовых скважин возникают нештатные ситуации, требующие принятия неотложных решений. Промедление с разрешением таких ситуаций может привести к тяжелым последствиям для персонала, оборудования, ствола скважины, к трате ресурсов на устранение этих последствий. Для контроля параметров бурения и принятия неотложных решений привлекаются высококвалифицированные специалисты [1; 2], чаще работающие в офисе и занимающиеся надзором над бурением сразу нескольких скважин. Выезд специалистов на буровые площадки нежелателен с экономической точки зрения и происходит только в экстренных случаях. Существует типичный набор средств, применяемых для решения задачи удаленного сопровождения бурения, но специалисты, работающие с ними, сталкиваются с ограничениями, связанными с особенностями каналов связи офиса и буровых, организацией вычислительных сетей предприятия, вопросами информационной безопасности, простотой применения данных средств. Например, при использовании спутникового канала связи ПО удаленного рабочего стола может в сжатые сроки истратить значительную часть отведенного на месяц работы трафика буровой. Использование мобильной связи накладыва-

ет ограничения на использование ПО, требующего прямого подключения к ПК буровой, из-за механизма преобразования сетевых адресов.

В рамках работы была поставлена задача разработать программное средство прозрачного удаленного доступа к технологическим параметрам бурения и данным скважинной телеметрической системы в условиях нестабильной связи с целью сопровождения бурения и оказания помощи инженерам наклонно направленного бурения. Задача была разбита на этапы:

1) определить основных организаторов и участников процесса удаленного сопровождения бурения и их роли; выделить функциональные требования к программному средству; основываясь на характеристиках каналов связи офиса и буровой, на предпочтениях участников процесса, определить нефункциональные требования;

2) провести анализ технических средств, пригодных для удовлетворения поставленных требований и выбрать из них наиболее подходящие для реализации программного средства;

3) спроектировать программное средство, основанное на выбранном наборе технических средств: понять, каким образом пользователи будут с ним взаимодействовать, на какие функциональные блоки оно будет декомпозировано, проработать сценарии внедрения и поддержки;

4) определить методику тестирования и наладки, подобрать необходимые инструменты;

5) реализовать программное средство;

6) провести тестирование и отладку, сделать замеры использования сетевых ресурсов, времени безотказной работы, проверить пригодность разработанного средства для решения поставленных задач;

7) апробировать разработанное программное средство для сопровождения бурения нефтяных скважин с телеметрической системой “Луч”.

Организация процесса сопровождения бурения

Обычно удаленное сопровождение бурения проходит с использованием существующих каналов связи с буровой. Далее приведены наиболее распространенные варианты организации связи с их характерными особенностями.

1. Спутниковая связь. Зачастую это единственный вариант, доступный в полевых условиях из-за удаленности от населенных пунктов. Задержка передачи данных при таком способе соединения достигает нескольких сотен миллисекунд. Скорость приема данных обычно лежит в интервале от 64 кбит/с до 6 Мбит/с, скорость передачи – от 32 кбит/с до 2 Мбит/с. Часто указанная пропускная способность разделяется между несколькими буровыми. Максимально возможная скорость приема-передачи данных зависит от множества физических параметров, от погодных условий, однако телекоммуникационное оборудование рассчитано на работу на скорости сотен мегабит в секунду. Обычно на буровых устанавливается спутниковый модем с маршрутизатором.

2. Мобильная связь (GPRS, 3G, 4G) вблизи населенных пунктов, в зонах покрытия сети. Она имеет меньшие задержки передачи данных, проще в организации. Как правило, устройство абонента не имеет адреса в Интернете, поэтому прямой доступ к ПК на буровой затруднен, и приходится использовать ПО, в котором соединения устанавливаются недоступной через Интернет стороной (например, Teamviewer).

3. Кабельная связь обладает наивысшей скоростью передачи данных, но применяется редко из-за дороговизны строительства сетей в труднодоступных районах. В таких случаях ее владельцем является нефтегазовая компания, и сеть организуется с учетом требований, связанных с удаленным доступом. Но, как правило, эта сеть недоступна подрядным сервисным организациям, в том числе занимающимся строительством новых скважин.

Параметрами бурения можно считать данные забойной телеметрической системы, данные геолого-технологических исследований, наземных датчиков, видео с камер наблюдения на буровой и т. д. В зависимости от набора используемого ПО эти данные можно получать как в специальных форматах, предназначенных для машинной обработки, так и только в виде

изображения пользовательского интерфейса. На основании того, какие форматы данных доступны, как устроена связь офиса с буровыми, выбираются инструменты для контроля параметров бурения.

Довольно распространенный способ организации контроля за бурением – накопление нужных данных на ПК буровой и периодическая (например, после пробуривания скважины на длину одной трубы) ручная отправка их по электронной почте заказчику буровых работ для анализа и сопоставления фактических результатов измерений с геологической моделью и планом строительства скважины. В случае несоответствия этих данных бурение может быть временно остановлено для корректировки модели или вообще прекращено. Для минимизации времени простоя буровой и оперативного выявления таких несоответствий полезной является возможность наблюдения за параметрами непосредственно в процессе бурения.

Анализ технических средств

Проведем краткий анализ инструментов, применяемых для удаленного контроля параметров бурения.

1. Специализированное клиентское ПО – программное обеспечение для контроля параметров бурения с клиент-серверной архитектурой, поддерживающее работу со специализированными форматами для машинной обработки. Данные каротажа обычно хранятся на сервере и могут быть представлены в разных форматах, например в текстовом LAS (Log ASCII Standard), PDF (каротажный материал в графическом виде). Доступ к данным, хранящимся на сервере, может осуществляться по различным протоколам – как открытым (WITS), так и проприетарным, специфичным только для определенного ПО. Примером такого ПО может служить RealDepth 5 для работы с телеметрической системой «Луч» [3], использующее специально разработанный протокол, который позволяет передавать не только глубинные, но и временные данные. Часто реализуются системы сбора и интерпретации данных, создаваемые для решения узкоспециализированных задач [4].

2. Удаленный рабочий стол – обобщенное название приложений для удаленного управления рабочим столом с клиент-серверной архитектурой (например, TightVNC, Radmin).

3. TeamViewer¹ – это приложение для удаленного управления и организации конференций. Отличие этого средства от удаленного рабочего стола заключается в использовании промежуточного сервера, доступного обеим сторонам, для установления соединения. Это позволяет подключаться к рабочим столам пользователей, чьи ПК недоступны из-за NAT.

4. VPN – это обобщенное название технологий, позволяющих обеспечить одно или несколько сетевых соединений (логическую сеть) поверх другой сети. Используется вместе с клиентским ПО из п. 1, 2.

5. SSH port forwarding – это технология и тип SSH-канала, предусмотренные стандартом протокола SSH². Поддержка таких каналов на стороне SSH-сервера позволяет перенаправлять TCP/IP соединения с определенного порта сервера на определенный адрес и порт со стороны клиента. Используется вместе с клиентским ПО из п. 1, 2.

Было проведено сравнение этих инструментов по нескольким параметрам, связанным с особенностями удаленного сопровождения бурения нескольких скважин (см. таблицу):

- 1) скорость – пригодность для использования в условиях низкой пропускной способности канала связи;
- 2) конфиденциальность – обеспечение конфиденциальности передаваемых данных;
- 3) прозрачность – возможность использовать специализированное клиентское ПО, установленное на ПК специалиста. Это может быть полезным, когда на ПК специалиста уже настроено свое привычное окружение;

¹ TeamViewer – удаленная поддержка, удаленный доступ, служба поддержки, совместная работа онлайн и со-вещания. <https://www.teamviewer.com/>

² RFC 4254 – The Secure Shell (SSH) Connection Protocol. <https://tools.ietf.org/html/rfc4254>

4) привилегии – отсутствие необходимости изменения системных параметров на ПК буровой или топологии существующей сети: не требуется привлечение системного администратора для настройки средства для удаленного доступа и поддержания его в работоспособном состоянии;

5) простота – простота настройки в полевых условиях инженерами наклонно направленного бурения;

6) интерфейс – наличие единого интерфейса для отображения информации о множестве буровых: состояние и параметры подключения, иная дополнительная информация.

Результаты сравнения инструментов

The Results of the Comparison Tools

Средство	Параметр					
	Скорость	Конфиденциальность	Прозрачность	Привилегии	Простота	Интерфейс
Специализированное клиентское ПО	?	?	+	+	+	–
Удаленный рабочий стол	?	?	–	?	–	–
TeamViewer	+	+	–	+	+	+
VPN	+	+	+	–	–	+
SSH portfwd	+	+	+	+	–	–

Примечание: знаки «+», «–», «?» обозначают соответствие, несоответствие, возможное соответствие (зависит от конкретного выбора) рассматриваемого средства выбранному параметру.

Note: the signs “+”, “–”, “?” Denote correspondence, nonconformity, possible correspondence (depending on the particular choice) of the considered tool to the selected parameter.

По результатам анализа (см. таблицу), ни одно из рассмотренных средств в полной мере не учитывает всех важных особенностей удаленного сопровождения бурения.

Анализ требований и проектирование

Разрабатываемое программное средство рассматривалось с нескольких точек зрения.

1. С точки зрения ведущих инженеров наклонно направленного бурения, работающих в офисе. В их интересах иметь средство для оперативного доступа к данным бурения, чтобы контролировать параметры совместно с полевыми инженерами наклонно направленного бурения. В разных ситуациях могут быть удобными разные решения: сеанс удаленного рабочего стола, если у полевого инженера возникли проблемы с настройкой ПО, или подключение клиентским ПО забойной телеметрической системы. По сравнению с использованием удаленного рабочего стола это позволяет экономно расходовать отведенный объем трафика. Поэтому было предложено с помощью разрабатываемого программного средства предоставлять доступ к нескольким сервисам, запущенным на ПК буровой. Было предложено реализовать монитор, отображающий состояние программного средства на ПК буровой и доступность указанных сервисов, способ подключения к этим сервисам. Кроме того, при разборе нештатных ситуаций возникает потребность понять, находился ли полевой инженер на рабочем месте в определенный момент времени, чем он был занят. Для этого было предложено собирать с ПК буровых снимки экранов и наличие пользовательского ввода, отображать эти дан-

ные в мониторе ведущего инженера и иметь возможность сформировать из них отчет за указанный промежуток времени.

2. С точки зрения системных администраторов, работающих в офисе, занимающихся разворачиванием и настройкой программного средства. В их задачи входит сетевое администрирование и формирование списка зарегистрированных ПК буровых, чтобы ведущие инженеры имели доступ только к нужным буровым. Месячный трафик нескольких буровых ограничен, поэтому для контроля над расходом трафика средством удаленного контроля было предложено встроить в него сбор статистики использования сети и инструмент для просмотра этой статистики за определенный временной интервал. В случае аномального расхода трафика средством удаленного контроля системные администраторы смогут установить причину данного явления.

3. С точки зрения полевых инженеров наклонно направленного бурения – инженеров телеметрической службы, ГТИ. Основная их работа – планирование и организация работ по геонавигационному сопровождению скважин, и непосредственно сопровождение. Предполагалось, что их участие в процессе настройки и эксплуатации программного средства удаленного контроля потребуется только в целях диагностики.

Основными нефункциональными требованиями были:

- 1) ограничение на объем передаваемых собственных данных программного средства;
- 2) сжатие и шифрование всего трафика программного средства;
- 3) простота в обращении для полевых инженеров – отсутствие надобности читать руководство по эксплуатации для выполнения основных действий с программным средством; в идеальном случае отсутствие какого-либо взаимодействия полевого инженера по наклонно направленному бурению с программным средством;
- 4) длительное время стабильной работы программного средства;
- 5) работоспособность программного средства в условиях плохой связи с частыми разрывами соединения;
- 6) возможность запуска на ОС Windows 7, 8, 10, Linux.

Общая схема работы программного средства

В основу разрабатываемого средства было решено положить технологию SSH port forwarding (перенаправление портов), так как ее настройка и использование не требуют изменения системных параметров. Из этого выбора следует, что архитектура разрабатываемого программного средства должна быть клиент-серверной, где клиентами будут являться узлы, расположенные на буровых, на которых запущены сервисы, предоставляющие по TCP/IP доступ к технологическим параметрам бурения и данным скважинной телеметрической системы (рис. 1). На сервере должна быть запущена командная оболочка, через которую клиенты будут передавать данные о своем состоянии. Настройки клиентов хранятся на сервере и отдаются по запросу, переданному в командную оболочку при установке соединения. Это позволяет задавать настройки каждого из клиентов через интерфейс администратора.

Интерфейс для отображения списка подключенных клиентов и их состояния было решено реализовать в виде постоянно обновляющейся веб-страницы.

При переезде или расформировании партии ИТС или ГТИ, при отправке неисправных компьютеров на ремонт возможен несанкционированный доступ к серверу. Поэтому в системе должна быть предусмотрена возможность закрывать указанному клиенту доступ на сервер. Было рассмотрено два технических решения, одно из которых основывается на работе с пользователями ОС, другое – на использовании списка авторизованных публичных ключей SSH. Для реализации был выбран второй вариант, так как он требует меньших привилегий в ОС. Кроме того, в реализации *OpenSSH*-сервера есть возможность гибко управлять ресурсами, доступными клиенту (параметры «Permit*»³), в том числе прошедшему аутентифика-

³ sshd_config(5) – OpenBSD manual pages. https://man.openbsd.org/sshd_config

цию с помощью заданного публичного ключа. Это немаловажно с точки зрения безопасности, так как можно быть уверенным, что клиент не использует не предназначенные для него ресурсы. Идентификация пользователя в системе осуществляется с помощью 128-битного отпечатка (fingerprint) публичного ключа.

Клиентское ПО состоит из функциональных блоков («сервисов»), которые включаются и настраиваются автоматически на основании ответа сервера на запрос получения конфигурации клиента.

Реализовано четыре типа сервисов, нужда в которых продиктована требованиями к программному средству.

1. RemoteTunnel запускает механизм перенаправления портов для указанных локальных TCP-серверов (например, RealDepth5 [5], служащего для чтения и обработки данных скважинной телесистемы и наземных датчиков, TightVNC (VNC) для удаленного управления компьютером и передачи файлов), отправляет состояние перенаправления (локальную доступность сервиса; ошибки при открытии порта на сервере) на сервер через командную оболочку.

2. Screenshot периодически делает снимки экрана и отправляет их через командную оболочку. В веб-интерфейсе реализована функциональность просмотра снимков экрана для заданного клиента, а также построения отчетов об активности клиентов во времени на основании наличия снимков экранов.

3. Netstat – ведет учет использования сети клиентом, периодически отправляет статистику на сервер через командную оболочку.

4. Bootstrap – сервис, создающий все остальные на основе конфигурации, полученной от сервера.

Видимое состояние сервисов состоит из двух логических переменных: исправность «локальной» и «удаленной» части сервиса. Данные понятия интерпретируются по-разному в зависимости от типа сервиса.

В результате была выработана архитектура программного средства для удаленного сопровождения бурения, показанная на рис. 1.

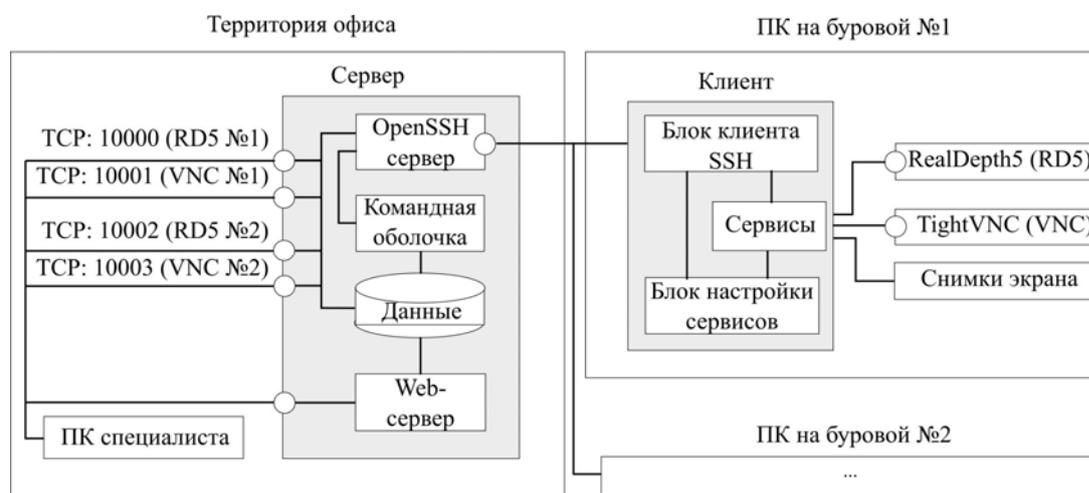


Рис. 1. Архитектура программного средства и схема подключения ПК буровых и ПК специалиста к серверу для оказания удаленной помощи. Каждому из клиентов на сервере автоматически выделяется диапазон TCP-портов для проброса сервисов. В данном примере для ПК на буровой № 1 выделен диапазон 10000–10001, для ПК на буровой № 2 выделен диапазон 10002–10003. Серым фоном обозначены разработанные компоненты

Fig. 1. The architecture of the software and the connection scheme of the drilling rig's PC and specialist's PC to the remote assistance server. Each of the clients on the server is automatically allocated a range of TCP ports for forwarding services. In this example, the range 10000–10001 is allocated for the PC at the drilling rig no. 1, and the range 10002–10003 is allocated for the PC at the drilling rig no. 2. The gray background indicates the developed components

С учетом возможностей и ограничений выбранных средств были разработаны сценарии взаимодействия клиента с сервером, пользователей с программным средством во время регистрации на сервере и в нормальном рабочем режиме, при котором происходит процесс удаленного сопровождения бурения.

Сценарий регистрации клиента на сервере.

1. Клиентское ПО генерирует пару ключей для шифрования.
2. Клиентское ПО проходит аутентификацию на сервере с заранее известным паролем и через командную оболочку передает серверу запрос на регистрацию с данными о себе: имя ПК, публичный ключ. При таком режиме аутентификации это единственный доступный клиенту запрос.
3. На сервере указанный клиент попадает в множество кандидатов на регистрацию.
4. Администратор или ведущий инженер через веб-интерфейс выбирает кандидатов на регистрацию и переносит их в основной список клиентов для сопровождения. После этого клиенту доступны аутентификация с помощью публичного ключа и множество запросов на передачу служебной информации.
5. Клиентское ПО переподключается к серверу автоматически и проходит аутентификацию с помощью публичного ключа, получает от сервера свою конфигурацию и проходит настройку в соответствии с конфигурацией. После этого клиентское ПО начинает периодически передавать на сервер служебную информацию и переходит в состояние готовности к удаленному подключению из офиса.

Типичный сценарий подключения к сервису на ПК буровой.

1. Клиентское ПО подключается автоматически к серверу и проходит описанный в предыдущем сценарии процесс настройки.
2. Ведущий инженер через веб-интерфейс наблюдает за состоянием (работоспособность, номер ТСР-порта) интересующего его сервиса на ПК полевого инженера. Если сервис доступен, ведущий инженер вводит настройки подключения клиентского ПО, адрес сервера удаленного сопровождения и номер соответствующего сервису порта, подключается к сервису на буровой и осуществляет сопровождение бурения в соответствии с установленными стандартами.

Методика тестирования и отладки

Разработанное программное средство нуждалось в тщательном тестировании на соответствие поставленным требованиям. При несоответствии критически важным требованиям, связанным со временем непрерывной работы ПО, расходом трафика буровой, разработанное программное средство не нашло бы практического применения.

Для проверки соответствия программного средства требованиям, связанным со скоростью передачи данных и их объемом, было предложено искусственно ограничивать максимальные скорости приема и передачи данных для клиентского приложения и наблюдать за его работоспособностью при выставленных ограничениях. Для этого использовалось ПО NetLimiter. Кроме того, с помощью отладочного вывода проверялось, возобновляется ли передача крупных посылок при восстановлении соединения после разрыва, типичного для спутникового или мобильного канала связи.

Вероятны были неполадки клиентского ПО, которые с трудом воспроизводились бы на ПК разработчика. Для диагностики причин этих неполадок и их устранения записывались файлы журнала, которые извлекались с клиентского ПК вместе с образом процесса приложения, а затем анализировались вручную.

В числе нефункциональных требований указано длительное время стабильной работы программного средства. Поскольку формальное доказательство корректности программы трудозатратно, был предложен следующий критерий: на ПК разработчиков программное средство должно проработать без ошибок в течение 10 дней. Это срок, сравнимый со временем нахождения телеметрической системы под столом ротора буровой. Режим использова-

ния ПК на буровой схож с режимом использования разработчиками их компьютеров, поэтому предполагалось, что такой критерий должен подойти.

Детали реализации

В ходе разработки приветствовалось использование свободно распространяемого программного обеспечения. Поэтому в качестве ОС для сервера был использован дистрибутив CentOS 7 Linux. Среди других он выделялся наибольшим сроком поддержки и выпуска обновлений. Файлы сервера и скрипты для установки были упакованы в пакет RPM, так как часть задач при установке сервера удаленного сопровождения бурения выполняет менеджер пакетов, а обновление или удаление можно провести штатными инструментами.

Для реализации программного средства был взят OpenSSH сервер версии 7.8, поскольку в настройках этой версии появились необходимые опции для ограничения возможностей подключенных клиентов. Например, используется параметр PermitListen для задания доступного клиенту диапазона портов сервера, на которых можно запускать механизм перенаправления. Также используется переменная среды SSH_USER_AUTH (появилась в версии 7.6), задаваемая процессу ssh, по значению которой можно узнать, какой механизм аутентификации был использован, идентифицировать клиента и определить набор команд, доступных ему на выполнение в командной оболочке. Для этого стандартная командная оболочка для соответствующего пользователя в ОС была заменена на свою, в которой проводится идентификация пользователя по этой переменной, генерируется и отправляется на стандартный вывод уникальный номер сессии, а в базу данных записываются параметры этой сессии.

Основная командная оболочка реализована в виде TCP-сервера, так как это проще в отладке, не требует создания и настройки псевдотерминала и борьбы с особенностями его работы (эхо, переносы строк). Клиенты получают к нему доступ через SSH local port forwarding, в посылки включают выданный первичной командной оболочкой номер сессии. Для предотвращения нехватки места на диске записи о старых неактивных сессиях стираются из базы данных.

Данные клиентов, хранящиеся в файловой системе, организованы в дерево с путями вида *<идентификатор пользователя>/<год>/<месяц>/<день>/<час>/*. С такой структурой легко удаляются данные конкретного клиента (все сразу или в указанном временном интервале), например, при его удалении через веб-интерфейс или при автоматической очистке старых данных. Очистка запускается по расписанию, и при нехватке места в файловой системе удаляются данные старше заданного возраста.

Серверная часть веб-интерфейса была реализована с использованием фреймворка Django, так как автор был с ним частично знаком, в комплекте шли библиотека для ORM и система шаблонов веб-страниц. Этот выбор определил основной язык для реализации остальных компонент сервера – Python 3. В качестве веб-сервера был выбран Apache 2 с включенным mod_wsgi, впоследствии на нем была настроена поддержка HTTPS с самоподписанным сертификатом. Клиентская часть (рис. 2) веб-интерфейса была реализована с использованием библиотеки jQuery, упрощающей разработку динамических страниц, и библиотеки стилей Bootstrap. Большинство запросов на сервер отправляются периодически с использованием AJAX.

Клиентская часть (см. рис. 2) была написана на языке C++14, так как библиотека libssh2 для работы с SSH реализована на языке C, и автор был знаком с фреймворком Qt 5 для построения графических пользовательских интерфейсов. Конфигурацию клиента (настройки подключения и сервисов) было решено не выносить в отдельный текстовый файл, а оставить в исполняемом файле, чтобы избежать случайной порчи рабочих настроек. Клиентское приложение при запуске сворачивается в область уведомлений и переходит на передний план лишь при необходимости ввода одноразового пароля для регистрации. Пара ключей для SSH генерируется один раз и хранится в директории для локальных файлов приложений. Там же находится файл с журналом событий, ограниченный размером 10 МБ, который может быть

использован для устранения неполадок в работе данного программного средства. При нормальной работе программного средства у оператора не должна возникать потребность открывать главное окно, однако при возникновении сбоев в работе оно может быть полезно, так как содержит состояния подключения и сервисов, которые можно передать разработчикам и получить указания по дальнейшим действиям. Клиент был скомпилирован и протестирован для ОС Windows 7, 10, Centos 7 Linux, Ubuntu 18.04 Linux.

Использование ФС сервера и время офиса

Название	Был активен, времени назад	Сервис	Порт	Последний статус	Снимок экрана (время UTC+7)
Dima-Vika@Dima-Vika-PC Версия 1.0.75.0.181022	1 с	Radmin	10481	✗	
		LSBons	10484	✗	
		TightVNC	10485	✓	11.03.2019, 12:36:45 Архив снимков
Litvinov litvinov@IM/WS2016 Версия 1.0.87.0.190117	1 с	Radmin	10541	✗	
		LSBons	10544	✗	
		TightVNC	10545	✗	05.03.2019, 14:03:55 Архив снимков
Litvinov v@DESKTOP-KFPFCO4 Версия 1.0.75.0.181022	90 дн 22 ч 50 мин 19 с	Radmin	10441	✗	
		LSBons	10444	✗	
		TightVNC	10445	✓	10.12.2018, 13:44:49 Архив снимков

Условный псевдоним для клиента и версия клиентского ПО

Цветом выделены строки таблицы, соответствующие запущенным клиентам

Интерфейс для выбора даты и времени

Снимки за указанный час

Сервис	Статус
Netstats	●
Screenshot-2	●
Tunnel: Radmin	●
Tunnel: TightV...	●
Tunnel: USBons	●

Версия 1.0.85.0.181112 [Лицензия](#)

Рис. 2. Главное окно веб-интерфейса со списком зарегистрированных клиентов, состоянием подключений и сервисов, а также последним снимком экрана (вверху). Окно просмотра архивных снимков экрана для выбранного клиента (внизу слева) и главное окно клиента (внизу справа)

Fig. 2. The main window of the web interface with a list of registered clients, the status of connections and services, as well as the latest screen shot. Bottom left is a window for viewing archived screenshots for a selected client. Bottom right is the main client window.

Поскольку требования к программному средству в будущем могут меняться, то может нарушаться совместимость клиентской и серверной частей. Для контроля версии ПО и свое-

временного его обновления на веб-интерфейсе отображается версия запущенного клиента. Версия присваивается автоматически во время сборки в системе непрерывной интеграции Jenkins.

Тестирование

В результате тестирования по описанной выше методике были получены следующие характеристики и выводы:

- 1) минимальная скорость, при которой удалось добиться стабильной работы программного средства, – 3 кбит/с на прием и передачу данных;
- 2) снимок экрана в формате PNG с максимальной степенью сжатия и 24 битами цвета занимает объем около 200 КБ при типичном для бурения наборе открытых окон и их компоновки;
- 3) в качестве заставки рекомендуется использовать сплошную заливку одним цветом, так как снимок такого рабочего стола лучше сжимается;
- 4) программное средство работает стабильно в течение трех-четырех недель, затем может произойти критический сбой, причина которого пока не выявлена.

Апробация

Представленное программное средство прошло этап внедрения в отдел программного обеспечения НПП ГА “Луч”. Данное внедрение имело во многом тестовый характер, в его рамках проводились испытания с ограничением скорости приема и передачи данных. Но нашлось и практическое применение для данного средства: оно было использовано для учета рабочего времени сотрудников и заполнения таблиц и для оказания поддержки в использовании телеметрической системы “Луч”. Программная часть данной телеметрической системы устанавливается на ПК полевого инженера наклонно направленного бурения. В нее входят сервер баз данных; сервис RealDepth5 [5], взаимодействующий непосредственно с аппаратной частью телесистемы, регистрирующий параметры бурения и предоставляющий к ним доступ по специальному протоколу; графический монитор полевого инженера, взаимодействующий с сервисом и отображающий параметры бурения скважины. В ходе эксплуатации телеметрической системы была обнаружена ошибка в работе программы, вызванная переходом базы данных в несогласованное состояние. С помощью разработанного средства удалось подключиться к СУБД и восстановить работоспособность телеметрической системы. При этом от полевого инженера не требовалось никаких действий по работе с этим средством. Кроме того, по запросу осуществляется подключение к сервису RealDepth5 для оказания консультаций по работе комплекса.

В ходе данного внедрения был создан образ виртуальной машины с серверной частью разработанного средства – с пустым списком зарегистрированных клиентов и отсутствием каких-либо записей об активности. Если возникает необходимость разграничить доступ ведущих инженеров к разным ПК буровых, то это можно осуществить, создав нужное количество копий образа виртуальной машины и изолировав их друг от друга.

В настоящее время проводится внедрение в телеметрическую службу ПАО «Сургутнефтегаз». В данной организации уже имеется сеть VPN, объединяющая буровые с офисом управления бурением, поэтому возможность удаленного доступа уже была там до внедрения. Предполагается, что разработанное программное средство найдет применение в качестве монитора состояния сервисов на буровых и как единая точка доступа к данным бурения. Кроме того, полезной будет информация об активности пользователей, например, при разборе нестандартных ситуаций.

Заключение

Разработано программное средство для прозрачного удаленного доступа к технологическим параметрам бурения и данным скважинной телеметрической системы в условиях нестабильной связи. За основу средства была взята клиент-серверная архитектура с сервером, расположенным в офисе, и клиентами, расположенными на буровой и имеющими прямой доступ к данным бурения. С применением данного средства ведущие инженеры наклонно направленного бурения для доступа к данным бурения могут использовать те же программные средства, что и полевые инженеры. Средство из-за набора технологий получилось универсальным, пригодным не только для удаленного доступа к данным бурения, а к любым данным, доступным по TCP/IP на ПК буровой. Реализован сбор снимков экранов и времени активности пользователей ПК буровой как дополнительного источника информации при разборе нештатных ситуаций.

Проведены тесты, по результатам которых определена пригодность средства для работы в полевых условиях с использованием спутниковой и мобильной связи буровой и офиса: программное средство работает стабильно при скорости приема и передачи данных, равной 3 кбит/с, и без сбоев в течение трех-четырех недель.

С помощью данного программного средства оперативно устранены неисправности в работе ПО забойной телеметрической системы «Луч» путем подключения к СУБД на ПК буровой и переводением ее в согласованное состояние. Разработанное программное средство используется для удаленной помощи в эксплуатации данного комплекса.

Из недостатков можно отметить возможную необходимость учитывать доступность TCP-портов сервера (а следовательно, и клиентских сервисов, предоставляющих доступ к данным бурения) из локальной сети офиса и предотвращать несанкционированный доступ к ним. На данный момент этот недостаток устраняется с помощью творческого подхода системных администраторов, например, используются межсетевые экраны.

Список литературы / References

1. **Amer A., Chinellato F., Collins S. et al.** Structural Steering – A Path to Productivity. *Oilfield Review* 25, 2013, no. 1, p. 14–31.
2. **Gorgone I., Gomez J., Uddenberg G.** Remote operations, the future of drilling: Directional drilling remote control. In: SPE/IADC Drilling Conference, Proceedings, 2008, vol. 1, p. 554–559.
3. **Власов А. А., Фаге А. Н., Тейтельбаум Д. В., Расковалов П. С.** Программа для предварительной обработки данных автономного каротажа RealDepth 5, №2014662063 от 21 ноября 2014 г.
Vlasov A. A., Fage A. N., Teitel'baum D. V., Raskovalov P. S. Programma dlya predvaritel'noi obrabotki dannykh avtonomnogo karotazha RealDepth 5, №2014662063, 21 November 2014. (in Russ.)
4. **Sun Q., Sun Y.** The application of real-time data remote transmission for LWD/MWD in oil field. In: Proceedings of the International Conference on E-Commerce, E-Business and E-Service, EEE, 2014, vol. 1, p. 155–158.
5. **Косонова Н. О., Баранов Д. В., Власов А. А., Тейтельбаум Д. В.** Информационное средство хранения, мониторинга и первичной обработки данных забойной телеметрической системы измерений процесса бурения // Интерэкспо ГЕО-Сибирь-2016. XII Междунар. науч. конгр. «Недропользование. Горное дело. Направления и технологии поиска, разведки и разработки месторождений полезных ископаемых. Геоэкология» (г. Новосибирск, 18–22 апреля 2016 г.): Сб. материалов: В 4 т. Новосибирск, 2016. Т. 2. С. 43–47.

Kosogova N. O., Baranov D. V., Vlasov A. A., Teitelbaum D. V. Informatsionnoe sredstvo khraneniya, monitoringa i pervichnoi obrabotki dannykh zaboinoi telemetricheskoi sistemy izmerenii protsesse bureniya. In: Interexpo GEO-Siberia-2016. XII International Exhibition and Scientific Congress. Novosibirsk, 2016, vol. 2, p. 43–47.

*Материал поступил в редколлегию
Received
15.03.2019*

Сведения об авторах / Information about the Authors

Литвинов Василий Сергеевич, магистрант факультета информационных технологий Новосибирского государственного университета (ул. Пирогова, 1, Новосибирск, 630090, Россия)

Vasiliy S. Litvinov, Master's Student, Faculty of Information Technologies, Novosibirsk State University (1 Pirogov Str., Novosibirsk, 630090, Russian Federation)

kvas.omsk@gmail.com

Власов Александр Александрович, кандидат технических наук, старший научный сотрудник, Институт нефтегазовой геологии и геофизики им. А. А. Трофимука СО РАН (пр. Академика Коптюга, 3, Новосибирск, 630090, Россия)

Alexander A. Vlasov, Candidate of Science (Techniques), Senior Researcher, Trofimuk Institute of Petroleum-Gas Geology and Geophysics SB RAS (3 Academician Koptyug Ave., Novosibirsk, 630090, Russian Federation)

vlasovaa@ipgg.sbras.ru

Тейтельбаум Дмитрий Владимирович, начальник отдела ПО, ООО Научно-производственное предприятие геофизической аппаратуры «Луч» (ул. 2-я Юргинская, 34, Новосибирск, 630051, Россия)

Dmitry V. Teytelbaum, head of IT department of SPE "Looch" (34 2-ya Yurginskaya Str., Novosibirsk, 630051, Russian Federation)

teytelbaum@gmail.com

Применение методов глубинного обучения для обнаружения вторжений

В. А. Нечахин¹, Б. Н. Пищик^{1,2}

¹Новосибирский государственный университет
Новосибирск, Россия

²Институт вычислительных технологий СО РАН
Новосибирск, Россия

Аннотация

Приведены результаты применения глубоких нейронных сетей для детектирования вредоносной активности в сетевом трафике. В процессе исследования реализованы два вида нейронной сети: рекуррентный автоэнкодер и генеративно-состязательная сеть. Приведены результаты исследования на наборе данных CICIDS2017.

Ключевые слова

обнаружение вторжений, глубинное обучение, CICIDS2017, автоэнкодер, генеративно-состязательная сеть.

Для цитирования

Нечахин В. А., Пищик Б. Н. Применение методов глубинного обучения для обнаружения вторжений // Вестник НГУ. Серия: Информационные технологии. 2019. Т. 17, № 2. С. 114–121. DOI 10.25205/1818-7900-2019-17-2-114-121

Using Deep Learning Methods for Intrusion Detection

V. A. Nechakhin¹, B. N. Pishchik^{1,2}

¹Novosibirsk State University
Novosibirsk, Russian Federation

²Institute of Computational Technologies SB RAS
Novosibirsk, Russian Federation

Abstract

One of the ways of ensuring information security are intrusion detection systems (IDS). IDS are used to detect malicious activity on the network. The standard approach to the detection of attacks is looking for specific patterns, such as byte sequences in network traffic, or known malicious instruction sequences used by malware. This approach is highly efficient, but it does not able to detect the attacks without patterns. Modern approaches to detection of attacks use deep learning.

The purpose of this work was to explore the possibility of building a universal classifier of network traffic based on a deep neural network.

For this, a recurrent autoencoder was trained on TCP packets from the CICIDS2017 dataset. During training the neural network was a model in which the expected vector was set the same as the original one. And learning was on normal traffic. The main idea was that a recurrent autoencoder trained in this way should recover anomalous traffic with a high loss. The TCP package is considered malicious if the recovery loss is above the threshold. However, the accuracy of recovering normal TCP packets was low due to the insufficient model capacity and the lack of the suitable representation learning method.

After the results analyzing, we proposed an approach that can improve accuracy of detection for some attacks. Based on this approach, the VAEGAN network was trained on normal network flows from CICIDS2017. The VAEGAN was used to detect malicious network flows: to calculate the anomaly score for flow; if score is above the threshold – the flow is malicious.

The VAEGAN network showed a high percentage of attacks detection and the F-score value – 0.933.

Keywords

anomaly detection, deep learning, CICIDS2017, autoencoder, generative adversarial network

For citation

Nechakhin V. A., Pishchik B. N. Using Deep Learning Methods for Intrusion Detection. *Vestnik NSU. Series: Information Technologies*, 2019, vol. 17, no. 2, p. 114–121. (in Russ.) DOI 10.25205/1818-7900-2019-17-2-114-121

Введение

Выявление вредоносной активности в сетевом трафике является актуальной задачей, к решению которой существует несколько подходов [1].

Классический подход состоит в написании шаблонов, по которым в сетевом трафике детектируется вредоносная активность. Данный подход является высокоэффективным, но не позволяет обнаруживать атаки, для которых отсутствует шаблон.

Современные подходы базируются на том, что детектирование вредоносного трафика является частным случаем классической задачи классификации, которая в настоящее время успешно решается с помощью машинного обучения, в частности, нейронных сетей. Однако простые алгоритмы машинного обучения являются эффективными только при правильном ручном извлечении признаков из исходных данных. А извлечение необходимых признаков во многих случаях само по себе является не тривиальной задачей. Применение методов *обучения признакам* позволяет автоматически выявлять признаки, которые необходимы для дальнейшей классификации. А для выявления высокоуровневых признаков используют методы глубинного обучения [2].

Цель работы – исследовать возможность построения универсального классификатора сетевого трафика на основе глубокой нейронной сети.

CICIDS2017

Для обучения нейронной сети был сформирован набор данных CICIDS2017¹, который является публично доступным и предназначен для исследований в области кибербезопасности и обнаружения вторжений. Набор данных CICIDS2017 распространяется в виде файлов 2-х форматов: PCAP и CSV. Первые хранят дампы трафика, охватывающие пять дней сетевой активности, вторые содержат описание для каждого сетевого потока: статистические признаки и класс, к которому принадлежит поток.

Поток – это последовательность сетевых пакетов, у которых совпадают следующие поля: IP источника, IP получателя, порт источника, порт получателя и протокол. Для каждого потока выделяется более 80 признаков². Общие данные о наборе CICIDS2017 представлены в табл. 1.

Рекуррентный автоэнкодер

В качестве глубокой нейронной сети был выбран рекуррентный автоэнкодер [2] как нейронная сеть, предназначенная для обработки последовательностей различной длины. Обучение нейронной сети проходило на TCP пакетах из дампов трафика. Архитектура рекуррентного автоэнкодера представлена в табл. 2.

¹ Intrusion Detection Evaluation Dataset (CICIDS2017) // University of New Brunswick, Canadian Institute for Cybersecurity. URL: <https://www.unb.ca/cic/datasets/ids-2017.html> (дата обращения 21.11.2017).

² CICFlowMeter // University of New Brunswick, Canadian Institute for Cybersecurity. URL: <https://www.unb.ca/cic/research/applications.html#CICFlowMeter> (дата обращения 21.11.2017).

Таблица 1

Общие данные о CICIDS2017
General Information about CICIDS2017

Table 1

Класс	Вредоносность	Количество потоков
Benign	-	2 358 036
DoS Hulk	+	231 073
Port Scan		158 930
DDoS		41 835
DoS GoldenEye		10 293
FTPPatator		7 938
SSHPatator		5 897
DoS Slow Loris		5 796
DoS Slow HTTP Test		5 499
Botnet		1 966
Web Attack: Brute Force		1 507
Web Attack: XSS		652
Infiltration		36
Web Attack: SQL Injection		21
Heartbleed		11

Таблица 2

Архитектура рекуррентного автоэнкодера

Table 2

Recurrent Autoencoder Architecture

Слой	Число нейронов	Функция активации	Dropout
Энкодер			
GRU	128	ReLU	0.2
GRU	64	ReLU	0.2
Декодер			
GRU	128	ReLU	0.2
GRU	256	Softmax	0
Размерность входного слоя		256	
Оптимизатор		$RMSProp(lr = 10^{-5})$	
Функция ошибки		MSE	
Размер пакетов (batch size)		64	
Количество эпох		100	

Для более равномерного последующего нормирования входных значений каждому нейрону на вход подается 1 байт пакета. А выбранный размер входного слоя (256) является предположительно оптимальным с точки зрения качества работы сети и сложности вычислений.

При обучении рекуррентный автоэнкодер представлял собой модель, в которой ожидаемый вектор устанавливался таким же, как и исходный. А обучение происходило исключительно на легитимном трафике. Идея заключалась в том, что обученный таким образом рекуррентный автоэнкодер должен восстанавливать аномальный трафик с *высокой степенью* ошибки. *Ошибка восстановления* рассчитывается как значение близости исходного TCP пакета и восстановленных данных.

Для того чтобы определить, является ли пакет вредоносным, необходимо ввести параметр «порог ошибки восстановления». Таким образом, корректируя значение порога ошибки вос-

становления можно настраивать точность выявления вредоносной активности. Однако во время обучения на сконструированном рекуррентном автоэнкодере не удалось получить точность восстановления более 60 %.

Предметная область (изображения, звук, текст) в существенной степени влияет на процесс обучения признакам. Например, сверточные нейронные сети в рамках задачи распознавания изображений используют иерархии объектов и базируются на особенностях работы зрительной коры головного мозга [3]. Для сетевого трафика такой концепции на данный момент не выработано. Отсутствие концепции обучения признакам и недостаточная ёмкость модели стали причинами низкой точности восстановления.

Сетевой трафик в наборе CICIDS2017 представляет собой совокупность потоков, в свою очередь, поток – это совокупность сетевых пакетов. Для обучения нейронной сети в качестве входных данных могут быть использованы сетевые пакеты или потоки в зависимости от того, какие классы атак необходимо детектировать. Достоинства и недостатки двух подходов можно проследить на примере обнаружения конкретных типов атак.

Атаки класса Brute Force сложно выявить инспекцией только отдельных сетевых пакетов. Ключевой особенностью таких атак является отправка множества однотипных пакетов за небольшой промежуток времени. Вредоносные и легитимные пакеты в данном случае слабо различимы.

Такие атаки, как SQL Injection или XSS, ключевая информация для детектирования которых содержится внутри полезной нагрузки, сложно детектировать на уровне потоков. При работе с сетевыми потоками информация заключенная внутри полезной нагрузки, теряется.

Вероятно, статистические признаки сетевого потока не позволяют закодировать достаточно много информации. Один поток может содержать сотни и тысячи сетевых пакетов, каждый из которых имеет собственные уникальные признаки. При агрегировании сетевых пакетов до уровня потоков теряется часть информации. Это то же самое, что взять предложения из некоторого текста и заменить вектором метаданных, который кодирует количество слов и символов в предложении.

Таким образом, перспективным можно считать подход, при котором для различных классов атак реализуется свой метод обнаружения, основанный на особенностях представления данных и применимости определенного типа нейронной сети. Это означает, что необходимо отказаться от идеи универсального классификатора и сосредоточить внимание только на определенной группе атак. Например, классы атак, представленные в CICIDS2017, можно разделить на две группы и использовать рекомендуемые типы нейронных сетей в соответствии с табл. 3.

Таблица 3

Рекомендуемые типы нейронных сетей для CICIDS2017

Table 3

Recommended Types of Neural Networks for CICIDS2017

Классы атак	Типы нейронной сети	Данные для обучения
DoS, DDoS, Port Scan, FTPPatator, SSHPatator, Bot, Web Attack: Brute Force, Infiltration, Heartbleed	Автоэнкодер и генеративно-состязательная сеть [4]	Сетевые потоки
Web Attack: SQL Injection и XSS	Seq2Seq автоэнкодер *	Сетевые пакеты

* Seq2seq-web-attack-detection // Positive Technologies. URL: <https://github.com/PositiveTechnologies/seq2seq-web-attack-detection> (дата обращения 22.02.2019).

VAEGAN

На основе предложенного подхода была разработана нейронная сеть, которая объединяет в себе вариационный автоэнкодер (Variational Autoencoder, VAE) и генеративно-состязатель-

ную сеть (Generative Adversarial Network, GAN). Сети такого типа были предложены в [5] и имеют название VAEGAN.

Обучение реализованной нейронной сети происходило без учителя, на легитимных сетевых потоках из набора CICIDS2017. В качестве тестовых данных были выбраны легитимные сетевые потоки, которые не входили в тренировочный набор, и сетевые потоки всех типов вредоносной активности. Общая информация о тестовой выборке представлена в табл. 4, архитектура реализованной VAEGAN сети – в табл. 5.

Таблица 4

Общая информация о тестовой выборке

Table 4

General Information about Test Dataset

Класс	Вредоносность	Количество потоков
Benign	–	1 000
Port Scan	+	1 000
DDoS		1 000
DoS		1 000
FTPPatator		1 000
SSHPatator		1 000
Botnet		1 000
Web Attack: Brute Force		1 000
Web Attack: XSS		652
Infiltration		36
Web Attack: SQL Injection		21
Heartbleed		11

Таблица 5

Архитектура VAEGAN

Table 5

VAEGAN Architecture

Слой	Число нейронов	Функция активации	Dropout
Энкодер (E)			
Dense	64	LeakyReLU($\alpha = 0.2$)	0.2
Dense	32	LeakyReLU($\alpha = 0.2$)	0.2
Dense	32	LeakyReLU($\alpha = 0.2$)	0.2
Декодер (G)			
Dense	32	LeakyReLU($\alpha = 0.2$)	0.2
Dense	32	LeakyReLU($\alpha = 0.2$)	0.2
Dense	64	LeakyReLU($\alpha = 0.2$)	0.2
Dense	80	Tanh	0
Дискриминатор (D)			
Dense	64	LeakyReLU($\alpha = 0.3$)	0.2
Dense	64	LeakyReLU($\alpha = 0.3$)	0.2
Dense	32	LeakyReLU($\alpha = 0.3$)	0.2
Dense	1	Sigmoid	0
Оптимизатор		$Adam(\alpha = 0.001, \beta_1 = 0.5)$	
Размер пакета (batch size)		64	
Размерность входного слоя энкодера и дискриминатора		80	
Размерность кода		20	
Начальная инициализация весов		Xavier normal	

В VAEGAN вариационный автоэнкодер получает на вход вектор признаков x для сетевого потока и учится восстанавливать максимально похожий вектор $\tilde{x} = G(E(x))$ на выходе. Дискриминатор принимает на вход восстановленный \tilde{x} и исходный x векторы и учится предсказывать, какой вектор является настоящим, а какой восстановленным.

Выявление вредоносной активности реализовано на основе подхода, представленного в [6]. Для того чтобы определить, является ли сетевой поток вредоносным, вычисляется значение аномальности:

$$\mathcal{L}(x) = 0.1 * \mathcal{L}_D + 0.9 * \mathcal{L}_R,$$

где

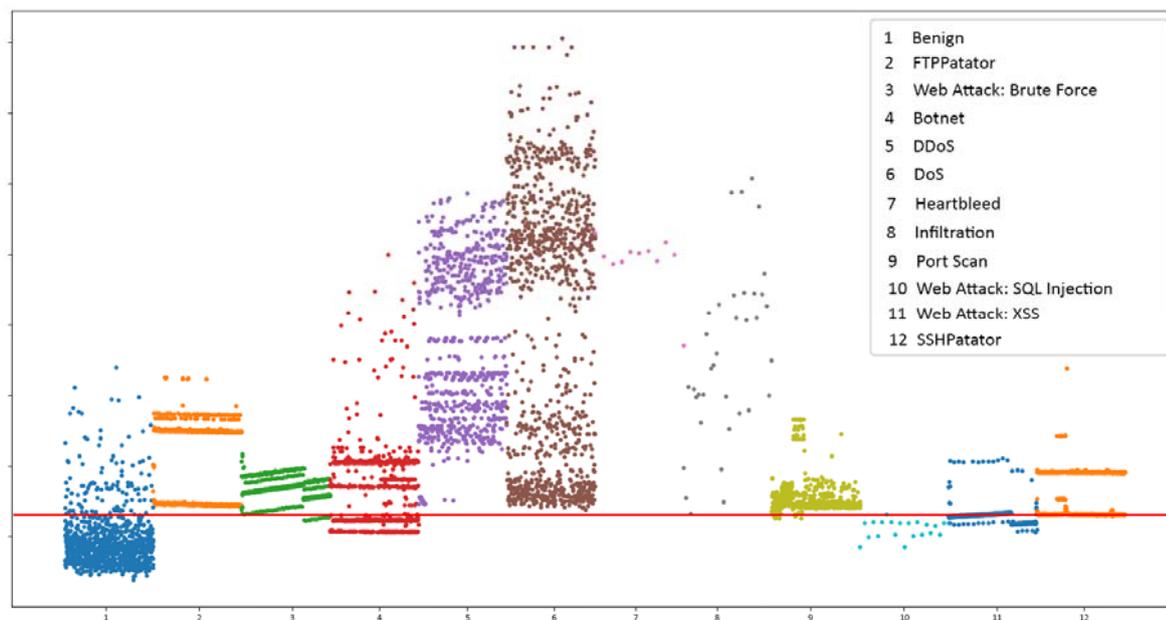
x – сетевой поток,

$\tilde{x} = G(E(x))$ – восстановленный вариационным автоэнкодером сетевой поток,

$\mathcal{L}_D(\tilde{x}) = \sum |D_l(x) - D_l(\tilde{x})|$ – ошибка на l -м слое дискриминатора,

$\mathcal{L}_R(\tilde{x}) = \sum |x - \tilde{x}|$ – ошибка вариационного автоэнкодера.

Поток считается вредоносным, если значение $\mathcal{L}(x)$ превышает установленный порог. Корректировать точность выявления вредоносной активности можно, регулируя значение порога аномальности. Полученные результаты представлены на рисунке.



Результат работы VAEGAN
The VAEGAN Results

Каждая точка на этом рисунке отображает значение аномальности для сетевого потока. Красная линия – установленный порог. В результате полученная точность составила 0,983, полнота – 0,889, значение F-меры – 0,933. Процент правильной классификации по классам представлен в табл. 6.

Таблица 6

Результат работы VAEGAN

Table 6

The VAEGAN Results

Класс	Правильно классифицированные потоки, %
Benign	88,2
Port Scan	98,3
DDoS	100
DoS	100
FTPPatator	100
SSHPatator	99,7
Botnet	63,9
Web Attack: Brute Force	98,6
Web Attack: XSS	48,3
Infiltration	100
Web Attack: SQL Injection	4,8
Heartbleed	100

Заключение

В настоящее время многие задачи классификации успешно решаются с помощью глубоких нейронных сетей. В работе был сконструирован рекуррентный автоэнкодер для задачи классификации сетевого трафика. Использование нейронной сети данного типа было обусловлено тем, что исходными данными являлись последовательные сетевые пакеты различной длины. Однако не было заложено никакой концепции обучения признакам. В результате точность восстановления оказалась низкой.

При анализе полученных результатов были рассмотрены ключевые особенности сетевого трафика в рамках задачи классификации. Это позволило сделать предположение о неэффективности универсального классификатора и предложить подход, повышающий точность обнаружения вредоносного трафика. На основе сформулированного подхода на легитимных сетевых потоках была обучена VAEGAN сеть. В результате удалось получить высокую точность выявления сетевых потоков для выделенных классов атак. Низкий процент детектирования оказался для атак SQL Injection, XSS и Botnet, для успешного выявления которых требуется инспекция полезной нагрузки и применение рекомендуемого в табл. 3 типа нейронной сети.

Список литературы / References

1. **Браницкий А., Котенко И.** Анализ и классификация методов обнаружения сетевых атак // Тр. СПИИРАН. 2016. №. 45. С. 207-244. DOI: <https://doi.org/10.15622/sp.45.13>
Branitskii A., Kotenko I. Analiz i klassifikatsiya metodov obnaruzheniya setevykh atak [Analysis and classification of network attack detection methods]. In: Trudy SPIIRAN (SPIIRAS Proceedings), 2016, vol. 45, p. 207–244. (in Russ.) DOI 10.15622/sp.45.13
2. **Goodfellow I., Bengio Y., Courville A.** Deep Learning. MIT Press, 2016, 802 с.
3. **Masakazu M., Mori K., Mitari Y., Kaneda Y.** Subject Independent Facial Expression Recognition With Robust Face Detection Using a Convolutional Neural Network. *Neural Networks*, 2003, no. 16, p. 555–559. DOI 10.1016/S0893-6080(03)00115-1

4. **Zenati H., Foo C. S., Lecouat B., Manek G., Chandrasekhar V. R.** Efficient GAN-based anomaly detection. In: 6th International Conference on Learning Representations, 2018.
5. **Larsen A. B. L., Sonderby S. K., Larochelle H., Winther O.** Autoencoding beyond pixels using a learned similarity metric. In: ICML'16 Proceedings of the 33rd International Conference on International Conference on Machine Learning, 2016, no. 48, p. 1558–1566.
6. **Schlegl T., Seeböck P., Waldstein S. M., Schmidt-Erfurth U., Langs G.** Unsupervised Anomaly Detection with Generative Adversarial Networks to Guide Marker Discovery. In: The proceedings of the international conference on Information Processing in Medical Imaging (IPMI), 2017. DOI 10.1007/978-3-319-59050-9_12

Материал поступил в редколлегию

Received

02.04.2019

Сведения об авторах / Information about the Authors

Нечахин Владислав Александрович, магистрант факультета информационных технологий Новосибирского государственного университета (ул. Пирогова, 1, 630090, Новосибирск, Россия)

Vladislav A. Nechakhin, Master's Student, Faculty of Information Technologies, Novosibirsk State University (1 Pirogov Str., Novosibirsk, 630090, Russian Federation)

v.nechakhin@g.nsu.ru

Пищик Борис Николаевич, кандидат технических наук, заведующий кафедрой компьютерных систем факультета информационных технологий Новосибирского государственного университета (ул. Пирогова, 1, 630090, Новосибирск, Россия), старший научный сотрудник Института вычислительных технологий СО РАН (пр. Академика Лаврентьева, 6, Новосибирск, 630090, Россия)

Boris N. Pishchik, Candidate of Science (Technology), Head of the Computer Systems Department at the Faculty of Information Technology, Novosibirsk State University (1 Pirogov Str., Novosibirsk, 630090, Russian Federation), Senior Researcher, Institute of Computational Technologies SB RAS (Pr. Academician Lavrentiev, 6, Novosibirsk, 630090, Russian Federation)

b.pishchik@nsu.ru

PishchikBN@ict.nsc.ru

УДК 004.9
DOI 10.25205/1818-7900-2019-17-2-122-137

Опыт разработки и внедрения информационной системы для поддержки процесса направления студентов на практику в НИУ МЭИ

С. А. Петров

*Национальный исследовательский университет МЭИ
Москва, Россия*

Аннотация

Рассмотрен обязательный для высших образовательных учреждений РФ процесс проведения практических занятий студентов, приведены ссылки на нормативные документы, описан подход, устоявшийся в НИУ МЭИ. До начала проведения практики каждый студент должен быть прикреплен к определенной базе практики – учреждению с соответствующей профилю подготовки обучающегося специализацией, которое будет осуществлять проведение и контроль практических занятий. Если база практики внешняя, т. е. не входит в организационную структуру образовательного учреждения, то с ней должен быть заключен договор. В статье представлен опыт разработки и внедрения в НИУ МЭИ информационной системы, автоматизирующей процесс распределения студентов по базам практик и подготовку сопутствующей документации. Программная реализация включает модуль импорта данных из других информационных систем, модуль редактирования календарного графика практик, модуль ведения справочной информации о предприятиях-партнерах, модуль формирования и согласования заявок с распределением студентов, модуль подготовки печатных форм документов, модуль построения статистической отчетности. Описана работа каждого из модулей, отмечаются их преимущества перед ручной обработкой данных. Раскрыты некоторые технические решения и характеристики программной системы, отдельное внимание уделено разграничению прав доступа.

Ключевые слова

практика студентов, согласование заявок, подготовка договоров, электронный документооборот

Для цитирования

Петров С. А. Опыт разработки и внедрения информационной системы для поддержки процесса направления студентов на практику в НИУ МЭИ // Вестник НГУ. Серия: Информационные технологии. 2019. Т. 17, № 2. С. 122–137. DOI 10.25205/1818-7900-2019-17-2-122-137

Experience in Development and Implementation of the Information System, Supporting Students Assignment for Practice to NRU MPEI

S. A. Petrov

*National Research University "Moscow Power Engineering Institute"
Moscow, Russian Federation*

Abstract

The article describes the arrangement of internship for students that is required for higher educational institutions of the Russian Federation; provides links to regulatory documents, describes the approach established at NRU MPEI. Prior to the beginning of the internship, each student should be assigned to a particular internship place – an institution with corresponding specialization that matches the student's major. That institution will conduct and supervise the internship. In case of external internship place, i.e. if it is not included into the organizational structure of an educational institution, then an agreement with this organization should be concluded. This article presents the experience of the development and implementation of the information system at NRU MPEI which automates the students' internship assignment and prepares the related documentation. The program implementation includes a module for data import from other information systems, a module for the internship schedule editing, a module for reference information on partner institutions, a module for the formation and coordination of internship assignment applications, a module for

© С. А. Петров, 2019

the preparation of the printed forms of documents, a module for statistical reporting. The article describes the functioning of each module with the observation of its advantages over the manual data processing. Some technical solutions and characteristics of the program system have been disclosed; special attention is paid to the access rights management.

Keywords

student practice, approval of applications, preparation of contracts, electronic document management

For citation

Petrov S. A. Experience in Development and Implementation of the Information System, Supporting Students Assignment for Practice to NRU MPEI. *Vestnik NSU. Series: Information Technologies*, 2019, vol. 17, no. 2, p. 122–137. (in Russ.) DOI 10.25205/1818-7900-2019-17-2-122-137

Введение

Неотъемлемой частью любых теоретических познаний является их применение на практике. Как бы подробно и детально не был разобран изучаемый вопрос теоретически, человек, успешно прошедший обучение, на практике может столкнуться с совершенно новыми и неизвестными для него трудностями. Это хорошо известный факт, который требует от современных учебных заведений в обязательном порядке включать в учебный процесс мероприятия, предлагающие учащимся на практике применить полученные знания для решения реальных задач. В статьях [1; 2] отмечается высокая роль учебной и производственной практик в формировании профессиональных компетенций и трудоустройстве выпускников. Проведение практики положительно оценивается как самими студентами, так и их будущими работодателями [3; 4].

В контексте высших учебных заведений, в частности НИУ МЭИ, следует упомянуть приказ Министерства образования и науки Российской Федерации (ныне – Министерство науки и высшего образования Российской Федерации) от 27 ноября 2015 г. № 1383, утверждающий «Положение о практике обучающихся, осваивающих основные профессиональные образовательные программы высшего образования»¹. Положение распространяется на все организации, осуществляющие основные профессиональные образовательные программы высшего образования (ОПОП ВО), и накладывает требования на программу практики, дает описание форм проведения практики, регламентирует деятельность лиц, участвующих в процессе направления и прохождения практики обучающимися, и характеризует другие аспекты, связанные с практикой. В свою очередь, федеральные государственные образовательные стандарты высшего образования (ФГОС ВО), являющиеся основой для любой программы подготовки, разрабатываемой в вузе, включают в программу блок «Практика» и требуют проведения различных типов практик в ходе обучения. Например, реализуемые в МЭИ ФГОС ВО для бакалавриата² и магистратуры³ по направлению подготовки 13.03.01 Теплоэнергетика и теплотехника.

До 27.01.2018 Положение о практике⁴ явно выделяло два вида практики – учебную и производственную. Первая, учебная, была ориентирована на приобретение первичных профессиональных умений и навыков и обычно проводилась в стенах вуза (являлась стационарной по способу проведения). Вторая, производственная (в том числе преддипломная), была направлена для получения углубленных профессиональных умений и опыта профессиональной деятельности и часто связана с направлением обучающихся на работу во внеш-

¹ Приказ Министерства образования и науки РФ от 27 ноября 2015 г. № 1383 «Об утверждении Положения о практике обучающихся, осваивающих основные профессиональные образовательные программы высшего образования» (с изменениями и дополнениями). <http://base.garant.ru/71288178/>

² ФГОС ВО – бакалавриат по направлению подготовки 13.03.01 Теплоэнергетика и теплотехника. <https://mpei.ru/umo/HigherEducation/Documents/standards/projects/13.03.01.pdf>

³ ФГОС ВО – магистратура по направлению подготовки 13.03.01 Теплоэнергетика и теплотехника. <https://mpei.ru/umo/HigherEducation/Documents/standards/projects/13.04.01.pdf>

⁴ Приказ Министерства образования и науки РФ от 27 ноября 2015 г. № 1383 «Об утверждении Положения о практике обучающихся, осваивающих основные профессиональные образовательные программы высшего образования» (с изменениями и дополнениями).

ние организации (являлась выездной по способу проведения). Согласно приказу⁵, начиная с 28.01.2018, организациям разрешено самостоятельно устанавливать виды практик и способы их проведения в соответствии с ФГОС ВО, однако многие вузы продолжают считать основными видами практик учебную и производственную⁶.

Более сложной, с юридической точки зрения, является практика, которая проводится вне стен учебного заведения – на базе внешней организации. В этом случае, помимо внутренних документов вуза, которые готовятся участниками в ходе проведения практики, с внешней организацией должен быть заключен официальный договор, устанавливающий обязанности и ответственность сторон⁷. Для большинства организаций-партнеров подходит некоторый устоявшийся формат договоренностей, и договор готовится по стандартной форме, принятой в вузе. Но в некоторых случаях необходима большая гибкость, и сторонами может быть сформирован особый договор в согласованной форме.

Все обучающиеся направляются на практику согласно календарному графику, в котором строго определены даты начала и окончания всех видов учебной деятельности в семестре, в том числе и практик. Все договоры с организациями-партнерами должны быть заключены строго до начала проведения практики.

Учитывая тот факт, что у вузов с большим количеством обучающихся, вероятнее всего, имеется и немалое количество организаций-партнеров, а также сжатость сроков для подготовки необходимых документов, естественным образом встает вопрос автоматизации процесса направления студентов на практику. Если в некотором вузе рассмотренная задача и решается вручную, то увеличение размера «исходных» данных (а учебное заведение стремится к увеличению количества обучающихся) с большой вероятностью приведет к повышению человеческих трудозатрат, появлению ошибок в документах, нарушению регламентов и, в целом, обнаружит нерентабельность «ручной» обработки.

Практика в НИУ МЭИ

Федеральное государственное бюджетное образовательное учреждение высшего образования «Национальный исследовательский университет МЭИ» (НИУ МЭИ) включает в себя 12 институтов и более 60 кафедр, которые принимают непосредственное участие в обучении более 13 000 студентов по программам бакалавриата, специалитета и магистратуры [5; 6].

Прохождение практик является неотъемлемой частью учебного процесса для всех студентов МЭИ и регламентируется локальным положением⁸. На различных курсах обучения студенты должны пройти учебную, производственную, преддипломную и научно-производственную практики. В качестве базы практики – учреждения, к которому прикрепляется студент, – могут выступать как подразделения МЭИ (на конец 2018 г. в организационной структуре МЭИ насчитывалось более 350 подразделений), так и внешние организации-партнеры. В ходе учебной практики студенты обычно выступают в роли ассистентов преподавателей кафедры и помогают им в проведении занятий. При проведении остальных видов

⁵ Приказ Министерства образования и науки РФ от 15 декабря 2017 г. № 1225 «О внесении изменений в Положение о практике обучающихся, осваивающих основные профессиональные образовательные программы высшего образования, утвержденное приказом Министерства образования и науки Российской Федерации от 27 ноября 2015 г. № 1383». <http://base.garant.ru/71854842/>

⁶ Положение о порядке проведения практики студентов ФГБОУ ВО «НИУ «МЭИ» от 30.10.2017. https://df.mpei.ru/about_filial/Documents/reg_11.PDF; Положение о проектной, научно-исследовательской деятельности и практиках студентов Национального исследовательского университета «Высшая школа экономики» от 05.09.2016. https://www.hse.ru/data/xf/588/638/1120/Положение_о_проектной,_научно-исследоват.итета_«Высшая_школа_экономики».docx

⁷ Приказ Министерства образования и науки РФ от 27 ноября 2015 г. № 1383 «Об утверждении Положения о практике обучающихся, осваивающих основные профессиональные образовательные программы высшего образования» (с изменениями и дополнениями).

⁸ Положение о порядке проведения практики студентов ФГБОУ ВО «НИУ «МЭИ» от 30.10.2017; Положение о проектной, научно-исследовательской деятельности и практиках студентов Национального исследовательского университета «Высшая школа экономики» от 05.09.2016.

практик студенты, по возможности, направляются для работы во внешние организации и решают поставленные там задачи. Для многих из них – это первый опыт работы, а для некоторых еще и успешное начало карьеры.

В НИУ МЭИ устоялся определенный процесс направления студентов на практику и подготовки сопроводительной документации. Каждая кафедра выбирает одного или нескольких ответственных за практику сотрудников для взаимодействия со студентами, организациями-партнерами и отделом занятости и практических форм обучения (ОЗ и ПФО). На плечи ответственных ложится достижение первоначальных договоренностей о прохождении практики конкретных студентов в конкретных организациях. В одних случаях инициатива исходит от студентов, в других – от организаций. Ответственные вносят эти договоренности в заявку на проведение практики и передают ее в ОЗ и ПФО. В заявке содержится информация о виде практике, ее сроках и распределении студентов по базам практик.

Получив заявку на практику, сотрудники ОЗ и ПФО готовят договоры для всех студентов, направляемых во внешние организации. Возможны следующие виды документов:

- 1) разовый договор – стандартная форма МЭИ для заключения соглашения о разовом прохождении практики студентов на базе внешней организации;
- 2) дополнительное соглашение – используется при действующем долгосрочном договоре с организацией, в рамках которого МЭИ может направлять студентов на практику. Все «направления» фиксируются в дополнительных соглашениях;
- 3) договор по форме предприятия – аналог разового договора для случая, когда стандартная форма договора МЭИ не подходит внешней организации.

Каждый из подготовленных документов в двух экземплярах со стороны МЭИ подписывается начальником ОЗ и ПФО. Затем ответственный от кафедр забирает подготовленные документы и направляет их для подписания уполномоченным лицам во внешние организации, после чего один экземпляр возвращается в ОЗ и ПФО. Когда все студенты распределены по базам практик и подписаны сопровождающие документы, готовится приказ по МЭИ о направлении студентов на практику.

Формально процесс согласования и подготовки документов можно представить с помощью диаграммы деятельности (рис. 1). На приведенной диаграмме для компактности изображения использованы узлы, обозначающие элементарные действия, но некоторые из них на самом деле представляют собой более сложные «составные» шаги. Например, подготовка любого из видов документов разбивается на несколько составляющих: перенос из заявки сведений о внешней организации (наименование, адрес, подписант и др.), сроках и виде практики, направляемых студентах; поиск в информационных системах МЭИ дополнительных данных о студентах и включение их в текст документа; подготовка сопроводительного письма-направления для организации; печать и подпись документа. В случае дополнительного соглашения добавляется поиск и указание ссылки на долгосрочный договор.

На практике при подготовке документов не всегда удается соответствовать утвержденному регламенту. У кафедр нет доступа к актуальному списку студентов группы, его приходится поддерживать самостоятельно, из-за чего возможны «потери» при подготовке заявок. В некоторых случаях ответственные самостоятельно формируют договор, подписывают и передают его вместе с заявкой в ОЗ и ПФО. Это может приводить к юридическим ошибкам и необходимости повторно заключать договор. Кроме этого, в силу объективных и субъективных причин могут затягиваться сроки подписания, изменяться базы прохождения практик, и возникать другие сложности, требующие оперативного решения.

С учетом большого количества студентов, кафедр и организаций-партнеров огромная нагрузка ложится на сотрудников ОЗ и ПФО, которым вручную крайне сложно отследить и обработать все поступающие к ним изменения и запросы. В конце 2017 г. было принято решение о необходимости автоматизации процесса подачи заявок на проведение практики кафедрами и их обработки ОЗ и ПФО.

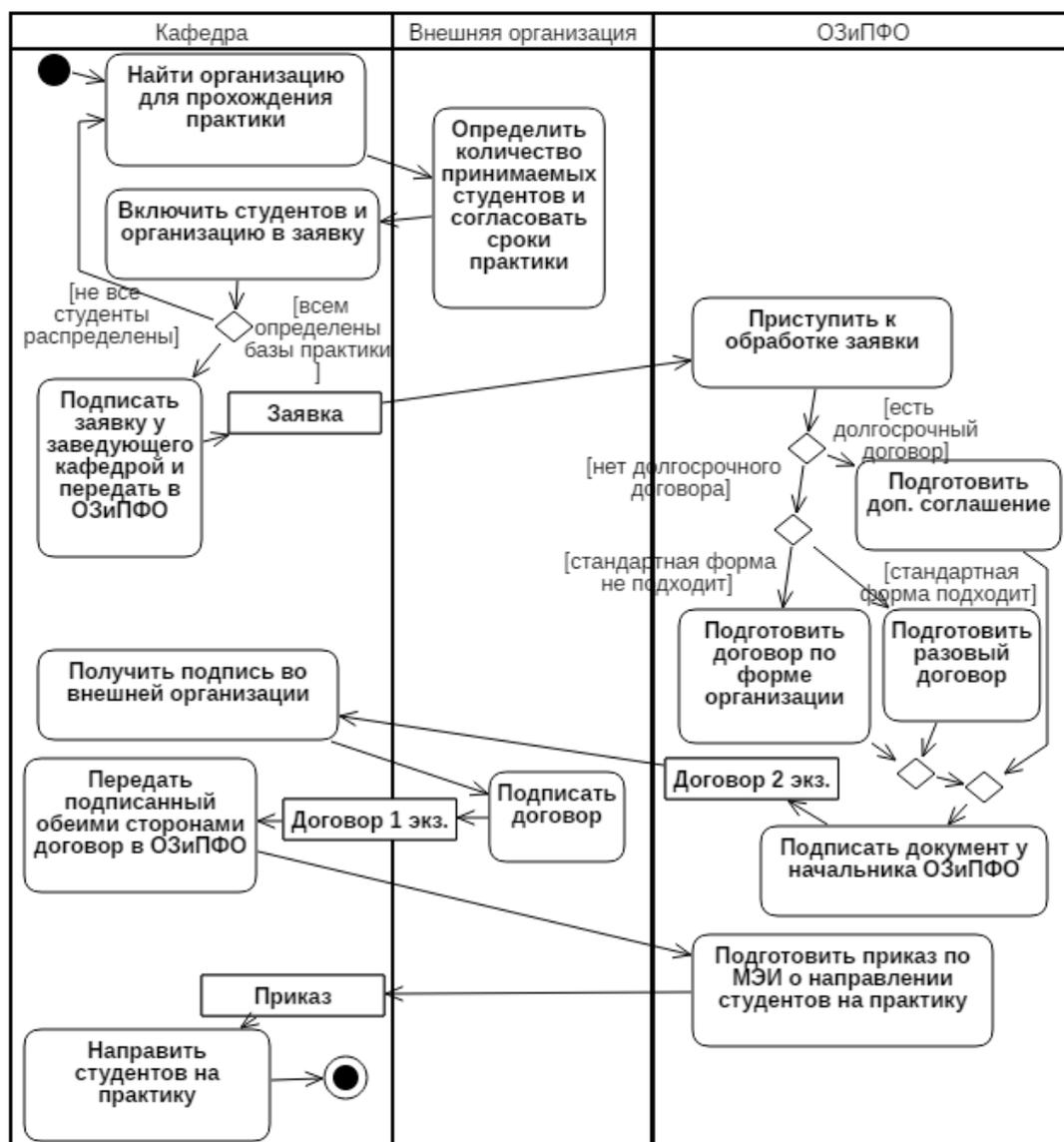


Рис. 1. Диаграмма деятельности для процесса согласования и подготовки документов на практику

Fig. 1. Activity Diagram for the Approval and Preparation Practice Document Process

ИС Практика

Процесс согласования заявок напоминает электронный документооборот и, вероятно, может быть автоматизирован за счет внедрения системы электронного документооборота (СЭД).

Основной недостаток использования «коробочных» СЭД – высокая стоимость приобретения, внедрения и обслуживания [7]. Само внедрение нередко «упирается» в консерватизм сотрудников и нежелание обучаться новым непривычным для них технологиям [8; 9]. При этом если СЭД не будет внедрена на всех рабочих местах, то и эффект от ее использования будет минимален [9]. Поэтому нередко организации приходят к решению использовать заказное программное обеспечение и, таким образом, разрабатывать решение «под себя» [10;

11]. Для принятия окончательного решения в каждом конкретном случае требуется оценить затраты на приобретение и разработку, а также потенциальную выгоду от внедрения нового ПО.

Согласно данным аналитического агентства «TAdviser», наиболее популярными СЭД на российском рынке являются: Directum, Elma, DocsVision, Дело (ЭОС), Тезис, 1С:Документооборот⁹. Но перечисленные решения слабо ориентированы на рассматриваемую задачу и не позволяют решить ее «из коробки». Для этого потребуются немалые доработки (например, разработка шаблонов документов) и реализация интеграционных процессов (например, для передачи данных о студентах). Конечно, большинство современных СЭД предоставляют возможности, упрощающие процессы адаптации продукта и внедрения в конкретной организации. Например, в состав СЭД Directum входит предметно-ориентированный инструмент разработки IS-Builder для гибкой настройки системы и Integration Toolset для интеграции с другими решениями. Однако все работы по внедрению должны проводить специалисты, обладающие высокой квалификацией в конкретной СЭД, а для НИУ МЭИ это означало бы привлечение внешних сотрудников, дополнительные затраты и увеличение сроков реализации проекта. Отметим, что у организации уже имеется опыт внедрения «коробочного» решения – платформы Alfresco, которая с 2015 г. используется в МЭИ качестве СЭД. Однако в настоящее время с системой работает лишь малая часть контингента вуза, что не позволяет рассматривать данный продукт для подготовки и согласования заявок на прохождение студентами практик.

В контексте автоматизации задач в сфере образования нельзя не упомянуть решения фирмы 1С – «1С:Университет» и «1С:Университет ПРОФ». Системы функционируют в таких вузах, как МЭСИ, МГТУ «СТАНКИН» и Санкт-Петербургский государственный институт кино и телевидения¹⁰. Однако, согласно описанию на официальном сайте, текущие возможности по управлению практиками ограничиваются подготовкой графиков распределения, приказов и протоколов распределения. Итерационное согласование несколькими пользователями системы будущих баз практик студентов (заявок на практику) пока не предусмотрено.

Другая проблема «готовых» решений – перегруженность и слабая адаптируемость пользовательского интерфейса. В основном они ориентированы на хорошо подготовленных узкоспециализированных операторов, а в согласовании заявок от кафедр задействованы преподаватели (в том числе по нетехническим дисциплинам), которым необходим минимальный, понятный и прозрачный интерфейс решения их задачи. В качестве примера «сложного» интерфейса можно привести решение 1С, включающее в себя панель разделов, панель «открытых», панель инструментов, панель функций текущего раздела, панель избранного, панель истории. Несомненным преимуществом продуктов 1С является возможность настройки интерфейса, что не всегда присуще решениям такого класса. Но, несмотря на это, многим конечным пользователям интуитивно сложно сориентироваться и сразу приступить к выполнению своей задачи.

К общим минусам «коробочных» продуктов были отнесены: высокая стоимость некоторых решений, ограничения бесплатных версий, вероятные сложности при интеграции с уже функционирующими системами. В итоге руководством НИУ МЭИ было принято решение о собственной разработке информационной системы под названием «Практика» (ИС Практика). Задача была направлена в отдел разработки и внедрения информационных систем, который взял на себя ответственность решить ее в сжатые сроки.

На начальном этапе проекта было утверждено техническое задание, разбивающее систему на несколько модулей.

1. Модуль импорта данных из существующих в МЭИ систем.

⁹ Аналитическое агентство «Тэдвайзер». Российский рынок СЭД/ECM. URL: <http://www.tadviser.ru/index.php/СЭД>.

¹⁰ Официальный сайт фирмы «1С». Внедренные решения. URL: <http://1c.ru/rus/partners/solutions/>.

2. Модуль «График проведения практики», предназначенный для ввода календарного графика практик по каждому институту.

3. Модуль «Предприятия», служащий для ведения справочной информации о предприятиях, с которыми взаимодействуют кафедры НИУ МЭИ.

4. Модуль «Заявки», предназначенный для формирования и обработки заявок на практику.

5. Модуль подготовки печатных форм.

6. Модуль построения статистической отчетности.

Такое разделение позволило вводить в опытную эксплуатацию ИС Практика поэтапно. Рассмотрим модули в порядке их запуска.

Модуль импорта ИС Практика обеспечивает ежедневный импорт сведений о студентах, учебных группах и организационной структуре МЭИ (кафедрах и вышестоящих институтах) из функционирующих в МЭИ систем кадрового учета студентов и сотрудников. Импортимые данные необходимы для начала процесса подготовки любой заявки. Доступ к актуальным спискам учебных групп существенно облегчает работу каждого ответственного от кафедры.

Модуль «График проведения практик» позволяет задать связь направления и курса обучения студентов с видом, трудоемкостью и сроками прохождения практики. Данная информация вводится в систему дирекциями институтов и утверждается учебным управлением МЭИ в начале каждого семестра. Выделение в ИС Практика графика проведения практик в отдельный набор таблиц, его утверждение и предоставление доступа к нему в виде справочника при подготовке заявок позволило устранить ошибки в указании сроков и в названии практик.

В модуле «Предприятия» все внешние организации разделены на две категории – партнеры МЭИ и партнеры кафедр.

Партнеры МЭИ – организации, с которыми заключены долгосрочные договоры. Долгосрочный договор обычно заключается на 5 лет и дает право взаимодействующим с организацией кафедрам направлять в нее студентов на практику по дополнительным соглашениям. Справочник партнеров МЭИ ведет сотрудник ОЗ и ПФО.

Партнеры кафедр – организации, с которыми заключаются разовые договоры на прохождение практики. У каждой кафедры в ИС Практика собственный справочник партнеров, к которому не имеют доступа другие подразделения. Исключением являются сотрудники ОЗ и ПФО, которые проверяют корректность указанных сведений, а также могут перевести организацию из категории «партнер кафедры» в категорию «партнер МЭИ» после заключения долгосрочного договора. Для борьбы с «дубликатами», возникающими в случае, если организация одновременно является партнером нескольких кафедр и переходит в категорию «партнер МЭИ», предусмотрены функции по слиянию нескольких записей «организация» и / или «подписант» в одну. Также ИС Практика контролирует появление «дублей» среди партнеров одной кафедры и партнеров МЭИ по ИНН.

Централизованный справочник организаций, с одной стороны, отображает и позволяет актуализировать ОЗ и ПФО сведения более чем о 600 внешних партнерах, а с другой – предлагает каждой кафедре личное пространство, не перегружая его чрезмерной информацией. Помимо основных сведений об организации (название, ИНН, юридический адрес и др.), система позволяет указывать список сотрудников-подписантов, отображает связанные с организацией документы и заявки (рис. 2).

Внесенные один раз в справочник партнеров данные могут повторно использоваться при распределении новых студентов по базам практик.

Модуль «Заявки» – основная часть ИС Практика. Так как процесс подготовки и обработки заявок тесно связан с появлением изменений, в системе реализована возможность итеративного согласования заявки между кафедрой и ОЗ и ПФО. Для успешной обработки заявка должна преодолеть цепочку из 4 статусов: «Подготовка на кафедре», «Подготовка в ОЗ и ПФО», «Обработана в ОЗ и ПФО», «Завершена».

Основные свойства Сотрудники Документы Заявки

Редактирование предприятия

Обращайте внимание на правильность сведений: названия организаций проверяйте в интернете (Заглавные и строчные буквы, ООО-ПАО, адрес).
По некорректно оформленным организациям создание договора невозможно!

Название: ООО "ААМ Автоматик"

Краткое название: ООО "ААМ Автоматик"

ИНН: 7722632361

Юридический адрес: 111250 Москва, Красноказарменная, 13, стр. 4.

Активно:

Сфера деятельности: IT и коммуникации

Подразделение-партнёр: НИУ МЭИ

Сохранить

Рис. 2. Фрагмент формы редактирования сведений о предприятии

Fig. 2. Fragment of the Company Editing Form

С каждым статусом ассоциируется определенная информация о состоянии заявки и полномочия различных пользователей.

1. Подготовка на кафедре (присваивается новой заявке). В этом статусе с заявкой работает ответственный от кафедры, включая в нее студентов и задавая им базы практики.

2. Подготовка в ОЗ и ПФО. Сотрудник ОЗ и ПФО проверяет корректность данных в заявке и готовит договоры и дополнительные соглашения для подписания.

3. Обработка ОЗ и ПФО – «сигнал» для ответственного от кафедры о том, что распечатанные и подписанные со стороны МЭИ документы можно забирать и направлять для подписи во внешнюю организацию.

4. Завершена. Сотрудник ОЗ и ПФО переводит заявку в этот статус, когда все документы, связанные с заявкой, подготовлены и подписаны заинтересованными сторонами.

При выявлении неточностей заявка может быть «возвращена» в предыдущий статус для доработки. Система позволяет добавлять комментарий при изменении статуса, давая понять, на что именно требуется обратить внимание. Для упрощения процесса согласования было принято организационное решение о возможности разделения одной группы студентов на несколько заявок.

Дополнительная сложность заявки – вложенность в нее множества документов, которые должны быть подписаны обеими сторонами. В отличие от статуса заявки, которую всегда первым подписывает заведующий кафедрой, для документа в общем случае неизвестна сторона, которая первой его подпишет. Поэтому для документа выделены признаки: «подписан МЭИ», «подписан организацией», «аннулирован». Данные характеристики проставляет со-

трудник ОЗ и ПФО при получении «бумажного» подтверждения свершившегося факта. Признак «аннулирован» может быть использован, например, при изменении базы практики – договор с первой организацией расторгается, а студент включается в повторную заявку с новой организацией.

Удобства ИС Практика для сотрудников кафедры уже отмечались – актуальный список студентов, доступ к утвержденному графику практик, справочник организаций-партнеров. Для сотрудников ОЗ и ПФО помимо указанных удобств реализованы и другие. Например, автоматическое включение в один договор с организацией всех связанных студентов в заявке (вручную можно разбить на несколько договоров); автоматический выбор внешнего подписанта; автоматическая подстановка долгосрочного договора при подготовке дополнительного соглашения; автоматическая нумерация заявок кафедр, облегчающая группировку, поиск и сортировку; высокоинформативные табличные представления данных (рис. 3).

Список заявок									
Поиск...		Q X		Все семестры		Все курсы		Фильтровать	
№	Дата	Инициатор НТ (02...)	Строка графика	Учебная группа	Статус Все	Документы	Комментарий к статусу		
1	15.05.2018	НТ Михайлова Инна Александровна	3 - 2015 Производственная, 11.06.2018 - 08.07.2018 14.03.01 Ядерная энергетика и теплофизика	ТФ-09-15	Завершена	ДП прил 4 от 18.05.2015 к дог. №65/2016 от 31.05.16 18.05.18 РД 503/2018 18.05.18 РД 504/2018 18.05.18			
10	24.07.2018	НТ Релецкая Ольга Павловна	3 - 2015 Производственная, 11.06.2018 - 08.07.2018 14.03.01 Ядерная энергетика и теплофизика	ТФ-14-15	Подготовка в ОЗиПФО				
2	20.09.2018	НТ Михайлова Инна Александровна	2 - 2017 Научно-производственная, 01.09.2018 - 30.12.2018 14.04.01 Ядерная энергетика и теплофизика , Магистр	ТФ-14м-17	Завершена	ДС 30 31.08.18 РД 812/2018 31.08.18			

Рис. 3. Фрагмент представления списка заявок

Fig. 3. Fragment of the List of Requests

Функции модуля подготовки печатных форм понятны из его названия. В ИС Практика утвержден перечень печатных форм, которые могут быть экспортированы в формате Microsoft Word. В перечень входят: график проведения практик, заявка на проведение практики (рис. 4), долгосрочный договор, дополнительное соглашение, разовый договор, сопроводительное письмо-направление. Для указанных документов генерируется текст, не требующий никаких поправок или изменений. Модуль закрывает все потребности пользователей по подготовке печатных версий документов, начиная с утверждения графика практик и заканчивая подготовкой сопроводительных писем во внешние организации.

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ МЭИ						
Заявка на проведение практики № 1 от 09.04.2018						
Институт	Институт электротехники					каф. ЭППЭ
Кафедра	Электроснабжения промышленных предприятий и электротехнологий					
Группа	ЭЛ-08-15					ЭЛ-08-15
Направление	13.03.02 Электроэнергетика и электротехника					3 курс
Курс	3					
Практика	Производственная					
Сроки практики	11.06.2018-08.07.2018					

№	Название предприятия	Должность и ФИО подписанта от предприятия	Юридический адрес	ФИО студентов	Руководитель практик и	№ договора
1	АО «Мосинжпроект»	Директор по управлению персоналом [Инициалы]	101990, г. Москва, Сверчков переулок, дом 4/1	1. [Инициалы] [Инициалы]	Никитин А.А.	РД №281/2018
2	ПАО «МОЭСК»	Заместитель генерального директора по работе с персоналом и административным вопросам [Инициалы]	115114, город Москва, проезд Павелецкий 2-й, дом 3, строение 2	1. [Инициалы] 2. [Инициалы] 3. [Инициалы] 4. [Инициалы] 5. [Инициалы]	Никитин А.А.	ДП №Прил. №8/1 8 к Дог. №1807 1-409 от 12.05.2014 г.

Рис. 4. Фрагмент печатной формы заявки

Fig. 4. Fragment of the Printed Request Form

Обычно совместно с переносом данных в электронный формат решаются задачи аналитики и построения отчетных форм. В ИС Практика для этих целей реализован **модуль построения статистической отчетности**. На данный момент разработаны следующие отчеты:

1) *сводка по заявкам на практику* позволяет отслеживать заявки с учетом заданных критериев, отображает количество заявок в различных статусах с учетом заданных критериев (рис. 5);

2) *сводка по видам баз практики* в разрезе кафедр предоставляет сведения о количестве студентов, проходящих практику в МЭИ и во внешних организациях;

3) *сводка по документам* позволяет найти документы, соответствующие заданным критериям, например за определенный период, по конкретной кафедре, по типу документа, по курсу учебной группы.

Сводка по заявкам на практику							
№	Кафедра	Учебная группа	Заявок всего	Подготовка на кафедре	Подготовка в ОЗиПФО	Обработана в ОЗиПФО	Завершена
1	АСУТП	ТФ-06-15	8	0	0	0	8
2	АСУТП	ТФ-07-15	6	0	0	0	6
3	АЭП	ЭЛ-01-15	3	0	0	0	3
4	АЭС	ТФ-11-15	2	0	0	0	2
5	АЭС	ТФ-12-15	3	0	0	0	3
6	ВМСС	А-07-15	1	0	0	0	1
7	ВМСС	А-08-15	1	0	0	0	1
8	ВМСС	А-12-15	1	0	0	0	1
9	ВТ	А-06-15	2	0	0	0	2
10	ГВИЭ	ИГ-01-15	4	0	0	0	4

Рис. 5. Фрагмент сводки по заявкам на практику

Fig. 5. Fragment of the Summary of Requests for Practice

Модуль отчетности поддерживает расширение – при появлении новых регламентированных (периодических) запросов разрабатываются отдельные отчетные формы. Ответ на нерегламентированный (разовый) запрос может подготовить человек, владеющий языком структурированных запросов T-SQL.

Разграничение прав в ИС Практика

При разработке любой многопользовательской системы обязательно встает вопрос о разграничении прав доступа. Разграничение прав основывается на идентификации, аутентификации и авторизации.

Под первым термином понимается присвоение элементам системы идентификаторов и сопоставление их с определенным ранее перечнем (например, присвоение реальному человеку учетной записи пользователя в ИС). Под вторым – проверка подлинности элементов системы (например, проверка предоставленных пользователем своих учетных данных). Под третьим – определение и проверка прав для выполнения конкретных действий в системе (например, контроль операций чтения и редактирования пользователей и их прав в ИС).

За идентификацию и аутентификацию в ИС Практика отвечает функционирующая в НИУ МЭИ служба каталогов Active Directory. Каждый пользователь имеет пару логин-пароль, которую должен использовать для работы с информационными сервисами НИУ МЭИ.

Все необходимые процедуры авторизации ИС Практика обеспечивает самостоятельно. В системе предусмотрено множество разрешений, в том числе параметризуемых, из которых складываются конкретные группы (роли) пользователей. Параметризуемость означает, что для разрешения можно задать определенную область действия (табл. 1).

Представленные разрешения могут быть назначены напрямую отдельному пользователю или группе пользователей и уже через нее «переданы» конкретным пользователям. Из указанных в табл. 1 разрешений «складываются» основные группы пользователей ИС Практика (табл. 2).

Разрешения и их возможные области их действия

Таблица 1

Permissions and Their Possible Scope

Table 1

Разрешение	Область действия
P1. Просмотр графика проведения практики	Все институты
P2. Редактирование графика проведения практики	Конкретный институт
P3. Просмотр предприятия	Все подразделения (МЭИ в целом и кафедры)
P4. Редактирование предприятия	Конкретное подразделение
P5. Просмотр заявки в статусе	Все статусы
P6. Редактирование заявки в статусе	Конкретный статус
P7. Просмотр заявки подразделения	Все подразделения
P8. Редактирование заявки подразделения	Конкретное подразделение

Таблица 2

Группы пользователей в ИС Практика

Table 2

User Groups in IS Practice

Группа (роль) ИС Практика	Разрешения
Сотрудники дирекции института «И1»	<ul style="list-style-type: none"> • P1, P2 для «И1» • P5 для «Завершена» • P7 для кафедр «К1», «К2», ..., «Kn», входящих в состав «И1»
Ответственные от кафедры «К1», входящий в состав института «И1»	<ul style="list-style-type: none"> • P1 для «И1» • P3, P4, P7, P8 для «К1» • P3 для «МЭИ» • P5 для «Все статусы» • P6 для «Подготовка на кафедре»
Сотрудники ОЗ и ПФО	<ul style="list-style-type: none"> • P1 для «Все институты» • P3, P4, P7, P8 для «Все подразделения» • P5 для «Подготовка в ОЗ и ПФО», «Обработана ОЗ и ПФО», «Завершена» • P6 для «Подготовка в ОЗ и ПФО», «Обработана ОЗ и ПФО»

В ИС Практика присутствуют и другие группы пользователей, которые используют как представленные, так и опущенные в данном тексте разрешения.

Реализованный механизм разграничения прав, основанный на разрешениях и области их действия, позволяет определять новые группы пользователей и не требует навыков программирования.

Другой аспект, связанный с безопасностью программных систем и реализованный в ИС Практика, – это протоколирование любых изменений данных и ведение соответствующего журнала аудита. Дополнительно ИС Практика фиксирует дату последнего входа в систему для каждого пользователя. На конец 2018 г. в журнале аудита зафиксировано более 200 000 фактов изменений.

Технические характеристики системы

Современные программные системы обычно работают согласно клиент-серверной архитектуре и предоставляют своим пользователям веб-интерфейс. Указанный подход имеет множество преимуществ, среди которых снижение аппаратных и программных требований к рабочим машинам конечных пользователей; повышение доступности – с системой можно работать из любой точки мира, имеющей доступ к глобальной сети Интернет, и с устройств различных видов и размеров; упрощение процедуры выпуска новых версий программного продукта.

ИС Практика реализована по клиент-серверной архитектуре на стеке технологий компании Microsoft. Сервер работает под управлением операционной системы Microsoft Windows Server 2012 R2 (64-разрядная версия) и имеет следующие аппаратные характеристики: процессор – Intel Xeon X5675 3.07GHz, объем оперативной памяти – 8 Гб, жесткий диск – 60 Гб. Помимо ИС Практика, указанный сервер обслуживает и другие информационные системы, функционирующие в НИУ МЭИ. За управление базой данных отвечает SQL Server 2012 SP 2, за обработку веб-запросов – Internet Information Services (IIS) 8.5. Задействованные технологии достаточно давно и успешно используются при решении аналогичных задач, они сопровождаются и активно развиваются компанией Microsoft и по праву занимают немалую долю рынка современных ИТ-решений.

При разработке ИС Практика использован фреймворк ASP.NET MVC, реализующий шаблон разделения исходного кода «Модель-Представление-Контроллер» (или «Model-View-Controller»). Для доступа к данным используется объектно-реляционное отображение с помощью технологии ADO.NET Entity Framework. Адаптивный пользовательский интерфейс строится на базе Bootstrap и использует входящие в его состав HTML и CSS шаблоны. Основными компонентами для реализации интерактивности на веб-страницах ИС Практика являются: язык JavaScript, библиотека JQuery и ряд плагинов Bootstrap и JQuery.

При формировании документов в формате Microsoft Word (DOCX) задействован набор средств разработки Open XML SDK, позволяющий работать с электронными документами в серии форматов Office Open XML, включающей DOCX. Для разработки сводных отчетов применяется служба SQL Server Reporting Services, являющаяся дополнением к системе управления базами данных SQL Server.

Наконец, собрать воедино все перечисленные технологии и многие другие позволяет среда разработки Microsoft Visual Studio (MSVS). MSVS, без сомнения, один из самых передовых, современных и мощных инструментов для разработки приложений любой сложности.

В отличие от многообразия и сложности технологий разработки ИС Практика для пользователя системы все значительно проще. Работать с ИС можно через любой современный веб-браузер (Internet Explorer, EDGE, Google Chrome, Firefox и др.).

Опыт внедрения

ИС практика запускалась поэтапно начиная с марта 2018 г. На первом этапе ответственные от дирекций институтов внесли и утвердили в системе графики прохождения практик. На втором этапе ответственным от кафедр и сотрудникам ОЗ и ПФО был открыт доступ к справочнику организаций-партнеров. Третьим этапом был запущен модуль подготовки и согласования заявок на проведение практик. Запуск этапов сопровождался публикацией на веб-портале МЭИ информационных писем. Они разъясняли регламент работы с новой системой и обязывали ответственных лиц работать в ИС Практика.

В рамках подготовки пользователей к работе в новой системе отделом разработки и внедрения информационных систем была подготовлена инструкция, рассматривающая решение задач различных категорий пользователей. Помимо этого, было проведено обучение ряда пользователей и предложены очные консультации по отдельным вопросам.

Большинство пользователей позитивно отнеслись к запуску ИС Практика. Система не была заменой уже функционирующей, а являлась принципиально новым подходом к автоматизации ряда рутинных задач. Возможно, именно поэтому при внедрении практически отсутствовало «сопротивление» со стороны пользователей.

В отдел разработки поступало немало предложений по улучшению функционала системы. В основном пожелания относились к повышению информативности реализованных веб-форм. Например, добавить в таблицу заявок отображение связанных документов (столбец «Документы» на рис. 3). Такие доработки оперативно вносились в функционал ИС Практика. Используемые при разработке проекта веб-технологии позволили без труда обновлять функционал, доступный конечным пользователям.

Замечаний к реализованному в системе бизнес-процессу направления студентов на практику не было. Конечно, этот процесс имел уже достаточно устоявшийся характер, но все-таки требовал некоторой формализации для воплощения в программной системе.

По результатам работы ИС Практика в двух семестрах – весеннем 2017/2018 и осеннем 2018/2019 – можно отметить следующие факты:

- в работе задействованы сотрудники ОЗ и ПФО, 11 институтов и почти 50 кафедр;
- к системе обращается более 110 пользователей;
- заведено более 650 организаций-партнеров;
- сформировано более 3 000 направлений студентов на практику;
- обработано почти 400 заявок кафедр;
- подготовлено более 900 документов.

В ходе запуска проекта система получила дополнительный функционал для сбора сведений от выпускающих кафедр о трудоустройстве студентов и была интегрирована с другими системами, функционирующими в МЭИ (подготовка приказов на проведение практики в системе учета контингента студентов – ИС Студент). Обработка данных в электронном формате позволила повысить скорость и точность подготовки статистических сводок.

Заключение

В современном мире информатизация стала неотъемлемой частью любого процесса, связанного с обработкой данных, а сами данные – ценным ресурсом их владельца. ИС Практика частично решает задачи, связанные с документооборотом, возникающим внутри МЭИ и между МЭИ и организациями-партнерами. У решения видны определенные перспективы и варианты развития – охват новых участников процесса (студенты, руководители практики, представители внешних организаций и др.) и автоматизация подготовки прочих документов (задание на практику, отчет по практике и др.).

Перед разработчиками системы не стояла задача охвата всевозможных процессов в сфере образования, связанных с подготовкой и согласованием документов. С одной стороны, это можно считать недостатком – область применения достаточно ограничена, но с другой – преимуществом. Специфика присутствует практически всегда, и тяжеловесные продукты широкого профиля зачастую не подходят для решения узконаправленных задач, тогда выбор падает на небольшие специализированные системы. Хотя они и не обладают множеством функций (в том числе и слабо востребованных) универсальных продуктов, но превосходят их ориентированностью на конкретную задачу и нередко позволяют решить ее наилучшим образом. ИС Практика следует относить к классу именно таких специализированных систем.

Современные образовательные учреждения РФ обязаны проводить практику для обучаемых студентов. Многие из них достаточно глубоко изучают и анализируют этот процесс и его особенности [1–4]. Их цель заключается в улучшении результатов проведения практики в целом и в совершенствовании процессов распределения студентов по базам практик и подготовки сопроводительной документации в частности [12; 13]. Представленный в статье опыт будет, несомненно, им интересен, а описанные технические решения могут быть адап-

тированы с учетом специфики конкретного образовательного учреждения и применены для автоматизации указанных выше процессов.

Список литературы / References

1. **Чиняков О. Е.** Роль учебной и производственной практик в формировании профессиональных компетенций и трудоустройстве выпускников // Мир науки и образования. 2017. № 3 (11). С. 1–9.
Chinyakov O. E. The role of educational and industrial practices in formation of professional competencies and employment of graduates. *World of science and education*, 2017, no. 3 (11), p. 1–9. (in Russ.)
2. **Архипова М. Н.** Роль учебной и производственной практик в процессе профессионального становления студентов // Электронный научно-методический журнал Омского ГАУ. 2017. № 1 (8). С. 1–4.
Arkhipova M. N. Role of Educational and Industrial Practices in the Course of Professional Formation of students. *Electronic science and methodology journal of the Omsk State Agrarian University*, 2017, no. 1 (8), p. 1–4. (in Russ.)
3. **Чекрышкина Л. А., Слепова Н. В., Дозморова Н. В., Березина Е. С., Турышев А. Ю.** Производственная практика и ее значимость в освоении профессиональных компетенций // Медицинская наука и образование Урала. 2017. Т. 18. № 4 (92). С. 166–169.
Chekryshkina L. A., Slepova N. V., Dozmorova N. V., Berezina E. S., Turyshev A. Yu. Proizvodstvennaya praktika i ee znachimost' v osvoenii professional'nyh kompetencij [Internship and its importance in development of professional competence]. *Medical science and education of Ural*, 2017, vol. 18, no. 4 (92), p. 166–169. (in Russ.)
4. **Колтунов В. А., Ловыгина М. Б.** Повышение качества подготовки специалистов-машиностроителей на основе целевой организации производственных практик // Современные проблемы науки и образования. 2015. № 5. URL: <http://www.science-education.ru/ru/article/view?id=22577>
Koltunov V. A., Lovygina M. B. The quality improvement training of industrial engineers based on the aimed practical training organization. *Modern problems of science and education*, 2015, no. 5. URL: <http://www.science-education.ru/ru/article/view?id=22577> (in Russ.)
5. **Petrov S. A., Afanasyeva A. O.** Information Services for Structural Units Provided in a Personal Account Within Moscow Power Engineering Institute. In: 2018 IV International Conference on Information Technologies in Engineering Education (Inforino). Moscow, 2018, p. 1–4. DOI 10.1109/INFORINO.2018.8581750
6. **Petrov S. A., Krepkov I. M., Ovsyannikova M. R.** Information System for Recording Student Personal Files at National Research University “Moscow Power Engineering Institute”. In: 2018 IV International Conference on Information Technologies in Engineering Education (Inforino). Moscow, 2018, p. 1–4. DOI 10.1109/INFORINO.2018.8581817
7. **Коломиец А. Ю.** Проблемы и перспективы развития СЭД // Экономика и управление в XXI веке: тенденции развития. 2014. № 19-2. С. 151–155.
Kolomiets A. Yu. Problemy i perspektivy razvitiya SED [Problems and prospects of development of EDMS]. *Economics and Management in the 21st Century: Development Trends*, 2014, no. 19-2, p. 151–155. (in Russ.)
8. **Лежнин С. А., Петров В. Ю.** Проблемы использования систем электронного делопроизводства // Фундаментальные исследования. 2016. № 9-3. С. 615–621.
Lezhnin S. A., Petrov V. Yu. The problem of the use of electronic record keeping. *Basic Research*, 2016, no. 9-3, p. 615–621. (in Russ.)

9. **Девайкина Т. А., Багаева А. П.** Проблемы внедрения СЭД и их решения // Актуальные проблемы авиации и космонавтики. 2015. Т. 1, № 11. С. 546–548.
Devaykina T. A., Bagaeva A. P. Implementation challenges and solutions EDS. *Actual problems of aviation and cosmonautics*, 2015, vol. 1, no. 11, p. 546–548. (in Russ.)
10. **Баталин К. В., Мамеев Н. С., Попова К. Ю., Рыжаков И. Д., Яхьяева Г. Э.** Программная система управления образовательным процессом ИТОС // Вестник НГУ. Серия: Информационные технологии. 2018. Т. 16, № 4. С. 20–30. DOI 10.25205/1818-7900-2018-16-4-20-30
Batalin K. V., Mameev N. S., Popova K. Yu., Ryzhakov I. D., Yakhyaeva G. E. Software System for Management of Educational Process ITOS. *Vestnik NSU. Series: Information Technologies*, 2018, vol. 16, no. 4, p. 20–30. (in Russ.) DOI 10.25205/1818-7900-2018-16-4-20-30
11. **Чичиль В. О., Королева И. Ю.** Разработка системы автоматизации документооборота выпускающей кафедры вуза // Изв. Волгоград. гос. техн. ун-та. 2016. № 11 (190). С. 116–121.
Chichil V. O., Koroleva I. Yu. Development of Workflow Graduating Departments University. *News of Volgograd State Technical University*, 2016, no. 11 (190), p. 116–121. (in Russ.)
12. **Шилович А. В.** Совершенствование организации производственных практик студентов // Вестник Белорус. гос. ун-та транспорта: Наука и транспорт. 2016. № 2 (33). С. 210–211.
Shilovich A. V. The improvements of the students practice training ability. *Bulletin of the Belarusian State University of Transport: Science and Transport*, 2016, no. 2 (33), p. 210–211. (in Russ.)
13. **Мягков А. Ю., Григорьева М. В., Журавлева И. В., Журавлева С. Л.** Производственная практика глазами студентов технического вуза (по материалам социологического исследования) // Образование и наука. 2015. № 4 (123). С. 100–113.
Myagkov A. Yu., Grigorieva M. V., Zhuravleva I. V., Zhuravleva S. L. Internship through the eyes of technical university students (based on sociological research). *Education and Science*, 2015, no. 4 (123), p. 100–113. (in Russ.)

*Материал поступил в редколлегию
Received
10.01.2019*

Сведения об авторах / Information about the Authors

Петров Сергей Андреевич, кандидат технических наук, ведущий программист, доцент кафедры ФГБОУ ВО «Национальный исследовательский университет МЭИ» (Красноказарменная ул., 14, Москва, 111250, Россия)

Sergey A. Petrov, Candidate of Technical Sciences, Leading Programmer, Associate Professor National Research University “Moscow Power Engineering Institute” (14 Krasnokazarmennaya Str., Moscow, 111250, Russian Federation)

petrovs@mpei.ru

УДК 004.89:004.4
DOI 10.25205/1818-7900-2019-17-2-138-152

Платформа комплексирования и тестирования средств трансформации естественно-языковых запросов в SPARQL-запросы

И. С. Постаногов, И. А. Турова

*Пермский государственный национальный исследовательский университет
Пермь, Россия*

Аннотация

Статья посвящена вопросу поддержки процесса создания средства трансформации естественно-языковых (ЕЯ) запросов в SPARQL-запросы (далее – средство трансформации).

Во введении описаны актуальность задачи понимания ЕЯ-запросов информационными системами, а также преимущества использования онтологий как средства представления знаний для решения данной задачи. Обозначена роль средств трансформации в системах с ЕЯ-запросами к реляционным базам данных. На основании анализа проблем, связанных с комплексированием и тестированием существующих средств трансформации, а также с целью поддержки создания и тестирования собственных модулей трансформации предложена концепция программной платформы, упрощающей эти задачи. Архитектура платформы удовлетворяет требованиям простоты подключения сторонних средств трансформации, переиспользования отдельных модулей, а также интеграции результирующих средств трансформации в другие системы, в том числе в системы тестирования.

Строительными блоками создаваемых систем трансформации служат отдельные модули трансформации, упакованные в Docker-контейнеры. Программный доступ к каждому модулю осуществляется при помощи gRPC. Загруженные в платформу модули могут быть выстроены в конвейер трансформации автоматически или вручную при помощи встроенного стороннего редактора диаграмм потоков данных SciVi. Совместимость отдельных модулей контролируется при помощи автоматического анализа программных интерфейсов. На основании спецификации конвейера в формате, поддерживаемом SciVi, а также gRPC-спецификаций отдельных модулей комплексировается единое многоконтейнерное приложение, которое может быть интегрировано в другие системы, а также протестировано на пополняемом наборе тестов. Ожидаемые и фактические результаты трансформации запроса доступны для просмотра в графическом виде в разработанном ранее средстве визуализации.

Ключевые слова

интеллектуальные информационные системы, онтология, естественно-языковой интерфейс, SPARQL, доступ к данным на базе онтологий, Docker, gRPC, Protocol Buffers, диаграмма потока данных

Для цитирования

Постаногов И. С., Турова И. А. Платформа комплексирования и тестирования средств трансформации естественно-языковых запросов в SPARQL-запросы // Вестник НГУ. Серия: Информационные технологии. 2019. Т. 17, № 2. С. 138–152. DOI 10.25205/1818-7900-2019-17-2-138-152

Platform for Integrating and Testing Tools which Transform Natural Language Queries into SPARQL Queries

I. S. Postanogov, I. A. Turova

*Perm State University
Perm, Russian Federation*

Abstract

In the paper we discuss how to support the process of creating tools which transform natural language (NL) queries into SPARQL queries (hereinafter referred to as a transformation tool).

© И. С. Постаногов, И. А. Турова, 2019

In the introduction, we describe the relevance of the task of understanding natural language queries by information systems, as well as the advantages of using ontologies as a means of representing knowledge for solving this problem. This ontology-based data access approach can be also used in systems which provide natural language interface to databases. Based on the analysis of problems related to the integration and testing of existing transformation tools, as well as to support the creation and testing own transformation modules, the concept of a software platform that simplifies these tasks is proposed. The platform architecture satisfies the requirements for ease of connecting third-party transformation tools, reusing individual modules, as well as integrating the resulting transformation tools into other systems, including testing systems.

The building blocks of the created transformation systems are the individual transformation modules packaged in Docker containers. Program access to each module is carried out using gRPC. Modules loaded into the platform can be built into the transformation pipeline automatically or manually using the built-in third party SciVi data flow diagram editor. Compatibility of individual modules is controlled by automatic analysis of application programming interfaces. The resulting pipeline is combined according to specified data flow into a single multi-container application that can be integrated into other systems, as well as tested on extendable test suites. The expected and actual results of the query transformation are available for viewing in graphical form in the visualization tool developed earlier.

Keywords

intelligent information systems, ontology, natural language interface, SPARQL, ontology-based data access, Docker, gRPC, Protocol Buffers, data flow diagram

For citation

Postanogov I. S., Turova I. A. Platform for Integrating and Testing Tools which Transform Natural Language Queries into SPARQL Queries. *Vestnik NSU. Series: Information Technologies*, 2019, vol. 17, no. 2, p. 138–152. (in Russ.) DOI 10.25205/1818-7900-2019-17-2-138-152

Введение

В современном мире все чаще используются естественно-языковые (ЕЯ) интерфейсы для взаимодействия с различными информационными системами. Примером этого может служить разговор на естественном языке с роботом-промоутером на выставке или в торговом центре, где любой человек вне зависимости от знаний в области информационных технологий может задать вопрос о товаре, услуге, получить ответ и, возможно, сразу же оформить новый заказ. Те же подходы к человеко-машинному взаимодействию активно применяются и в виртуальных ассистентах, работающих не только в мобильных устройствах (Яндекс Алиса, Google Ассистент, Apple Siri), но и в умных колонках, являющихся центральными узлами систем домашней автоматизации (Яндекс Станция, Google Home, Apple HomePod). Взаимодействие с такими системами может быть не только устное, но и письменное при помощи приложений обмена мгновенными сообщениями.

Для внутренних нужд компаний корпоративного сектора естественно-языковой интерфейс используется при общении с чат-ботами в корпоративных мессенджерах. Традиционным способом реализации таких чат-ботов является сопоставление текста запроса с образцом и выдача результата, полученного предопределенным для сопоставленного шаблона способом. Для предоставления возможности задания более сложных, нерегламентированных запросов следует использовать знания, которые бы определяли процесс вывода ответа на запрос.

В качестве средства представления знаний можно использовать онтологии. Онтология – это точная спецификация концептуализации [1], ключевыми элементами которой являются понятия и связи между ними. К настоящему моменту существуют стандарт OWL для записи онтологий в машиночитаемом виде, средства программной обработки и вывода на онтологиях. Активно развивается исследовательское направление организации доступа к данным, основанного на онтологиях (Ontology-based Data Access, OBDA) [2]. Существенная часть исследований в области OBDA посвящена предоставлению виртуального доступа к данным, хранящимся в реляционных базах данных, т. е. трансформации запросов на языке SPARQL в запросы на языке SQL.

Ранее в работе [3] была предложена концепция обогащения унаследованных информационных систем сервисом запросов на естественном языке *Reply*. Основными идеями концепции были двухэтапная трансформация ЕЯ-запроса в SQL-запрос с использованием промежуточного представления в виде SPARQL-запроса, а также автоматизация конфигурирования как используемых OBDA-фреймворков, так и всей системы в целом.

Reply разрабатывался как сервис для трансформации традиционных информационных систем в интеллектуальные системы с запросами на естественном языке безотносительно предметной области исходной информационной системы, а также к целевому языку запросов (русский, английский...). В то же время для получения более качественного и полного ответа на запросы пользователей в каждом конкретном случае следует предусмотреть возможность выбора оптимального средства трансформации ЕЯ-запроса в SPARQL-запрос либо сборки нового средства на основе существующих модулей. В данной статье описана разработанная платформа комплексирования и тестирования средств трансформации ЕЯ-запросов в SPARQL-запросы для поддержки этого процесса. Под комплексированием в работе понимается процесс объединения системных элементов для производства полной системы¹.

Платформы создания средств трансформации ЕЯ-запросов в SPARQL-запросы

В настоящем разделе описываются экосистема, современные подходы и существующие реализации средств трансформации.

Обзор архитектур средств трансформации

Идея, положенная в основу вопросно-ответной системы AskNow [4], заключается в использовании в качестве промежуточного представления особой синтаксической структуры, названной авторами нормализованной структурой запроса (Normalized Query Structure – NQS). На первом этапе исходный ЕЯ-запрос должен быть преобразован в соответствующую ему NQS. Для этого с помощью модуля Query Processor запрос разбивается на отдельные токены, которые впоследствии помечаются тегами частей речи (part-of-speech tag, POS-тег). Модуль Auxiliary Relation Handler выделяет вспомогательное отношение – отношение, которое содержит лексические формы выражений *is*, *is kind of*, *much*, *might be*, *does*. Модуль Token Merger объединяет токены в сегменты исходя из их POS-тегов и положения в запросе. Выделенные сегменты проходят через модуль NQS Instance Generator, который строит итоговую NQS по определенным правилам. После этого построение SPARQL-запроса по нормализованной структуре запроса происходит также в несколько этапов. Модуль NQS Parser определяет тип запроса по его NQS и сопоставляет с сегментами запроса наиболее подходящие элементы онтологии (DBpedia) с помощью сервиса DBpedia Spotlight. Для решения проблем с отображением сегментов на онтологию используется электронный тезаурус WordNet. В нем осуществляется поиск синонимов для неопределенных сегментов. Используя результаты предыдущих этапов, модуль SPARQL Generator преобразует NQS в итоговый SPARQL-запрос.

Работа вопросно-ответной системы Sina [5] разделяется на четыре этапа. На первом этапе производится предобработка запроса: выделяются токены в виде отдельных ключевых слов, удаляются стоп-слова, выполняется лемматизация. На втором этапе ключевые слова различным образом объединяются в сегменты (N-граммы), каждый сегмент проверяется на возможность соответствия некоторому элементу онтологии. Далее на третьем этапе для всех подходящих сегментов ведется поиск соответствующего ресурса в онтологии по свойству *rdfs:label*. На этапе устранения неоднозначностей (disambiguation) выбирается подграф

¹ ГОСТ Р ИСО/МЭК 12207-2010. Информационная технология. Системная и программная инженерия. Процессы жизненного цикла программных средств. М.: Стандартинформ, 2011. 106 с.

из базы знаний, наиболее соответствующий исходному запросу. Далее по полученному подграфу строится SPARQL-запрос.

Вопросно-ответная система Xser [6] представляет собой конвейер из двух этапов: выделение из запроса возможных семантических структур – экземпляров, классов, отношений, и отображение выделенных структур на элементы онтологии. На первом этапе происходит разметка исходного запроса с использованием меток «начало фразы-отношения», «продолжение фразы-экземпляра» и т. п. По полученной разметке строится ациклический ориентированный граф зависимостей фраз. На втором этапе полученный граф преобразуется в запрос к онтологии. Так как с каждой фразой, являющейся элементом графа зависимостей, может быть сопоставлено более одного элемента онтологии, все возможные отображения ранжируются, и выбирается наилучшее. В качестве метрики выступает произведение нормированных результатов функции схожести для каждого слова, полученных с помощью Freebase Search API (для запросов к базе знаний Freebase).

Хотя в описанных системах используются разные методы и средства для преобразования ЕЯ-запросов в SPARQL-запросы, в процессе трансформации можно выделить одинаковые этапы [7]. Этап анализа естественно-языкового запроса может включать определение типа запроса, выделение именованных сущностей, определение зависимостей между частями предложения, построение синтаксического дерева запроса и т. д. На этапе отображения сегментов запроса на элементы онтологии происходит поиск подходящих определенной части запроса экземпляров, классов и отношений онтологии. Так как из-за наличия неоднозначностей в естественном языке (многозначность, омонимия и т. п.) с одним сегментом запроса могли быть сопоставлены несколько ресурсов онтологии, в каждом случае необходимо определить наиболее подходящий к контексту ресурс. Эти проблемы решаются на этапе устранения неоднозначностей. На этапе построения SPARQL-запроса определяется необходимый тип запроса (SELECT, ASK), строится основной графовый шаблон запроса.

Обзор существующих платформ создания средств трансформации

Возможность выделения отдельных этапов при решении задач обработки естественного языка привела к появлению универсальных платформ по анализу и преобразованию текстов. Примерами таких платформ служат NLTK [8] и TAISim [9]. Кроме того, существуют платформы, позволяющие конструировать программные средства, решающие более частные задачи данной области знаний, в том числе задачу трансформации ЕЯ-запроса в SPARQL-запрос.

Qanary [10] – система для комплексирования средств трансформации ЕЯ-запросов из отдельных компонентов. Согласно методологии разработки вопросно-ответных систем, предложенной авторами Qanary, компоненты системы должны взаимодействовать посредством аннотаций, определенных в словаре Qanary, построенном на основе модели аннотаций WADM. Qanary Pipeline – центральный компонент системы – представляет собой сервер, к которому подключаются остальные компоненты посредством RESTful интерфейса с помощью фреймворка Spring. Qanary Pipeline имеет веб-интерфейс для выбора компонентов для текущего конвейера, определения их порядка и запуска конвейера с некоторым исходным запросом. Пользовательские компоненты также должны быть реализованы в виде сервисов. Разработчики Qanary предоставляют шаблон Maven-приложения на Java для создания компонента. Шаблон включает код для запуска Spring-сервиса и пустую реализацию Qanary-компонента. При запуске сервиса отправляется запрос на регистрацию компонента в системе. При вызове компонента из Qanary Pipeline он получает входные данные и сохраняет результат своей работы с помощью извлечения и добавления аннотаций в служебную онтологию посредством выполнения SPARQL-запросов. При необходимости использования других технологий разработчик может вызвать свою реализацию из Java-компонента. Собранная с помощью Qanary-фреймворка система может быть интегрирована в другие системы как Spring-сервис.

В работе [11] представлен фреймворк QALL-ME, предоставляющий переиспользуемую архитектуру для создания мультязычных вопросно-ответных систем, учитывающих пространственно-временной контекст запроса. Авторы декомпозируют процесс трансформации на этапы и сопоставляют с каждым этапом отдельный тип компонентов, например аннотаторы именованных сущностей для различных языков. В QALL-ME компоненты разделяются на три типа: языкозависимые, пространственно-зависимые и общесистемные компоненты, реализация которых одинакова для всех языков и местоположений. Фреймворк поддерживает сервис-ориентированную архитектуру: компоненты системы реализуются в виде веб-сервисов, а их программный интерфейс описывается на языке WSDL (Web Service Description Language). QALL-ME не является средством комплексирования средств трансформации. Цель создания фреймворка – упрощение разработки мультязычных вопросно-ответных систем. В частности, реализация содержит методы для вызова компонента определенного типа, подходящего для языковых и пространственных характеристик запроса. Архитектура средства трансформации должна быть определена заранее, и последовательный вызов компонент должен быть запрограммирован. QALL-ME позволяет расширять возможности системы посредством написания компонентов для других языков и местоположений и регистрации пользовательских компонентов в QAPanner (ядре системы) с помощью отправки запросов. Так как компоненты являются веб-сервисами, они могут быть реализованы с использованием любых языков программирования и технологий.

openQA [12] – расширяемый вопросно-ответный фреймворк. Он нацелен на улучшение качества ответов на естественно-языковые запросы путем комбинирования результатов работы нескольких систем. Обработка пользовательского запроса происходит в четыре этапа: интерпретация запроса в запрос на формальном языке, получение результатов запроса, синтез и представление результатов. На первом этапе ЕЯ-запрос обрабатывается всеми активными модулями интерпретации, например, в качестве таких модулей в openQA присутствуют реализации вопросно-ответных систем Sina и TBSL. Далее из источника извлекаются результаты выполнения полученных запросов на формальных языках (SPARQL, SQL). На третьем этапе результаты ранжируются по релевантности, на четвертом – представляются пользователю. К openQA можно подключить пользовательские модули для каждого этапа, однако они должны быть реализованы в виде Java-плагинов. Фреймворк не поддерживает возможность комплексирования модуля интерпретации из нескольких компонент. Собранный сервис представляет собой сервис и доступна через RESTful интерфейс.

Сравнение существующих платформ создания средств трансформации

Основными критериями для сравнения существующих платформ были выбраны:

- 1) предъявляемые требования к степени связывания как непосредственно между модулями, так и между модулями и платформой;
- 2) наличие средств поддержки процесса развертывания создаваемых систем.

Основной принцип *слабого* связывания состоит в уменьшении числа допущений, которые делают друг о друге взаимодействующие стороны [13]. Возможность слабого связывания трактуется как преимущество платформы, поскольку упрощается процесс подмены реализации, тем самым ускоряется поиск наилучшей комбинации модулей средства трансформации.

Среди рассмотренных выше существующих платформ построения средств трансформации системой с сильным связыванием является система openQA, требующая оформления отдельных модулей системы в виде Java-плагинов, тем самым ограничивающая возможности для использования прочих языков программирования. Даже при наличии возможности написания программной обертки на языке Java, разработчику требуется самостоятельно организовывать коммуникацию между программами на языке Java и языке программирования ядра модуля трансформации.

Qanary позволяет создавать системы на принципах слабого связывания, однако коммуникация между отдельными модулями осуществляется при помощи модификации глобального состояния – множества аннотаций онтологии. Для модулей отсутствует формальное описание того, какие именно категории аннотаций будут добавлены, удалены, изменены в процессе работы. Корректность создаваемого конвейера в случае зависимости одного модуля от результатов работы другого модуля в данной платформе может быть проверена только на этапе исполнения.

QALL-ME также позволяет создавать системы на принципах слабого связывания, причем API каждого из сервисов описывается на языке WSDL. Наличие формального описания позволяет генерировать клиентские и / или серверные части взаимодействующих компонентов. Недостатком данного подхода является выбор технологий веб-сервисов для взаимодействия компонент. У данной технологии есть проблемы с производительностью, объясняемые использованием XML и HTTP 1.1 для передачи сообщений. Кроме того, в QALL-ME существенно ограничены типы модулей, для которых можно предоставить собственную реализацию.

Рассмотренные платформы не ориентированы на поддержку процесса развертывания итогового средства трансформации с учетом требований к среде исполнения отдельных модулей, а также с учетом потенциальной противоречивости этих требований. Например, несколько модулей могут зависеть от разных версий одной программной библиотеки, которая должна располагаться по одному и тому же предопределенному системному пути. В таком случае без дополнительного вмешательства на этапе сборки или запуска средства трансформации использование комбинации этих модулей будет невозможно.

Тестирование средств трансформации

Для сравнения средств трансформации необходимо производить их тестирование. На данный момент существуют тестовые наборы запросов и соответствующих эталонных SPARQL-запросов к различным онтологиям. Большинство тестовых наборов [14; 15] содержат запросы к обширным базам знаний общего назначения, таким как DBpedia, Freebase, YAGO. Например, с 2011 г. в рамках серии соревнований QALD [16] для оценки вопросно-ответных систем организаторами было разработано более десяти тестовых наборов запросов к DBpedia, среди которых помимо англоязычных наборов присутствует несколько мультиязычных. Тестовые наборы составляются вручную группой людей и включают в себя запросы различных типов и сложности, поэтому с их помощью можно провести качественную оценку вопросно-ответных систем.

Наличие в платформе комплексирования средств трансформации возможности проверки собранных систем на существующих тестовых наборах запросов, а также наглядного просмотра результатов таких тестов существенно упростит процесс разработки.

Архитектура предлагаемой платформы

На основании анализа недостатков описанных ранее программных средств было принято решение о разработке собственной платформы комплексирования и тестирования средств трансформации.

Основные требования к платформе:

- 1) минимальность предъявляемых ограничений на технологии, используемые модулями трансформации;
- 2) проверка совместимости интерфейсов модулей на этапе комплексирования;
- 3) эффективность технологий межмодульной коммуникации;
- 4) возможность проведения тестирования средств трансформации и визуализации его результатов;
- 5) простота встраивания системы в сторонние интеллектуальные информационные системы.

Для удовлетворения этих требований в качестве ключевых технологий были выбраны программная платформа контейнеризации Docker² и система удаленного вызова процедур gRPC³.

Использование технологии контейнеризации позволяет разработчику модуля однократно настроить изолированное воспроизводимое окружение исполнения программного кода. Кроме того, существующие средства согласованного запуска нескольких контейнеров позволяют автоматизировать процесс сборки и запуска созданного при помощи платформы средства трансформации, в том числе как часть сторонних систем.

Система удаленного вызова процедур gRPC требует формального описания сервисов на языке описания интерфейсов Protocol Buffers⁴. Использование этого описания позволяет автоматически проверять совместимость интерфейсов сервисов, а также автоматически генерировать программный код по комплексированию отдельных сервисов. Требование реализации gRPC-интерфейса существенно не ограничивает разработчика модулей трансформации. Поддержка gRPC реализована для всех наиболее популярных современных языков программирования, а значительная часть программного кода для сетевого взаимодействия может быть сгенерирована автоматически. Еще одним преимуществом gRPC является эффективность при передаче сообщений за счет жесткости протокола, использования бинарной сериализации и HTTP 2.0 в качестве транспорта.

На этапе проектирования были выделены следующие сценарии использования платформы:

- 1) управление модулями трансформации;
- 2) комплексирование средств трансформации;
- 3) проведение тестирования средств трансформации;
- 4) анализ результатов тестирования средств трансформации;
- 5) выгрузка созданного средства трансформации.

В сценарии *Управление модулями трансформации* основной операцией является загрузка новых модулей трансформации в платформу. Для этого пользователь заблаговременно загружает в приватный Docker-репозиторий платформы Docker-образ программного модуля, предоставляющий после запуска gRPC-сервис. В пользовательском интерфейсе платформы пользователь добавляет новый модуль, указывая его имя, категорию, а также формальную спецификацию gRPC-сервиса при помощи proto-файла. Впоследствии пользователь может отредактировать эти параметры.

В сценарии *Комплексирование средств трансформации* пользователь выстраивает конвейер трансформации из загруженных в платформу модулей. Доступные модули отображаются в палитре модулей редактора диаграмм потоков данных системы научной визуализации SciVi [17]. Для пополнения этой палитры разработанная платформа на основании загруженного формального описания интерфейса каждого модуля на языке описания сервисов gRPC Protobuf автоматически формирует правила конфигурации в онтологическом формате, используемом редактором. В редакторе пользователь может перетаскать необходимые модули из палитры в область редактирования диаграммы, после чего соединить соответствующие сокеты (входы и выходы) узлов вершин (рис. 1). Редактор позволяет отфильтровывать только совместимые с выбранным модули трансформации, а также проверять корректность создаваемых дуг. Построенную диаграмму трансформации можно сохранить в платформе с целью последующего редактирования.

² URL: <https://www.docker.com/>

³ URL: <https://grpc.io/>

⁴ URL: <https://developers.google.com/protocol-buffers/>

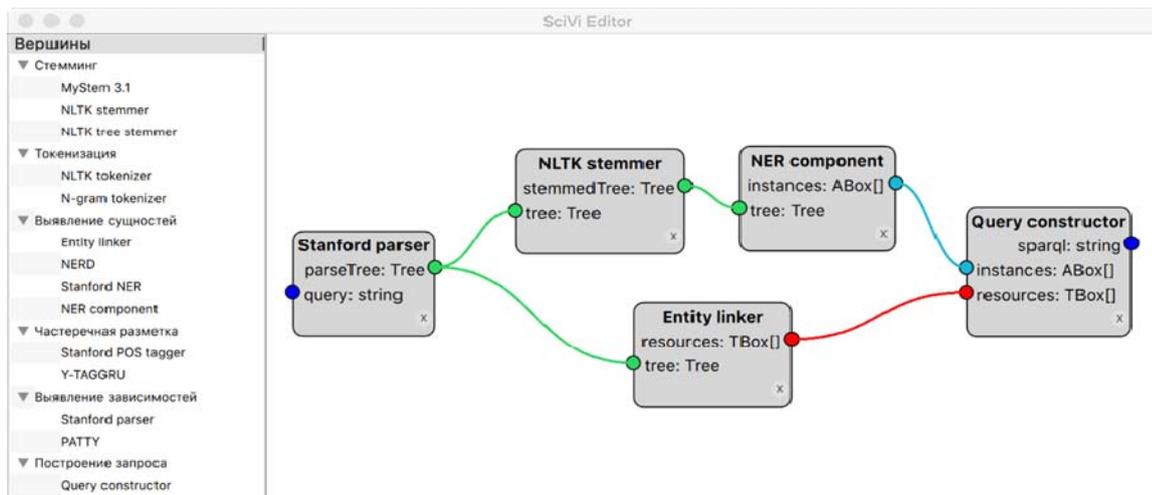


Рис. 1. Диаграмма средства трансформации, построенная в системе SciVi

Fig. 1. Diagram of Transforming Tool Built in SciVi System

После сохранения диаграммы в фоновом режиме запускается процесс комплексирования средства трансформации, соответствующего этой диаграмме. Для этого в копию заранее подготовленного шаблона проекта на языке Scala добавляются proto-файлы с описанием всех сервисов модулей, использованных в диаграмме. Затем автоматически генерируется программный код на языке Scala, который последовательно либо параллельно (в зависимости от дуг в диаграмме) осуществляет соответствующие gRPC-вызовы. Полученный проект компилируется и упаковывается в Docker-контейнер. После этого автоматически формируется соответствующий диаграмме конфигурационный файл для утилиты Docker Compose, позволяющей установить зависимости между Docker-контейнерами всех использованных модулей трансформации, а также настроить их совместный запуск. В платформе имеется возможность автоматической генерации нескольких средств трансформации по одной диаграмме. В таком случае генерируются альтернативные конвейеры трансформации, в которых отдельные модули заменены на другие совместимые по интерфейсу модули того же класса.

В сценарии *Проведение тестирования средств трансформации* пользователь выбирает одно или несколько комплексированных ранее средств трансформации и запускает их тестирование на одном или нескольких заранее подготовленных тестовых наборах. В настоящий момент тестовые наборы представляют собой множества из кортежей следующей структуры:

- 1) исходный естественно-языковой запрос;
- 2) ожидаемый результирующий SPARQL-запрос;
- 3) онтология, к которой задается запрос.

В сценарии *Анализ результатов тестирования средств трансформации* пользователь может просмотреть результаты тестирования в срезах по средствам трансформации и по тестовым наборам. Для каждого тестового случая пользователь может, не выходя из приложения, просмотреть ожидаемый и фактический результат трансформации ЕЯ-запроса в SPARQL-запрос в наглядном виде с использованием разработанного ранее средства визуального графического представления [18].

В сценарии *Выгрузка созданного средства трансформации* пользователь скачивает созданный на этапе комплексирования файл конфигурации для утилиты Docker Compose (docker-compose.yml). Благодаря принятым архитектурным решениям этого файла достаточно для автоматической загрузки и развертывания комплексированного средства трансформации на любом компьютере пользователя (при возможности подключения к приватному Docker-репозиторию платформы).

Особенности реализации

При загрузке модуля в платформу пользователь сопровождает его спецификацией сервиса на языке описания интерфейсов Protocol Buffers. Спецификация содержит имя сервиса, названия реализуемых сервисом серверных частей удаленных процедур, форматы входных и выходных сообщений. Встроенный сторонний редактор диаграмм потоков данных SciVi принимает на вход онтологическое описание модулей для добавления их в палитру редактора, отображения входных и выходных параметров модуля и анализа их совместимости. Для разбора файла спецификации и преобразования описания сервиса в требуемое SciVi онтологическое описание были использованы классы FileDescriptorSet, FileDescriptorProto и DescriptorProto из официальной библиотеки Protocol Buffers, позволяющие получить объектное представление спецификации сервисов.

В качестве языка программирования для автоматически генерируемого кода конвейера трансформации был выбран язык Scala. Статическая типизация и компилируемость этого языка позволяла на этапе разработки платформы быстрее находить некоторые ошибки в сгенерированном программном коде без необходимости запуска тестов. Кроме того, в системе сборки Scala build tool (sbt) были использованы релевантные для задачи плагины ScalaPB и sbt Native Packager. ScalaPB позволяет автоматически сгенерировать программный код gRPC-клиента на этапе компиляции без необходимости вызова сторонних программ генерации. sbt Native Packager позволяет автоматически упаковать Scala-приложение в Docker-контейнер и опубликовать его в задаваемом в параметрах проекта репозитории.

Исходя из того, что на данном этапе развития платформы предполагается, что пользователь в редакторе диаграмм потоков данных может строить только направленные ациклические графы, программный код конвейера трансформации генерируется с учетом возможных параллельных дуг графа. В таком случае более эффективным является параллельное выполнение запросов к отдельным модулям трансформации в асинхронном режиме. Для представления результатов асинхронных вызовов удаленных процедур gRPC в Scala используется класс Future, представляющий собой программную обертку над значением, которое в данный момент может быть еще не вычислено. Несмотря на возможное отсутствие значений, экземпляры классов Future можно комбинировать между собой в неблокирующем режиме. Благодаря этому весь генерируемый программный код представляет собой цепочку из неблокирующих вызовов с соответствующим параллельным исполнением запросов (см. пример в следующем разделе).

Пример комплексирования и тестирования средства трансформации с использованием разработанной платформы

Для демонстрации разработанной платформы была рассмотрена задача создания системы для трансформации ЕЯ-запросов в SPARQL-запросы к базе знаний Geobase⁵. База знаний предоставляется в виде фактов на языке Prolog. Для преобразования Prolog-фактов в OWL были использованы правила отображения, представленные в работе [19]. В качестве модулей системы были выбраны сторонние компоненты: стеммер NLTK Snowball stemmer (Python) [8], парсер зависимостей Stanford Parser (Java) [20]. Кроме того, были созданы демонстрационные компоненты NER Component (Kotlin), Entity Linker (Kotlin), Query Constructor (Kotlin), решающие задачи отображения токенов запроса на ресурсы онтологии и построения SPARQL-запроса по множеству найденных ресурсов.

Для каждого модуля была создана программная обертка, предоставляющая gRPC-сервис. Каждый модуль был загружен в Docker-репозиторий и зарегистрирован в разработанной платформе. Для каждого модуля автоматически сформирован файл с онтологическим описа-

⁵ URL: <http://www.cs.utexas.edu/users/ml/nldata/geoquery.html>

нием в формате использованного редактора диаграмм потоков данных. На рис. 2 приведен скриншот графического представления фрагмента автоматически сформированной онтологии (графическое представление создано при помощи редактора онтологий ОНТОЛИС 2.0 [21]).

После этого при помощи редактора диаграмм потоков данных был построен граф трансформации ЕЯ-запроса в SPARQL-запрос (см. рис. 1). На основании этого графа был автоматически сгенерирован и скомпилирован Scala-проект, выполняющий трансформацию запроса посредством удаленного вызова процедур использованных модулей в соответствии со спецификацией пользователя.

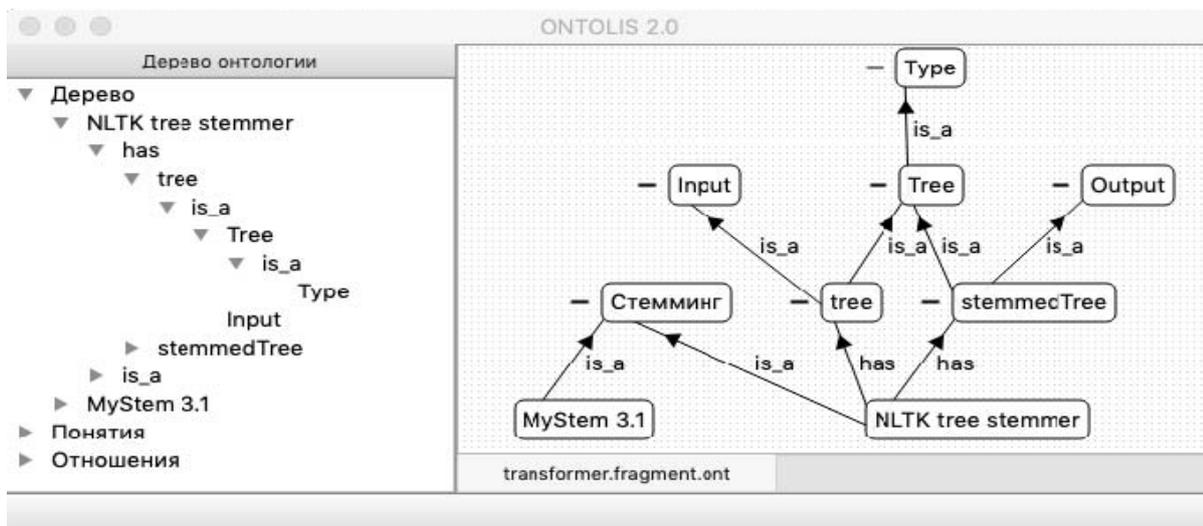


Рис. 2. Фрагмент автоматически сформированной онтологии

Fig. 2. Fragment of automatically built ontology

В листингах 1–5 приведены результаты работы всех модулей трансформации в виде текстового представления сообщений Protocol Buffers на примере запроса «What states does the Mississippi run through?».

Листинг 1. Результат работы модуля Stanford parser

Listing 1. Stanford Parser Module's Result

```
Tree: {
  tokens: [
    token: { index: 1, word: "What", pos: "WDT" },
    token: { index: 2, word: "states", pos: "NNS" },
    token: { index: 3, word: "does", pos: "VBZ" },
    token: { index: 4, word: "the", pos: "DT" },
    token: { index: 5, word: "Mississippi", pos: "NNP" },
    token: { index: 6, word: "run", pos: "NN" },
    token: { index: 7, word: "through", pos: "IN" },
    token: { index: 8, word: "?", pos: "." }
  ],
  dependencies: [
    dependency: { dep: "ROOT", governor: 0, dependent: 3 },
    dependency: { dep: "det", governor: 2, dependent: 1 },
    dependency: { dep: "nmod", governor: 3, dependent: 2 },
```

```

    dependency: { dep: "det", governor: 6, dependent: 4 },
    dependency: { dep: "compound", governor: 6, dependent: 5 },
    dependency: { dep: "dobj", governor: 3, dependent: 6 },
    dependency: { dep: "case", governor: 2, dependent: 7 }
  ]
}

```

Листинг 2. Результат работы модуля Entity linker
Listing 2. Entity Linker Module's Result

```

TBox: [
  iri: "http://www.fluz.sp.owl#State",
  iri: "http://www.fluz.sp.owl#run_through"
]

```

Листинг 3. Результат работы модуля NLTK Snowball stemmer
Listing 3. NLTK Snowball Stemmer Module's Result

```

Tree: {
  tokens: [
    token: { index: 1, word: "what", pos: "WDT" },
    token: { index: 2, word: "state", pos: "NNS" },
    token: { index: 3, word: "doe", pos: "VBZ" },
    token: { index: 4, word: "the", pos: "DT" },
    token: { index: 5, word: "Mississippi", pos: "NNP" },
    token: { index: 6, word: "run", pos: "NN" },
    token: { index: 7, word: "through", pos: "IN" },
    token: { index: 8, word: "?", pos: "." }
  ],
  dependencies: [
    dependency: { dep: "ROOT", governor: 0, dependent: 3 },
    dependency: { dep: "det", governor: 2, dependent: 1 },
    dependency: { dep: "nmod", governor: 3, dependent: 2 },
    dependency: { dep: "det", governor: 6, dependent: 4 },
    dependency: { dep: "compound", governor: 6, dependent: 5 },
    dependency: { dep: "dobj", governor: 3, dependent: 6 },
    dependency: { dep: "case", governor: 2, dependent: 7 }
  ]
}

```

Листинг 4. Результат работы модуля NER component
Listing 4. NER Component Module's Result

```

ABox: [
  iri: "http://www.fluz.sp.owl#mississippi_river"
]

```

Листинг 5. Результат работы модуля Query constructor
Listing 5. Query Constructor Module's Result

```

sparql: "PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX geo: <http://www.fluz.sp.owl#>
SELECT ?A {

```

```

    geo:mississippi_river geo:run_through ?A .
    ?A rdf:type geo:State
  }"

```

Собранный проект был автоматически упакован в Docker-контейнер и загружен в приватный Docker-репозиторий платформы.

Листинг 6. Фрагмент автоматически сгенерированного кода вызова модулей трансформации
Listing 6. Fragment of Automatically Generated Code which Calls Transformation Modules

```

override def transform(request: TransformRequest): Future[TransformResponse] = {
  val parseResponseF = stubDependencyParser.parse(
    ParseRequest(request.ontologySource, request.query))
  val stemmerResponseF = parseResponseF.flatMap(pr =>
    stubStemmer.stem(StemRequest(request.ontologySource, pr.parseTree)))
  val linkerResponseF = parseResponseF.flatMap(pr =>
    stubLinker.link(LinkRequest(request.ontologySource, pr.parseTree)))
  val nerResponseF = stemmerResponseF.flatMap(sr =>
    stubNer.ner(NerRequest(request.ontologySource, sr.stemmedTree)))
  val constructorResponseF = (for {
    nr <- nerResponseF
    lr <- linkerResponseF
  } yield stubConstructor.construct(ConstructRequest(request.ontologySource,
    nr.instances, lr.resources))).flatten
  constructorResponseF.map(cr => TransformResponse(cr.sparql))
}

```

С целью демонстрации пользовательского интерфейса просмотра результатов тестирования комплексированное средство трансформации было протестировано на наборе из 250 запросов Geoquery. На рис. 3 приведен пример просмотра результата модели на наборе тестов, а на рис. 4 – пример просмотра ожидаемых и фактических результатов тестового случая.

№	Запрос	Вердикт	Результат модели	Эталон
77	what is the largest river in washington state?	Неверно	SELECT ?river WHERE { ?ri...	SELECT ?river WHERE { ?...
78	what is the population of illinois?	Верно	SELECT ?population WHE...	SELECT ?population WHE...
79	which state borders the most states?	Неверно	SELECT * WHERE { }	SELECT ?state1 WHERE { ...
80	which rivers flow through alaska?	Верно	SELECT ?river WHERE { ?ri...	SELECT ?river WHERE { ?...
81	what city has the most people?	Неверно	SELECT ?state WHERE { ?...	SELECT ?city WHERE { ?c...
82	which states does the mississippi run through?	Верно	SELECT ?state WHERE { g...	SELECT ?state WHERE { ...
83	what is the capital of washington?	Верно	SELECT ?city WHERE { ?cit...	SELECT ?city WHERE { ?c...
84	what is the smallest city in the us?	Неверно	SELECT * WHERE { }	SELECT ?city WHERE { ?c...
85	what are the major cities in texas?	Неверно	SELECT ?city WHERE { ?cit...	SELECT ?city WHERE { ?c...
86	which state has the highest population density?	Неверно	SELECT ?state WHERE { ?...	SELECT ?state WHERE { ...
87	what state contains the highest point in the us?	Неверно	SELECT * WHERE { }	SELECT ?state WHERE { ...
88	what states does the delaware river run throu...	Верно	SELECT ?state WHERE { g...	SELECT ?state WHERE { ...
89	which states capital city is the largest?	Неверно	SELECT ?state WHERE { ?...	SELECT ?state WHERE { ...

Рис. 3. Пример просмотра результатов модели

Fig. 3. User Interface for Viewing Model's Testing Results

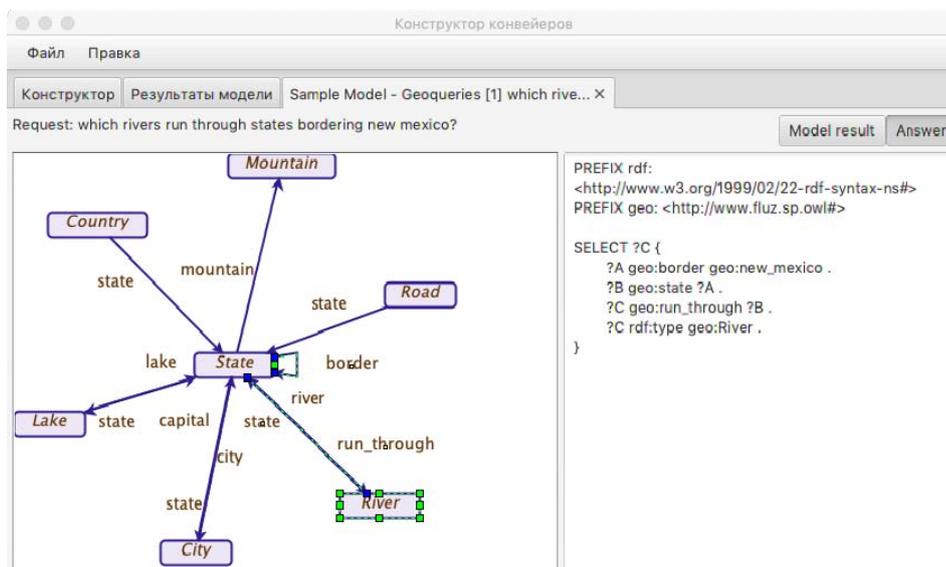


Рис. 4. Пример просмотра информации о результатах трансформации

Fig. 4. User Interface for Viewing Test's Detailed Information

Заключение

В работе были рассмотрены некоторые из современных средств трансформации ЕЯ-запросов в SPARQL-запросы. На основании схожести этапов их работы, а также по причине выявленных ограничений существующих платформ создания средств трансформации была сформулирована задача разработки собственной платформы комплексирования и тестирования средств трансформации.

Удовлетворение сформулированных к платформе требований было достигнуто при помощи использования двух ключевых технологий: программной платформы контейнеризации *Docker* и системы удаленного вызова процедур *gRPC*. Их использование позволило реализовать поддержку комплексирования переносимой системы из модулей, использующих наиболее подходящие для того или иного этапа программные технологии.

Описанная платформа автоматизирует многие процессы разработки средств трансформации: созданы средства поддержки связывания отдельных модулей с поддержкой автоматической проверки корректности, создана система автоматического комплексирования средств трансформации на основе графа трансформации, создана система тестирования, интегрированная с разработанной ранее системой визуализации результатов трансформации.

В будущем планируется автоматически генерировать средства трансформации произвольной структуры в дополнение к реализованной возможности генерации альтернативных средств заданной структуры, а также отображать различия между ожидаемым и фактическим результатами трансформации в наглядном виде при помощи визуального графического представления онтологии.

Список литературы / References

1. **Gruber T. R.** A Translation Approach to Portable Ontology Specifications. *Knowledge Acquisition*, 1993, vol. 5, no. 2, p. 199–220. DOI 10.1006/knac.1993.1008.

2. **Xiao G., Calvanese D., Kontchakov R., Lembo D., Poggi A., Rosati R., Zakharyashev M.** Ontology-based data access: A survey. In: Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, 2018, p. 5511–5519. DOI 10.24963/ijcai.2018/777.
3. **Чуприна С. И., Постаногов И. С.** Концепция обогащения унаследованных информационных систем сервисом запросов на естественном языке // Вестник Перм. ун-та. Математика. Механика. Информатика. 2015. Т. 2. С. 78–86.
Chuprina S. I., Postanogov I. S. Kontseptsiya obogashcheniya unasledovannykh informatsionnykh sistem servisom zaprosov na estestvennom yazyke [Enhancing Legacy Information Systems with a Natural Language Query Interface Service]. *Vestnik Permskogo universiteta. Matematika. Mekhanika. Informatika*, 2015, vol. 2, p. 78–86. (in Russ.)
4. **Dubey M., Dasgupta S., Sharma A., Höffner K., Lehmann J.** Asknow: A framework for natural language query formalization in SPARQL. In: International Semantic Web Conference. Cham., 2016, p. 300–316. DOI 10.1007/978-3-319-34129-3_19.
5. **Shekarpour S., Marx E., Ngomo A. C. N., Auer S.** SINA: Semantic Interpretation of User Queries for Question Answering on Interlinked Data. In: Web Semantics: Science, Services and Agents on the World Wide Web, 2015. DOI 10.1016/j.websem.2014.06.002.
6. **Xu K., Zhang S., Feng Y., Zhao D.** Answering natural language questions via phrasal semantic parsing. In: Natural Language Processing and Chinese Computing, 2014, p. 333–344. DOI 10.1007/978-3-662-45924-9_30.
7. **Diefenbach D., Lopez V., Singh K., Maret P.** Core techniques of question answering systems over knowledge bases: a survey. *Knowledge and Information systems*, 2018, vol. 3, no. 55, p. 529–569. DOI 10.1007/s10115-017-1100-y
8. **Bird S., Klein E., Loper E.** Natural language processing with Python: analyzing text with the natural language toolkit. O'Reilly Media, Inc., 2009, 504 p.
9. **Kostareva T., Chuprina S., Nam A.** Using Ontology-Driven Methods to Develop Frameworks for Tackling NLP Problems. In: AIST (Supplement), 2016, p. 102–113.
10. **Both A., Diefenbach D., Singh K., Shekarpour S., Cherix D., Lange C.** Qanary – a methodology for vocabulary-driven open question answering systems. In: The Semantic Web. Latest Advances and New Domains, 2016, p. 625–641. DOI 10.1007/978-3-319-34129-3_38.
11. **Ferrandez O., Spurk C., Kouylekov M., Dornescu I., Ferrandez S., Negri M., Magnini B.** The QALL-ME Framework: A specifiable-domain multilingual Question. *Answering architecture. Web semantics: Science, services and agents on the world wide web*, 2011, vol. 2, no. 9, p. 137–145. DOI 10.1016/j.websem.2011.01.002
12. **Marx E., Usbeck R., Ngomo A.C.N., Höffner K., Lehmann J., Auer S.** Towards an open question answering architecture. In: Proceedings of the 10th International Conference on Semantic Systems, 2014, p. 57–60. DOI 10.1145/2660517.2660519.
13. **Hohpe G.** Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions. Prentice Hall, 2004, 683 p.
14. **Trivedi P., Maheshwari G., Dubey M., Lehmann J.** Lc-quad: A corpus for complex question answering over knowledge graphs. In: Proceedings of the 16th International Semantic Web Conference (ISWC), 2017, p. 210–218. DOI 10.1007/978-3-319-68204-4_22.
15. **Bordes A., Usunier N., Chopra S., Weston J.** Large-scale simple question answering with memory networks. In: arXiv preprint arXiv:1506.02075, 2015.
16. **Usbeck R., Gusmita R. H., Saleem M., Ngomo A. C. N.** 9th challenge on question answering over linked data (QALD-9). In: CEUR Workshop Proceedings, 2018, vol. 2241, p. 58–64.
17. **Рябинин К. В., Чуприна С. И., Бортников А. Ю.** Автоматизация настройки систем научной визуализации на специфику разнообразных источников данных // Научная визуализация. 2016. Т. 8, № 4. С. 1–14.

- Ryabinin K. V., Chuprina S. I., Bortnikov A. Yu.** Avtomatizatsiya nastroiки sistem nauchnoi vizualizatsii na spetsifiku raznoobraznykh istochnikov dannykh [Automated Tuning of Scientific Visualization Systems to Varying Data Sources]. *Nauchnaya vizualizatsiya*, 2016, vol. 8, no. 4, p. 1–14. (in Russ.)
18. **Турова И. А., Постановов И. С.** Использование визуального графического представления в системах трансформации естественно-языковых запросов в SPARQL-запросы // Математика и междисциплинарные исследования – 2018. Пермь, 2018. С. 111–114.
Turova I. A., Postanogov I. S. Ispol'zovanie vizual'nogo graficheskogo predstavleniya v sistemakh transformatsii estestvenno-yazykovykh zaprosov v SPARQL-zaprosy [Using the Visual Graphical Representation in Systems that Transform Natural Language Queries into SPARQL Queries]. In: *Matematika i mezhdistiplinarnye issledovaniya* – 2018. Perm, 2018, p. 111–114. (in Russ.)
19. **Luz F. F., Finger M.** Semantic Parsing Natural Language into SPARQL: Improving Target Language Representation with Neural Attention. In: arXiv preprint arXiv:1803.04329. – 2018.
20. **Chen D., Manning C.** A fast and accurate dependency parser using neural networks. In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, p. 740–750. DOI 10.3115/v1/D14-1082.
21. ОНТОЛИС 2.0: Свидетельство о государственной регистрации программ для ЭВМ / И. С. Постановов, С. И. Чуприна. № 2017610729; дата регистрации 16.01.2017.
ONTOLIS 2.0: svidetel'stvo o gosudarstvennoi registratsii programm dlya EVM. I. S. Postanogov, S. I. Chuprina. № 2017610729; data registratsii 16.01.2017. (in Russ.)

*Материал поступил в редколлегию
Received
04.02.2019*

Сведения об авторах / Information about the Authors

Постановов Игорь Сергеевич, аспирант, старший преподаватель кафедры математического обеспечения вычислительных систем Пермского государственного национального исследовательского университета (ул. Букирева, 15, Пермь, 614068, Россия)

Igor S. Postanogov, PhD Student, Senior Lecturer, Department of Software Computing Systems, Perm State University (15 Bukirev Str, Perm, 614068, Russian Federation)
ipostanogov@outlook.com

Турова Ирина Алексеевна, студентка кафедры математического обеспечения вычислительных систем Пермского государственного национального исследовательского университета (ул. Букирева, 15, Пермь, 614068, Россия)

Irina A. Turova, Student, Department of Software Computing Systems, Perm State University (15 Bukirev Str, Perm, 614068, Russian Federation)
turovaia@yandex.ru

Правила оформления текста рукописи

Авторы представляют статьи на русском или английском языке объемом от 0,5 авторского листа (20 тыс. знаков) до 1 авторского листа (40 тыс. знаков), включая иллюстрации (1 иллюстрация форматом 190 × 270 мм = 1/6 авторского листа, или 6,7 тыс. знаков). Публикации, превышающие указанный объем, допускаются к рассмотрению только после индивидуального согласования с редакцией журнала.

Текст рукописи должен быть представлен в редколлегию в виде файла MS Word (.doc, .docx). Гарнитура Times New Roman, размер шрифта 11, межстрочный интервал 1, размеры полей – стандартные значения текстового редактора. Форматирование – выравнивание по ширине страницы, переносы слов включены, каждый новый абзац начинается с красной строки. Не допускается ручное форматирование абзацев (пробелами, лишними переводами строк, разрывами страниц).

Структура статьи

- Индекс УДК (универсальной десятичной классификации). Выравнивание по левому краю
- Название статьи. Выравнивание по центру, полужирный шрифт
- ФИО авторов (инициалы, фамилия). Выравнивание по центру, полужирный шрифт
- Места работы всех авторов. Выравнивание по центру, курсив
- Аннотация статьи
- Ключевые слова, не более 10
- Благодарности, сведения о финансовой поддержке
- Название статьи **на английском языке**. Выравнивание по центру, полужирный шрифт
- ФИО авторов **на английском языке** (инициалы, фамилия). Выравнивание по центру, полужирный шрифт
- Места работы авторов **на английском языке**. Выравнивание по центру, курсив
- Аннотация статьи **на английском языке (Abstract)**, 200–250 слов
- Ключевые слова **на английском языке (Keywords)**, не более 10
- Благодарности, сведения о финансовой поддержке **на английском языке**, если есть соответствующий раздел на русском языке (**Acknowledgements**)
- Основной текст
- Список литературы / **References**
- Сведения об авторах

Требования к оформлению основного текста и иллюстративных материалов

Основной текст должен быть представлен в структурированном виде, рекомендуется использовать подзаголовки – например: Введение, Методика..., Выводы, Результаты, Заключение.

Подзаголовки отделяются и набираются полужирным шрифтом. В целях выделения частей текста и отдельных слов и словосочетаний допускается использование курсива или полужирного шрифта. Подчеркивание, разрядка, изменение основного кегля и выделение цветом не используются.

Иллюстрации к рукописи статьи должны быть приложены в виде отдельных файлов. При этом в тексте должно содержаться включенное изображение с указанием имени файла. Все иллюстрации, содержащие схемы, графики, алгоритмы и т. п., должны быть представлены

в векторном виде (.ai, .eps, .cdr). Скриншоты и другие растровые изображения должны быть представлены в максимально высоком качестве, без каких-либо потерь и искажений (.jpg, .tif). Все иллюстрации должны иметь подрисовочную подпись – свое название. Надписи к таблицам и подписи к иллюстрациям приводятся **на двух языках (русском и английском).**

Примеры:

Рис. 1. Диаграмма производительности...

Fig. 1. Performance diagram...

Таблица 1

Сравнение алгоритмов...

Table 1

Comparison of algorithms...

Нумерация последовательная и неразрывная от начала статьи. Не допускается использование других наименований, кроме «Рис.» / «Fig.», «Таблица» / «Table», и усложнение нумерации (например, «Рис. 3.2.»). Ссылка на иллюстрацию в тексте должна быть приведена в круглых скобках, например: (рис. 1), (табл. 1).

Формулы должны быть набраны с использованием редактора MathType либо встроенного редактора формул MS Word. Кегль основных символов – 11, греческие символы набираются прямым шрифтом, латинские – курсивом. Нумеруются только те формулы, на которые автор ссылается в тексте.

Abstract

Аннотация статьи на английском языке (Abstract) не должна быть дословным переводом русскоязычной аннотации. Раздел Abstract, как и основной текст, должен быть структурирован, в нем должно содержаться описание цели работы, методов исследования, научной значимости, выводов / результатов. Требуется качественный перевод на английский язык (при необходимости просим авторов обращаться к профессиональным переводчикам). **Объем Abstract 200–250 слов.**

Список литературы / References

Список литературы и список литературы на английском языке (References) размещаются в общем разделе. Рекомендуемое количество цитируемых в статье источников – не менее 10, в список желательно включать ссылки на актуальные работы по теме исследования, особенно в иностранных периодических изданиях.

В тексте статьи ссылки на литературу указываются цифрами в квадратных скобках, при необходимости указываются номера страниц, например: [2; 3. С. 15].

Список литературы нумеруется в порядке цитирования и оформляется в соответствии с ГОСТ Р 7.0.5-2008 на библиографическое описание (знаки тире в описании опускаются). Ссылки на неопубликованные работы, а также на Интернет-ресурсы (кроме электронных изданий, поддающихся библиографическому описанию) оформляются в виде сноски.

В Список литературы ссылки на источники следует включать на оригинальном языке опубликования. Каждый источник должен быть также оформлен на английском языке (References) по международному стандарту для публикаций в области информатики IEEE Style со следующими отличиями:

- инициалы авторов указываются после фамилии;
- название статьи не берется в кавычки, отделяется точкой;
- отсутствует союз «and» перед фамилией последнего автора;
- в диапазоне страниц указывается одна «р» (вместо «pp. 2–9» – «р. 2–9»);
- год издания указывается после места издания (для книг) и сразу после названия журнала (для периодики).

Перевод источника на английский язык:

- если источник имеет выходные данные на английском языке, то для формирования References **следует использовать именно эти данные;**

• если оригинальная публикация не содержит выходных данных на английском языке, то допускается транслитерация названия материала на латинский алфавит в сочетании с переводом на английский язык в квадратных скобках. В конце описания указывается, на каком языке написана эта работа, например, (in Russ.). При транслитерации можно воспользоваться Интернет-ресурсом <http://ru.translit.ru/>, рекомендуется выбрать стандарт BSI. Место издания не транслитерируется, указывается полностью на английском языке, например: Moscow. Название издательства / издателя, как правило, транслитерируется. Для журналов, у которых есть официальное название на английском языке, – использовать его (проверить на сайте журнала, или, например, в библиотеке WorldCat), если названия на английском языке нет, использовать транслитерацию по системе BSI. Не следует самостоятельно переводить названия журналов.

Если у цитируемого источника есть **цифровой идентификатор DOI** (<https://search.crossref.org>), его требуется обязательно указывать в конце библиографической ссылки.

Примеры оформления ссылок. Каждый источник в том же пункте дублируется на английском языке (References).

Источник на русском языке, перевод на английский доступен в метаданных статьи

1. Журавлев С. С., Рудометов С. В., Окольнішников В. В., Шакиров С. Р. Применение модельно-ориентированного проектирования к созданию АСУ ТП опасных промышленных объектов // Вестник НГУ. Серия: Информационные технологии. 2018. Т. 16, № 4. С. 56–67. DOI 10.25205/1818-7900-2018-16-4-56-67

Zhuravlev S. S., Rudometov S. V., Okolnishnikov V. V., Shakirov S. R. Model-Based Design Approach for Development Process Control Systems of Hazardous Industrial Facilities. *Vestnik NSU. Series: Information Technologies*, 2018, vol. 16, no. 4, p. 56–67. (in Russ.) DOI 10.25205/1818-7900-2018-16-4-56-67

Источник на английском языке. Оформляем согласно требованиям для References. Приводим только 1 раз.

2. Telnov V. I. Optimization of the Beam Crossing Angle at the ILC for E + e- and yy Collisions. *Journal of Instrumentation*, 2018, vol. 13, no. 03, p. P03020–P03020. DOI 10.1088/1748-0221/13/03/p03020

Метаданные источника доступны только на русском языке

3. Жижимов О. Л., Федотов А. М., Шокин Ю. И. Технологическая платформа массовой интеграции гетерогенных данных // Вестник НГУ. Серия: Информационные технологии. 2013. Т. 11, вып. 1. С. 24–41ю

Zhizhimov O. L., Fedotov A. M., Shokin Yu. I. Tekhnologicheskaya platforma massovoi integratsii geterogennykh dannyykh [Technology Platform For the Mass Integration of Heterogeneous Data]. *Vestnik NSU. Series: Information Technologies*, 2013, vol. 11, no. 1, p. 24–41. (in Russ.)

Сведения об авторах

Последний раздел статьи – информация об авторе / авторах **на русском и английском языках:**

- ФИО полностью, ученая степень, ученое звание, должность, место работы, адрес места работы
- e-mail
- идентификаторы автора, такие как ORCID или ResearcherID (всем авторам рекомендуется использовать данные сервисы для ведения актуального списка своих публикаций)
- контактный телефон (не публикуется)

Если статья представляется на английском языке, необходимо приложить перевод на русский язык названия, аннотации, ключевых слов, сведений об авторе.

Доставка материалов

Материалы предоставляются в редакцию по электронной почте inftech@vestnik.nsu.ru.

Порядок рецензирования

Все статьи сначала проходят проверку на заимствование и только после этого отправляются на рецензирование. Редакционный совет не допускает к публикации материал, если имеется достаточно оснований полагать, что он является плагиатом.

Тип рецензирования статей – двухуровневое, одностороннее анонимное («слепое»).

Для каждой статьи редколлегией выбираются рецензенты, научная деятельность которых связана с темой представленного материала. Ответственный секретарь журнала обращается к ним с просьбой дать экспертную оценку статье либо помочь организовать рецензирование.

Рецензии для журнала «Вестник НГУ. Серия: Информационные технологии» составляются по единой схеме и подразумевают оценку по следующим критериям: соответствие тематике журнала, оригинальность и значимость результатов, качество изложения материала.

Заполненный бланк рецензии высылается на электронный адрес редакции. В зависимости от экспертных заключений статья может быть принята редакционным советом к опубликованию, рекомендована автору к доработке (с последующим повторным рецензированием либо без него) или отклонена (с предоставлением автору мотивированного отказа). Автору на электронный адрес высылается текст рецензии без указания ФИО рецензента и его контактных данных.

Все рецензии хранятся в редакции журнала не менее 5 лет. Редколлегия журнала обязуется при поступлении соответствующего запроса направлять копии рецензий в Министерство науки и высшего образования Российской Федерации.