

ТРУДЫ

**ИНСТИТУТА СИСТЕМНОГО
ПРОГРАММИРОВАНИЯ РАН**

**PROCEEDINGS OF THE INSTITUTE
FOR SYSTEM PROGRAMMING OF THE RAS**

ISSN Print 2079-8156
Том 34 Выпуск 2

ISSN Online 2220-6426
Volume 34 Issue 2

Институт системного
программирования
им. В.П. Иванникова РАН

Москва, 2022

ИСП **РАН**

Труды Института системного программирования РАН Proceedings of the Institute for System Programming of the RAS

Труды ИСП РАН – это издание с двойной анонимной системой рецензирования, публикующее научные статьи, относящиеся ко всем областям системного программирования, технологий программирования и вычислительной техники. Целью издания является формирование научно-информационной среды в этих областях путем публикации высококачественных статей в открытом доступе.

Издание предназначено для исследователей, студентов и аспирантов, а также практиков. Оно охватывает широкий спектр тем, включая, в частности, следующие:

- операционные системы;
- компиляторные технологии;
- базы данных и информационные системы;
- параллельные и распределенные системы;
- автоматизированная разработка программ;
- верификация, валидация и тестирование;
- статический и динамический анализ;
- защита и обеспечение безопасности ПО;
- компьютерные алгоритмы;
- искусственный интеллект.

Журнал издается по одному тому в год, шесть выпусков в каждом томе.

Поддерживается открытый доступ к содержанию издания, обеспечивая доступность результатов исследований для общественности и поддерживая глобальный обмен знаниями.

Труды ИСП РАН реферируются и/или индексируются в:

Proceedings of ISP RAS are a double-blind peer-reviewed journal publishing scientific articles in the areas of system programming, software engineering, and computer science. The journal's goal is to develop a respected network of knowledge in the mentioned above areas by publishing high quality articles on open access.

The journal is intended for researchers, students, and practitioners. It covers a wide variety of topics including (but not limited to):

- Operating Systems.
- Compiler Technology.
- Databases and Information Systems.
- Parallel and Distributed Systems.
- Software Engineering.
- Software Modeling and Design Tools.
- Verification, Validation, and Testing.
- Static and Dynamic Analysis.
- Software Safety and Security.
- Computer Algorithms.
- Artificial Intelligence.

The journal is published one volume per year, six issues in each volume.

Open access to the journal content allows to provide public access to the research results and to support global exchange of knowledge. **Proceedings of ISP RAS** is abstracted and/or indexed in:



Редколлегия

Главный редактор - [Аветисян Арутюн Ишханович](#), академик РАН, доктор физико-математических наук, профессор, ИСП РАН (Москва, Российская Федерация)

Заместитель главного редактора - [Кузнецов Сергей Дмитриевич](#), д.т.н., профессор, ИСП РАН (Москва, Российская Федерация)

Члены редколлегии

[Воронков Андрей Анатольевич](#), доктор физико-математических наук, профессор, Университет Манчестера (Манчестер, Великобритания)

[Вирбицкайте Ирина Бонавентуровна](#), профессор, доктор физико-математических наук, Институт систем информатики им. академика А.П. Ершова СО РАН (Новосибирск, Россия)

[Коннов Игорь Владимирович](#), кандидат физико-математических наук, Технический университет Вены (Вена, Австрия)

[Ластовецкий Алексей Леонидович](#), доктор физико-математических наук, профессор, Университет Дублина (Дублин, Ирландия)

[Ломазова Ирина Александровна](#), доктор физико-математических наук, профессор, Национальный исследовательский университет «Высшая школа экономики» (Москва, Российская Федерация)

[Новиков Борис Асенович](#), доктор физико-математических наук, профессор, Санкт-Петербургский государственный университет (Санкт-Петербург, Россия)

[Петренко Александр Федорович](#), доктор наук, Исследовательский институт Монреаля (Монреаль, Канада)

[Черных Андрей](#), доктор физико-математических наук, профессор, Научно-исследовательский центр CICESE (Энсенада, Баха Калифорния, Мексика)

[Шустер Ассаф](#), доктор физико-математических наук, профессор, Технион — Израильский технологический институт Technion (Хайфа, Израиль)

Адрес: 109004, г. Москва, ул. А. Солженицына, дом 25.

Телефон: +7(495) 912-44-25

E-mail: info-isp@ispras.ru

Сайт: <http://www.ispras.ru/proceedings/>

Editorial Board

Editor-in-Chief - [Arutyun I. Avetisyan](#), Academician of RAS, Dr. Sci. (Phys.–Math.), Professor, Ivannikov Institute for System Programming of the RAS (Moscow, Russian Federation)

Deputy Editor-in-Chief - [Sergey D. Kuznetsov](#), Dr. Sci. (Eng.), Professor, Ivannikov Institute for System Programming of the RAS (Moscow, Russian Federation)

Editorial Members

[Igor Konnov](#), PhD (Phys.–Math.), Vienna University of Technology (Vienna, Austria)

[Alexev Lastovetsky](#), Dr. Sci. (Phys.–Math.), Professor, UCD School of Computer Science and Informatics (Dublin, Ireland)

[Irina A. Lomazova](#), Dr. Sci. (Phys.–Math.), Professor, National Research University Higher School of Economics (Moscow, Russian Federation)

[Boris A. Novikov](#), Dr. Sci. (Phys.–Math.), Professor, St. Petersburg University (St. Petersburg, Russian Federation)

[Alexandre F. Petrenko](#), PhD, Computer Research Institute of Montreal (Montreal, Canada)

[Assaf Schuster](#), Ph.D., Professor, Technion - Israel Institute of Technology (Haifa, Israel)

[Andrei Tchervnykh](#), Dr. Sci., Professor, CICESE Research Centre (Ensenada, Baja California, Mexico).

[Irina B. Virbitskaite](#), Dr. Sci. (Phys.–Math.), The A.P. Ershov Institute of Informatics Systems, Siberian Branch of the RAS (Novosibirsk, Russian Federation)

[Andrew Voronkov](#), Dr. Sci. (Phys.–Math.), Professor, University of Manchester (Manchester, United Kingdom)

Address: 25, Alexander Solzhenitsyn st., Moscow, 109004, Russia.

Tel: +7(495) 912-44-25

E-mail: info-isp@ispras.ru

Web: <http://www.ispras.ru/en/proceedings>

Подход к генерации тестовых программ для верификации когерентности памяти микропроцессоров «Эльбрус» <i>Агафонов В.А., Фролов П.В., Мешков А.Н.</i>	7
Алгоритм генерации тестов из функциональной спецификации с использованием модели Open IE и кластеризации <i>Кобышев К.С., Молодяков С.А.</i>	17
Поиск ошибок в бинарном коде методами динамической символической интерпретации <i>Вишняков А.В., Кобрин И.А., Федотов А.Н.</i>	25
Паттерны проектирования для аналитической платформы, основанной на знаниях <i>Заякин В.С., Лядова Л.Н., Рабчевский Е.А.</i>	43
Акторный подход к управлению бизнес-процессами разработки аналитических систем <i>Прибылев Т.М., Зайцев М.Н., Викентьева О.Л.</i>	57
Архитектура программной системы для проектирования надежных бизнес-процессов <i>Самойлова К.В., Замятина Е.Б.</i>	67
Стратегии семплирования текста для прогнозирования недостающих библиографических ссылок <i>Краснов Ф.В., Смазневич И.С., Баскакова Е.Н.</i>	77
Функции потерь для обучения моделей сегментации изображений документов <i>Перминов А.И., Турдаков Д. Ю., Беляева О.В.</i>	89
Система управления заданиями автоматизированного сбора данных из сети Интернет <i>Лазарев В.А., Варламов М.И., Яцков А.К.</i>	111
Подход автоматического управления объектным хранилищем с помощью Operator SDK и Custom Resource Definition <i>Стоноженко К. М., Никифоров И. В., Устинов С. М.</i>	123
Определение присутствия человека с помощью фильтра Колмогорова-Винера и нейронных сетей с управляемыми рекуррентными блоками методом Wi-Fi-сканирования <i>Шибяев П.П., Чупахин А.А.</i>	135
Автоматизация процесса экологического мониторинга в особо охраняемых природных территориях российской Федерации на базе системы управления контентом Directum RX <i>Рудько И.Д., Хорошко Л.Л., Кейно П.П.</i>	145
Реализация функций управления задачами и ресурсами высокопроизводительной вычислительной системы в «СПО Супер-ЭВМ» <i>Игнатьев А.О., Калинин А.А, Мокшин С.Ю.</i>	159

Веб-приложение для продвижения донорства крови в России <i>Смирнов П.А., Малиновская В.В., Воинов Н.В.</i>	179
Модификация метода расчета полигенных рисков с использованием графа вариации <i>Кондратьева О.А., Карпулевич Е.А.</i>	191
Подход к отображению данных для пользователей телемедицинских систем раннего обнаружения заболеваний по ЭКГ <i>Бусенков А.А., Холодов Э.А., Новиков Р.С., Позин Б.А.</i>	201

Table of Contents

An Approach to Test Program Generation for Memory Coherence Verification of “Elbrus” Microprocessors <i>Agafonov V.A., Frolov P.V., Meshkov A.N.</i>	7
An algorithm of test generation from functional specification using Open IE model and clustering <i>Kobyshev K.S., Molodyakov S.A.</i>	17
Error detection in binary code with dynamic symbolic execution <i>Vishnyakov A.V., Kobrin E.A., Fedotov A.N.</i>	25
Design Patterns for a Knowledge-Driven Analytical Platform <i>Zayakin V.S., Lyadova L.N., Rabchevskiy E.A.</i>	43
Actor-oriented approach for business-process management of analytical system development <i>Pribylev T.M., Zaytsev M.N., Vikentyeva O.L.</i>	57
Architecture a software system for designing robust business processes <i>Samoylova K.V., Zamyatina E.B.</i>	67
Text sampling strategies for predicting missing bibliographic links <i>Krasnov F.V., Smaznevich I.S., Baskakova E.N.</i>	77
Loss functions for train document image segmentation models <i>Perminov A.I., Turdakov D. Y., Belyaeva O.V.</i>	89
Job management system for automated data collection from the Internet <i>Lazarev V.A., Varlamov M.I., Yatskov A.K.</i>	111
Automated Object Storage Management Approach with Operator SDK and Custom Resource Definition <i>Stonozhenko K. M., Nikiforov I. V., Ustinov S. M.</i>	123
Wi-Fi Sensing Human Detection with Kolmogorov-Wiener Filter and Gated Recurrent Neural Networks <i>Shibaev P.P., Chupakhin A.A.</i>	135
Automation of the process of environmental monitoring in specially protected natural territories of the Russian Federation based on the content management system Directum RX <i>Rudko I.D., Khoroshko L.L., Keyno P.P.</i>	145
Task and resources management function in HPC operation system «SPO Super-EVM» <i>Ignatyev A.O., Kalinin A.A., Mokshin S.Yu.</i>	159
A Web Application to Promote Blood Donation in Russia <i>Smirnov P.A, Malinovskaya V.V., Voinov N.V.</i>	179
Modification of the Method for Calculating Polygenic Risks with Variation Graph <i>Kondrateva O.A., Karpulevich E.A.</i>	191

Approach to Displaying Data to Users of Telemedicine Systems for Early Detection of Diseases by ECG

Busenkov A.A., Kholodov E.A., Novikov R.S., Pozin B.A.201

DOI: 10.15514/ISPRAS-2022-34(2)-1



An Approach to Test Program Generation for Memory Coherence Verification of “Elbrus” Microprocessors

^{1,2} V.A. Agafonov, ORCID: 0000-0001-6632-7434 <Vladimir.A.Agafonov@mcst.ru>

^{1,2,3} P.V. Frolov, ORCID: 0000-0002-9810-2210 <Pavel.V.Frolov@mcst.ru>

^{2,3} A.N. Meshkov, ORCID: 0000-0002-8117-7398 <Aleksey.N.Meshkov@mcst.ru>

¹ Moscow Institute of Physics and Technology (National Research University),
9 Institutskiy per., Dolgoprudny, Moscow Region, 141701, Russia

² MCST,

24 Vavilova str., Moscow, 119334, Russia

³ INEUM,

24 Vavilova str., Moscow, 119334, Russia.

Abstract. One of the key aspects of the correctness of the memory subsystem of a microprocessor is its functioning in accordance with the memory coherence protocol. This article presents an approach to test program generation for memory coherence verification of “Elbrus” microprocessors. Requirements for memory coherence tests are considered. The memory map structure allowing to describe the memory areas used in tests and the types of accesses to these areas in a flexible way is presented. The method of test program generation based on the memory map structure is described. The method of automatic memory map generation is proposed. Generated tests have been used for verification of RTL models and FPGA-based prototypes.

Keywords: system verification; memory coherence verification; pseudorandom test generation; Elbrus

For citation: Agafonov V.A., Frolov P.V., Meshkov A.N. An Approach to Test Program Generation for Memory Coherence Verification of “Elbrus” Microprocessors. Trudy ISP RAN/Proc. ISP RAS, vol. 34, issue 2, 2022, pp. 7-16. DOI: 10.15514/ISPRAS-2022-34(2)-1

Подход к генерации тестовых программ для верификации когерентности памяти микропроцессоров «Эльбрус»

^{1,2} В.А. Агафонов, ORCID: 0000-0001-6632-7434 <Vladimir.A.Agafonov@mcst.ru>

^{1,2,3} П.В. Фролов, ORCID: 0000-0002-9810-2210 <Pavel.V.Frolov@mcst.ru>

^{2,3} А.Н. Мешков, ORCID: 0000-0002-8117-7398 <Aleksey.N.Meshkov@mcst.ru>

¹ Московский физико-технический институт,
Россия, 141701, Московская область, г. Долгoprудный, Институтский пер., 9

² АО «МЦСТ»

Россия, 119334, Москва, ул. Вавилова, д. 24

³ ПАО «Институт электронных управляющих машин им. И.С. Брука»

Россия, 119334, Москва, ул. Вавилова, д. 24

Аннотация. Одним из ключевых аспектов корректности подсистемы памяти микропроцессора является ее функционирование в соответствии с протоколом когерентности памяти. В данной статье представлен подход к генерации тестовых программ для верификации когерентности памяти микропроцессоров семейства “Эльбрус”. Рассмотрены требования к тестовым программам

когерентности памяти. Представлена структура карты памяти, позволяющая гибко описывать используемые в тестовых программах области памяти и типы обращений к ним. Описан метод генерации тестовых программ на основе карты памяти. Предложен способ автоматической генерации карт памяти. Сгенерированные тестовые программы использовались при проведении верификации на RTL-моделях и прототипах на базе ПЛИС.

Ключевые слова: системная верификация; верификация когерентности памяти; генерация псевдослучайных тестов; Эльбрус.

Для цитирования: Агафонов В.А., Фролов П.В., Мешков А.Н. Подход к генерации тестовых программ для верификации когерентности памяти микропроцессоров «Эльбрус». Труды ИСП РАН, том 34, вып. 2, 2022 г., стр. 7-16. DOI: 10.15514/ISPRAS-2022-34(2)-1

1. Introduction

Memory subsystems of modern microprocessors provide support for various address spaces with address translation and include various levels of cache memory, means of ensuring data coherence, numerous buffers and switches [1]. To ensure the consistency of states of caches, the computing nodes of the system are combined into a single system and exchange messages in accordance with a cache coherence protocol [2]. All this determines high combinatorial complexity of verification [3], which sharply restricts the use of formal methods to individual bottlenecks or devices.

To increase the probability of reproducing various dynamic situations that occur when many devices are functioning simultaneously in the system, increasing the number of tests is necessary. In order to automate the development of test scenarios and their implementations in the form of ready-made test programs, automatic test generation is actively used. In this case, the source code of the test program is generated randomly, considering the specified parameters [3]. Parameterization allows fitting tests to reproduce certain situations with certain sets of random parameters [4]. The selection of test generation control parameters is performed at the development stage of each tool separately and is determined by the test generation algorithm.

Currently, MCST JSC is designing multi-core microprocessors with general-purpose cores of “Elbrus” architecture version 6. The development of “Elbrus” architecture has led to significant additions to the instruction set architecture and changes in the operation of devices included in the memory subsystem. In particular, the transition to a new coherence protocol affected the functionality of existing memory accesses. This led to inapplicability of the existing test generation algorithm to verify the memory subsystem of the developing microprocessors. For these reasons, development of a new algorithm for generating memory coherence tests and its implementation in the form of a pseudo-random Assembly test generator was required. This article discusses a new approach that is the basis for solving this problem.

2. “Elbrus” architecture overview

“Elbrus” architecture introduces a set of memory types determining the system behavior of memory accesses. The system behavior of memory access is characterized by a combination of specified properties. Each memory type describes a unique combination of these properties.

Memory access instructions are represented by instructions of store and load types. The size of the addressing memory fragment and the source/destination register format are determined by the memory access instruction format. “Elbrus” architecture uses an operation code extension – memory address specifier (MAS) for each memory access instruction. MAS defines the additional specific properties of the memory access, the method for storing data in different cache levels and affects the memory type. Thus, the set of memory accesses types is determined by the following formula:

$$T \subseteq I \times F \times M \quad (1)$$

where:

T – set of memory accesses types,

I – set of memory access instruction types,
 F – set of memory access instruction formats,
 M – set of MAS.

According to the generally accepted classification “Elbrus” is a VLIW (Very Long Instruction Word) architecture [5]. Each VLIW contains a set of instructions. Instructions placed in the same VLIW are executed in parallel.

In “Elbrus” architecture VLIW consists of 6 instruction channels. The sets of instructions supported by each channel are different. Memory access instructions are supported by only 4 instruction channels. However, store instructions are supported by only 2 of these channels. The use of certain memory access types is moreover supported by only certain channels. Due to architectural restrictions, the placement of several instructions in the same VLIW is limited.

Modern “Elbrus” microprocessors are Systems-on-a-Chip (SoC) with multiple unified general-purpose cores. The unified general-purpose core includes private level 1 instruction cache (L1I), level 1 data cache (L1D) and level 2 cache (L2). SoC of various configurations are being developed. The number of unified general-purpose cores, the presence of shared level 3 cache (L3), the number of interprocessor communication channels (IPCC), the number of memory controllers and other options are defined by the SoC configuration. Multiple microprocessors can be combined in a multiprocessor system with coherent shared memory based on ccNUMA principle by IPCC.

Among the innovations of “Elbrus” instruction set version 6 can be noted: the introduction of additional memory access properties, the transition to new memory types, the introduction of additional MAS with extended caching hints and the elimination of Input-Output memory space coherence support. In addition, hardware innovations include the transition to unified general-purpose cores, cache memory policies modification and optimization for unaligned memory accesses. The coherence protocol has also been significantly changed.

3. Test requirements

The following requirements for generated tests were formed.

Since “Elbrus” microprocessors contain multiple cores and can be combined in a variety of multiprocessor configurations, memory coherence tests should be designed for verification of multiprocessor systems. Before the execution of a test sequence, the system under test should be initialized. In order to check different operating modes of memory subsystem devices and cover more dynamic situations, the implementation of pseudo-random initialization of memory subsystem devices settings within acceptable limits is necessary. The system initialization procedure should end with a cores synchronization procedure to ensure that all of the cores of the system under test are ready for execution of a test sequence.

The existing test development environment provides a unified parameterized system initialization program and implementations of commonly used test procedures, such as cores synchronization and exit code output to the using test bench. Using the test development environment is a great way to simplify test development.

After the execution of the system initialization procedure each core executes a test sequence of instructions. In the case of memory subsystem verification tests, the test sequence should contain memory access instructions of various memory access types and perform the operation of cores with shared memory. In order to check the functionality of individual cache memories, the organization of memory access sequences at addresses that lead to cache lines eviction is necessary. To test the memory coherence mechanisms, a stream of parallel requests to the same cache memory lines from different cores should be formed. In addition, the generation of VLIWs with various combinations of memory access instructions needs to be supported in order to check the VLIWs execution.

The test sequence of instructions for each core should end with a self-check procedure that checks the correctness of register values and data in the tested memory areas. This approach allows using

tests to verify RTL models, FPGA-based prototypes [6], and to test manufactured chips. The intention is to use a self-checking code generator for the self-check procedures generation. This tool is based on the functional model of the system under test and allows to generate the code for comparing registers with the reference values obtained as a result of the test execution by the functional model [7][8].

Debugging of generated tests is supposed to be performed using a functional model of the system under test and a trace comparator. The results of the test execution on the functional model and RTL model are the executed instructions traces. Firstly, to achieve a successful test execution on the functional model is necessary. This is followed by debugging the test on the RTL model. To speed up the search for differences between the functional model and RTL model executed instruction traces the trace comparator should be used. Due to the unavailability of the execution instruction trace, debugging tests on the FPGA-based prototype is difficult. For this reason, the test exit code should localize the failed self-check code statement. The reasons for the test fail can be both the test errors and the system under test errors.

4. Shared memory interaction

The operation of multiple cores with shared memory allows using memory coherence maintain mechanisms and detecting errors in their implementations. There are two ways of sharing memory: *true sharing* and *false sharing*. Both of them are presented on fig. 1.

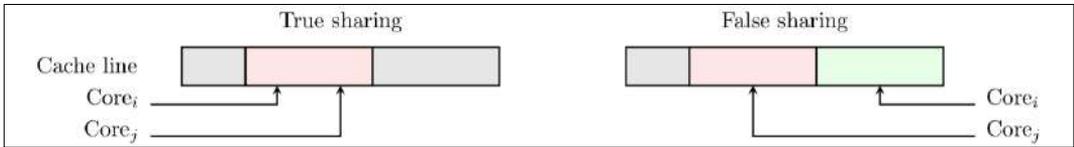


Fig. 1. Methods of memory sharing

True sharing is the operation of multiple cores with overlapping memory fragments. The implementation of true sharing in tests is limited due to the need to ensure deterministic test execution. To ensure determinism of the true sharing usage, synchronization of code execution by the cores before and after each modification of the overlapping memory fragments is necessary. Since the cores synchronization procedure requires multiple memory accesses from each of the synchronized cores and blocks the test sequence execution, use of true sharing can lead to high overhead costs.

False sharing is the operation of multiple cores with non-overlapping memory fragments. One of the cores is designated for each memory fragment – the core-owner. Only the core-owner can operate with its own memory fragments. Since cache lines change their states according to the coherence protocol, this way of sharing memory allows loading memory coherence mechanisms without the overhead of ensuring the determinism: multiple cores use the same cache line, however, the order of memory accesses of the core-owner is guaranteed by the hardware according to the used memory type and other cores do not affect the content of the memory fragment.

In comparison with true sharing, false sharing does not provide real operation with shared data. Thus, some ways of changing the states of cache lines are not checked. In addition, using true sharing can change the dynamics of the test execution.

For these reasons, the combination of both approaches was proposed: mainly to organize data separation in tests by false sharing, but at the same time to implement operation with common data in some volume. This is achieved by forming a testing sequence from a parameterized number of sections. The full test structure is shown in fig. 2. At the beginning of each section, all of the cores are synchronized. After synchronization procedures, a random sequence of VLIWs with different combinations of memory access instructions is generated for each core. Within each section of the test sequence, cache lines are split between the cores using false sharing in different ways. Therefore,

when switching between test sections occurs, some memory fragments are passed to other cores-owners for management. Thus, the operation with shared memory fragments by several microprocessor cores is implemented in cores synchronization procedures and during transitions between the test sections.

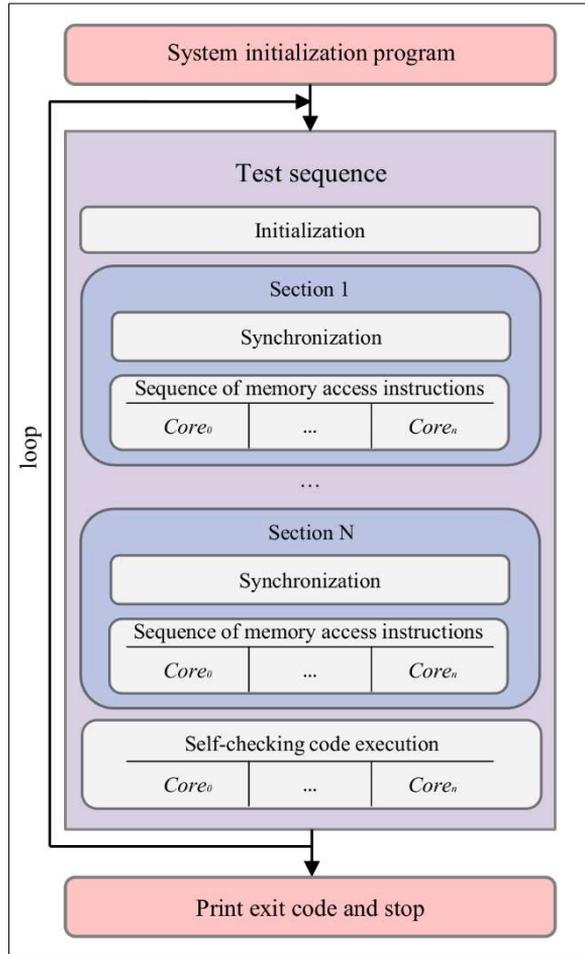


Fig. 2. Test structure

5. Description of memory accesses

A special structure – a memory map has been developed to describe memory accesses. The memory map contains a list of address ranges corresponding to memory fragments and maps the memory fragments to their parameter sets:

$$[a_k^{begin}, a_k^{end}] \mapsto \{I_k, F_k, M_k, c_k, r_k^{st/ld}, p_k^u\}, k = \overline{1, K} \quad (2)$$

where:

a_n^{begin} – the address of the n th memory fragment beginning,

a_n^{end} – the address of the n th memory fragment ending,

I_n – the n th set of memory access instruction types,

F_n – the n th set of memory access instruction formats,

M_n – the n th set of MAS,

c_n – the n th core-owner,

$r_n^{st/lid}$ – the n th store-to-load ratio,

p_n^u – the n th priority of use,

K – the number of the memory fragments in the memory map.

All addresses are presented in terms of bytes. The memory map describes non-overlapping memory fragments:

$$a_1^{begin} \leq a_1^{end} < a_2^{begin} \leq a_2^{end} < \dots < a_K^{begin} \leq a_K^{end} \quad (3)$$

Adjacent memory fragments with equivalent parameter sets are combined into a single memory fragment in the memory map. Priorities of use memory fragments allow managing the frequency of memory accesses to each memory fragment, as well as store-to-load ratios control the frequency of using different types of memory accesses. This allows configuring spatio-temporal profiles of cores with different memory areas in a flexible way [9].

The need to ensure a deterministic state of memory imposes restrictions on the combination of using memory access types. Using coherent and non-coherent memory accesses to the memory fragments located in the same cache line generally results in an undefined cache line memory value. For this reason, the set of valid memory access types for each memory fragment should be selected for deterministic reasons.

A memory map is used to describe the requests generated in a test section. For multi-section tests, a separate memory map for each section of the test should be used. In this case, ensuring deterministic transitions between the test sections by adding additional procedures to the test is necessary.

6. Code generator

The code generator implements the described in the memory maps memory accesses as a ready-made Assembly test in accordance with the test structure presented in fig. 2.

At the beginning of the test, the code generator defines initialization parameters for the parameterized system initialization program. Depending on the test generation parameters, default or random settings for memory subsystem devices can be used.

The test sequence is executed in a loop. The number of iterations of the loop is parameterized. Due to the operation of the cache memory, different dynamics of the test program execution at different iterations is achieved. Consequently, with a slight increase in the size of the test, the execution time and the number of situations being tested can be increased many times. This is convenient when performing a test on an FPGA-based prototype, where the test load time is significant.

Memory access instructions are generated based on the memory map of the current test section. The algorithm for generating a test sequence using a memory map runs independently for each core. To check the correctness of the store-to-load bypass in the L1D cache, random generation of the load instruction corresponding to the previous store instruction by address, format and MAS is provided. Random generation of "wait" instructions is also implemented. For core c the algorithm consists of the following steps:

- 1) Retrieving the set A of memory fragments owned by core c ;
- 2) An empty VLIW creating;
- 3) Constructing a memory fragments probability distribution based on the priorities of use p^u for memory fragments $a \in A$;
- 4) If bypass-load is scheduled to be inserted into this VLIW, restore the memory access type t , the memory access address \tilde{a} and go to step 13;
- 5) With specified probability of wait instruction generation randomly choose an instruction from the set of acceptable wait instructions and go to step 14;
- 6) Random choice of the memory fragment $a_m \in A$ according to the constructed memory fragments probability distribution;

- 7) Constructing a store-to-load probability distribution based on the store-to-load ratio $r_m^{st/ld}$;
- 8) Random choice of the instruction type $i \in I_m$ according to the constructed store-to-load probability distribution;
- 9) Retrieving the set $\widetilde{F}_m \subset F_m$ of suitable memory access instruction formats $f: \text{size}(f) \leq \text{size}(a_m), \forall f \in \widetilde{F}_m$;
- 10) Retrieving the set T_m of suitable memory access types using (2) in accordance with the requirements of the instruction set architecture:

$$T_m \subseteq i \times \widetilde{F}_m \times M_m \quad (4)$$

- 11) Random choice of a memory access type $t \in T_m$;
- 12) Random choice of the memory access address $\tilde{a} \in a_m$ considering the memory access alignment and the borders of a_m ;
- 13) Selecting the data register for the generated memory access instruction;
- 14) Placing the generated instruction in the current VLIW or in a new empty VLIW if the placing in the current VLIW is impossible;
- 15) If i is store, with the specified probability of store-to-load bypassing save the memory access address \tilde{a} , the memory access type t and randomly choose the VLIW to bypass-load insertion based on the specified VLIW skipping range;
- 16) Repeating steps 4-16 until the specified number of memory access instructions is reached.

The register and memory random values initialization code is placed in the beginning of the first section of the test for each core. In this case, a prohibition is introduced on the generation of load instructions, only uninitialized memory fragments are written with store instructions of the maximum possible format.

The mode of generation of unaligned addresses is optional, its use is determined by the generation parameters of the test.

Code generation for each test section is performed independently using different memory maps. The code generator ensures the correctness of the transition between test sections. If the memory access types of a cache line are changed from coherent to non-coherent, the cache line should be flushed out of all the caches when switching between the test sections. Before starting a new test section, the code generator places the cores synchronization procedure and a sequence of cache line flush instructions for such cache lines. Changing the memory access types of a cache line from non-coherent to coherent does not violate determinism; therefore, no additional code is generated in this case. As a result of using this approach, the functionality of evicting the specified cache lines is additionally checked.

The final step of the test is checking whether the register and memory values match the reference values. The self-checking code is generated independently for each core. The memory map of the last section of the test is used as a description of requests. In this way, each core checks its own memory fragments. If a discrepancy between the test data and the reference data is detected, the test execution is terminated with the output of diagnostic information to the user (exit code or debug printing when executed on an FPGA-based prototype).

7. Automation of the memory maps formation

Memory maps are a large and detailed description of memory accesses. Manual compilation of memory maps requires high labor costs. In practice, describing memory requests in high detail, up to fragments, is not always necessary. In order to minimize the effort involved in creating memory maps for large memory areas with the same allowed memory access types, this process has been automated.

The user is given the opportunity to describe memory maps at a more general level: the description is made for arbitrary memory areas. In accordance with each memory area A_k the parameter set are placed:

$$A_k \mapsto \{I_k, F_k, M_k, C_k, S_k, r_k^{st/ld}, p_k^u\}, k = \overline{1, K} \quad (5)$$

where:

I_n – the n th set of memory access instruction types,

F_n – the n th set of memory access instruction formats,

M_n – the n th set of MAS,

C_n – the n th set of cores-owners,

S_n – the n th set of memory fragment sizes,

$r_n^{st/ld}$ – the n th store-to-load ratio,

p_n^u – the n th priority of use,

K – the number of described memory areas.

No restrictions are imposed. The resulting parameter set of overlapping memory areas is a union of the overlapping memory areas parameter sets. All conflicts will be resolved automatically. If the conflicts cannot be resolved, test generation ends with an error message.

Then the symbol A is used for the described memory areas set. The algorithm for generating a memory map consists of the following steps:

- 1) Splitting of the memory area set A into K fragments $\widetilde{A}_k \in A, k = \overline{1, K}$ of cache line size and alignment with the parameter set inheritance as $\{\widetilde{I}_k, \widetilde{F}_k, \widetilde{M}_k, \widetilde{C}_k, \widetilde{S}_k, r_k^{st/ld}, \widetilde{p}_k^u\}$;
- 2) Analysis of the MAS sets $\widetilde{M}_k, k = \overline{1, K}$ for simultaneous presence of coherent and non-coherent types of MAS in M_k sets;
- 3) Elimination of coherent or non-coherent types of MAS in a random way for each M_k set, which contains coherent and non-coherent MAS simultaneously;
- 4) Splitting the fragments $\widetilde{A}_k, k = \overline{1, K}$ into fragments $a_n^k \in \widetilde{A}_k, n = \overline{1, N(k)}$ of \widetilde{S}_k sizes in a random way, where $N(k)$ is a resulting number of memory fragments a_n^k in \widetilde{A}_k ;
- 5) Mapping the memory map parameter set $\{\widetilde{I}_k, \widetilde{F}_k, \widetilde{M}_k, \widetilde{c}_{nk}, r_k^{st/ld}, \widetilde{p}_k^u\}$ with random core-owner $\widetilde{c}_{nk} \in \widetilde{C}_k$ to each memory fragment a_n^k .

The result of this algorithm is a memory map that describes the memory requests for a test section.

The memory maps are generated independently for each section of the test using the presented algorithm. Ensuring true sharing of memory and variations in request types used in different sections of the test is achieved due to the randomness of the memory map generation algorithm. The code generator provides determinism support during transitions between the test sections.

To check the functionality of individual caches, memory areas are selected based on the organization of the target cache. Requests to lines with the same indexes lead to evictions from the cache memory, and the number of lines used must exceed the associativity of the target cache memory. To automate the compilation of memory areas aimed at creating evictions in various cache memory levels, a memory areas generator has been developed. The memory areas generator allows generating a memory area that corresponds to a given number of lines of the target cache level with the same indexes and different tags.

8. Results

A method for describing memory accesses using a memory map was developed. The test generator of self-checking Assembly tests for memory coherence verification of “Elbrus” architecture microprocessors implementing the memory accesses described in the memory maps was developed in C++.

The memory areas generator and the memory maps generator have been developed for the convenience of creating memory maps that are used for checking the functionality of individual cache memories. Automating the formation of memory maps allows reducing the volume and complexity of describing the memory maps. In this case, the user can simultaneously use an arbitrary number of memory areas generators and manually specify memory areas. Moreover, the parameters for generating memory maps can be set separately for each obtained memory area. The implementation of the described tools for automating generation of memory maps does not lead to the loss of the ability to describe requests at the fragment level, but in some cases significantly simplifies the process of configuring the test generator.

Currently, the developed test generator is used for verification of RTL models and FPGA-based prototypes of developing microprocessors with general-purpose cores of “Elbrus” architecture version 6. As a result of using the test generator, 74 logical errors were detected in the following hardware units: L1D-cache, L2-cache, L3-cache, Translation Lookaside Buffer (TLB), Memory Access Unit (MAU), On-Chip Network (OCN), Home Memory Unit (HMU), Memory Controller (MC), EFUSE. In addition, 3 malfunctions were found in the hardware of the FPGA-based prototype. These malfunctions are not errors of the original design and are specific only to the implementation of the FPGA-based prototype. Logical errors were manifested in the following manner: data corruption, deadlock, RTL assertion failure (available only in RTL-simulation).

Conducting verification on the FPGA-based prototype allows using tests with a large number of the test sequence iterations (> 100) and, therefore, obtaining a variety of the test sequence execution dynamics, which is difficult to achieve with RTL-simulation. For this reason, some logical errors were found only during verification on the FPGA-based prototype. Detailed statistics on detected logical errors are presented in Table 1.

The plan for further development of the test generator consists of work in the following areas:

- Test scenarios development;
- Support for virtual addressing;
- Development of algorithms for verification of memory consistency.

Table 1. Statistics on detected logical errors

Unit	Number of logical errors	
	<i>Detected on the FPGA-based prototype</i>	<i>Total</i>
L1D-cache	2	19
L2-cache	1	10
L3-cache	2	10
TLB	0	5
MAU	0	13
OCN	0	4
HMU	0	6
MC	1	6
EFUSE	0	1

References / Список литературы

- [1]. Kim A.K., Perekatov V.I., Ermakov S.G. Microprocessors and computer systems of the Elbrus family. SPB., «Piter» Publishing house, 2013, 272 p. (in Russian) / Ким А.К., Перекатов В.И., Ермаков С.Г. Микропроцессоры и вычислительные комплексы семейства «Эльбрус». СПб., Издательство «Питер», 2013 г., 272 стр.

- [2]. Lebedev D., Petrochenkov M. Test environment for verification of multi-processor memory subsystem unit. *Trudy ISP RAN/Proc. ISP RAS*, vol. 31, issue 3, 2019. pp. 67-76. DOI: 10.15514/ISPRAS-2019-31(3)-6.
- [3]. Adir A., Almog E. et al. Genesys-Pro: Innovations in Test Program Generation for Functional Processor Verification. *IEEE Design & Test of Computers*, vol. 21, no. 2, 2004, pp. 84-93.
- [4]. Frolov P.V. Random System-Level Test Generation for Elbrus Architecture Microprocessors. *Voprosy radioelektroniki*, vol. 4, issue 3, 2014, pp. 38-46 (in Russian) / Фролов П.В. Генерация случайных тестов системного уровня для микропроцессоров с архитектурой «Эльбрус». *Вопросы радиоэлектроники*, сер. ЭВТ, том 4, вып. 3, 2014 г., стр 38-46.
- [5]. A.S. Tanenbaum, T. Austin. *Structured Computer Organization*. 6th ed. Pearson, 2012, 808 p.
- [6]. S.V. Yurlin, I.N. Bychkov. FPGA prototyping for functional verification of multi-core processors. *Problems of advanced micro- and nanoelectronic systems development (MES)*, no. 4, 2014, pp. 45-50 (In Russian) / С.В. Юрлин, И.Н. Бычков. Прототипирование на основе ПЛИС для верификации многоядерных микропроцессоров. *Проблемы разработки перспективных микро- и наноэлектронных систем (МЭС)*, no. 4, 2014 г., стр. 45-50.
- [7]. A.N. Meshkov, M.P. Ryzhov, V.A. Shmelev. The development of the verification tools of the Elbrus-2S microprocessor. *Voprosy radioelektroniki*, vol. 4, no. 3, 2014, pp. 5-17 (in Russian) / А.Н. Мешков, М.П. Рыжов, В.А. Шмелев. Развитие средств верификации микропроцессора «Эльбрус-2S». *Вопросы радиоэлектроники*, сер. ЭВТ, том 4, no. 3, 2014 г. стр. 5-17.
- [1]. K.L. Gurin, A.N. Meshkov et al. Memory architecture development in the Elbrus series computer models. *Voprosy radioelektroniki*, no. 3, 2010 (in Russian) / К.Л. Гурин, А.Н. Мешков и др. Развитие модели подсистемы памяти вычислительных комплексов серии «Эльбрус». *«Вопросы радиоэлектроники»*, сер. ЭВТ, вып. 3, 2010 г.
- [2]. V.A. Agafonov. Using a memory card in the generation of system tests of the microprocessor memory subsystem. In *Proc. of the 61st All-Russian Scientific Conference of the Moscow Institute of Physics and Technology*, 2018, p. 21 (in Russian) / В.А. Агафонов. Использование карты памяти при генерации системных тестов подсистемы памяти микропроцессора. *Труды 61-й Всероссийской научной конференции МФТИ*, 2018 г., стр. 21.

Information about authors / Информация об авторах

Vladimir Andreevich AGAFONOV – lead engineer of the sector of system-level verification ("verification and modeling" department at JSC MCST), postgraduate student at MIPT. Main research interests: functional hardware verification, test program generation methods.

Владимир Андреевич АГАФОНОВ – ведущий инженер сектора системной верификации АО МЦСТ, аспирант МФТИ. Сфера научных интересов: логическая верификация аппаратуры, функциональная верификация, методы генерации тестовых программ.

Pavel Viktorovich FROLOV – head of the sector of system-level verification ("verification and modeling" department at JSC MCST). Main research interests: functional hardware verification, dynamic verification and formal methods.

Павел Викторович ФРОЛОВ – начальник сектора системной верификации АО МЦСТ. Профессионально-научные интересы: логическая верификация аппаратуры, методы динамической верификации и формальные методы верификации ПО и цифровой аппаратуры.

Alexey Nikolaevich MESHKOV – Ph.D. in Technology, head of "verification and modeling" department at JSC MCST. Main research interests: system verification, functional verification, simulation of hardware, system level simulation.

Алексей Николаевич МЕШКОВ – кандидат технических наук, начальник отдела "Моделирование и верификация" в АО "МЦСТ". Сфера научных интересов: системная верификация, функциональная верификация, моделирование аппаратуры, симулятор системного уровня.

DOI: 10.15514/ISPRAS-2022-34(2)-2



An algorithm of test generation from functional specification using Open IE model and clustering

*K.S. Kobyshev, ORCID: 0000-0002-1120-9569 <kobyshev2.ks@edu.spbstu.ru>
S.A. Molodyakov, ORCID: 0000-0003-2191-9449 <molodyakov_sa@spbstu.ru>
Peter the Great Saint Petersburg Polytechnic University,
29, Politechnicheskaya st., St. Petersburg, 195251, Russia*

Abstract. Automated test coverage is a widespread practice in long-live software development projects **for now**. According to the test development approach, each automated test should reuse functions implemented in test framework. The provided research is aimed at improving the test framework development approach using natural language processing methods. The algorithm includes the following steps: preparation of test scenarios; transformation of scenario paragraphs to syntax tree using pretrained OpenIE model; test steps comparison with test framework interfaces using GloVe model; transformation of the given semantic tree to the Kotlin language code. The paper contains the description of a prototype of the system automatically generating Kotlin language tests from natural language specification.

Keywords: automatic test; computational linguistics; relation extraction; open information extraction; dependency tree parsing; natural language processing; clustering; E2E test; GloVe; Kotlin

For citation: Kobyshev K.S., Molodyakov S.A. An algorithm of test generation from functional specification using Open IE model and clustering. Trudy ISP RAN/Proc. ISP RAS, vol. 34, issue 2, 2022, pp. 17-24. DOI: 10.15514/ISPRAS-2022-34(2)-2

Алгоритм генерации тестов из функциональной спецификации с использованием модели Open IE и кластеризации

*K.C. Кобышев, ORCID: 0000-0002-1120-9569 <kobyshev2.ks@edu.spbstu.ru>
С.А. Молодяков, ORCID: 0000-0003-2191-9449 <molodyakov_sa@spbstu.ru>
Санкт-Петербургский Политехнический университет Петра Великого,
Россия, 195251, Санкт-Петербург, Политехническая ул., 29*

Аннотация. Автоматизированное тестовое покрытие на данный момент является широко распространенной практикой в долгосрочных проектах разработки программного обеспечения. Согласно подходу к разработке тестов, каждый автоматизированный тест должен повторно использовать функции, реализованные в тестовой среде. Представленное исследование направлено на совершенствование подхода к разработке тестовой среды с использованием методов обработки естественного языка. Алгоритм включает следующие этапы: подготовка тестовых сценариев; преобразование абзацев сценария в синтаксическое дерево с использованием предварительно обученной модели OpenIE; сравнение шагов тестирования с интерфейсами тестового фреймворка с использованием модели GloVe; преобразование заданного семантического дерева в код языка Kotlin. Статья содержит описание прототипа системы автоматической генерации языковых тестов Kotlin из спецификации на естественном языке.

Ключевые слова: автоматический тест; компьютерная лингвистика; извлечение отношений; извлечение открытой информации; разбор дерева зависимостей; обработка естественного языка; кластеризация; E2E-тест; GloVe; Kotlin

Для цитирования: Кобышев К.С., Молодяков С.А. Алгоритм генерации тестов из функциональной спецификации с использованием модели Open IE и кластеризации. Труды ИСП РАН, том 34, вып. 2, 2022 г., стр. 17-24. DOI: 10.15514/ISPRAS-2022-34(2)-2

1. Introduction

Automated test coverage is a widespread practice in long-live software development projects for now. The coverage can be implemented on different levels of testing pyramid: unit tests, integration tests, API (Application programming interface) tests, E2E (End-to-End) tests [1]. The practice of test coverage allows us to decrease complexity of code refactoring process, also tests can be used as primary code documentation according to the Test-Driven Development methodology [2].

Also, there is another popular approach of test framework development. This approach complements the automated test coverage approach. According to the test development approach, each automated test should reuse functions implemented in test framework [3]. Therefore, after implementation of test framework, we have a clear architecture of test infrastructure without code duplication and with reusable test steps and objects.

The provided research is aimed at improving the test framework development approach, at reducing labor costs of this approach using natural language processing methods. We consider the existing test automation approaches, define their shortcomings and analyze how these shortcomings can be addressed.

2. Problems of existing testing automation approaches

When a programming system is quite complex, usually, analysts prepare a document describing the system behavior called a functional specification. Usually, in case of complex and long living projects, the functionality should be delivered by short release cycles, a program build should be delivered immediately after functionality implementation. In this case it is necessary to check not only the new functionality, but also existing earlier. In other words, it is necessary to complete the automated regression testing in such cases. Consider the testing automation methods presented in Table 1 and define their disadvantages.

Table 1. Existing approaches characteristics

Approach	Classic	BDD	Formal verification methods	Neural Network training
Characteristic				
Test structuredness	-	++	++	--
Analyst participation	--	++	++	+
Source code complexity resistance	++	++	--	++
Reliability	+	+	++	--
Automation	--	--	+	++

According to the classic testing automation approach, analyst should prepare a functional specification that is used for automatic test preparation by QA engineers (Quality Assurance engineer). Automatic tests are prepared manually. This method forces analyst and QA engineers to work separately. Participation of analysts is minimal and interaction between analysts and QA engineers is done over the document – functional specification. Also, QA engineers are responsible of test framework structure support. This approach excludes the full automation of test preparation. The BDD approach (Behavior-Driven Development) is based on test framework interfaces preparation by analyst with using of domain-oriented language [4]. Analysts prepare structure of test framework and QA engineers implements the test framework. This approach allows to achieve the high level of test structuredness. This approach like the classic approach, excludes the full automation of test preparation.

A set of formal verification methods allows us to check completely the program correctness according to functional specification requirements, made with, for example, language of temporal

logic [5]. The performance of verification process significantly degrades with increasing of cyclomatic complexity of program source code. The formal verification process is a check of all possible program states, which can cause the “combinatorial explosion”. Therefore, the formal verification usually applied for prototype of program instead of the source program.

Also, there is an approach based on the training of neural network [6]. Authors proposed to train neural network by random input data for program and given from its output data. This approach does not take in account analyst participation and testing is based on already prepared program. But this approach cannot guarantee the reliability because it is impossible to make the completely correct trained neural network model. Also, it is impossible to continue the model training with new program changes.

So, the following problems were found out during the existing methods analysis:

- Chaotic state, low level of test structuredness.
- Analysts work separately from QA engineers, absence of correct unified understanding of expected system behavior. Their work can be done only through documents, functional specification, consisted of non-strict natural language texts.
- Low testing system performance when source code of checked program is complex.
- Absence of guarantee that automatic testing system is completely correct.
- A lot of manual work on preparation of test infrastructure.

Consider the proposed approach, and how this approach allows us to deal with these enumerated shortcomings.

3. Proposed test development automation approach

Consider the solution proposed in the current research schematically presented in fig. 1.

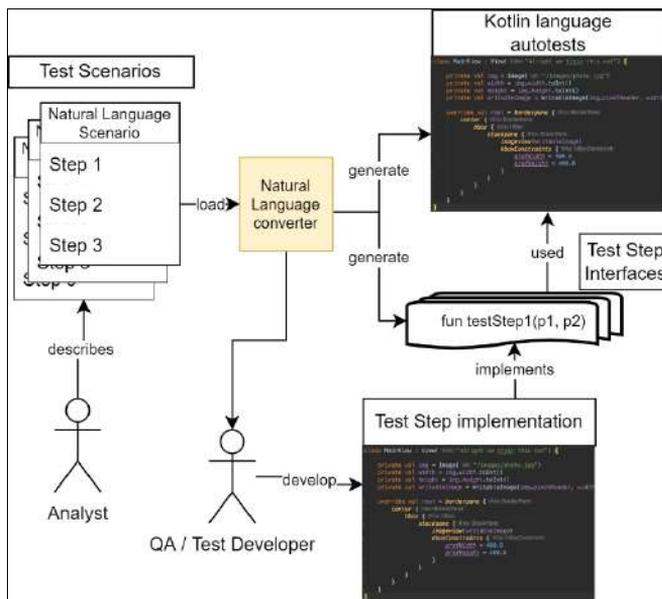


Fig. 1. The proposed solution for automatic test generation

We proposed to organize the development process by the following way:

- Analysts prepare natural language scenario set.
- Natural language test scenarios are transformed to interfaces of test steps by the proposed automatic software tool and to tests that are using interfaces from the generated test framework.
- QA engineer implements given test step interfaces on Kotlin programming language.

So, the main idea is to convert automatically non-strict natural language test scenarios to the stricter Kotlin programming language using existing natural language processing methods. Thanks to automated natural language processing of test scenarios, we achieve such advantages of the considered BDD test automation approaches as good test structuredness, consolidated understanding of system behavior between analysts and test developers, high reliability of testing system. Also, we decreased the manual work on test infrastructure preparation with automated test scenario processing. All, that test developer should do is implementation of test framework interfaces. The structure of tests and tests themselves will be prepared automatically.

Consider the proposed solution in detail, each step of the proposed test scenario processing algorithm.

4. Test generation algorithm steps

Consider the proposed algorithm steps schematically presented in fig. 2.

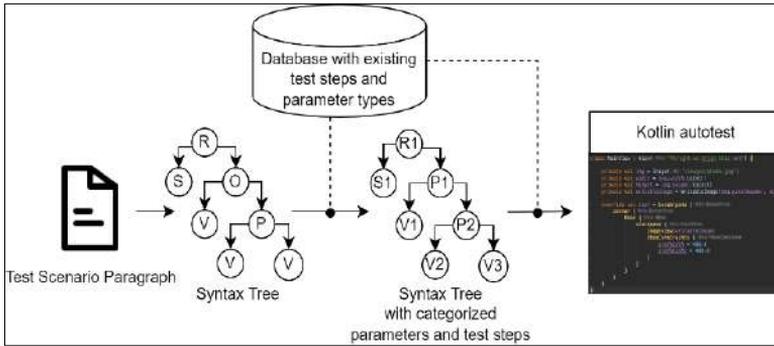


Fig. 2. Steps of the proposed test generation algorithm

The proposed method includes the following steps:

1. A test scenario name is taken as a test method name.
2. The test scenario is divided to sentences. Each sentence will be transformed to the one line of final code.
3. Each sentence is transformed to the syntax tree using the pretrained OpenIE model [7].
4. Test step, parameter group and separate parameter names are associated with test step, parameter group and parameter types using GloVe model [8, 9].
5. The given semantic tree is transformed to the Kotlin language code.

Consider steps 3, 4, 5 in detail.

5. Syntax tree preparation

OpenIE model is used to build the syntax tree from test scenario sentence [7]. Before OpenIE processing, the text data should be prepared by the following algorithms: tokenization [10], lemmatization [11], part-of-speech definition [12], building the dependency tree D [13]. Triplets are formed with using of OpenIE according to the expression (1), where s is a subject, R is a relation, o is an object:

$$T_i = s_i R_i o_i \tag{1}$$

In some cases, an object contains a set of several interconnected natural language words. The object can be presented in a form of a part of dependency tree, therefore according to the expression (2):

$$a_i \in D_i \tag{2}$$

This view allows us to present the object as a hierarchic structure of different parameters, that will make automatic tests more descriptive. The dependency tree can be presented by expressions (3) and (4), where P are tree nodes, and V are leaves. In other words, these leaves are values V of parameters P . And parameters P can include other parameters P or values V , so o can be presented in a form of hierarchic structure, so tests will contain trees of parameters P with values V :

$$o_i = P \cup V = (P_1, P_2, \dots, P_k) \cup (V_1, V_2, \dots, V_k) \quad (3)$$

$$\forall n, P_n = (P_x, P_{x+1}, \dots, P_y) \cup (V_m, V_{m+1}, \dots, V_l) \quad (4)$$

For now, when the current step is completed, found subjects, relationships, parameter sets, and values are not associated with any types. In the next step, they will be classified to form interfaces of test framework.

6. Test element type definition

As a result of the previous step, we got a hierarchically connected subjects s , relationships R , parameter sets P , values V . Each s, R, P, V is associated with some source natural language word or word set. Any natural language word can be presented in a form of coordinates vector in semantic space. Close s, R, P, V can be grouped to clusters associated with test framework interfaces.

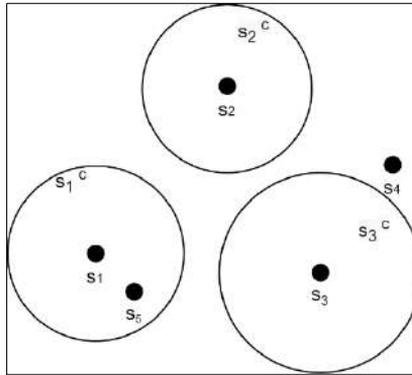


Fig. 3. Clustering of natural language words in semantic space

For now, there are many ways to get natural language word coordinates in semantic space. The most used for today models presenting word semantic coordinates are: RNNLM [14], word2vec [8], GloVe [15], fastText [16]. The GloVe model was used in the proposed method because this model takes in account in significant degree word cooccurrence frequency, that is important for our clustering.

As it was discussed earlier, we got a syntax tree D and a set (s, R, P, V) . Also, before clustering, we have a set $(s^{c0}, R^{c0}, P^{c0}, V^{c0})$, associated with a cluster set (s'_0, R'_0, P'_0, V'_0) found earlier on clustering of previous test scenario sentence words.

Each subset s, R, P, V is divided to clusters separately. Consider an example in the fig. 3 in two-dimensional space, when clusters s_1^c, s_2^c already found from previous test scenario sentences and for now we want to parse 3 remaining sentences and define their s, R, P, V types or clusters.

After parsing of three remaining sentences, as a result, algorithm extracts subjects s_3, s_4, s_5 from these three sentences. Clusters of these subjects are defined in the following way. So, we get a point in the two-dimensional semantic space. If there are no clusters in radius r from the given point, then the cluster with radius r will be placed at this point and the point will be a cluster center. If the point is in the other cluster zone, then this point will be associated with that cluster. If the point is not in cluster, but the r -radius circle from this point intersects with any cluster, then the point will be associated with the closest cluster.

We can see on the fig. 3 that clusters s_1^c, s_2^c were found at the beginning. Then algorithm accepted the point s_3 , that was associated with the cluster s_3^c , because the r -radius circle from this point is not

intersected with any existing r-radius clusters. The r-radius circle of point s_4 is intersected with cluster s_3^c , that is why it was associated with the cluster s_3^c . The point s_5 was associated with the cluster s_1^c because it was inside of the r-radius circle of this cluster.

The last remaining step is to get the Kotlin language code from the given semantic tree.

7. Semantic tree transformation to the Kotlin language code

The last step is to get the Kotlin language code from the given typed semantic tree. As a result, we will get an automatic test on the domain-oriented language and interfaces of the test framework. Consider transformation rules presented in the Table 2, where you can see examples of the parsed sentence in the “before” column and prepared automatic test code fragment in the “after” column.

Table 2. Transformation rules to the Kotlin language code

Transformation rule	Before	After
Subject	User paid free package User - subject	<pre>user { ...paid free package... }</pre>
Subject grouping	User paid free package. User got payment bill.	<pre>user { ...paid free package, got payment bill... }</pre>
Relationship	User paid free package	<pre>user { paid(...) }</pre>
Object	User paid free package	<pre>user { paid(Package(...)) }</pre>
Parameter	User paid free package	<pre>user { paid(Package(type=...)) }</pre>
Value	User paid free package	<pre>user { paid(Package(type=FREE)) }</pre>
Test Scenario	Payment flow: User paid free package. User got payment bill.	<pre>@Test fun paymentFlow() { user { paid(Package(type=FREE)) got(PaymentBill()) } }</pre>

The found subject is transformed to the lambda expression with context. QA engineer should implement the context class. If the same subject is appeared in two test scenario sentences, then those subject lambda expressions will be grouped to the one lambda expression. The found relationship is transformed to the method call, and that method should be implemented. Parameters are transformed to the class field names. Values are transformed to the primitive types of the Kotlin language or Strings. Then all code is wrapped to the test method having the name like the test scenario name.

8. Prototype of the proposed solution

A prototype of the proposed solution was implemented on Java language. The developed system uses a pretrained OpenIE model in a form of Maven package manager dependency called Stanford NLP. A pretrained GloVe model was used. This model was given from Wikipedia of 2014 year and

Gigaword text corpuses. The model contains 400 thousand words and their coordinates in 100-dimensional space and takes 822 Mb of memory. The GloVe model was stored and indexed in Mongo database. For now, the prototype gives true results for simple test scenarios, however, we found that it does not work correctly in some complex test scenarios including multiple words in subjects and relationships. Therefore, we need to investigate more and improve clustering stage of the proposed algorithm for now.

9. Conclusion

In the provided research we analyzed existing automated testing approaches and defined their disadvantages. After the analysis we proposed solution based on natural language processing of test scenarios and transformation of them to the Kotlin language autotests and test framework.

As a result of research, we implemented a prototype of the proposed algorithm on Java language in a form of Maven open-source library. The developed solution includes a pretrained OpenIE model from Stanford NLP library. A pretrained GloVe model was used to automate a search of test items categories. This model was given from Wikipedia of 2014 year and Gigaword text corpuses. The model contains 400 thousand words and their coordinates in 100-dimensional space and takes 822 Mb of memory. The GloVe model was stored and indexed in Mongo database.

For now, the prototype gives true results for simple test scenarios, however, we found that it does not work correctly in some complex test scenarios including multiple words in subjects and relationships. Therefore, we need to investigate more and improve clustering stage of the proposed algorithm for now.

References / Список литературы

- [1] N. Radziwill, G. Freeman Gr. Reframing the Test Pyramid for Digitally Transformed Organizations. *Software Quality Professional*, vol. 22, issue 4, 2020, pp. 18-25.
- [2] I. Karac and B. Turhan. What Do We (Really) Know about Test-Driven Development? *IEEE Software*, vol. 35 issue 4, 2018, pp. 81–85.
- [3] M.F. Fontoura. A systematic approach for framework development. PhD thesis, Pontifical Catholic University of Rio de Janeiro, 1999.
- [4] M. Irshad, R. Britto and K. Petersen. Adapting Behavior Driven Development (BDD) for large-scale software systems. *Journal of Systems and Software*, vol. 177, 2021, article no, 110944, 20 p.
- [5] W. Wasira. Existing Tools for Formal Verification and Formal Methods. 2020. DOI: 10.13140/RG.2.2.12162.22721.
- [6] A.D. Danilov and V.M. Mugatina. Verification and testing of complex software products based on neural network models. *VSTU Bulletin*, vol. 12, no. 6, 2016, pp. 62-67 (in Russian) / А.Д. Данилов, В.М. Мугатина. Верификация и тестирование сложных программных продуктов на основе нейросетевых моделей. *Вестник Воронежского государственного технического университета*, том 12, no. 6, 2016 г., стр. 62-67.
- [7] G. Angeli, M. Premkumar, C. Manning. Leveraging Linguistic Structure for Open Domain Information Extraction. In Proc. of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, 2015, pp. 344–354.
- [8] L. Ma and Y. Zhang. Using Word2Vec to process big text data. 2015 IEEE International Conference on Big Data, 2015, pp. 2895-2897.
- [9] A.D. Kovalev, I.V. Nikiforov, P.D. Drobincev. Automated approach to semantic search through software documentation based on Doc2Vec algorithm. *Information and control systems*, no. 1, 2021, pp. 17-27 (in Russian) / А.Д. Ковалев, И.В. Никифоров, П.Д. Дробинцев. Автоматизированный подход к семантическому поиску по программной документации на основе алгоритма Doc2Vec. *Информационно-управляющие системы*, no. 1, 2021 г., pp. 17-27.
- [10] R.M. Garcia-Teruel, H. Simon-Moreno. The digital tokenization of property rights. A comparative perspective. *Computer Law & Security Review*, vol. 41, issue 2, 2021, pp. 1-16.
- [11] B. Vimala, E. Lloyd-Yemoh. Stemming and Lemmatization: A Comparison of Retrieval Performances. *Lecture Notes on Software Engineering*, vol. 2, no. 3, 2014, pp. 262-267.

- [12] S. Chotirat, P. Meesad. Part-of-Speech tagging enhancement to natural language processing for Thai wh-question classification with deep learning. *Heliyon*, vol. 7, issue 10, 2020, article no. e08216, 13 p.
- [13] R. Zmigrod, T. Vieira, R. Cotterell. On Finding the K-best Non-Projective Dependency Trees. In Proc. of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, 2021, pp. 1324-1337.
- [14] G. Lecorve, P. Motlicek. Conversion of Recurrent Neural Network Language Models to Weighted Finite State Transducers for Automatic Speech Recognition. In Proc. of the 13th Annual Conference of the International Speech Communication Association, 2012, 4 p.
- [15] J. Pennington, R. Socher, C. Manning. Glove: Global Vectors for Word Representation. In Proc. of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014, pp. 1532-1543.
- [16] I.N. Khasanah. Sentiment Classification Using Fast Text Embedding and Deep Learning Model. *Procedia Computer Science*, vol. 189, 2021, pp. 343-350.

Information about authors / Информация об авторах

Kirill Sergeevich KOBYSHEV – postgraduate student at High School of Software Engineering. Research interests: computational linguistics, natural language processing.

Кирилл Сергеевич КОБЫШЕВ – аспирант Высшей школы программной инженерии. Область научных интересов: компьютерная лингвистика, обработка естественного языка.

Sergey Aleksandrovich MOLODYAKOV – Doctor of Technical Sciences, Professor of High School of Software Engineering. Research interests: image recognition, digital signal processing, video processors.

Сергей Александрович МОЛОДЯКОВ – доктор технических наук, профессор Высшей школы программной инженерии. Область научных интересов: распознавание изображений, цифровая обработка сигналов, видеопроцессоры.



Поиск ошибок в бинарном коде методами динамической символьной интерпретации

¹А.В. Вишняков, ORCID: 0000-0003-1819-220X <vishnya@ispras.ru>

^{1,2}И.А. Кобрин, ORCID: 0000-0002-6035-0577 <kobrineli@ispras.ru>

¹А.Н. Федотов, ORCID: 0000-0002-8838-471X <fedotoff@ispras.ru>

¹Институт системного программирования им. В.П. Иванникова РАН,
109004, Россия, г. Москва, ул. А. Солженицына, д. 25

²Московский государственный университет имени М.В. Ломоносова,
119991, Россия, Москва, Ленинские горы, д. 1.

Аннотация. Современное программное обеспечение стремительно развивается, принося новые ошибки, и все больше компаний следуют безопасному циклу разработки ПО. Одними из самых популярных средств для поддержки безопасного цикла разработки являются фаззинг и символьная интерпретация программ, позволяющие автоматически тестировать программу и искать в ней ошибки. Гибридный фаззинг – наиболее эффективный подход, который заключается в применении комбинации этих двух техник, при котором две техники работают совместно. Другим способом искать программные ошибки является символьная интерпретация с использованием предикатов безопасности – условий на входные данные, при выполнении которых будет проявлена ошибка. В этой работе мы предлагаем метод автоматизированного поиска ошибок с помощью динамической символьной интерпретации, совмещающий гибридный фаззинг с проверкой предикатов безопасности. Гибридный фаззинг требуется для получения большого количества различных входных данных, а ошибки работы с памятью и неопределенного поведения в программах ищут предикаты безопасности, которые позволяют находить ошибки деления на ноль, выхода за границы массива, целочисленного переполнения и другие. Результаты работы предикатов безопасности верифицируются с помощью санитайзеров, чтобы отбросить ложно положительные срабатывания. В результате практического применения предложенного метода к программам с открытым исходным кодом было найдено 11 различных новых ошибок в 5 разных проектах.

Ключевые слова: динамическая символьная интерпретация; DSE; фаззинг; предикат безопасности; автоматическое обнаружение ошибок; безопасный цикл разработки; SDL; бинарный код; санитайзер; ошибка; CWE

Для цитирования: Вишняков А.В., Кобрин И.А., Федотов А.Н. Поиск ошибок в бинарном коде методами динамической символьной интерпретации. Труды ИСП РАН, том 34, вып. 2, 2022 г., стр. 25-42. DOI: 10.15514/ISPRAS-2022-34(2)-3

Благодарности. Работа поддержана грантом РФФИ № 20-07-00921 А.

Error detection in binary code with dynamic symbolic execution

¹A.V. Vishnyakov, ORCID: 0000-0003-1819-220X <vishnya@ispras.ru>

^{1,2}E.A. Kobrin, ORCID: 0000-0002-6035-0577 <kobrineli@ispras.ru>

¹A.N. Fedotov, ORCID: 0000-0002-8838-471X <fedotoff@ispras.ru>

¹*Ivannikov Institute for System Programming of the RAS,
25, Alexander Solzhenitsyn Str., Moscow, 109004, Russia*

²*Lomonosov Moscow State University,
GSP-1, Leninskie Gory, Moscow, 119991, Russian Federation*

Abstract. Modern software is rapidly developing, revealing new program errors. More and more companies follow security development lifecycle (SDL). Fuzzing and symbolic execution are among the most popular options for supporting SDL. They allow to automatically test programs and find errors. Hybrid fuzzing is one of the most effective ways to test programs, which combines these two techniques. Checking security predicates during symbolic execution is an advanced technique, which focuses on solving extra constraints for input data to find an error and generate an input file to reproduce it. In this paper we propose a method for automatically detecting errors with the help of dynamic symbolic execution, combining hybrid fuzzing and checking security predicates. Firstly, we run hybrid fuzzing, which is required to increase number of corpora seeds. Then we minimize corpora. Thus, it would give the same coverage as the original corpora. After that we check security predicates on minimized corpora. Thus, security predicates allow to find errors like division by zero, out of bounds access, integer overflow, and more. Security predicates results are later verified with sanitizers to filter false positive results. As a result of applying the proposed method to different open source programs, we found 11 new different errors in 5 projects.

Keywords: dynamic symbolic execution; DSE; fuzzing; security predicate; automatic error detection; security development lifecycle; SDL; binary code; sanitizer; bug; error; CWE

For citation: Vishnyakov A.V., Kobrin E.A., Fedotov A.N. Error detection in binary code with dynamic symbolic execution. *Trudy ISP RAN/Proc. ISP RAS*, vol. 34, issue 2, 2022, pp. 25-42 (in Russian). DOI: 10.15514/ISPRAS-2022-34(2)-3

Acknowledgements. This work was supported by RFBR grant 20-07-00921 A.

1. Введение

Современное программное обеспечение стремительно развивается. Новый код неизбежно приносит с собой новые ошибки и уязвимости [1]. Всё больше и больше индустриальных компаний следуют безопасному циклу разработки [2-4] для улучшения качества программного обеспечения и защиты его от злонамеренных атак. Главной целью безопасного цикла разработки является поиск ошибок в разрабатываемом программном обеспечении, включающих в себя как простые ошибки, приводящие к неправильной работе программы в несущественных местах, так и те, которые приводят к аварийному завершению работы программы или открывают доступ для различных злонамеренных атак. В результате применения безопасного цикла разработки программное обеспечение становится безопаснее, надежнее и качественней.

Одной из наиболее распространенных технологий для поддержки безопасного цикла разработки остается фаззинг с обратной связью по покрытию [5-6]. Гибридный фаззинг извлекает выгоду из динамической символьной интерпретации [7-17], открывая сложные состояния программы, трудно достигаемые с помощью простого фаззинга.

Фаззинг представляет собой метод автоматического тестирования программ, при котором программе на вход подается множество наборов входных данных, после чего анализируется реакция программы и генерируются новые входные данные. Целью фаззинга является поиск входных данных, которые приведут выполнение программы к аварийному завершению, переполнению по памяти или зависанию.

Символьная интерпретация представляет собой метод автоматического тестирования, при котором происходит интерпретация программы, где конкретным значениям переменных, зависящих от входных данных, сопоставляются символьные переменные, принимающие произвольные значения. Анализируемый путь выполнения программы может быть описан предикатом безопасности – системой уравнений и неравенств, зависящих от символьных переменных, решение которой обеспечивает прохождение потока управления по тому же пути. Предикатом безопасности мы называем дополнительные условия на предикат пути, которые позволяют проявить ошибочную ситуацию в программе.

В данной работе мы описываем разработанный метод поиска ошибок работы с памятью и неопределенного поведения в программном обеспечении, использующий гибридный фаззинг программ и предикаты безопасности, и представляем результаты практического применения этого метода на программах с открытым исходным кодом.

2. Обзор

2.1 KLEE

KLEE – это инструмент для символьной интерпретации, способный автоматически генерировать тесты для достижения высокого покрытия и поиска ошибок на разнообразном наборе сложных программ [18]. Во время символьной интерпретации KLEE для потенциально опасных операций генерирует условные переходы, которые проверяют, существует ли какой-либо набор входных данных, который может привести к проявлению ошибки. Например, инструкция деления генерирует условный переход, проверяющий, может ли делитель быть равен нулю. Такие условные переходы будут работать так же, как и обычные условные переходы: при выполнимости предиката пути, содержащего такой условный переход, будет сгенерирован входной файл для воспроизведения ошибки и будет порожден новый процесс работы KLEE, который продолжит анализ пути с условием, обратным условию для проявления ошибки. Таким образом, даже если проверка успешна (то есть обнаружена ошибка), KLEE продолжает работу по пути, на котором ошибка не проявляется, добавляя отрицание условия проверки в качестве ограничения на путь.

2.2 Mayhem

Mayhem – это система для автоматического обнаружения программных ошибок в бинарных файлах [19]. Каждая ошибка, найденная с помощью Mayhem, сопровождается работающей эксплуатацией этой ошибки. Mayhem генерирует эксплойты для каждой возможной перезаписи счетчика инструкций, обычно вызываемой переполнением буфера. Когда Mayhem находит символьный адрес инструкции, он пытается сгенерировать эксплуатацию перехода по регистру. В таких случаях счетчик инструкций должен указывать на инструкции по типу `jmp eax`, а регистр `eax` должен указывать на место в памяти, где может быть размещен какой-то произвольный код. Такие условия конструируются в виде формулы, которая передается математическому решателю. Если решение существует, то ошибка может быть проэксплуатирована. Если решения не существует, то генерируется более простая формула, где счетчик инструкций должен указывать конкретно на место в памяти, где может быть размещен произвольный код.

Для форматных строк Mayhem проверяет, содержат ли аргументы для форматной строки или сама форматная строка символьные байты. Если так, то Mayhem пытается разместить в аргументе данные, которые приведут к перезаписи адреса возврата из формирующей функции.

2.3 Google Sanitizers

Google Sanitizers – это набор инструментов с открытым исходным кодом для динамического анализа кода [20], позволяющие обнаруживать различные ошибочные ситуации во время выполнения программы.

Реализация санитайзеров основана на инструментации программного обеспечения на этапе его компиляции. Например, при использовании `AddressSanitizer` различные манипуляции с памятью в процессе выполнения программы одновременно отображаются определенным способом в отдельной структуре данных, называемой теневой памятью, позволяющей следить за корректностью выполнения различных операций с памятью. При использовании `UndefinedBehaviorSanitizer` в код программы на этапе компиляции встраиваются различные проверки на целочисленные переполнения и другие ошибки неопределенного поведения.

Существует несколько санитайзеров: `AddressSanitizer`, `LeakSanitizer`, `ThreadSanitizer`, `UndefinedBehaviorSanitizer` и `MemorySanitizer`.

`AddressSanitizer` позволяет обнаруживать ошибки работы с памятью во время ее выполнения. Он способен обнаружить переполнение буфера на стеке, куче, а также переполнение глобальных буферов. Также он позволяет обнаруживать использования указателей после освобождения памяти, на которую они ссылаются, и обнаруживает двойные и некорректные освобождения памяти, а также другие ошибки работы с памятью.

`LeakSanitizer` позволяет обнаруживать утечки памяти. Для использования этого санитайзера отдельно не требуется инструментации программы, однако также он встроен в `AddressSanitizer`, что позволяет использовать эти два санитайзера совместно.

`ThreadSanitizer` позволяет обнаруживать ошибки гонки данных, которые происходят, когда разные потоки выполнения обращаются к одним и тем же областям памяти без синхронизации и где хотя бы одно из обращений является записью в эту область памяти.

`UndefinedBehaviorSanitizer` позволяет обнаруживать ошибки неопределенного поведения, являющиеся результатом выполнения операций с неспецифицированными семантиками, таких как деление на нуль, разыменованное нулевого указателя, использование неинициализированной нестатической переменной и т.п.

`MemorySanitizer` позволяет обнаруживать ошибки неинициализированного чтения из памяти. Этот санитайзер находит случаи, когда производится чтение из памяти, выделенной на стеке или в куче, до того, как в эту память было что-либо записано.

2.4 SAVIOR

SAVIOR – это фреймворк для гибридного тестирования программного обеспечения, нацеленный на поиск ошибок [21]. Работа SAVIOR основывается на двух механизмах: механизме распределения приоритетов, основанном на ошибках, и механизме верификации с учетом ошибок.

Суть механизма распределения приоритетов заключается в том, чтобы производить символьную интерпретацию не на всех наборах входных данных, а отдавать приоритет таким входным данным, которые с большей вероятностью приведут к обнаружению программных ошибок. До тестирования программного обеспечения SAVIOR анализирует исходный код и статически помечает потенциально опасные места в программе. Также SAVIOR высчитывает набор базовых блоков, достижимых от каждого условного перехода. Во время символической интерпретации SAVIOR считает приоритетными те наборы входных данных, которые смогут привести к посещению наибольшего числа важных условных переходов. Таким образом, SAVIOR ускоряет обнаружение новых ошибок в программах.

Суть механизма верификации заключается в следующем. Получая наборы входных данных в результате фаззинг-тестирования, SAVIOR выполняет программу на этих входных данных и для каждого ранее помеченного потенциально опасного места верифицирует

соответствующий ему предикат, проверяя его с учетом пути выполнения. Если предикат удовлетворим, то ошибка подтверждена. Это позволяет SAVIOR генерировать доказательство обнаружения ошибки в виде конкретных входных данных или ее несуществования на конкретном пути выполнения программы.

Пометка потенциально опасных мест производится с помощью Undefined Behavior Sanitizer. На основе инструментации от санитайзера SAVIOR инструментированы потенциально опасное места в программе предикатами, удобными для математического решателя, которые будут проверяться на этапе верификации. Таким образом, SAVIOR поддерживает предикаты для ошибок выхода за границы массива, битовых сдвигов на слишком большое значение, знакового и беззнакового целочисленного переполнения. С помощью статического анализа SAVIOR учитывает знаковость операндов и их размеры при составлении предикатов для дальнейшей проверки на этапе верификации.

2.5 ParmeSan

ParmeSan – инструмент для фаззинга с обратной связью по санитайзерам, который оптимизирован специально для покрытия ошибок [22]. Основной идеей является использование инструментации, производимой санитайзерами, для обеспечения эффективного механизма поиска интересных базовых блоков для обратной связи с фаззером. В отличие от фаззеров с обратной связью по покрытию, которые вслепую покрывают все базовые блоки программы, ParmeSan направлен на исследование путей выполнения программы, имеющих наибольший шанс проявления ошибок за кратчайшее время.

Первым делом ParmeSan исследует интересные цели из данного санитайзера, которым была проинструментирована программа, убирая неинтересные проверки. Затем динамически конструируется граф потока управления, чтобы направить фаззинг на конкретные цели. Сам процесс фаззинга основан на двух этапах: фаззинга для конструирования графа потока управления и фаззинга целевых точек программы, нацеленного на поиск ошибок в программе. Для ускорения работы второго этапа ParmeSan использует доступную из анализа потока данных информацию. Одной из основных идей, заложенных в работу ParmeSan, является то, как система направляет фаззинг по путям выполнения, ведущим к интересным блокам. ParmeSan в процессе работы строит граф потока управления. ParmeSan выполняет программу на исходных наборах входных данных и смотрит, до каких базовых блоков дошло выполнение программы. Основываясь на построенном графе потока управления, ParmeSan с помощью введенной авторами метрики вычисляет дистанцию от пройденных базовых блоков до целевых блоков, и на основе полученных значений метрики и графа потока управления направляет фаззинг так, чтобы как можно скорее дойти до целевых блоков, содержащих потенциальную ошибку. Для этого фаззер, представленный ParmeSan, при выборе следующего набора входных данных для мутации выбирает тот набор, при выполнении которого были посещены базовые блоки, имеющие наименьшую дистанцию с целевыми блоками, и повышает приоритет для инвертирования тех условных переходов, которые наиболее близко расположены к целевым базовым блокам.

Санитайзеры инструментуют программу двумя разными способами. Некоторые инструментации просто обновляют внутренние структуры данных (например, теневую память), другие инструментации используются, когда санитайзеры обнаруживают ошибку, используя условие условного перехода, которое взаимодействует с внутренними структурами данных или непосредственным состоянием программы. Цель ParmeSan – направить фаззинг в те точки программы, где санитайзер обновляет свои внутренние структуры данных, и решать предикаты, вставленные санитайзерами, выполнимость которых означает обнаружение ошибки, с помощью мутаций на основе анализа потока данных.

Таким образом, процесс работы ParmeSan состоит из трех фаз. Первая фаза – быстрое исследование путей исполнения с опорой на покрытие и этап трассировки, чтобы получить

граф потока управления. В течение первой фазы ParmeSan собирает трассы и пытается построить граф потока управления настолько точно, насколько это возможно. Вторая фаза – направленное исследование, чтобы достичь целевых базовых блоков. Третья фаза – генерация входных данных, проявляющих ошибку, – начинается, когда целевой блок был достигнут.

2.6 IntScope

IntScope – инструмент для обнаружения ошибок целочисленного переполнения при помощи символьной интерпретации программного обеспечения [23]. IntScope предлагает следующий подход для обнаружения ошибок целочисленного переполнения. При помощи символьной интерпретации анализируются инструкции, реализующие арифметические операции. Далее найденные места с потенциальной ошибкой целочисленного переполнения проверяются лениво, то есть арифметическая инструкция проверяется не сразу, а лишь когда помеченное символьное значение используется в чувствительных точках, таких как выделение памяти функциями `malloc`, `calloc`. IntScope предупреждает о потенциальной ошибке целочисленного переполнения, только если помеченное символьное значение, используемое в чувствительных местах, может переполниться.

Недостовверный источник – арифметическая инструкция в программе, где потенциально может произойти ошибка целочисленного переполнения, и один из операндов получен из помеченных данных. Помеченные данные могут быть получены из недоверенных источников входных данных, таких как сеть, файлы или опции командной строки. Функциями, с помощью которых могут быть получены помеченные данные, считаются `read`, `fread`, `recv`.

Сток ошибки – место в программе, в котором использование переполненного значения может привести к дальнейшим программным уязвимостям. В IntScope выделяются следующие стоки ошибок:

- выделение памяти (использование переполненного значения в аргументах таких функций, как `malloc`);
- доступ к памяти (переполненное значение использовано в качестве индекса или смещения при разыменовании адреса);
- условный переход (использование переполненного значения в условном переходе, который может привести к пропуску проверок на безопасность).

При достижении потенциального стока ошибки, использующего уже найденный источник ошибки, происходит соответствующая проверка. Если сток – аргумент функции выделения памяти, то значение проверяется как беззнаковое. Также информацию о знаковости символьного значения IntScope получает с помощью инструкций условных переходов, таких как `JG` (знаковый), `JA` (беззнаковый) и подобных.

2.7 Обнаружение целочисленных уязвимостей переполнения в двоичном коде программного обеспечения

В данной работе авторы предлагают свой подход для поиска ошибок целочисленного переполнения в бинарных файлах [24]. Подход основан на символьной интерпретации и двойном отображении памяти. Для этого строится усеченный граф потока управления, основанный на машинном коде. Слои графа проверяются на выполнение условий для обнаружения уязвимости.

Главная идея подхода – использовать символьную интерпретацию кода для конструирования условий, проявляющих уязвимость, на входные переменные. После эти условия нужно будет решить.

Представленный подход можно разделить на следующую последовательность фаз.

- **Построение графа потока управления.** Входной файл используется для построения графа потока управления тестируемой программы, в результате которого должны быть обнаружены вершины ввода и вывода. Вершина ввода отображает части кода, где происходит ввод данных, а вершина вывода отображает части кода, где происходит вызов функций выделения памяти.
- **Отсевивание неиспользованного пути в графе потока управления.** Все вершины, которые не были использованы в любом из путей от вершины ввода до вершины вывода должны быть удалены.
- **Построение символьного входа.** Входные данные и инициализированные ячейки памяти полагаются символьными. Каждая ячейка изначально отображается как пара, состоящая из символьного числа и символьного адреса.
- **Символьная эмуляция.** Каждый путь обрабатывается независимо. Обработка включает в себя символьную эмуляцию машинных инструкций. Она затрагивает обе части отображения ячеек (число и адрес). Символьные ячейки могут потерять один из этих параметров, если в процессе интерпретации становится понятно, является ли символьная переменная адресом или числом. Например, если переменная была разыменована, то она является адресом, тогда параметр числа отбрасывается.
- **Построение системы условий.** Система условий описывает доступность вершины вывода от вершины ввода. Ограничения добавляются в систему по мере исследования путей выполнения. Условия для проверки возможности ошибки используются в качестве последнего условия в системе.
- **Проверка существования решения системы.** Созданные условия проверяются с помощью автоматической системы доказательства теорем. Если система имеет решение, то программа уязвима, а сгенерированное решение является доказательством найденной ошибки.

Символьная эмуляция программы применяется к каждому пути в графе, приводящему к назначенной вершине вывода. Уравнения и неравенства на символьные переменные, из которых составляется система, которую позднее будет решать математический решатель, могут встречаться между вершинами графа и являться условиями операторов ветвления или условиями на переполнение в инструкциях, реализующих арифметические операции, таких как `add`, `sub`, `mul`, `imul`, `shl` и т.д.

Составленная система уравнений и неравенств на символьные переменные проверяется решателем последовательно по достижению вершины вывода, то есть вызова некоторой опасной функции, использование переполненного значения в которой может привести к серьезным последствиям. Если система решается, то это говорит об уязвимости вызываемой функции.

3. Символьные предикаты безопасности

Предикаты безопасности – это дополнительные условия на предикат пути, которые позволяют обнаружить ошибку в программе. Для построения предиката безопасности во время символьной интерпретации программы анализируются определенные ее инструкции и места, которые считаются опасными. Если такое место находится, то составляется предикат безопасности, представляющий собой уравнения и неравенства над символьными переменными и константными значениями. Затем предикат безопасности и предикат пути совместно образуют систему уравнений и неравенств, которая передается в качестве входных данных SMT-решателю [25]. Решение такой системы уравнений представляет собой набор входных данных, которые обеспечивают как прохождение потока управления по исследуемому пути, так и проявление конкретной программной ошибки. Таким образом,

запуск программы на этих входных данных воспроизводит найденные ошибки. Предикаты безопасности были разработаны в рамках инструмента Sydr [26].

Одним из разработанных предикатов безопасности является предикат безопасности для поиска ошибок целочисленного деления на нуль. Во время символьной интерпретации бинарного файла каждая символьная инструкция деления, такая как `div` или `idiv`, проверяется предикатом безопасности для поиска ошибки деления на нуль. Если делитель инструкции является символьным, то составляется уравнение на равенство делителя нулю. Затем получившееся уравнение конъюнктуруется с предикатом пути и получается соответствующий предикат безопасности, который далее проверяется по вышеописанной схеме.

3.1 Выход за границы массива

Выход за границы массива – одна из самых опасных и распространенных программных ошибок [27]. Для обнаружения таких ошибок мы строим предикат безопасности для каждого разыменования символьного адреса.

Для того, чтобы построить предикат безопасности для ошибки выхода за границы массива, сначала нужно определить границы массива [*lower_bound*; *upper_bound*). Определив границы массива, мы строим предикат безопасности, который возвращает истину, если символьный адрес выходит за границы массива. Далее получившиеся неравенства конкатенируются с предикатом пути и итоговый предикат проверяется на выполнимость. Если предикат выполним, то генерируются входные данные, приводящие к выходу за границу массива.

В случае, если Sydr не может обнаружить обе границы массива, составляется предикат безопасности для выхода за нижнюю границу массива, которую можно получить эвристическими методами. Например, в адресном выражении [`rdx + rax`], где `rax` является конкретным адресом базы, а `rdx` является символьным индексом, Sydr полагает конкретную часть регистра `rax` за нижнюю границу массива.

Для определения символьных границ адресов мы поддерживаем теньевую кучу и теневой стек. Sydr оборачивает все функции для работы с динамической памятью, такие как `malloc`, `calloc`, `realloc`, `free` и т.д., и при вызове таких функций обновляет теньевую кучу, которая содержит все границы выделенных в памяти буферов. Для каждой встреченной инструкции `call` Sydr кладет адрес, по которому располагается адрес возврата, на теневой стек; а для каждой инструкции возврата `ret` Sydr снимает элементы с теневого стека в соответствии с текущим значением указателя стека.

Соответственно, когда происходит разыменование символьного адреса, Sydr обнаруживает границы соответствующего буфера следующим образом. Если текущее конкретное значение адреса находится в теневой куче, то обе границы можно получить из нее. Если адрес указывает на стек, то ближайший адрес на теневом стеке, больший текущего адреса, будет считаться верхней границей массива. Нижняя граница вычисляется эвристически из конкретной части формулы символьного адреса. Главной идеей такого способа является суммирование конкретных частей символьной формулы.

Также мы оборачиваем функции копирования, такие как `memcpy`, `memmove`, `memset` и т.д., чтобы обнаруживать переполнения буфера, к которым может привести небезопасное использование таких функций. Если аргумент размера копирования является символьным, Sydr пытается сделать его таким, чтобы выйти за верхнюю границу.

Перед решением предиката безопасности мы составляем конъюнкцию с дополнительными условиями на предикат безопасности, которые позволяют обнаружить ошибку, которая вероятнее приведет к аварийному завершению программы, перезаписав адрес возврата при обращении к памяти или обращаясь к отрицательному адресу. Если такой более сильный

предикат не удовлетворим, Sydr возвращается к решению изначального предиката безопасности. Если при этом и адрес, и значение, которое будет записано по адресу, являются символическими, Sydr предупреждает о возможности ошибки write-what-where для такого выхода за границы.

3.2 Целочисленное переполнение

Ошибка целочисленного переполнения является одной из самых типичных программных ошибок [27], однако она достаточно часто встречается в бинарном коде. Sydr будет работать слишком долго, если он будет проверять каждое место в программе, где может возникнуть целочисленное переполнение. Более того, в некоторых ситуациях, таких как вычисление значения хэш-функции, целочисленное переполнение является корректным. Поэтому мы выделяем только критические части программы и для них проверяем предикаты безопасности. В отличие от других предикатов безопасности, мы отделяем сток ошибки от ее источника. Источник – это инструкция, где может произойти целочисленное переполнение (например, различные арифметические инструкции, такие как сложение или умножение). Сток – это место, в коде, где предшествующее целочисленное переполнение может привести к критической ошибке.

Мы решаем предикаты безопасности для поиска ошибок целочисленного переполнения для стоков ошибки, которые используют потенциально переполненное значение. Например, стоками могут быть условные переходы (изменение потока управления в зависимости от переполненного значения), адреса для обращения к памяти и аргументы функций. Особенно критично может быть использование переполненного значения в аргументах таких функций, как `malloc`, `memcpy` и т.п. [23, 24] Мы оборачиваем некоторые функции стандартной библиотеки и полагаем все их символичные аргументы потенциальными стоками. Для остальных функций мы проверяем первые три аргумента в соответствии со стандартным соглашением о вызовах.

При анализе конкретной инструкции мы проверяем, является ли она потенциальным источником, то есть является ли она арифметической и является ли хотя бы один из ее операндов символическим. Если так, то мы строим предикаты безопасности для беззнакового (флаг `CF`) и знакового (флаг `OF`) целочисленного переполнения. Для большинства арифметических инструкций предикат безопасности будет истинным, когда соответствующий флаг выставлен в единицу. Исключением являются инструкции битовых сдвигов, для которых мы составляем предикат безопасности отдельным способом так, чтобы он соответствовал целочисленному переполнению в таких случаях.

Затем мы проверяем, задействовано ли потенциально переполненное значение, то есть источник ошибки, в вычислении стока. Для этого мы проверяем, является ли абстрактное синтаксическое дерево источника ребенком абстрактного синтаксического дерева стока. Если так, то вычисление стока содержит потенциально переполненное значение.

Далее мы узнаем знаковость арифметической операции с помощью обратного слайсинга [12]. Мы проверяем инструкции условных переходов, начиная с последнего в текущем предикате пути, пока какая-либо такая инструкция не сможет нам дать информацию о знаковости какого-либо из операндов (например, инструкция `JL` говорит о том, что вычисления знаковые) [23]. Также мы можем узнать знаковость вычислений с помощью реализованных нами семантик функций вида `strto*l` [26].

Кроме того, мы составляем дополнительные условия для функций выделения памяти, таких как `malloc`, `calloc`, и функций копирования, таких как `memcpy`, так как переполнение в таких функциях может привести к более серьезным последствиям. Для функций выделения памяти мы составляем дополнительное условие, чтобы значение, получившееся в результате переполнения, оказалось меньше чем значение, на котором происходило конкретное выполнение, но при этом не равнялось нулю, что позволяет переполнить размер выделяемой

памяти и при этом ее успешно выделить (если переполненное число будет больше количества памяти на машине, то `malloc` вернет нуль). Для функций копирования мы хотим получить переполненное значение больше чем то, что было при конкретном выполнении программы. Если предикат с дополнительными условиями не удовлетворим, мы проверяем изначальный предикат безопасности.

Также мы обрабатываем случай, когда арифметическая операция производится с типами, размер которых меньше чем `sizeof(int)`. В таком случае в силу расширения целочисленных типов данных при вычислениях оригинальная инструкция не может привести к переполнению, поэтому мы самостоятельно строим новые предикаты безопасности для целочисленного переполнения с корректным размером операндов.

4. Метод автоматизированного поиска ошибок символьными предикатами безопасности

Для применения символьных предикатов безопасности был разработан следующий метод автоматизированного поиска ошибок. Сначала производится гибридный фаззинг выбранного проекта, при котором совместно работают фаззер и инструмент динамической символьной интерпретации Sydr [12]. После этого производится минимизация корпуса входных данных, полученных в результате гибридного фаззинга. Затем идет этап проверки предикатов безопасности на всех файлах, полученных после минимизации корпуса. Результаты предикатов безопасности верифицируются с помощью исполняемого файла, собранного с санитайзерами. Верифицированные результаты оцениваются человеком на критичность и корректность. Также возможна автоматизированная дедупликация, кластеризация и оценка критичности аварийных завершений с помощью Casr [28], если таковые были найдены на каких-либо наборах входных данных.

Для фаззинга мы использовали libFuzzer [5]. Для подготовки к этапу гибридного фаззинга требуется собрать два исполняемых файла для каждой фаззинг-цели проекта: первый должен быть собран с санитайзерами и libFuzzer'ом, второй – без санитайзеров, но с отладочной информацией (он будет использоваться Sydr'ом). После подготовки начинается этап гибридного фаззинга, в результате которого получается корпус входных данных, полученных от фаззера и Sydr'a.

Во время работы предикатов безопасности анализируется лишь конкретный путь исполнения, поэтому для проверки как можно большего количества инструкций программы необходимо проверить как можно больше входных файлов. Однако в получившемся в результате работы этапа гибридного фаззинга корпусе может содержаться большое количество файлов, похожих друг на друга и покрывающих один и тот же код в целевом проекте. Поэтому перед началом этапа проверки предикатов безопасности необходимо минимизировать корпус, чтобы оставить в нем наиболее эффективное подмножество файлов, дающее такое же покрытие кода программы, но при этом содержащее намного меньше файлов, чем изначальный корпус.

После минимизации корпуса начинается этап проверки предикатов безопасности. Проверка предикатов безопасности запускается для каждого файла из минимизированного корпуса. Новые файлы, генерируемые предикатами безопасности, верифицируются с помощью исполняемого файла, собранного с санитайзерами. Если при запуске этого исполняемого файла санитайзеры выдают предупреждение для строки программы, аналогичной той, для которой был сгенерирован проверяемый набор входных данных, то результат считается верифицированным и сохраняется отдельно. Также, если запуск собранного с санитайзерами исполняемого файла на проверяемом наборе приводит к срабатыванию санитайзеров, отсутствовавшему при запуске этого исполняемого файла на оригинальном файле (т.е. файле, на котором проводилась символьная интерпретация и проверка предикатов безопасности),

результат также считается верифицированным. Отдельно проверяется, приводит ли проверяемый набор входных данных к аварийному завершению.

После этапа проверки предикатов безопасности стоит вручную проверить, какие именно были найдены ошибки. Зачастую найденные ошибки действительно приводят к срабатыванию санитайзеров, но с точки зрения логики работы программы могут ошибками не являться. Также, если в результате проверки предикатов безопасности были найдены наборы данных, приводящие к аварийному завершению, стоит провести их автоматический анализ и кластеризацию с помощью инструмента Casg [28], который позволяет получать подробный отчет об аварийных завершениях.

5. Результаты применения метода к проектам с открытым исходным кодом

Разработанный метод автоматизированного поиска ошибок был применен к ряду проектов с открытым исходным кодом. В результате применения разработанного метода в них были найдены новые ошибки.

5.1 FreeImage

FreeImage – это библиотека с открытым исходным кодом [29], которая поддерживает работу с изображениями различных форматов, таких как PNG, BMP, JPEG, TIFF и т.д. С помощью разработанного метода в коде FreeImage было найдено 5 ошибок целочисленного переполнения на этапе проверки предикатов безопасности [30].

```
unsigned off_head, off_setup, off_image, i, temp;
...
fseek(ifp, off_setup + 792, SEEK_SET);
```

Листинг 1. Беззнаковое целочисленное переполнение в `fseek` (BMP)

Listing 1. Unsigned integer overflow in `fseek` (BMP)

На листинге 1 приводится фрагмент кода, относящийся к обработке изображений формата BMP, в котором была найдена ошибка беззнакового целочисленного переполнения при вызове функции `fseek`, смещающей текущую позицию в файле. Так как в коде отсутствуют проверки введенных значений, выражение `off_setup + 792` может привести к переполнению, что в свою очередь может привести к неправильной позиции в файле, измененной с помощью функции `fseek`. Источником ошибки, определенным проверкой предикатов безопасности, является инструкция сложения, реализующая вычисление выражения `off_setup + 792`, а стоком ошибки является аргумент смещения в функции `fseek`.

```
fseek(ifp, offset + length - 4, SEEK_SET);
```

Листинг 2. Знаковое целочисленное переполнение в `fseek` (TIFF)

Listing 2. Signed integer overflow in `fseek` (TIFF)

Аналогичная ошибка представлена на листинге 2, где также была найдена ошибка целочисленного переполнения при вызове функции `fseek`, но уже в части кода, отвечающей за обработку изображений формата TIFF. Найденная ошибка целочисленного переполнения происходит в аргументе смещения функции `fseek` при вычислении выражения `offset + length - 4`.

На листинге 3 представлен фрагмент кода FreeImage, где в аргументе функции `parse_tiff` происходит неявное преобразование типа `long` к типу `int`, где переменная `thumb_offset` имеет тип `long`. Так как предикат безопасности для поиска ошибок целочисленного переполнения определяет размер операндов по размеру типа стока, являющегося аргументом

функции, то предикат безопасности посчитал эту ошибку как целочисленное переполнение, которое по факту является неявным преобразованием типа, где значение, получающееся из выражения `thumb_offset + 12` типа `long`, не может быть помещено в аргумент функции типа `int` меньшего размера.

```
int parse_tiff(int base);
...
parse_tiff(thumb_offset + 12);
```

Листинг 3. Неявное преобразование в `parse_tiff`
Listing 3. Implicit conversation in `parse_tiff`

```
if (*len * tagtype_dataunit_bytes[(*type <= LIBRAW_EXIFTAG_TYPE_IFD8)
? *type : 0] > 4)
fseek(ifp, get4() + base, SEEK_SET);
```

Листинг 4. Беззнаковое переполнение в условном переходе
Listing 4. Unsigned integer overflow in branch condition

На листинге 4 приводится фрагмент кода, где в условии оператора ветвления была найдена ошибка беззнакового целочисленного переполнения при умножении `*len` на `tagtype_dataunit_bytes[(*type <= LIBRAW_EXIFTAG_TYPE_IFD8) ? *type : 0]`. Инструкцией источника является умножение, а стоком – инструкция условного перехода. Были подобраны такие входные данные, что при целочисленном переполнении меняется поток управления, что приводит к ошибочной работе программы.

На листинге 5 приводится фрагмент кода, где вычисляется некоторая ширина. При ее вычислении может произойти ошибка беззнакового целочисленного переполнения. Значение позже используется в условных переходах и многих других местах, и найденная ошибка может привести к неправильной работе программы.

```
width = raw_width - left_margin - (get4() & 7);
```

Листинг 5. Беззнаковое целочисленное переполнение при вычислении ширины
Listing 5. Unsigned integer overflow in width computation

5.2 xInt

`xInt` – это библиотека с открытым исходным кодом для языка C++ для управления электронными таблицами и их чтения/записи из/в файлы формата XLSX [31]. С помощью разработанного метода в `xInt` была найдены две ошибки целочисленного переполнения [32, 33] и ошибка выхода за границы массива [34].

```
in_>seekg(static_cast<std::ptrdiff_t>(sector_data_start() +
sector_size() * static_cast<std::size_t>(id)));
std::vector<byte> sector(sector_size(), 0);
```

Листинг 6. Беззнаковое целочисленное переполнение в `xInt`
Listing 6. Unsigned integer overflow in `xInt`

На листинге 6 приведен фрагмент кода, где были найдены обе ошибки беззнакового целочисленного переполнения в аргументе функции `seekg` в выражении `sector_data_start() + sector_size() * static_cast<std::size_t>(id)` при умножении и сложении соответственно. Источником ошибки является вычисление приведенного выражения, а стоком – аргумент вызываемой функции, являющийся результатом вычислений. Сама ошибка не приводит к серьезным последствиям, однако

подобранные для воспроизведения переполнения значения могут привести к следующим исходам.

Был найден набор входных данных, при котором значения `sector_size()` и `static_cast<std::size_t>(id)` достаточно велики, чтобы при перемножении привести к переполнению. Так как значение `sector_size()` очень велико, то вызов конструктора `std::vector<byte> sector(sector_size(),0)` приводит к аварийному завершению работы программы из-за невозможности выделить слишком большое количество памяти. Также был найден набор входных данных, при котором значение `id` равняется `-1`, и при приведении этого значения к типу `size_t` получается очень большое значение. Это также приводит к переполнению в указанном выражении. Однако, так как значение `sector_size()` не было подобрано большим, то аварийного завершения программы при вызове конструктора на следующей строке не происходит. Но далее в том же файле исходного кода в функции `read_directory` происходит обращение на чтение из памяти: `entries_[static_cast<std::size_t>(current_entry_id)]`. При обращении в память в силу подобранных для переполнения значений происходит выход за границы массива и аварийное завершение.

```
compound_document::follow_chain(sector_id start,
const sector_chain &table)
{
    auto chain = sector_chain();
    auto current = start;
    while (current >= 0)
    {
        chain.push_back(current);
        current = table[static_cast<std::size_t>(current)];
    }
    return chain;
}
```

Листинг 7. Выход за границы массива в `xlnt`
Listing 7. Out of bounds access in `xlnt`

На листинге 7 приведен фрагмент кода, где была найдена ошибка выхода за границы буфера. Ошибка происходит при обращении по адресу на строке `current = table[static_cast<std::size_t>(current)]`. Значение `current` было подобрано равным `2147483647`, что приводит к обращению в память за границами буфера. Ошибка была найдена независимо как фаззером, так и предикатами безопасности.

```
int sign = 0;
uint32_t i = 0;
uint32_t seconds = 0;
for (*endptr = nptr; **endptr; (*endptr)++) {
    switch (**endptr) {
        ...
        case '9':
            i *= 10;
        ...
    }
}
```

Листинг 8. Беззнаковое целочисленное переполнение в `unbound`
Listing 8. Unsigned integer overflow in `unbound`

5.3 unbound

unbound – это проверяющий, рекурсивный, кэширующий распознаватель DNS [35]. В коде проекта unbound при помощи разработанного метода была найдена ошибка беззнакового целочисленного переполнения. Фрагмент кода, в котором была найдена ошибка, представлен на листинге 8.

Переполнение происходит в функции `sldns_str2period`, которая обрабатывает строку и переводит ее в некоторое значение времени. Переполнение происходит в переменной `i` внутри цикла обработки строки. Переполнение может привести к неправильной работе со строкой и вследствие к неправильному результату работы функции. Ошибка была исправлена автором проекта [36].

5.4 hdp

HDF – это библиотека, предоставляющая набор утилит для работы с файлами формата hdf, который используется для хранения научных данных [37]. Одной из таких утилит является `hdp`, которая обеспечивает быстрое получение основной информации из hdf-файлов. В ходе применения разработанного метода автоматизированного поиска ошибок в `hdp` была найдена ошибка целочисленного деления на ноль, приводящая к аварийной остановке работы утилиты [38].

```
int32 buf_size;
/* we are bounded above by VDATA_BUFFER_MAX */
buf_size = MIN(total_bytes, VDATA_BUFFER_MAX);
/* make sure there is at least room for one record in our buffer */
chunk = buf_size / hsize + 1;
```

Листинг 9. Деление на ноль в hdp

Listing 9. Division by zero in hdp

На листинге 9 приведен фрагмент кода, в котором происходит деление на ноль. Код приведен из функции `Vsread`, которая считывает данные в некоторый буфер. Приведенный фрагмент кода вычисляет, какое количество элементов можно считать за раз. Так как в коде отсутствует проверка на равенство делителя нулю, на этапе проверки предикатов безопасности подбирается значение `hsize`, равное -1, что приводит к ошибке деления на ноль.

5.5 miniz

miniz – это высокопроизводительная библиотека сжатия данных без потерь в одном исходном файле, реализующая стандарты формата сжатия данных `zlib` и `Deflate` [39]. Также `miniz` входит в зависимости `PyTorch` – популярного фреймворка для машинного обучения [40]. В результате применения разработанного метода тестирования в `miniz` была найдена ошибка целочисленного переполнения [41].

```
if (cdir_size < pZip->m_total_files * MZ_ZIP_CENTRAL_DIR_HEADER_SIZE)
    return mz_zip_set_error(pZip, MZ_ZIP_INVALID_HEADER_OR_CORRUPTED);
```

Листинг 10. Целочисленное переполнение в miniz

Listing 10. Integer overflow in miniz

На листинге 10 приведен фрагмент кода, в котором может произойти целочисленное переполнение. Приведенное условие проверяет заголовок файла на корректность, и в случае, если заголовок некорректен, возвращается ошибка. В результате применения разработанного метода удалось найти входные данные, которые приводят к ошибке целочисленного переполнения. Однако для условных переходов мы накладываем дополнительные условия на

предикат безопасности, чтобы при проявлении ошибки был изменен поток управления. Так как при конкретном выполнении проверка проходила успешно, то были подобраны такие входные данные, при которых происходит ошибка целочисленного переполнения и меняется поток управления, в результате чего происходил возврат из функции с сообщением об ошибке. Для такого случая мы поменяли дополнительные условия на предикат безопасности так, чтобы при проявлении ошибки поток управления сохранялся. В результате этого эксперимента мы получили такие входные данные, при которых заголовок файла является некорректным, но благодаря целочисленному переполнению булево выражение в условии принимает значение истины, и проверка проходит успешно.

6. Заключение

В данной работе был предложен метод поиска ошибок в бинарном коде методами динамической символической интерпретации, заключающийся в фаззинге для генерации корпуса входных данных и проверке предикатов безопасности на сгенерированных данных. С помощью разработанного метода можно находить ошибки целочисленного переполнения, выхода за границы массива, деления на нуль и другие. В результате практического применения разработанного метода к программам с открытым исходным кодом были найдены 11 различных новых ошибок в 5 разных проектах.

Список литературы / References

- [1] CWE – common weakness enumeration. URL: <https://cwe.mitre.org/>.
- [2] M. Howard and S. Lipner. The Trustworthy Computing Security Development Lifecycle. Microsoft, 2006. URL: <http://msdn.microsoft.com/en-us/library/ms995349.aspx>.
- [3] ISO/IEC 15408-3:2008. Information technology – Security techniques – Evaluation criteria for IT security – Part 3: Security assurance components.
- [4] ГОСТ Р 56939-2016: Защита информации. Разработка безопасного программного обеспечения. Общие требования. Национальный стандарт РФ, 2016. / GOST R 56939-2016: Information Protection. Secure Software Development. General Requirements. National Standard of Russian Federation, 2016 (in Russian).
- [5] K. Serebryany. Continuous fuzzing with libFuzzer and AddressSanitizer. In Proc. of the 2016 IEEE Cybersecurity Development (SecDev), 2016, p. 157.
- [6] A. Fioraldi, D. Maier et al. AFL++: combining incremental steps of fuzzing research. In Proc. of the 14th USENIX Workshop on Offensive Technologies (WOOT 20), 2020, 12 p.
- [7] J.C. King. Symbolic execution and program testing. Communications of the ACM, vol. 19, issue 7, 1976, pp. 385-394.
- [8] J. Salwan. Triton Under the Hood. 2015, URL: <https://shell-storm.org/blog/Triton-under-the-hood/#8>.
- [9] N. Stephens, J. Grosen et al. Driller: augmenting fuzzing through selective symbolic execution. In Proc. of the Network and Distributed System Security Symposium, volume 16 of number 2016, 16 p.
- [10] I. Yun, S. Lee et al. QSYM: a practical concolic execution engine tailored for hybrid fuzzing. In Proc. of the 27th USENIX Security Symposium, 2018, pp. 745–761.
- [11] R. Baldoni, E. Coppa et al. A survey of symbolic execution techniques. ACM Computing Surveys, vol. 51, issue 3, 2018, Article no. 50, 39 p.
- [12] A. Vishnyakov, A. Fedotov et al. SYDR: cutting edge dynamic symbolic execution. In Proc. of the 2020 Ivannikov ISPRAS Open Conference (ISPRAS), 2020, pp. 46-54.
- [13] S. Poeplau and A. Francillon. Symbolic execution with SymCC: don't interpret, compile! In Proc. of the 29th USENIX Security Symposium, 2020, pp. 181-198.
- [14] S. Poeplau and A. Francillon. SymQEMU: compilation-based symbolic execution for binaries. In Proc. of the 2021 Network and Distributed System Security Symposium, 2021, 18 p.
- [15] L. Borzacchiello, E. Coppa, and C. Demetrescu. FUZZOLIC: mixing fuzzing and concolic execution. Computers & Security, vol. 108, 2021, article no. 102368.
- [16] D. Kuts. Towards symbolic pointers reasoning in dynamic symbolic execution. In Proc. of the 2021 Ivannikov Memorial Workshop (IVMEM), 2021, pp. 42-49.
- [17] J. Chen, J. Wang et al. JIGSAW: Efficient and Scalable Path Constraints Fuzzing. In Proc. of the 2022 IEEE Symposium on Security and Privacy (SP), 2022, pp. 1531-1531.

- [18] C. Cadar, D. Dunbar, and D. R. Engler. KLEE: unassisted and automatic generation of high-coverage tests for complex systems programs. In Proc. of the 8th USENIX Conference on Operating Systems Design and Implementation, 2008, pp. 209-224.
- [19] S.K. Cha, T. Avgerinos et al. Unleashing Mayhem on binary code. In Proc. of the 2012 IEEE Symposium on Security and Privacy, 2012, pp. 380-394.
- [20] Google sanitizers. URL: <https://github.com/google/sanitizers>.
- [21] Y. Chen, P. Li et al. SAVIOR: towards bug-driven hybrid testing. In Proc. of the 2020 IEEE Symposium on Security and Privacy, 2020, pp. 1580-596.
- [22] S. Österlund, K. Razavi et al. ParmeSan: sanitizer-guided greybox fuzzing. In Proc. of the 29th USENIX Security Symposium, 2020, pp. 2289-2306.
- [23] T. Wang, T. Wei et al. IntScope: automatically detecting integer overflow vulnerability in x86 binary using symbolic execution. In Proc. of the Network and Distributed System Security Symposium, 2009, 14 p.
- [24] R. Demidov, A. Pechenkin, and P. Zegzhda. Integer overflow vulnerabilities detection in software binary code. In Proc. of the 10th International Conference on Security of Information and Networks, 2017, pp. 101-106.
- [25] A. Niemetz and M. Preiner. arXiv: 2006.01621, 2020, 2 p.
- [26] A. Vishnyakov, V. Logunova et al. Symbolic security predicates: hunt program weaknesses. In Proc. of the 2021 Ivannikov ISPRAS Open Conference (ISPRAS), 2021, pp. 76-85.
- [27] 2021 CWE top 25 most dangerous software weaknesses. URL: https://cwe.mitre.org/top25/archive/2021/2021_cwe_top25.html.
- [28] G. Savidov and A. Fedotov. Casr-Cluster: crash clustering for Linux applications. In Proc. of the 2021 Ivannikov ISPRAS Open Conference (ISPRAS), 2021, pp. 47-51.
- [29] FreeImage project. URL: <https://freeimage.sourceforge.io/>.
- [30] FreeImage: Integer overflow errors in differnet places. URL: <https://sourceforge.net/p/freeimage/bugs/347/>.
- [31] xlnet project. URL: <https://github.com/xfussell/xlnet>.
- [32] xlnet: Integer overflow in compound_document::read_sector. URL: <https://github.com/xfussell/xlnet/issues/616>.
- [33] xlnet: Integer Overflow in compound_document::read_sector leads to Heap Buffer Overflow. URL: <https://github.com/xfussell/xlnet/issues/626>.
- [34] xlnet: Segmentation fault in xlnet::detail::compound_document::follow_chain(). URL: <https://github.com/xfussell/xlnet/issues/595>.
- [35] Unbound project. URL: <https://github.com/NLnetLabs/unbound>.
- [36] Unbound: Integer Overflow in sldns_str2period function. URL: <https://github.com/NLnetLabs/unbound/issues/637>.
- [37] HDF4 library. URL: <https://support.hdfgroup.org/products/hdf4/whatishdf.html>.
- [38] Hdp from hdf4-tools division by zero. URL: <https://bugs.launchpad.net/ubuntu/+source/libhdf4/+bug/1915417>.
- [39] Miniz library. URL: <https://github.com/richgel999/miniz>.
- [40] PyTorch project. URL: <https://github.com/pytorch/pytorch>.
- [41] Miniz: Fix integer overflow in header corruption check. URL: <https://github.com/richgel999/miniz/pull/238>.

Информация об авторах / Information about authors

Алексей Вадимович ВИШНЯКОВ – младший научный сотрудник отдела компиляторных технологий, закончил бакалавриат и магистратуру ВМК МГУ в 2020 году. Сфера научных интересов: компьютерная безопасность, жизненный цикл безопасной разработки (SDL), анализ бинарного кода, символьная интерпретация, фаззинг, автоматическое обнаружение ошибок и компиляторы.

Alexey Vadimovich VISHNYAKOV works for the Compiler Technology Department, obtained BSc degree and M.D. in the Faculty of Computational Mathematics and Cybernetics at Lomonosov Moscow State University. Research interests: computer security, security development lifecycle (SDL), binary analysis, symbolic execution, fuzzing, automatic error detection, and compilers.

Илай Александрович КОБРИН работает в отделе компиляторных технологий в ИСП РАН, получает степень бакалавра ВМК МГУ. Сфера научных интересов: компьютерная

безопасность, анализ бинарного кода, символьная интерпретация, фаззинг, операционные системы.

Eli Aleksandrovich KOBRIN works for the Compiler Technology Department at ISP RAS. He is a BSc student in the Faculty of Computational Mathematics and Cybernetics at Lomonosov Moscow State University. Research interests: computer security, binary analysis, symbolic execution, fuzzing, operating systems.

Андрей Николаевич ФЕДОТОВ – старший научный сотрудник отдела компиляторных технологий, закончил НИЯУ МИФИ в 2013 году, кандидат технических наук с 2017 года. Сфера научных интересов: информационная безопасность, символьная интерпретация, оценка критичности ошибок, обратная инженерия, поиск ошибок, языки программирования, динамический анализ.

Andrey Nikolaevich FEDOTOV works for the Compiler Technology Department, graduated from National Research Nuclear University МЕРФИ (Moscow Engineering Physics Institute) in 2013. Obtained PhD degree in 2017. Research interests: information security, symbolic execution, error severity estimation, reverse engineering, error search, programming languages, dynamic analysis.

DOI: 10.15514/ISPRAS-2022-34(2)-4



Design Patterns for a Knowledge-Driven Analytical Platform

^{1,2} V.S. Zayakin, ORCID: 0000-0002-5236-7360 <vszayakin@yandex.ru>

¹ L.N. Lyadova, ORCID: 0000-0001-5643-747X <lnlyadova@gmail.com>

² E.A. Rabchevskiy, ORCID: 0000-0001-9700-5577 <e.rabchevskiy@seuslab.ru>

¹ HSE University,

38, Studencheskaya st., Perm, 614070, Russia

² SEUSLAB LLC,

111, Shosse Kosmonavtov (building 27), Perm, 614066, Russia

Abstract. The development and support of knowledge-based systems for experts in the field of social network analysis (SNA) is complicated because of the problems of viability maintenance that inevitably emerge in data intensive domains. Largely this is the case due to the properties of semi-structured objects and processes that are analyzed by data specialists using data mining techniques and others automated analytical tools. In order to be viable a modern knowledge-based analytical platform should be able to integrate heterogeneous information, present it to users in an understandable way and to support tools for functionality extensibility. In this paper we introduce an ontological approach to information integration and propose design patterns for developing analytical platform core functionality such as ontology repository management, domain-specific languages (DSLs) generation and source code round-trip synchronization with DSL-models.

Keywords: information integration; knowledge bases; databases; system viability; analytical platforms; open data; data analysis.

For citation: Zayakin V.S., Lyadova L.N., Rabchevskiy E.A. Design Patterns for a Knowledge-Driven Analytical Platform. Trudy ISP RAN/Proc. ISP RAS, vol. 34, issue 2, 2022, pp. 43-56. DOI: 10.15514/ISPRAS-2022-34(2)-4

Паттерны проектирования для аналитической платформы, основанной на знаниях

^{1,2} В. С. Заякин, ORCID: 0000-0002-5236-7360 <vszayakin@yandex.ru>

¹ Л. Н. Лядова, ORCID: 0000-0001-5643-747X <lnlyadova@gmail.com>

² Е. А. Рабчевский, ORCID: 0000-0001-9700-5577 <e.rabchevskiy@seuslab.ru>

¹ Национальный исследовательский университет «Высшая школа экономики»,
614070, Россия, г. Пермь, ул. Студенческая, д. 38

² ООО «СЕУСЛАБ»,

614066, Россия, г. Пермь, ш. Космонавтов, д. 111, к. 27

Аннотация. Разработка и эксплуатация систем, основанных на знаниях, осложняется рядом проблем обеспечения жизнеспособности, которые неизбежно возникают в областях с интенсивным использованием данных таких, например, как, анализ социальных сетей и медиа (Social Network Analysis, SNA). Во многом это связано со свойствами слабоструктурированных объектов и процессов, которые подвергаются анализу специалистами с привлечением методов автоматизированной обработки данных (data mining, машинное обучение, ВІ-анализ и пр.). Для того чтобы аналитическая платформа, основанная на знаниях, была жизнеспособной, она должна обеспечивать интеграцию разнородной информации, включать интерпретацию информации в понятном для разных категорий пользователей виде и поддерживать инструменты для расширения функциональности. В данной работе предложены онтологический подход к интеграции информации и паттерны проектирования для разработки ядра

аналитической платформы, отвечающего за управление репозиторием онтологий, генерацию предметно-ориентированных языков (Domain-Specific Languages, DSL) и синхронизацию программного кода с DSL-моделями.

Ключевые слова: интеграция информации; базы знаний; базы данных; жизнеспособность систем; аналитические платформы; открытые данные; анализ данных.

Для цитирования: Заякин В.С., Лядова Л.Н., Рабчевский Е.А. Паттерны проектирования для аналитической платформы, основанной на знаниях. Труды ИСП РАН, том 34, вып. 2, 2022 г., стр. 43-56. DOI: 10.15514/ISPRAS-2022-34(2)-4

1. Introduction

The development of knowledge-driven analytical systems in data-intensive domains (e. g. social network analysis) is inevitably accompanied by problems of maintaining viability of those systems. Such class of intelligent systems shares common properties of development and support processes that result in common design challenges.

Firstly, logical models of knowledge bases (KB) and databases (DB), used by intelligent system for modeling the analyzed objects and processes, as well as to describe the ways of their physical representation, are constantly expanding and being modified, which creates problems of information integration. To a large extent, this is due to the properties of semi-structured objects and processes being analyzed, as well as the intensive use (gathering, processing, generation, etc.) of data.

Secondly, a wide range of specialists are involved in the development of intelligent system such as domain experts, knowledge engineers, data analysts, software engineers, etc. Each of these groups of specialists may interpret the integrated information differently, depending on the domain context in which they operate. Thus, it is crucial for analytical platform to successfully solve problems of interpreting same data according to different tasks and domain models.

Thirdly, the analytical platforms that are used in data-intensive domains are required to be extensible and allow users to implement or specify algorithms for data processing (including the use of data mining and machine learning techniques) themselves. Considering that the majority of end users (in particular, domain experts) do not have programming skills needed to implement such functionality, this creates problems of platform's extensibility and requires providing users with domain-specific modeling tools.

For developing a consistent and comprehensive solution to the mentioned groups of problems in this paper we identify common design challenges that arise in the process of development of knowledge-driven analytical platforms in data-intensive domains. As a result, this research proposes a conceptual ontological approach to information integration based on rules as well as design patterns for knowledge-driven analytical platform core functionality based on OWL 2 models of its abstract syntax.

2. Requirements and design challenges

Tools for development of viable knowledge-driven systems should be able primarily to integrate heterogeneous information, support changes traceability in data and knowledge models and interpret information according to the domain model that is understandable to the data consumer [1]. However, the development of an analytical platform is not feasible for an ordinary developer, since it involves the development of methods for solving specialized problems and the design of domain models that could include fuzzy relationships, fuzzy terms, temporal and spatial knowledge. In this regard, the solution of such a complex problem should be focused primarily on the creation of core tools, invariant to the domains in which they are used and that could be used to implement base functionality of a platform [2].

The following common requirements can be applied to the development tools of knowledge-driven analytical platforms in data-intensive domains:

- 1) Ensuring extensibility of logical models when descriptions of new sources of the information, such as data sources and domain models are included. It also requires automatic checks of logical integrity of the integrated model.
- 2) Ensuring integration of subject data with results of their analysis when algorithms of automated data processing (e. g. data mining techniques) are applied.
- 3) Providing tools for independent interpretation of data and results of their processing according to different domain models.
- 4) Ensuring traceability of changes in metamodels to support the relevance of semantic annotation of stored information.
- 5) Providing tools for declarative specification of data processing as well as input/output data structures used to extract/write data from/into DBs before/after executing data processing algorithms.
- 6) Providing tools for integration of data and knowledge models with software modules which implement data processing algorithms.

Related work in the field of knowledge-driven analytical platforms development includes the development of approaches to information integration and data federation. The vast majority of existing approaches are based on ontologies [3]. These approaches combine ontological descriptions of integrated sources and the descriptions of mappings and transformations between ontologies and information sources into a multifaceted ontology [4, 5].

Existing approaches are specifically designed to independently model and integrate various aspects of an intelligent system, namely, data structures (databases, event logs, texts), domains, data processing tasks, domain-specific languages, etc. This approach also provides a straightforward extension of the integration model, allowing to embed a new ontology by describing axiomatic relations between its elements and elements of previously formed ontologies. The implementations of multifaceted ontology-based information integration are used to design data architecture for networked enterprises [6], integrate spatial DBs [7], construct or transform queries to distributed DBs [8, 9, 10], describe and form datasets for machine learning tasks [11].

Based on the analysis of modern tools for the development of knowledge-driven analytical platforms, following common design challenges could be stated:

- 1) Integration of information (data sources models, data processing problems descriptions, domain models) at the level of source metamodels which are described independently of each other. It allows for metamodels to be described by different teams or to be imported from the Web and reused in the process of system development and maintenance.
- 2) Automated interpretation of data based on formal description of logical constraints and domain rules. It allows to use single inference subsystem to restructure data according to different models which helps to deduplicate and/or reduce the volume of stored information.
- 3) DSL integration for specification of data processing modules for gathering, preprocessing, analyzing and interpreting data. It requires the development of DSM subsystem. In exchange this provides users with tools to extend the functionality of the platform independently of its developers.
- 4) Integration of problem solvers based on declarative specifications of platform modules. It allows to generate specified versions of scripts and applications and to call them by providing a declarative description of input data, and parametrized attributes.
- 5) Composition of data processing modules into pipelines based on input and output data structures descriptions. It allows to automatically infer in which sequence data processing modules can be called. This can be used to orchestrate data processing tasks and manage data flows by such systems as Apache NiFi.

3. Ontological approach to information integration

The problem of information integration appears to be central in knowledge-driven analytical platform development. Without solving this problem, it is problematic to provide users with data to analyze in a transparent and comprehensible way which creates difficulties for them when choosing analytical techniques, methods and tools for data to be processed. Considering that in data-intensive domains data models are iteratively evolving it is crucial to develop an approach to information integration which would serve as a framework for managing multiple sources description.

Though existing approaches [6-11] allow to model and integrate source descriptions, they are restricted in terms of managing different versions of the same models resulting in changes traceability difficulties. For this reason, we introduce an ontological approach to information integration which helps to overcome described problem.

The proposed approach to information integration helps to organize information that is used in the processes of applying methods of automated data analysis and interpreting the results of analysis using expert knowledge of domain (fig. 1).

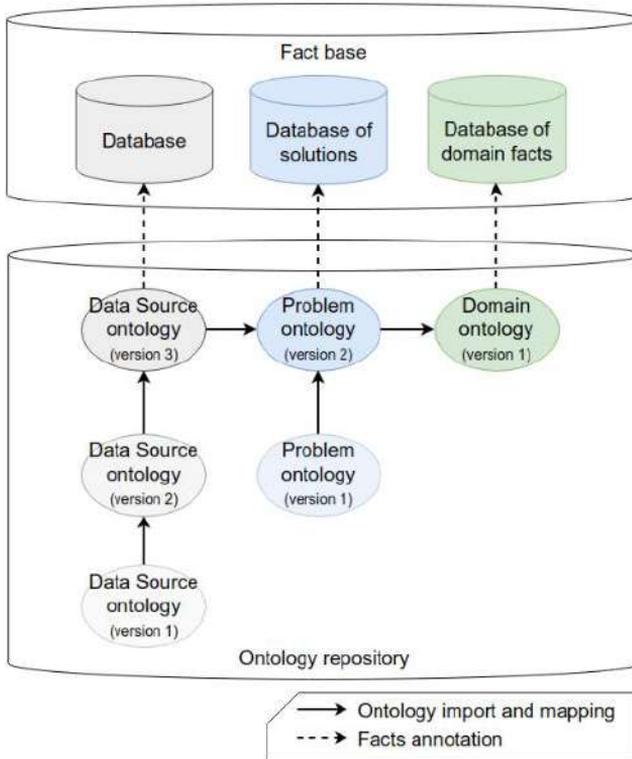


Fig. 1. The model of process of data analysis and interpretation

The process of data analysis and interpretation includes following steps:

- 1) A data analyst, based on the analytical problem being solved, determines the method for solving it.
- 2) A software engineer implements algorithms for selected method(s).
- 3) A domain expert and a knowledge engineer formulate rules and logical restrictions for interpreting the results of data analysis based on the possible outputs of the analysis methods.
- 4) A domain expert formulates new knowledge about the domain after interpreting the results.
- 5) Corresponding knowledge models then can be refined by a knowledge engineer by describing new patterns based on the extracted knowledge.

Knowledge base (fig. 2) is structurally divided into three groups of ontologies according to the type of modeled information:

- 1) Knowledge about data and data sources (*Data Source ontologies*). They may include information about data types, data storage formats, relations arity, attributes, etc.
- 2) Knowledge about data processing problems that use subject data to infer new facts and extract new knowledge (*Problem ontologies*). They may include information about the structure of input and output data for data processing procedures (e. g. data mining algorithms), links to externally executed scripts, the sequence of procedure calls, etc.
- 3) Domain knowledge (*Domain ontologies*). They may include concepts of the domain, as well as axiomatic statements that model the limitations of the domain.

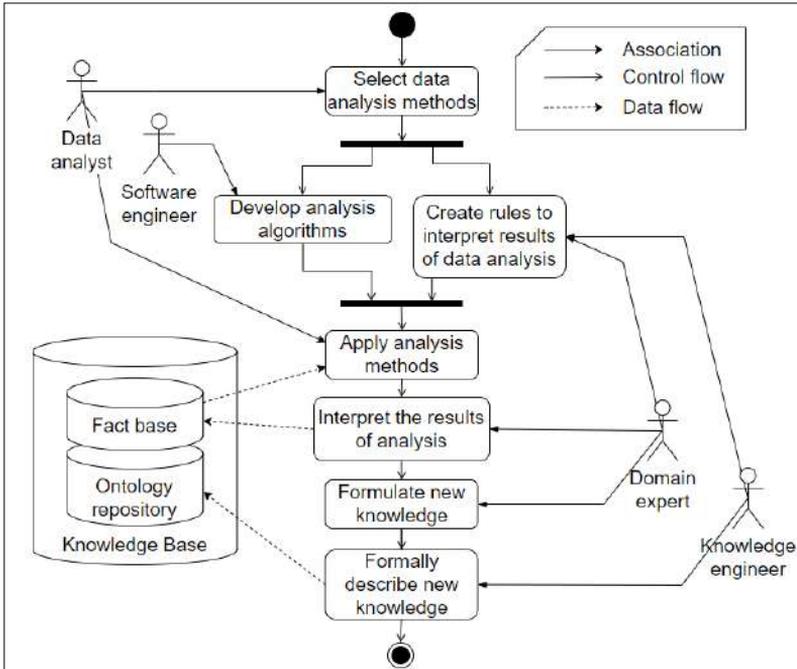


Fig. 2. The basic structure of knowledge base

The main idea of the approach is to model various information sources (using relevant terminology for context of the source) independently using ontologies. Then, elements of integrated ontologies are linked (or mapped) together using production rules or logical restrictions which allows automatic inference using semantic reasoners (e. g. Pellet, ELK, Hermit, etc.). There is a set of languages that could be used to formally describe ontologies and rules such as OWL 2, SWRL and RIF.

This approach can be called interpretive. Initially, ontology individuals are annotated according to the concepts in the ontology that describes some data source. When used to solve analytical problem, individuals are interpreted in terms of an ontology that models the conceptual scheme of this problem, according to the rules or logical restrictions. Similarly, the results of the analysis which are annotated using the concepts of the problem ontology, then are interpreted in the context of the ontology that models domain defined by experts and conceptualized by knowledge engineer.

In practice, the proposed approach allows to avoid data duplication. Data interpretation is determined while processing user query according to the context defined by ontologies. At the same time, it is possible to trace which facts and at what stage of processing was fetched or inferred. Facts gathered from a source (for example, from a specific social network) would be annotated using an ontology that describes this source. The results of data analysis (for example, the identifiers of duplicated objects) would be annotated using concrete problem ontology. This increases the

transparency of the data and allows to present the same information to different platform users according to the terminology they are familiar with.

Moreover, the proposed approach allows to manage and integrate different versions of ontologies to ensure traceability of changes in the same fashion as different types of ontologies which is not supported by existing approaches. We can simply copy an old version of ontology, modify it and define new rules and logical restrictions to bind elements of two versions of the same ontology.

4. Description of design patterns

According to [13], in order to develop a modern knowledge-driven analytical platform it is crucial to define its core functionality and to create tools which will allow to implement its base modules for gathering, preprocessing, analyzing and interpreting data. Then it will be possible to extend platform specifying new functionality by adapting its base modules.

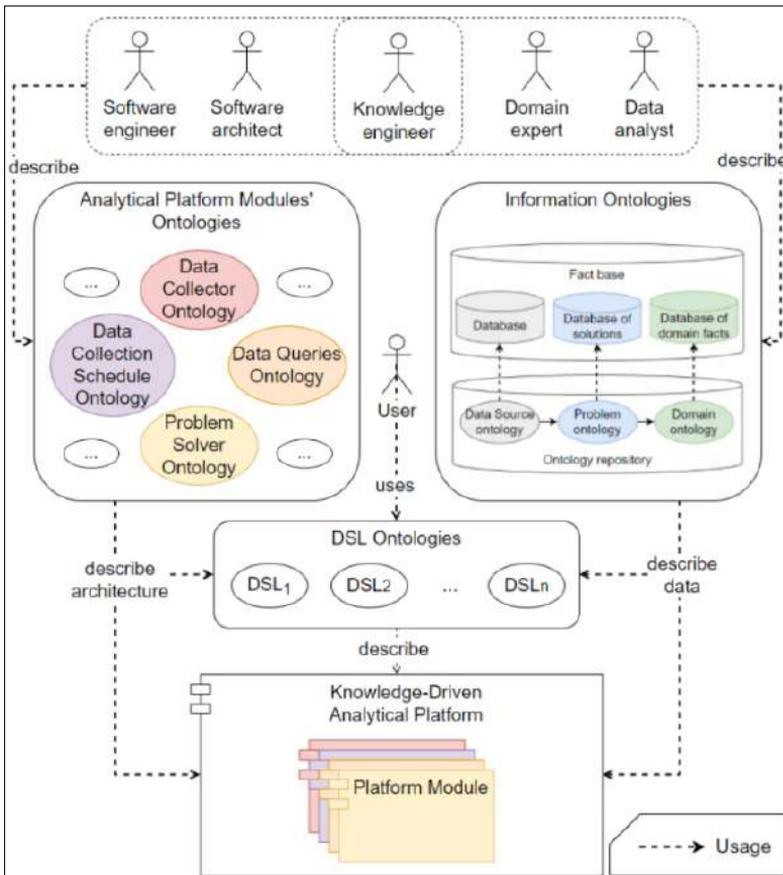


Fig. 3. An approach to analytical platform development

In this paper we propose (fig. 3) that new modules of an analytical platform are specified by users using a set of visual domain-specific languages generated on top of ontologies which are integrated in the knowledge base [5]. As a result, using generated DSLs, users who are not familiar with programming will be able to specify the modules and functions of the platform adapting existing functionality to their specialized problems [14]. At the same time, to support the uniformity of the resources description, DSLs themselves can also be represented in ontologies.

An approach to information integration, in that regard, is used to describe data structures of inputs and outputs that could be queried from fact base and passed into platform modules instances.

Besides, analytical platform modules' ontologies form base functionality of concrete analytical platform (for instance, platform for social network analysis) to be implemented using core functionality of a platform and could import, reuse and extend integrated information ontologies as well.

In order to execute approach to analytical platform development core functionality should allow to:

- manage ontologies;
- generate DSLs and visual editors for them;
- synchronize DSL-models with source code of the platform modules;
- execute problem solvers;
- manage problem solvers pipelines.

Thus, it is important to develop design patterns for the common core of different analytical platforms which could be developed using this approach. The specification of OWL 2 language [12] is used for this purpose. It contains the set of UML class diagrams and allows to extend them introducing new elements (which are highlighted in bold) to form the patterns.

4.1 Pattern for integrating ontologies

The pattern for integrating ontologies (Fig. 4) describes the taxonomy of integrated sources descriptions, as well as the types of logical restrictions that can be used to link elements of several ontologies.

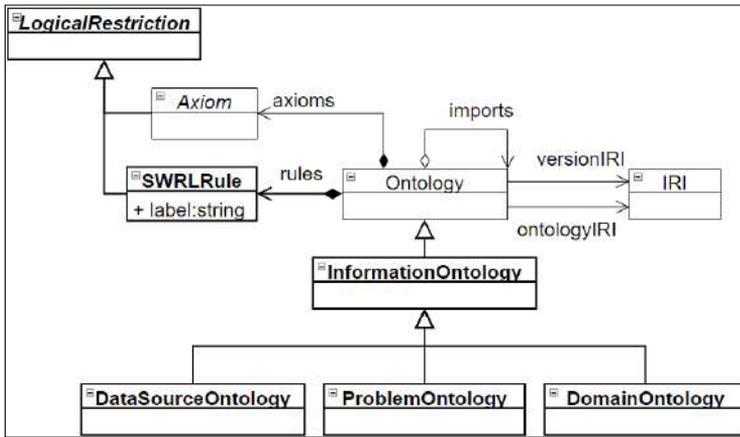


Fig. 4. The model of the pattern for integrating ontologies

Ontology which describes an integrated source of information is modeled by InformationOntology class which has three subclasses (DataSourceOntology, ProblemOntology, DomainOntology) according to every type of ontologies in the knowledge base. In order to identify ontology and all its versions IRIs are used.

Elements of ontologies (classes, object properties, datatype properties, etc.) are linked using either SWRLRules or Axioms that could be modeled using OWL 2 expressive capabilities. SWRLRules are one of the ways to define ontological mappings and logical restrictions within ontology. Just like axioms, they can be interpreted by reasoners (e.g., Pellet) to obtain new facts based on the information that gathered from data sources and stored in the fact base.

4.2 Pattern for ontology-based metamodeling

The pattern for ontology-based metamodeling (Fig. 5) is intended to be able to create metamodels are defined in the form of ontologies for describing DSLs and input/output data structures for problem solvers on top of integrated in knowledge base ontologies.

For each Ontology could be defined SubOntology that specify it. It is used to choose elements of the specified ontology that will form a model of an integrated DSL. It is constrained that SubOntology only can specify existing elements on an ontology, which is why elements added into SubOntology can only be SubEntities (e. g. subclasses and subproperties).

AssertionSubOntology which is the subtype of SubOntology is also used to specify ontologies but only by adding individuals and assertions involving them. It allows to ensure that new elements and axioms in AssertionSubOntology do not change the structure of metamodel that is defined by specified ontology. AssertionSubOntologies are supposed to be used to define concrete models according to the DSL and specify data structures of inputs and outputs for platform modules.

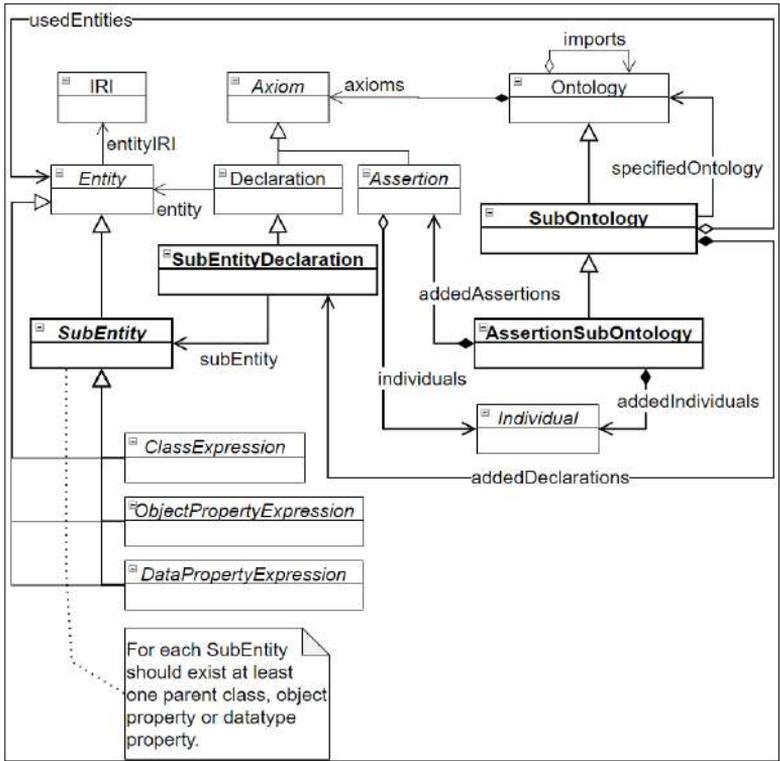


Fig. 5. The model of the pattern for ontology-based metamodeling

4.3 Pattern for integrating DSLs

The pattern for integrating DSLs (fig. 6) specify the ontological metamodeling pattern for generating and integrating DSLs to create languages to specify data processing modules in order to formally describe and manage the architecture of the analytical platform.

Classes and properties of ontologies that are specified by SubOntology can be used to form DSLMetaModel, and DSLConcreteModel is used to define models using DSL. DSLConcreteModel is a subclass of AssertionSubOntology, thus, added elements of DSL-models are individuals. The definition of DSLEntities as ontology classes and properties allows integrating them for inference of new facts which could be then automatically conceptualized in terms of InformationOntologies.

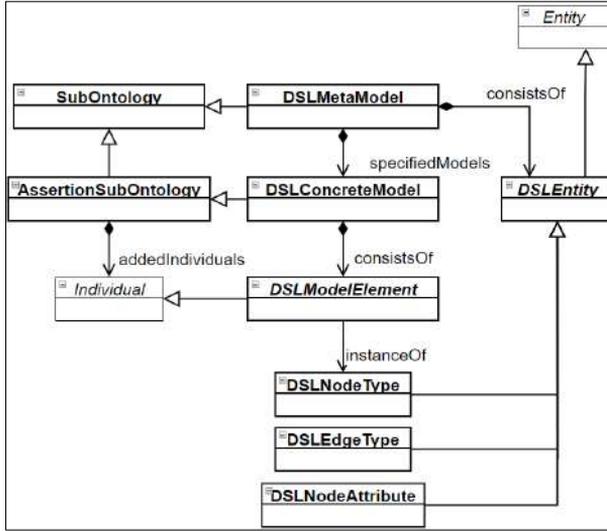


Fig. 6. The model of the pattern for integrating DSLs

4.4 Pattern for integrating problem solvers

The pattern for integrating problem solvers (Fig. 7) describes the structure of entities that are used to implement data processing modules of a platform using generated DSLs. The implementation of this pattern requires coordinating the processes of designing platform modules using DSL, implementing algorithms for problem solvers, and modeling the structure of input and output data for the problems to be solved by platform modules.

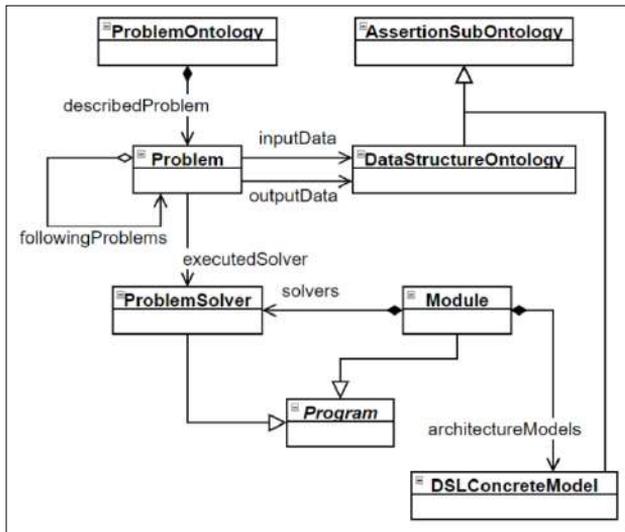


Fig. 7. The model of the pattern for integrating problem solvers

A set of DSLConcreteModels is used to describe architecture of a platform’s Modules which are interpreted by platform’s core. Every Module includes ProblemSolvers that are executed by the core. Each ProblemSolver are linked to the Problem description which is defined by some ProblemOntology. DataStructureOntology specifies input and output data structures of a Problem which allows inferring if one Problem can be solved following another one when defining a data processing pipeline.

5. Functionality of analytical platform core

It is assumed that the developed design patterns can be used to develop the analytical platform core. In order to demonstrate the principal possibility of creating such subsystem, in this section we describe the functionality of the core.

We propose that the knowledge engineer is responsible for managing the ontology repository (Fig. 8). He can create ontologies in a form of files corresponding to the OWL 2 with the assignment of a URI for the ontology and all its elements (classes, properties, etc.). Once created, the ontology file should be available for opening externally in an ontology editor (e.g., Protege) to directly fill the ontology with axioms and rules.

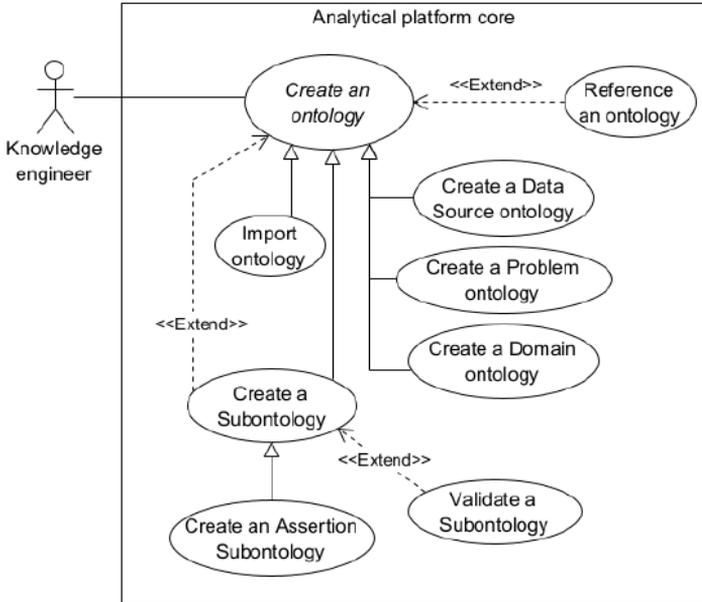


Fig. 8. The functionality of the tools for managing the repository of ontologies

It is also possible to create the special kinds of ontologies based on the existing ones. As was mentioned, a subontology (Create a Subontology) allows definition of a new element (class, property) if it is located within the existing taxonomies of an original ontology. In an assertion subontology (Create an Assertion Subontology) it is strictly individuals that are available for creation, as well as the assignment of property values to them.

In the process of creating subontology, the knowledge engineer selects a part of the taxonomies of classes and properties that are transferred to the subontology. This requires performing a reasoning to restore domains and ranges of properties if they are represented by classes that are removed from the taxonomy of the original ontology.

In addition, it is crucial to be able to Validate a subontology to verify the consistency of the creation of any elements in a subontology (or an assertion subontology) within the taxonomies of the original ontology.

If it is needed to use an existing ontology, it can be imported into the repository (Import ontology) from a remote server or a local file. After the import, a duplicate ontology is created separately within repository not to cause any conflict with files in the internal file system of the host operating system. For the purpose of building ontologies using existing ones there is a use case Reference an ontology, which allows elements of an existing ontology to be reused in the selected ontology.

According to an approach to analytical platform development, the software architect can use the platform core to create DSLs (Fig. 9) which provides users with tools to specify platform modules in a declarative way using visual editors. Using the pattern for integrating DSLs we can easily define

which elements of selected ontologies (i.e. which parts of taxonomies) form a concrete DSL metamodel using the level of abstraction that is provided by subontologies.

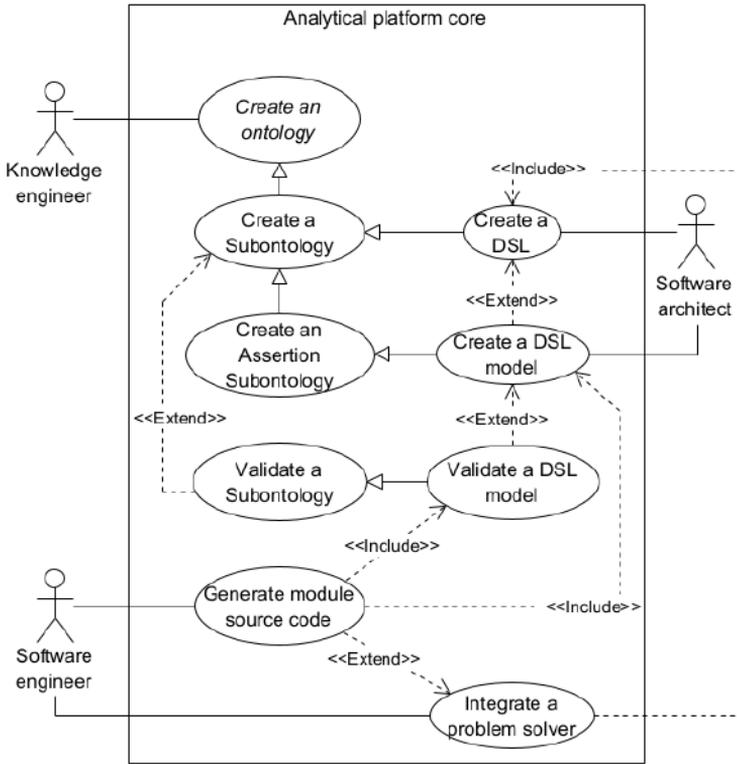


Fig. 9. The functionality of integrating DSLs and problem solvers

It is proposed that classes of the ontology that describe a DSL are used to represent classes of model elements (node types), object properties are used to depict relationships between classes (edge types) and datatype properties are used to list classes' attributes. Thus, a concrete model that is created according to DSL can only contain elements, edges between them and attribute values that can be represented as an assertion subontology.

In addition, the core allows the software engineer to generate source code of platform modules. For instance, we can transform an ontology as a set of classes that are translated to Python 3 programming language to be used as a blueprint of a problem solver.

Similar to DSL metamodels, the data engineer (fig. 10) can define metamodels of input and output data structure (Describe data structure for a problem) as a subontologies. Then, based on them the data analyst can describe and extract a dataset that is passed into a problem solver when the problem is being solved. Also, he has the possibility to validate the dataset according to the metamodel that defines its structure. When the data processing task is completed, validation can also be applied to the result of data processing.

The core users (namely, data analysts) can extend the platform functionality by simply specifying DSL models of a new module that is relied on a problem solver. This helps to reduce their dependence on software engineers and increase an adaptivity of a platform resulting in its improved viability.

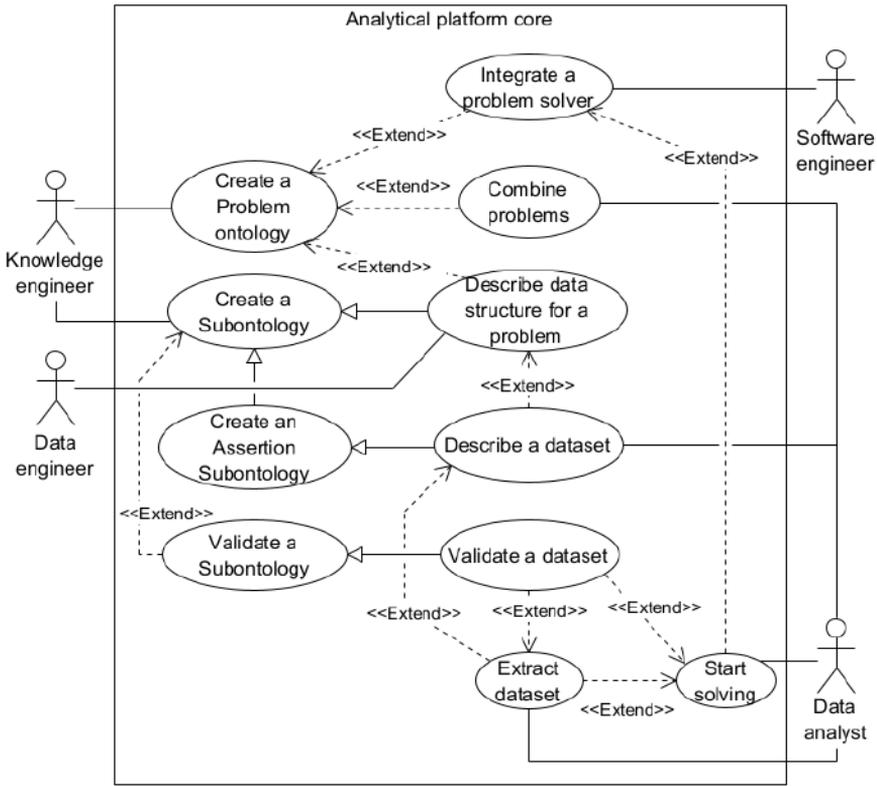


Fig. 10. The functionality of managing problem solving

6. Conclusion and future work

In this paper we proposed an ontological approach to information integration that allows to integrate different types of ontologies which are relevant to data-intensive domains (data source metamodels, problem descriptions, domain models). An interpretive approach to integration helps to avoid data duplication, ensure changes traceability of ontologies and automatically interpret data and the results of data analysis to provide them to different groups of users according to terminology that they are familiar with.

In order to create a basis for solving the design challenges of developing knowledge-based analytical platforms we presented design patterns based on the models of OWL 2 language. It is expected that the developed models will be used to implement analytical platform core functionality that will help to provide users with the tools to extend platform functionality with minimal reliance on software engineers.

Further work in this field is aimed at implementing software prototypes of the platform's core functionality based on described patterns and ontological approach to information integration.

References / Список литературы

- [1]. Gribova V.V., Moskalenko F.M. et al. Viable Intelligent Systems Development with Controlled Declarative Components. *Informational and mathematical technologies in science and management*, 2018, vol. 3, issue 11, pp. 6-17 (in Russian). DOI: 10.25729/2413-0133-2018-3-01 / Грибова В.В., Москаленко Ф.М. и др. Создание жизнеспособных интеллектуальных систем с управляемыми декларативными компонентами. *Информационные и математические технологии в науке и управлении*, том 3, вып. 11, 2018 г., стр. 6-17.

- [2]. Gribova V.V., Shalfeeva E.A. Systems based on ontological knowledge bases as the basis for the creation of modern artificial intelligence systems. In Proc. of the Eighteenth National Conference on Artificial Intelligence with International Participation, 2020, pp.12-19 (in Russian) / Грибова В.В., Шалфеева Е.А. Системы на основе онтологических баз знаний как основа для создания современных систем искусственного интеллекта. Труды Восемнадцатой Национальной конференции по искусственному интеллекту с международным участием, 2020 г., стр. 12-19.
- [3]. Alizadeh M., Shahrezaei M.H., Tahernezhad-Javazm F. Ontology Based Information Integration: a Survey. arXiv preprint, arXiv:1909.12372, 2019.
- [4]. Lyadova L.N., Zayakin V.S., Smirnov M.A. Formation of Event Series Using Multifaceted Ontologies. In Proc. of the X International Scientific and Technical Conference on Technologies for the Development of Information Systems, 2020, pp. 297-303 (in Russian) / Лядова Л.Н., Заякин В.С., Смирнов М. А. Формирование событийных рядов с использованием многоаспектных онтологий. Труды X Международной научно-технической конференции «Технологии разработки информационных систем», 2020 г., стр. 297-303.
- [5]. Lyadova L.N., Suvorov N.M., Vasiljuk V.A. The Architecture of the Knowledge-Based DSM Platform. In Proc. of the X International Scientific and Technical Conference on Technologies for the Development of Information Systems, 2020, pp. 304-311 (in Russian) / Лядова Л.Н., Заякин В.С., Смирнов М.А. Архитектура DSM-платформы, основанной на знаниях. Труды X Международной научно-технической конференции «Технологии разработки информационных систем», 2020 г., стр. 304-311.
- [6]. Tel'nov Yu.F., Kazakov V.A., Trembach V.M. Developing a Knowledge-Based System for the Design of Innovative Product Creation Processes for Network Enterprises. Business Informatics, 2020, vol. 14, issue 3, pp. 35-53 (in Russian). / Тельнов Ю.Ф., Казаков В.А., Трэмбач В.М. Разработка системы, основанной на знаниях, для проектирования инновационных процессов создания продукции сетевых предприятий. Бизнес-информатика, ном. 14, вып. 3, 2020 г., стр. 35-53.
- [7]. Pavlov S.V., Efremova O.A. Ontological Model for Integration of Structurally Heterogeneous Spatial Databases of Various Subject Areas into a Uniform Regional Database. *Ontology of Designing*, vol. 7, issue 3 (25), 2017, pp. 323-333 (in Russian). / Павлов С.В., Ефремова О.А. Онтологическая модель интеграции разнородных по структуре и тематике пространственных баз данных в единую региональную базу данных. Онтология проектирования, том 7, вып. 3, 2017 г., стр. 323-333.
- [8]. Asfand-E-Yar M., Ali R. Semantic Integration of Heterogeneous Databases of Same Domain Using Ontology. *IEEE Access*, vol. 8, 2020, pp. 77903-77919.
- [9]. Xiao G., Hovland D. et al. Efficient Ontology-Based Data Integration with Canonical IRIs. *Lecture Notes in Computer Science*, vol. 10843, 2018, pp. 697-713.
- [10]. Chuprina S.I., Postanogov I.S. Enhancing Legacy Information Systems with a Natural Language Query Interface Service. *Perm University Herald. Maths. Mechanics. Informatics*, 2015, vol. 2, pp. 78-86 (in Russian). / Чуприна С.И., Постаногов И.С. Концепция обогащения унаследованных информационных систем сервисом запросов на естественном языке. Вестник Пермского университета. Математика. Механика. Информатика, ном. 2, 2015 г., стр. 78-86.
- [11]. Kumar V.S., Cuddihy P., Aggour K.S. NodeGroup: A Knowledge-Driven Data Management Abstraction for Industrial Machine Learning. In Proc. of the 3rd International Workshop on Data Management for End-to-End Machine Learning, 2019, pp. 1-4.
- [12]. OWL 2 Web Ontology Language Structural Specification and Functional-Style Syntax (Second Edition). Available at: <https://www.w3.org/TR/2012/REC-owl2-syntax-20121211>, accessed: 07.04.2022.
- [13]. Gribova V.V., Kleshchev A.S. et al. Extensible Toolkit for the Development of Viable Systems with Knowledge Bases. *Software Engineering*, vol. 9, issue 8, pp. 339-348 (in Russian). DOI: 10.17587/prin.9.339-348 / Грибова В.В., Клещев А.С. и др. Расширяемый инструментарий для создания жизнеспособных систем с базами знаний. Программная инженерия, том 9, вып. 8, 2018 г., стр. 339-348.
- [14]. Lyadova L.N., Zayakin V.S., Smirnov M.A. The Architecture of the System for Analyzing Data from Internet Sources. *Informatization and communication*, no. 8, 2021, pp. 48-52 (in Russian). / Лядова Л.Н., Заякин В.С., Смирнов М.А. Архитектура системы анализа данных, получаемых из Интернет-источников. Информатизация и связь, no. 8, 2021 г., стр. 7-10.

Information about authors / Информация об авторах

Viktor Sergeevich ZAYAKIN – Master’s Student at National Research University Higher School of Economics, Data Engineer at SEUSLAB LLC. Research interests: modeling languages, information integration, ontologies and knowledge bases, semantic web.

Виктор Сергеевич ЗАЯКИН – студент магистратуры Национального исследовательского университета «Высшая школа экономики», инженер данных ООО «СЕУСЛАБ». Сфера научных интересов: языки моделирования, интеграция информации, онтологии и базы знаний, семантические технологии.

Lyudmila Nikolaevna LYADOVA – PhD in Computer Science, Associate Professor of the Department of Information Technologies in Business, HSE. Research interests: modeling languages, domain specific modeling, language toolkits, CASE tools, simulation systems.

Людмила Николаевна ЛЯДОВА – кандидат физико-математических наук, доцент кафедры информационных технологий. Сфера научных интересов: языки моделирования, предметно-ориентированное моделирование, языковые инструментарии, CASE-средства, системы имитационного моделирования.

Evgeny Andreevich RABCHEVSKIY – CEO of SEUSLAB LLC. Research interests: methods of information retrieval, big data, and artificial intelligence for information and psychological warfare, countering extremism and terrorism ideology on the Internet, assessing the military and political situation, information and analytical support of intelligence activities.

Евгений Андреевич РАБЧЕВСКИЙ – генеральный директор ООО «СЕУСЛАБ». Сфера научных интересов: применение технологий информационного поиска, больших данных и искусственного интеллекта в задачах информационно-психологического противоборства, противодействия идеологии экстремизма и терроризма в сети Интернет, оценки военно-политической обстановки, информационно-аналитического обеспечения оперативно-разыскной и разведывательной деятельности.

DOI: 10.15514/ISPRAS-2022-34(2)-5



Actor-oriented approach for business-process management of analytical system development

^{1,2} T.M. Pribylev, ORCID: 0000-0002-6110-0350 <timofey.pribylev@glowbyteconsulting.com>

^{2,3} M.N. Zaytsev, ORCID: 0000-0002-2974-1465 <mikhail.zaytsev@glowbyteconsulting.com>

¹ O.L. Vikentyeva, ORCID: 0000-0002-8991-4719 <ovikenteva@hse.ru>

¹ National Research University Higher School of Economics,
20, Myasnitskaya st., Moscow, 101000, Russia

² Glowbyte Consulting,

5/19, Nizhny Susalny Lane, Moscow, 105064, Russia

³ Lomonosov Moscow State University,

GSP-1, Leninskie Gory, Moscow, 119991, Russia

Abstract. This paper aims at investigating the feasibility of an actor-oriented approach for modelling analytical systems development business processes. The study analyzes existing management challenges of analytical systems development processes, identifies key business process modeling approaches, and proposes a modeling approach based on actor-oriented approach with high flexibility and enhanced control over business artifacts. The article also describes examples of possible applications of this approach in a business process management tool.

Keywords: business process; business process management; actor; artifact; analytical system

For citation: Pribylev T.M., Zaytsev M.N., Vikentyeva O.L. Actor-oriented approach for business-process management of analytical system development. Trudy ISP RAN/Proc. ISP RAS, vol. 34, issue 2, 2022, pp. 57-66. DOI: 10.15514/ISPRAS-2022-34(2)-5

Акторный подход к управлению бизнес-процессами разработки аналитических систем

^{1,2} Т.М. Прибылев, ORCID: 0000-0002-6110-0350 <timofey.pribylev@glowbyteconsulting.com>

^{2,3} М.Н. Зайцев, ORCID: 0000-0002-2974-1465 <mikhail.zaytsev@glowbyteconsulting.com>

¹ О.Л. Викентьева, ORCID: 0000-0002-8991-4719 <ovikenteva@hse.ru>

¹ Национальный исследовательский университет «Высшая школа экономики»,
101000, Россия, г. Москва, ул. Мясницкая, д. 20.

² Glowbyte Consulting

105064, Россия, г. Москва, пер. Нижний Сусальный, д. 5, стр. 19

³ Московский государственный университет имени М.В. Ломоносова,
119991, Россия, Москва, Ленинские горы, д. 1

Аннотация. Целью данной работы является исследование возможности применения акторного подхода для моделирования бизнес-процессов разработки аналитических систем. В исследовании анализируются существующие проблемы управления процессами разработки аналитических систем, определяются основные подходы к моделированию бизнес-процессов и предлагается подход к моделированию на основе акторного подхода, обладающий высокой гибкостью и улучшенным контролем над бизнес-артефактами. В статье также описаны примеры возможного применения данного подхода в инструменте управления бизнес-процессами.

Ключевые слова: бизнес-процесс; управление бизнес-процессами; актор; артефакт; аналитическая система

Для цитирования: Прибылев Т.М., Зайцев М.Н., Викентьева О.Л. Акторный подход к управлению бизнес-процессами разработки аналитических систем. Труды ИСП РАН, том 34, вып. 2, 2022 г., стр. 57-66. DOI: 10.15514/ISPRAS-2022-34(2)-5

1. Introduction

Background. Business process modelling and management is one of the most important tools for analysts and project managers developing software systems [1], including analytical systems and machine learning models (ML models). The main notation for business process modelling over the last decade is BPMN [2, 3].

Problems of business process modelling arise in many projects; an overview of these problems is given in [1, 3]. Different software tools are used for further management of modelled business processes. There are many tools on the market with different functionalities, their analysis is conducted in [2] and it highlights a few problems of existing tools.

Professional Significance. This study is a follow-up to the study [7] which proof of concept of an actor-oriented approach to business process modelling. The present study intends to confirm the feasibility of using an actor-oriented approach to further business process management and to advance the field of analytical systems lifecycle management, including by combining the artifact- and actor-oriented approaches.

Article structure. The remainder of this article is organized as follows: Section 2 introduces the motivation of this study, Section 3 reviews the related work, Sections 4 describe the problem statement and research methods, Section 5 describes the problems of analytical systems development, Section 6 highlights the key features of main approaches to business process management, Section 7 discusses an addition to the actor-oriented approach and the actor model for business process management of analytical systems development. Conclusion and future work are given in Section 8.

2. Motivation

The BPMN methodology has a few limitations [1, 3, 4], including the lack of flexibility in business process management. This problem is especially relevant when building analytical systems, as this area is characterized by a high proportion of data manipulation and experimentation when developing and implementing analytical models and ML models [5, 6].

Many different approaches to business process design have been proposed to address the above-mentioned problems. This paper discusses actor-oriented approach and combines it with artifact-oriented approach [7].

Actor-oriented approach is based on describing business processes through the interaction of actors that have their own state and have the ability to asynchronously exchange messages, process received data and generate new actors [8]. The artifact-oriented approach is based on describing business processes through describing the flow of process business artifacts and organizing the execution of tasks based on this flow [9]. By a business artifact (simply artifact) in this case is meant data records relating to key business-relevant objects, their lifecycles, and their use in carrying out process tasks

The use of the actor-oriented approach is also required in the development of an enterprise product, and this study was conducted for the purpose of designing an element of the system being developed.

3. Related works

Business process modelling is a key task for businesses [1] because with process modelling, further analysis, and management the performance of a business system can be significantly improved. This

is particularly important in the development of analytical systems and ML models, as will be shown below. Furthermore, there are a few approaches in business process modelling whose applicability to the lifecycle management of ML models can be shown since the identified features of this cycle. The development of analytical systems and especially ML models stands out for the high complexity of data management business processes, experimentation in model development and implementation in the final systems. The complexities of data management processes are shown in [5]. This process involves data exploration, validation and cleaning, and the process can be repeated several times to introduce additional features into the dataset. This leads to many business process artifacts that need to be tracked.

The cyclicity of analytical systems development processes is related to the large number of experiments required to build an accurate model [6, 10]. During the development process, the data scientist may test the applicability of multiple data processing algorithms and must use many input features to produce a result. Because of this, the development process often returns to previous phases for additional research or data preparation. This leads to the need to model highly flexible business processes. Moreover, the large number of experiments generates even more artifacts and metadata of the business process.

Once an analytical system is built, it goes through verification and implementation phases [10], which can also lead to changes in business processes. Meanwhile, an important part of the life cycle of ML models is monitoring and modification of models in case deviations are identified [10], which leads the model to return to the previous phases within the business processes. Thus, the main challenges for business process modelling and management of analytical systems development are the high number of experiments in the domain, which requires high process flexibility, and the huge number of process related artifacts that need to be manipulated.

To solve the highlighted issues, existing approaches and tools for business process modelling need to be explored. Nowadays, there is much research in this field in different directions. Thus, the paper [11] provides a survey of 405 articles devoted to business process modeling and management research of the ML model life cycle, highlighting the main research topics. According to this study, "Model Management" is the topic of about one third of all research, but the "Experiment Management" aspect is only addressed in 4 articles. "Data Management" is investigated in the 19 articles used in this research. This shows a low degree of investigation of the business process management challenges of developing analytical systems.

Tools are needed to use any business process modelling approach. A survey of 83 tools [2] shows a similar situation to the research in this area. According to the study, functionality for "Process lifecycle management" is represented in 4 tools and functionality for "Configurable meta model" in 6 tools. Thus, the existing tools do not satisfy the identified requirements for business process modelling and management of analytical systems development.

To respond to the highlighted challenges, it is necessary to identify the main approaches to business process modelling. Currently, action-, artifact- and actor-oriented approaches are distinguished.

The action-oriented approach means modelling a business process as a flow of tasks performed by process actors. The main modelling language for this approach is BPMN. This approach is currently the most used in real-world projects [2]. However, the action-oriented approach has several critical issues. Source [3] lists disadvantages of BPMN and other action-oriented business process notations. Among other issues, it highlights difficulties with resource management, process control flexibility and process inter-relationships, which do not allow effective use of the action-oriented approach for business processes management of analytical systems development. The papers [4, 12] also highlight the shortcomings of the action-oriented approach and provides a brief survey of the artifact-oriented approach concept.

The artifact-oriented approach offers the business processes management through a set of artifacts used and generated in these processes. The advantage of this approach over the action-oriented approach is the high flexibility of the built processes and the possibility of micro-manage of the data

used in the business processes [4], which is extremely important for business processes management of developing analytical systems. The authors of this study also highlight disadvantages of this approach, including difficulties in maintaining the connection between business process and business strategy. The study [9] presents the developed framework for this approach and confirms the possibility of using this concept for business process modelling.

A third approach is also proposed for modelling business processes. According to the actor-oriented approach, a business process is supposed to be considered as an interaction of intelligent business objects (IBO) [8]. Each IBO must be able to exchange messages asynchronously with other IBOs, process received messages, change its state, and generate new IBOs. This approach allows asynchronous interaction of business process parts and significantly increases the flexibility of the business process. The challenge of this approach is the difficulty to visualize the business process. The paper [7] presents the proof of concept of the actor-oriented approach.

4. Problem statement

The BPMN methodology has a few limitations [1, 3, 4], including the lack of flexibility in business process management. This problem is especially relevant when building analytical systems, as this area is characterized by a high proportion of data manipulation and experimentation when developing and implementing analytical models and ML models [5, 6].

This study intends to address the following research questions:

RQ1: What would be the benefits of an actor-oriented approach for business process modelling of analytical systems?

RQ2: How to develop an actor-oriented approach using elements of other approaches to business process management?

The outcome of the study is expected to be the answers to these research questions, as well as an actor model for the simple prototyping of business process management system for development of analytical systems.

5. Development analytical systems challenges

Articles [5, 6, 10] show the main challenges of analytical systems development processes. The following are important for this article:

1. **The high experimental nature of the field.** When developing analytical systems, it is necessary to research the domain of the analytical model, prepare data for the model, test the applicability of several algorithms in the domain and on existing data, develop the model and implement it in the runtime environment, and conduct continuous validation of the model's performance quality on updated data. Each of the steps described can be repeated several times, since at any of them it may be necessary to return to the previous step. This causes significant difficulties in the design of continuous business processes and their further management.

2. **The need to track data flows.** The main resource of the model work is the data on which it is applied. For proper functioning of the model, it is necessary to correctly process the raw data, to choose the algorithm for its analysis and to track the changes of data during the model's operation. To do this, it is critical to retain information about the data that is used as part of the business process of developing the model.

3. **The creation of many business artifacts.** The development of analytical systems generates many artifacts as part of experiments as well as data processing for the model. To effectively manage the processes, it is necessary to retain information about the artifacts that appear.

Thus, the business process management approach to developing analytical systems should allow flexibility to change the built business process in accordance with the current work and track the flow of data and artifacts.

6. Features of other approaches

When highlighting the main features of each of the approaches to business process management, it is necessary to operate on the needs of the area under study.

To address the challenges identified in the previous section, this article points out the following features of the approaches:

1. Simple process visualization (necessary for any human process management).
2. High flexibility (covers the high experimental nature of the field).
3. Control of data flows.
4. Possibility of asynchronous operation of business process elements (also covers high experimental nature of the field).
5. The ability to control individual tasks (adds flexibility to the approach).
6. Possibilities to control artifacts.

This article discusses three basic approaches to business process management - action-, artifact- and actor-oriented approaches. Their comparison according to the selected features is given in Table I.

The comparison shows that actor- and artifact-oriented approaches roughly equally cover the requirements of the study field. However, the most successful in terms of visualization remains the action-oriented approach.

Table 1. Approaches comparison

	Action-oriented	Artifact-oriented	Actor-oriented
Visualization	+	+/-	-
Flexibility	-	+	+
Asynchronous operation	+/-	+/-	+
Control of data-flow	-	+	+/-
Control of artifacts	-	+	-
Control of individual tasks	+	-	+

7. Actor-oriented approach

The actor-oriented approach fits well with the microservice architecture. A separate service can be created for each actor to handle the messages received by the actor. On this basis, this approach was chosen as the key approach of this paper.

The comparison of the approaches showed that the most promising development of the actor-oriented approach is to combine it with the artifact-oriented approach. For this purpose, the creation of a separate class of actors for artifacts is proposed. Thus, the actor model builds a system for supporting actors and processing their states.

The problem of business process visualization in this approach remains unresolved. When developing an enterprise product, it is possible to visualize using standard BPMN methods.

To build an actor model based on this combined approach, the following entities are also needed:

- 1) User controller.
- 2) Business process - for each business process an actor is created in the system.
- 3) Artifact controller.
- 4) Task controller.
- 5) Restriction controller – for working with user-defined restrictions on task execution.
- 6) Business process architect – for creating a business process according to the parameters entered by the user.

The actor model created based on the selected entities is shown in Fig. 1.

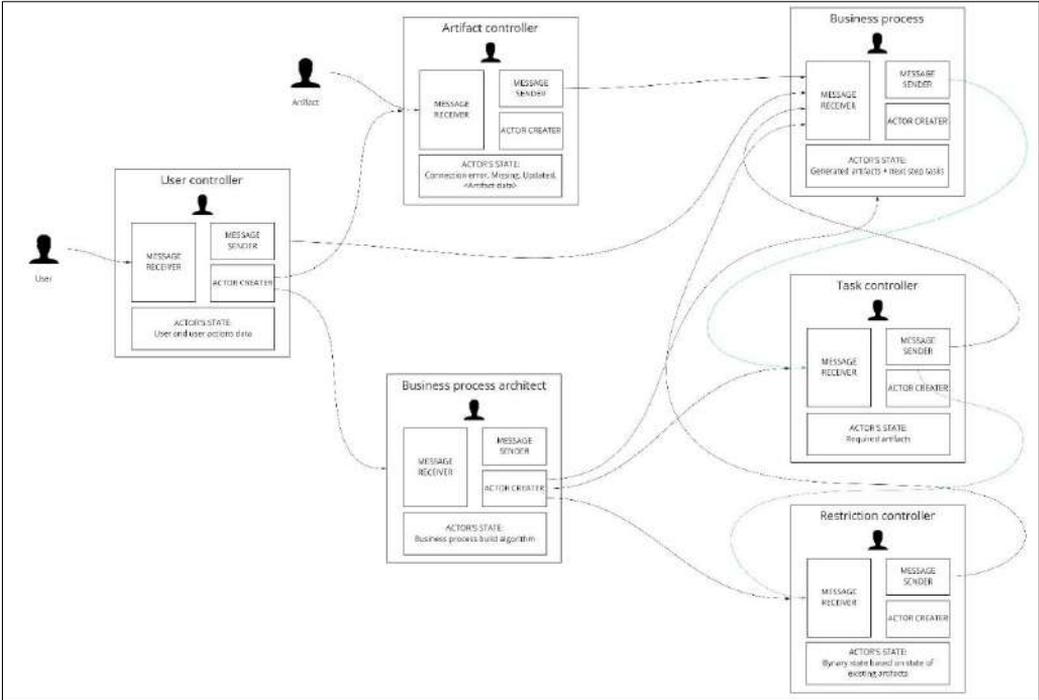


Fig. 1. Actor model

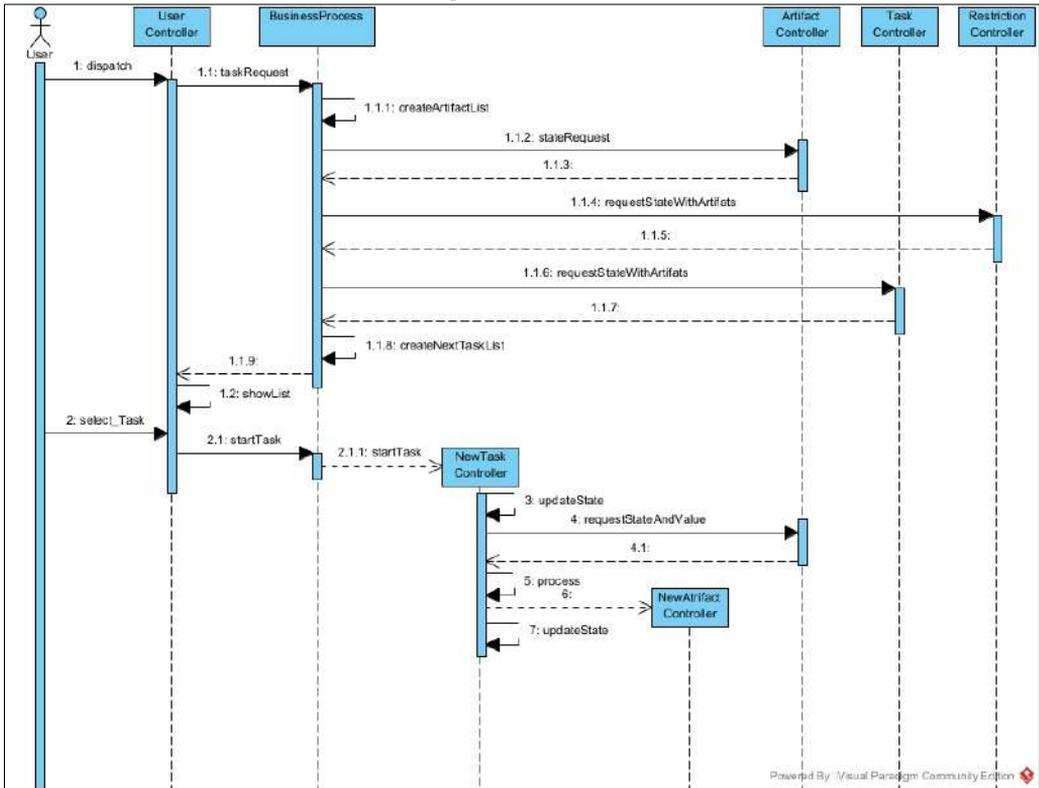


Fig. 2. Task selection and execution, UML sequence diagram

8. Performance examples

The proposed approach can create many different cases that meet the needs of a particular implementation. Two typical situations are designed for this article.

The first case is shown in Fig. 2. It represents the selection of a task by the user and its further execution. In the diagram, each of the objects is a previously described actor. Business process interrogates groups of actors of artifacts, tasks and constraints, on the basis of the responses it generates a list of available tasks to run. After selection by the user, the business process creates a new actor for the started task, and it performs the necessary actions for its execution.

The second case is presented in Fig. 3. This diagram simulates the situation of an artifact state change and the system response to this change. The artifact actor notifies the business processes and tasks actors subscribed to it. The end actors analyze these changes and update their state as needed. This process can be extended by a more complete description of the logic of each actor's actions, depending on the implementation context.

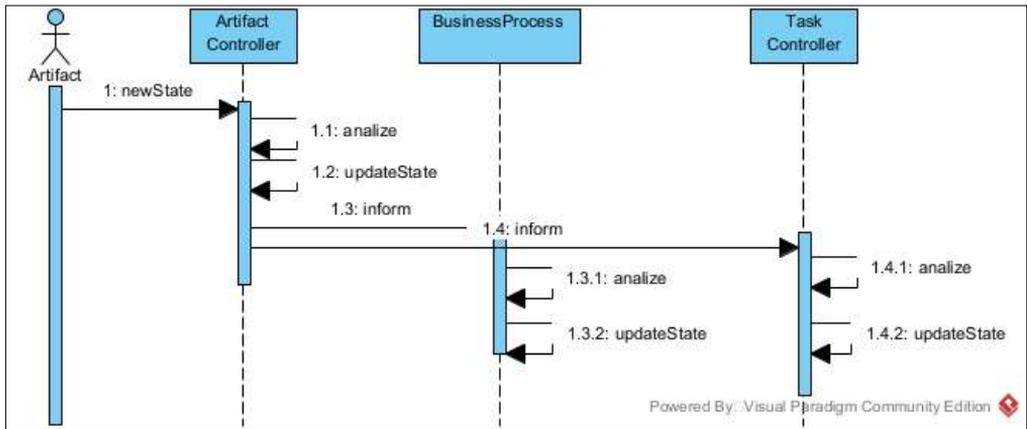


Fig. 3. Artifact update, UML sequence diagram

9. Conclusion

9.1 Answers to research questions

The following research questions were answered in this article:

RQ1: What would be the benefits of an actor-oriented approach for business process modelling of analytical systems?

A: The actor-oriented approach does not have decisive advantages over other business process management approaches to developing analytic systems, but it does make it easy to incorporate aspects of other approaches into its model. Moreover, the actor-oriented approach fits well with microservice architecture. These advantages make it possible to build business process management systems based on this approach.

RQ2: How to develop an actor-oriented approach using elements of other approaches to business process management?

A: The actor-oriented approach fits well with the artifact-oriented approach. This combination covers all the major challenges of business processes management of analytical systems development.

The article also provides an actor model for the system based on the developed approach.

9.2 Future work

The main area of further work will be to create a prototype system based on the developed approach to business process management. Another area of work could be the further combination of the created approach with others. For example, there are approaches based on Petri nets [13, 14], which were not considered in this article, but offers great functionality for process mining.

Finally, the study does not consider the problems of visualization of business processes built based on this approach. To create a visualization notation would require either combining the developed approach with BPMN diagrams or developing a new notation.

References

- [1]. M. Indulska, J. Recker et al. Business process modeling: Current issues and future challenges. *Lecture Notes in Computer Science*, vol. 5565, 2009, pp. 501-514.
- [2]. B. Zuhaira, N. Ahmad. Business process modeling, implementation, analysis, and management: the case of business process management tools. *Business Process Management Journal*, vol. 27, no. 1, 2021, pp. 145-183.
- [3]. E. Börger. Approaches to modeling business processes: a critical analysis of BPMN, workflow patterns and YAWL. *Software & Systems Modeling*, vol. 11, 2011, pp. 305-318.
- [4]. R. Hull. Artifact-centric business process models: Brief survey of research results and challenges. *Lecture Notes in Computer Science*, vol. 5332, 2008, pp. 1152-1163.
- [5]. N. Polyzotis, S. Roy et al. Data Management Challenges in Production Machine Learning. In *Proc. of the 2017 ACM International Conference on Management of Data*, 2017, pp. 1723-1726.
- [6]. S. Schelter, F. Biessmann et al. On Challenges in Machine Learning Model Management. *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*, vol. 41, 2018, pp. 5-15.
- [7]. F. Unterberger, R. Singer. Actor Based Business Process Automation via Intelligent Business Objects. In *Proc. of the 9th Conference on Subject-Oriented Business Process Management*, 2017, pp. 1-5.
- [8]. A. Fleischmann, W. Schmidt et al. *Subject-Oriented Business Process Management*. Springer. 2012, 392 p.
- [9]. S. Yongchareon, C. Liu et al. A View Framework for Modeling and Change Validation of Artifact-Centric Inter-Organizational Business Processes. *Information Systems*, vol. 47, 2015, pp. 51-81.
- [10]. A. Paleyes, R.-G. Urma, N.D. Lawrence. Challenges in Deploying Machine Learning: a Survey of Case Studies. *ACM Computing Surveys*, accepted on April 2022, 26 p.
- [11]. Y. Xie, L. Cruz et al. Systematic Mapping Study on the Machine Learning Lifecycle. In *Proc. of the IEEE/ACM 1st Workshop on AI Engineering - Software Engineering for AI (WAIN)*, 2021, pp. 70-73.
- [12]. J. Kunchala, J. Yu et al. Synthesis of Artifact Lifecycles from Activity-centric Process Models. In *Proc. of the IEEE 19th International Enterprise Distributed Object Computing Conference*, 2015, pp. 29-37.
- [13]. B. Xie and X. Li. Hybrid petri nets modeling for software development process management. In *Proc. of the 2nd IEEE International Conference on Computer Science and Information Technology*, 2009, pp. 546-550.
- [14]. W.M.P. van der Aalst. Using Free-Choice Nets for Process Mining and Business Process Management. In *Proc. of the 16th Conference on Computer Science and Intelligence Systems (FedCSIS)*, 2021, pp. 9-15,

Информация об авторах / Information about authors

Тимофей Михайлович ПРИБЫЛЕВ – студент 4 курса бакалавриата образовательной программы «Программная инженерия» Пермского филиала НИУ ВШЭ, кафедра информационных технологий в бизнесе. Сфера научных интересов: аналитика бизнес-процессов, подходы к исполнению бизнес-процессов.

Timofey Mikhailovich PRIBYLEV is a 4th-year undergraduate student of the Software Engineering educational program at the Perm Branch of the HSE University, Department of Information Technology in Business. His research interests include business process analytics, approaches to business process execution.

Михаил Николаевич ЗАЙЦЕВ является аспирантом кафедры теоретической информатики МГУ. Его научные интересы включают в себя распознавание звука и распознавание ключевых слов с помощью методов машинного обучения.

Mikhail Nikolaevich ZAYTSEV is a graduate student of the Department of theoretical computer science of the MSU. His research interests include speech and sound recognition, keyword spotting.

Ольга Леонидовна ВИКЕНТЬЕВА – кандидат технических наук, доцент кафедры информационных технологий в бизнесе НИУ ВШЭ. Является научным руководителем образовательной программы «Программная инженерия» Пермского филиала НИУ ВШЭ с 2021 года. Ее научные интересы включают CASE-технологии, анализ и моделирование бизнес процессов, объектно-ориентированное моделирование, проектирование систем, управление проектами.

Olga Leonidovna VIKENTYEVA is Candidate of Technical Sciences, Associate Professor of the Department of Information Technologies in Business, HSE University. She has been the Academic Supervisor of the Software Engineering educational program at the Perm Branch of the HSE University since 2021. Her researching interests include CASE technologies, business process analysis and modeling, object-oriented modeling, system design, project management.

DOI: 10.15514/ISPRAS-2022-34(2)-6



Architecture of a software system for designing robust business processes

K.V. Samoylova, ORCID: 0000-0002-6867-076X <katarinasamoilova@yandex.ru>

E.B. Zamyatina, ORCID: 0000-0001-8123-5984 <e_zamyatina@mail.ru>

HSE University,

38, Studencheskaya st., Perm, 614070, Russia

Abstract. Nowadays, in order for a company to remain competitive, efficient and attractive to investors it needs to have reliable and threat-resistant business processes. The question of methods for building such business processes remains relevant. This paper proposes a software system, which involves the use of methods and tools of DSM (Domain Specific Modeling), ontological approach, simulation modeling methods, mass service theory, Petri nets. As an example, the logistics process of ship boarding in the port is considered. Software tools implementing simulation modeling and DSM are ANYLOGIC and METALANGUAGE.

Keywords: multi-model approach; reliable business process; business process risks; simulation modeling; domain-based modeling; queuing systems; Petri nets; ontologies

For citation: Samoylova K.V., Zamyatina E.B. Architecture of a software system for designing robust business processes. Trudy ISP RAN/Proc. ISP RAS, vol. 34, issue 2, 2022, pp. 67-76. DOI: 10.15514/ISPRAS-2022-34(2)-6

Архитектура программной системы для проектирования надежных бизнес-процессов

K.B. Самойлова, ORCID: 0000-0002-6867-076X <katarinasamoilova@yandex.ru >

Е.Б. Замятина, ORCID: 0000-0001-8123-5984 <e_zamyatina@mail.ru>

*Национальный исследовательский университет «Высшая школа экономики»,
614070, Россия, г. Пермь, ул. Студенческая, д. 38*

Аннотация. В наше время для обеспечения конкурентоспособности, эффективности и привлекательности для инвесторов компаниям нужны надежные и устойчивые к угрозам бизнес-процессы. Остается актуальным вопрос о методах построения таких бизнес-процессов. В данной статье предлагается программная система, которая предполагает использование методов и инструментов DSM (Domain Specific Modeling), онтологического подхода, методов имитационного моделирования, теории массового обслуживания, сетей Петри. В качестве примера рассматривается логистический процесс погрузки судов в порту. Программными средствами, реализующими имитационное моделирование и DSM, являются ANYLOGIC и METALANGUAGE.

Ключевые слова: мультимодельный подход; надежный бизнес-процесс; риски бизнес-процессов; имитационное моделирование; доменное моделирование; системы массового обслуживания; сети Петри; онтологии

Для цитирования: Самойлова К.В., Замятина Е.Б. Архитектура программной системы для проектирования надежных бизнес-процессов. Труды ИСП РАН, том 34, вып. 2, 2022 г., стр. 67-76. DOI: 10.15514/ISPRAS-2022-34(2)-6

1. Introduction

Nowadays, in order for a company to remain competitive, efficient and attractive to investors, it needs to have reliable and threat-resistant business processes. At the same time, modern processes tend to have more complex and multi-level architectures. Therefore, in the case of a risky situation, the consequences can seriously affect not only the company owning the process, but also the associated companies, thus determining the relevance and necessity of developing sustainable processes.

Under the risks (threats) for a business process is usually considered a set of conditions, actions and factors, the implementation of which cause any damage or hinder its implementation and achievement of the goal of these business processes. You should consider how these threats will affect the operation of a business process, whether they are triggers for each other, and what the consequences of a particular risk will be. Therefore, at the stage of analysis and process design, it is necessary to decide on the modeling method, the way to identify threats and ways to prevent them, and the order of bottlenecks identification.

The description of your own business processes is necessary first of all because you can describe your activities, adequately assess the processes, identify problem areas, as well as represent the relationships between the participants and the elements. To analyze a business process, it is necessary to study its model. Under the model we will take a set of tools that can describe, document, and design the area under consideration. As a rule, several classes of models are considered, but often the business processes are described by analysts, using graphic models in one or another notation: IDEF, BPMN, UML, etc. There are papers that present methodologies (ROPE [1], INMOTOS [2], PoSem [3]) for detecting risks in business processes with subsequent application of countermeasures for their elimination.

The authors present a software system that proposes the use of simulation modeling methods to study the functioning of business processes over time. In this case, an automated transformation of the conceptual model M_X , where $X = \{IDEF, BPMN, \dots, UML\}$ (M_X is a graphical model of a business process whose description is performed using one of the notations: IDEF[4], BPMN[5], UML[6], etc.) into a simulation model M^Z , where $Y = \{GPSS, AnyLogic, \dots, NetLogo\}$. Here M_Y is a business process model that is described using simulation languages: GPSS[7], AnyLogic[8], NetLogo[9], etc. $Z = \{QT, PN, \dots, MC\}$ – a set of mathematical schemes that underlie this or that simulation system (queue theory, Petri nets, Markov chains, graph theory, etc.). Thus, a multi-model approach can be used to study business processes [10], namely, the business process model is transformed into a simulation model, which is based on a particular mathematical scheme. The transformation is performed using DSM (Domain Specific Modeling) Meta Language software [11,12]. As a result of the simulation experiment the risks of business processes are determined. For their identification and neutralization, the ontological approach is used [13]. Then, an automated modification of the original M_X model according to the specified countermeasures is performed and the actions associated with business process research are repeated (transformation of the conceptual M_X model into a simulation M_Y , simulation experiment, risk identification and countermeasures, application of countermeasures for business process reengineering, etc. Thus, the construction of a robust business process is iterative in nature.

The distinctive feature of the scheme proposed by the authors of building a reliable business process is the possibility of using a multi-model approach, involving the use of different mathematical apparatuses for dynamic research of business process functioning, analysis of countermeasure effectiveness, ontological approach for risk identification. In addition, any simulation modeling tool available to the researcher can be used for research, and any of the notations can be used to describe the business process.

In the following, the existing solutions for building reliable business processes, transformation rules, ontological approach for risk identification and countermeasures for their elimination, modeling results are discussed in detail.

2. Existing solutions

One of the methods that meets current conditions is the application of the ROPE methodology. Risk-Oriented Process Evaluation or ROPE is a three-layer model: synthesis of business process modeling, risk coordination and business continuity representation [1].

The model contains:

- A business process layer that describes the core activity of the process.
- Activity component definition layer - CARE (Condition, Action, Resource, Environment) diagram [1].
- Layer defining the consequences of occurred risks and possible threats, as well as making business decisions aimed at minimizing possible losses.

Because ROPE is aimed at revealing and removing of threats caused by the lack of resources, this method is applicable only to the coordination of the elementary processes, because more complex processes can lack the existing functionality.

Based on the ROPE model and taking into account the need to evaluate risks on the basis of several approaches, assessments and perceptions, the INMOTOS methodology was developed [2].

In general, the INMOTOS methodology begins with the modeling of business processes, using the CARE - diagram, followed by decomposition. Also, the necessary resources to perform the process are taken into account. During the threat prediction phase, an auxiliary agent is introduced, capable of identifying risks or their consequences, and triggers a process of predefined countermeasures. The process is completed by preventing threats and mitigating their impact on resources. However, the INMOTOS model does not take into account that threats as rules manifest themselves in complex ways and are most often triggers for each other.

POSeM (Process Oriented Security Models) methodology aims at exploring the relationship between business process re-engineering or general process modeling and methods to strengthen their resilience to risks, or to facilitate the choice of taking countermeasures to prevent risks by forming recommendations derived from the process description [3].

The implementation of this methodology follows two basic rules. Firstly, by means of the developed SEPL language, the requirements for business process resilience are provided. Secondly, a couple of rule bases are used to check the resilience of business processes to risks.

This method does not pay enough attention to external threats, the sources of which are not directly related to the process itself.

3. The example of constructing a reliable business process

So, we have considered a number of methodologies related to enhancing the reliability of business processes. In contrast, the authors propose to use models described using common notations with subsequent transformation (by DSM methods [11,12]) into simulation models developed in one of the popular simulation languages, as the ones studied for stability of business processes. The study of business processes for sustainability is performed by simulation modeling methods. As a result of risk detection, countermeasures for their elimination and automated modification of the original model-business process are determined. The general scheme of research of business processes on reliability is shown in fig. 1.

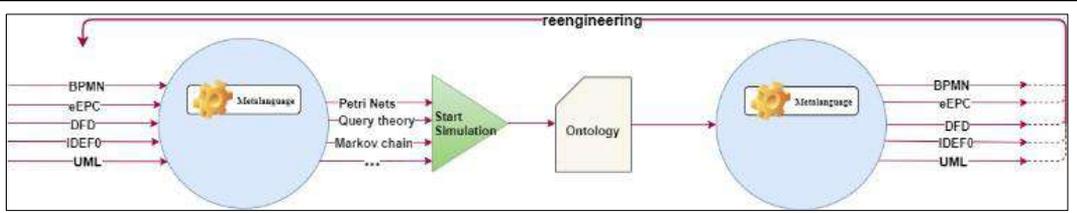


Fig. 1. Scheme for determining the sustainability of a business process, identifying and eliminating risks

Therefore, the proposed software system has metamodels that can be used to represent a business process in a particular notation and perform the transformation into a simulation model. As an example, we will consider the logistics process described in BPMN 2.0 notation (see fig. 2.) and its transformation into a simulation model (here we use AnyLogic [8], and Petri nets as the mathematical apparatus).

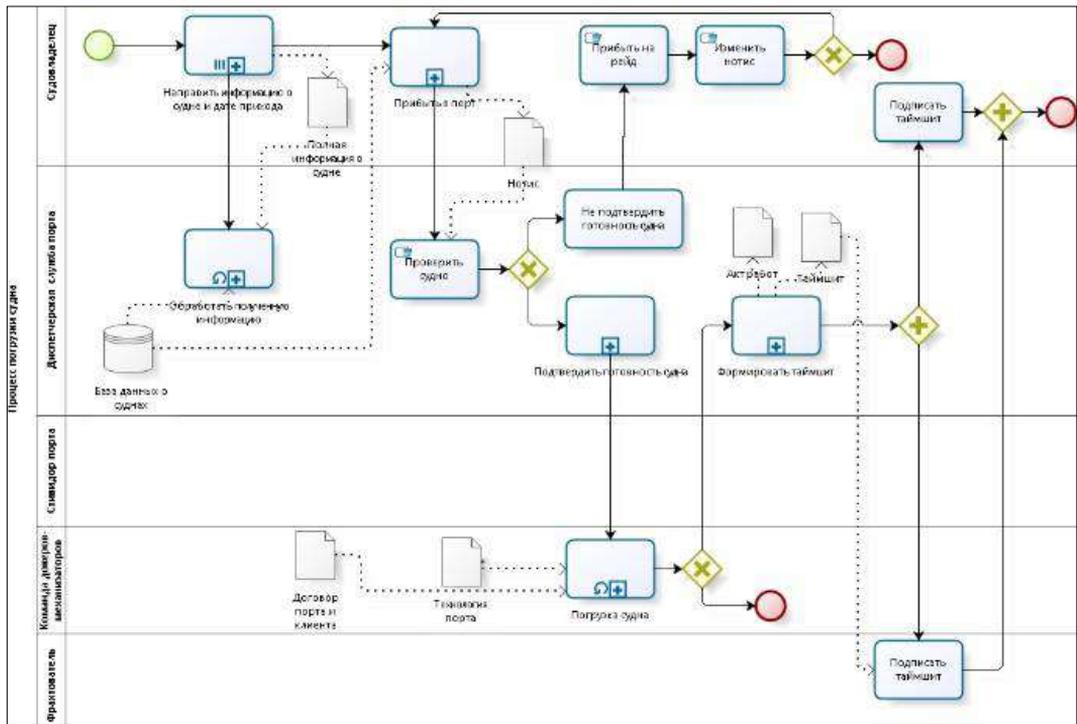


Fig. 2. Logistic process "Ship boarding in the port"

It is well-known that the rules of transformation from BPMN notation to Petri net notation were mentioned in literature [14].

Transformation of the model is performed using the Metalanguage language toolkit [12]. MetaLanguage was developed at Perm State University and meets all the key requirements for DSM platforms. The toolkit allows developing modeling languages of any subject area with the possibility of quick editing, implementing multilevel and multilingual modeling. Our task is to represent a logistic process described in BPMN notation with a list of rules into a model that can be represented by a Petri net (the mathematical apparatus that we chose to study the reliability of the process). The following are the rules of transformation $T: M_x \rightarrow M_Y^Z$ (M_Y^Z is the description of a business process in terms of temporal Petri nets (Z), the simulation experiment must be performed on one of the simulation platforms (Y – AnyLogic)) (see fig.3)

The platform for the implementation of simulation of the logistics process is the development of a Russian company – AnyLogic. AnyLogic is a domestic simulation modeling system widely used by specialists both in Russia and abroad. Models built in AnyLogic are scalable. When using this modeling system AnyLogic, the number of model elements is not limited [8].

As a result of transformations (Table 1), the initial process takes the following form (see fig. 3).

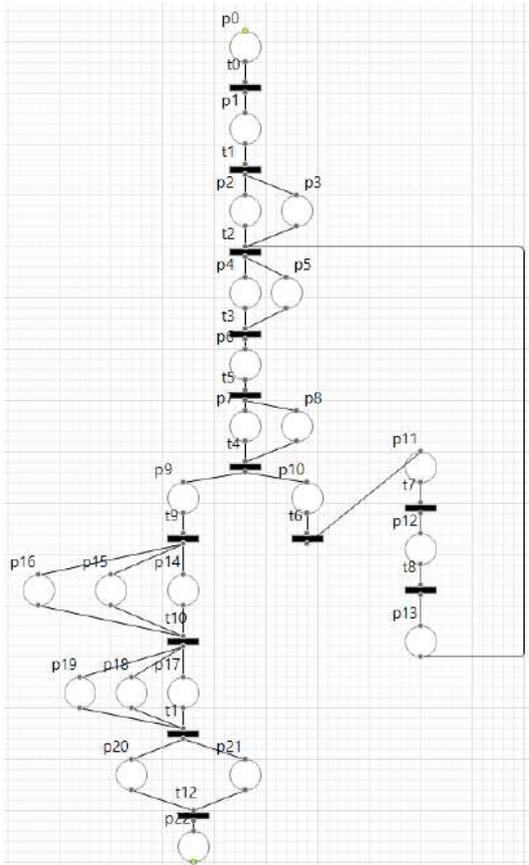


Fig. 3. Logistic process "Ship boarding in the port" (temporary Petri nets)

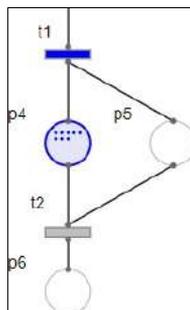


Fig. 4. The result of the simulation experiment - a transition that cannot be performed is identified

Next, a simulation experiment is performed, as a consequence of the simulation experiment the errors in the section t1-p4-p5-t2 are determined (fig. 4). To trigger transition t2, tokens must be present at place p4 and p5. Since transition t1 transfers tokens to place p4, and place p5 remains

empty – transition t2 fails. So, place p5 displays the use of the database of ships unloaded in the port. Thus, transition t2 cannot work if there is a database failure.

In order to determine what actions (countermeasures) should be taken to avoid the risk of an unregulated business process execution, we will use the ontology (fig. 5) and SPARQL query.

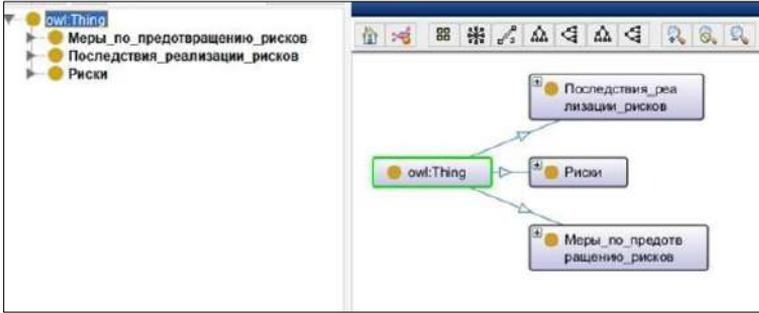


Fig. 5. Risks Ontology

The results of the SPARQL query are presented in fig. 6.

```

SPARQL query
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX SP: <http://www.semanticweb.org/katar/ontologies/2021/5/seaport_risk_1.0.0#>
SELECT *
WHERE { ?s SP:has_relation SP:Сбой_базы_данных.
?s rdfs:subClassOf ?o.
}

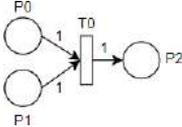
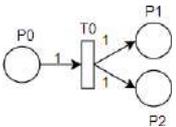
```

s	
Резервное_копирование_данных	Меры_по_предотвращению_рисков
Судно_не_впускают_в_порт	Последствия_реализации_рисков

Fig. 6. SPARQL

Table 1. Part of the result transformation rules

Name of the rule	Left part of the rule	Right part of the rule
Place0Transition0 Place1_Task	Place0Transition0Place 1 	Task

	Transition	Events
Transition_Events		
Place0Transition0 Place1Transition0 Place2_ ParallelJoinGate way	Place0Transition0Place1 Transition0Place2 	ParallelJoinGateway 
Place0Transition0 Place1Transition0 Place2_ ParallelForkGate way	Place0Transition0Place1 Transition0Place2 	ParallelForkGateway 

Then, using measures to prevent risks, the main process is restructured, and all stages are repeated again (see fig. 7).

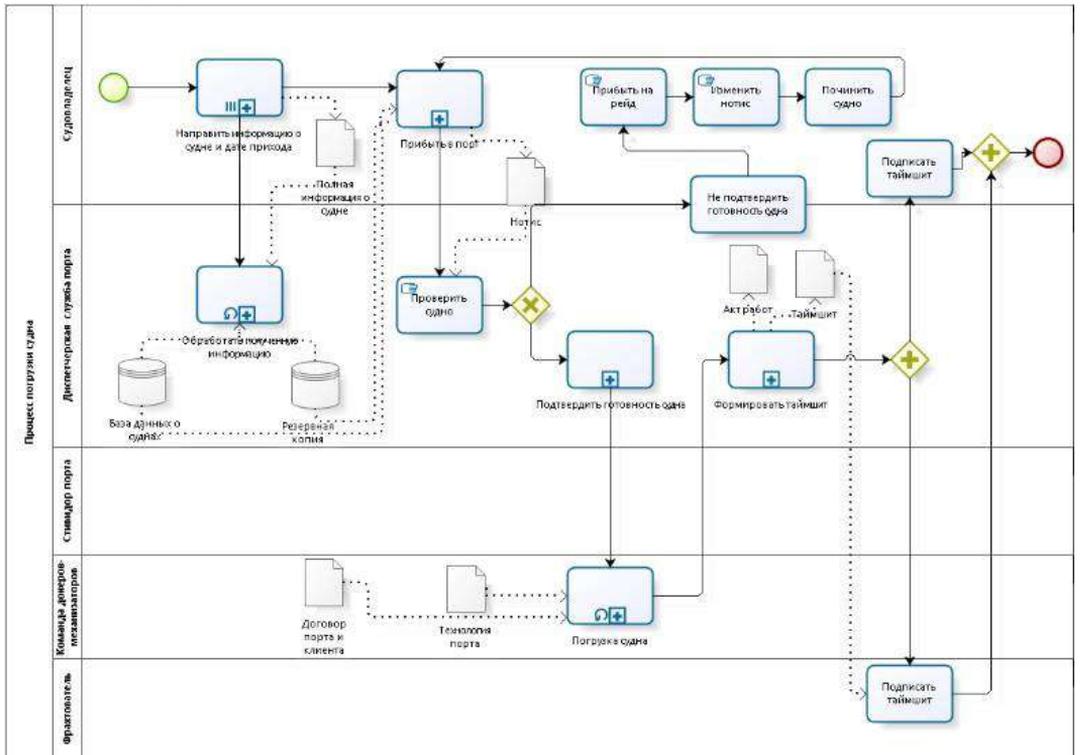


Fig. 7. Restructured logistic process "Ship boarding in the port"

4. Conclusion

The paper presents a methodology and software system that allows to identify risks in the business process model and perform reengineering actions in accordance with the recommendations for the application of certain countermeasures. In contrast to the methodologies discussed in the review part, the authors propose to use any well-known business process notations (DFD, EPC, BPMN and so on) made using notations, transform it into any simulation languages (AnyLogic, NetLogo for example) and use any mathematical apparatus (Queue network, Petri nets and so on). Thus, DSM methods are used to transform a business process model into a simulation model and back.

References / Список литературы

- [1] Jakoubi S., Tjoa S., Quirchmayr G. Rope: A Methodology for Enabling the Risk-Aware Modelling and Simulation of Business Processes. In Proc. of the 15th European Conference on Information Systems, 2007, pp. 1596-1607.
- [2] Zechner L., Kieseberg P., Weippl E. INMOTOS: Extending the ROPE-methodology. In Proc. of the 14th International Conference on Information Integration and Web-based Applications & Services, 2012, pp 272-277.
- [3] Rohrig S. Using Process Models to Analyse IT Security Requirements. PhD Thesis. Universität Zürich, 2003, 289 p.
- [4] Ledet J., Teran-Somohano A. et al. Toward Model-Driven Engineering Principles and Practices to Support Model Replicability. In Proc. of the Summer Simulation Multi-Conference (SummerSim'14), 2014, article no. 6, 6 p.
- [5] Menzel C., Mayer R. J. The IDEF family of languages. In Handbook on Architectures of Information Systems, Springer, 1998, pp. 209-241.
- [6] Chinos M., Trombetta A. BPMN: an introduction to the standard. Computer Standards & Interfaces, vol. 34, no. 1, 2012, pp. 124-134.
- [7] Booch G., Rumbaugh J., Jacobson I. The unified modeling language user guide, second edition. Addison-Wesley, 2005, 475 p.
- [8] Varzhapetyan A.G. Simulation modeling on GPSS/H. SPb., SUAI, 2007, 384 p. (in Russian) / Варжапетян А. Г. Имитационное моделирование на GPSS/H. СПб., ГУАП, 2007 г., 384 стр.
- [9] Karpov Yu.G. Simulation modeling systems. Introduction to modeling with AnyLogic 5. Saint-Petersburg, BHV, 2005. 390 p. (in Russian) / Карпов Ю.Г. Имитационное моделирование систем. Введение в моделирование с AnyLogic 5. СПб., BHV, 2005 г., 390 стр.
- [10] Chaudhry Q.A. An introduction to agent-based modeling modeling natural, social, and engineered complex systems with NetLogo: a review. Complex Adaptive Systems Modeling, vol. 4, 2016, article no. 11. 2 p.
- [11] Sokolov B.V., Yusupov R.M. The conceptual and technical basics for qualimetry of models and multiple-model complexes, SPIRAS Proceedings, vol. 1, issue 2, , 2004, pp. 10-35 (in Russian) / Соколов Б.В., Юсупов Р.М. Концептуальные и методические основы квалиметрии моделей и полимодельных комплексов. Труды СПИИРАН, том 1, вып. 2, 2004 г., стр. 10-35.
- [12] Rodríguez A., Durán F. et al. Executing Multilevel Domain-Specific Models in Maude. Journal of Object Technology, vol. 18, issue 2, 2019, article no. 4, 21 p.
- [13] Sukhov A.O., Lyadova L.N. Horizontal Transformations of Visual Models in MetaLanguage System. In Proc. of the 7th Spring/Summer Young Researchers' Colloquium on Software Engineering (SYRCoSE 2013). 2013, pp. 31-40.
- [14] Zamyatina E., Mikov A., Lanin V. Automation of Simulation Steps using Ontological Approach. In Proc. of the 10th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2018), 2018, pp. 223-230.
- [15] Kalenkova A.A., van der Aalst W.M.P. et al. Process mining using BPMN: relating event logs and process models//Software & Systems Modeling, vol. 16, issue 4, pp. 1019-1048.

Information about authors / Информация об авторах

Katarina Vadimovna SAMOYLOVA – 1st year student of master's degree program «Information Analytics in Enterprise Management». Research interests: business process optimization, ontology, natural language processing.

Катарина Вадимовна САМОЙЛОВА – студентка 1 курса магистратуры образовательной программы «Информационная аналитика в управлении предприятием». Научные интересы: оптимизация бизнес-процессов, онтология, обработка естественного языка.

Elena Borisovna ZAMYATINA – Candidate of Physical and Mathematical Sciences, Associate Professor of the Department of Information Technology in Business. Research interests: ontologies, distributed and parallel algorithms, pattern recognition, artificial intelligence, simulation, agent programming.

Елена Борисовна ЗАМЯТИНА – кандидат физико-математических наук, доцент, доцент кафедры информационных технологий в бизнесе. Научные интересы включают онтологии, распределенные и параллельные алгоритмы, распознавание образов, искусственный интеллект, моделирование, агентное программирование.

DOI: 10.15514/ISPRAS-2022-34(2)-7



Стратегии семплирования текста для прогнозирования недостающих библиографических ссылок

*Ф.В. Краснов, ORCID: 0000-0002-9881-7371 <fkrasnov2@yandex.ru>
И.С. Смазневич, ORCID: 0000-0002-5996-4635 <ismaznevich@naumen.ru>
Е.Н. Баскакова, ORCID: 0000-0002-7071-8961 <enbaskakova@naumen.ru>
NAUMEN,*

620028, Россия, Екатеринбург, ул. Татищева, 49А, БЦ «Татищевский», 4 этаж

Аннотация. В статье исследуются различные стратегии семплирования текстовых данных при выполнении автоматической классификации предложений с целью обнаружения недостающих библиографических ссылок. Построение семплов осуществляется на основе предложений в качестве семантических единиц текста, к которым добавляется их непосредственный контекст, состоящий из нескольких соседних предложений. Исследуется ряд стратегий семплирования, которые различаются размером и положением контекста. Эксперимент проведен на данных из сборника научных работ по естественнонаучной и инженерной тематике. Показано, что включение контекста предложений в семплы улучшает результат классификации предложений. Предложен метод автоматического определения оптимальной стратегии семплирования для данной текстовой коллекции: оптимальная стратегия определяется результатом голосования одинаковых классификаторов, получающих на вход одни и те же данные, семплированные различными способами. Семплирование с учетом контекста предложения в сочетании с процедурой жесткого голосования (hard voting) показало точность классификации 98% (оценка F1). Предложенный подход к обнаружению недостающих библиографических ссылок может использоваться в рекомендательных модулях прикладных интеллектуальных информационных систем.

Ключевые слова: семплирование текста; стратегия семплирования; анализ цитирования; прогнозирование библиографических ссылок; классификация предложений

Для цитирования: Краснов Ф.В., Смазневич И.С., Баскакова Е.Н. Стратегии семплирования текста для прогнозирования недостающих библиографических ссылок. Труды ИСП РАН, том 34, вып. 2, 2022 г., стр. 77-88. DOI: 10.15514/ISPRAS-2022-34(2)-7

Text sampling strategies for predicting missing bibliographic links

*F.V. Krasnov, ORCID: 0000-0002-9881-7371 <fkrasnov2@yandex.ru>
I.S. Smaznevich, ORCID: 0000-0002-5996-4635 <ismaznevich@naumen.ru>
E.N. Baskakova, ORCID: 0000-0002-7071-8961 <enbaskakova@naumen.ru>
NAUMEN,*

49A, Tatishcheva st., Yekaterinburg, 620028, Russia

Abstract. The paper proposes various strategies for sampling text data when performing automatic sentence classification for the purpose of detecting missing bibliographic links. We construct samples based on sentences as semantic units of the text and add their immediate context which consists of several neighbouring sentences. We examine a number of sampling strategies that differ in context size and position. The experiment is carried out on the collection of STEM scientific papers. Including the context of sentences into samples improves the result of their classification. We automatically determine the optimal sampling strategy for a given text collection by implementing an ensemble voting when classifying the same data sampled in different ways. Sampling strategy taking into account the sentence context with hard voting procedure leads to the classification

accuracy of 98% (F1-score). This method of detecting missing bibliographic links can be used in recommendation engines of applied intelligent information systems. Keywords: text sampling, sampling strategy, citation analysis, bibliographic link prediction, sentence classification.

Keywords: text sampling; sampling strategy; citation analysis; prediction of bibliographic references; proposition classification

For citation: Krasnov F.V., Smaznevich I.S., Baskakova E.N. Text sampling strategies for predicting missing bibliographic links. *Trudy ISP RAN/Proc. ISP RAS*, vol. 34, issue 2, 2022, pp. 77-88 (in Russian). DOI: 10.15514/ISPRAS-2022-34(2)-7

1. Введение

Научное исследование невозможно без соотнесения полученных результатов с работами других ученых: их следует упомянуть, вставив в статью библиографические ссылки. Специалисты в области наукометрии по-разному обосновывают необходимость установления таких связей между исследованиями и формулируют различные теории цитирования.

Нормативная теория цитирования, которая опирается на принципы научной этики, сформулированные Мертоном (Robert K. Merton) [1], предполагает, что ссылки в научных статьях призваны указывать на работы, которые являются основой для исследования, связаны тематически, описывают используемые методы или необходимы для обсуждения результатов. Согласно рефлексивной теории, связи между научными работами указывают на состояние науки и помогают создать ее формализованное представление, например, карты науки [2-3].

Таким образом, бенефициаром корректности научного цитирования является все научное сообщество, как исследователи, создающие статьи о своих результатах, так и администраторы, отслеживающие достижения в различных областях науки. Упоминание актуальных и значимых результатов других ученых является одним из основных требований при построении научных текстов, в частности, с точки зрения редакторов научных журналов. Эти требования отмечены в руководствах по академическому письму [4-6] и подтверждаются на практике, что описано, например, в результатах исследований публикационной активности в высокорейтинговых международных журналах [7].

Авторы научных работ самостоятельно выбирают источники для цитирования и позиции для ссылок в тексте, и в настоящее время этот процесс никак не автоматизирован. В данной работе исследуется возможность создания рекомендательного алгоритма, позволяющего находить недостающие библиографические ссылки в научной статье, то есть выявлять те фрагменты текста, где необходимо упомянуть другую исследовательскую работу. Для этой цели оценивается вероятность наличия ссылок во фрагментах текста, используя обучение с частичным привлечением учителя. Формальная постановка рассматриваемой задачи заключается в следующем: требуется автоматически находить в тексте научной статьи те фрагменты (предложения), где ссылка отсутствует, но необходима, используя в качестве обучающих данных набор размеченных фрагментов со ссылками и без ссылок.

Задача классификации фрагментов текста в зависимости от наличия в них ссылок методологически схожа с задачей анализа тональности, в рамках которой тексты автоматически классифицируются как позитивные и негативные (в основном) в соответствии с их эмоциональными характеристиками. В дополнение к классификации фрагментов на позитивные и негативные принцип анализа тональности используется для выделения других классов, включая определение значимости цитирования [8-11]. Задача выявления недостающих или ненужных ссылок в тексте также может рассматриваться аналогично анализу тональности, тогда искомым настроением здесь является потребность автора подтвердить сформулированное утверждение.

Другим близким направлением исследований является распознавание именованных сущностей (Named Entity Recognition, NER) на основе предсказания классификатора. Аналогичная задача рассматривается в работе [12], где сущности выделяются через определение спанов (span prediction). Задача NER может быть решена в два этапа: идентификация фрагментов с высокой вероятностью содержания сущностей и определение точного положения этих сущностей [13-14]. Некоторые методы NER также учитывают контекст сущностей, как локальный, так и глобальный, а также внешний [15].

Задача классификации предложений с учетом их ближайшего контекста обсуждалась в ряде исследований. В работе [16] использовались контекстно-зависимые эмбединги, созданные языковыми моделями, высокое качество которых достигается ценой скорости. Автором [17] изучалась тематическая классификация и было показано, что модели, получающие контекст среди входных данных, работают лучше моделей без контекста. В этих работах размер контекста определяется единожды исходя из некоторых предположений и для конкретного текстового корпуса он может оказаться неоптимальным.

Метод, представленный в данной работе, также можно рассматривать как своего рода ресемплирование (англ. resampling). До сих пор оно выполнялось в основном для балансировки распределения классов в обучающей выборке с целью повышения точности предсказания, на которую несбалансированность данных влияет негативно. Методы балансировки подразделяются на три типа, а именно: сокращение количества объектов мажоритарного класса (undersampling), увеличение количества объектов миноритарного класса (oversampling) и гибридные методы. Первый подход подразумевает исключение некоторых данных мажоритарного класса (см., например, [18-19]); второй предполагает воспроизведение существующих экземпляров миноритарного класса либо создание новых [20-22], а гибридные методы направлены на объединение преимуществ обоих подходов [23]. Локальные и глобальные контексты учитываются современными архитектурами нейронных сетей для анализа текста. Поскольку текст представляет собой однонаправленный список термов, контекст обычно понимается как некоторая совокупность соседних слов до или после рассматриваемого термина [24-25]. В сверточных нейронных сетях увеличение размера контекста приводит к значительному увеличению размерности тензоров и, как следствие, количества параметров модели, что в свою очередь требует увеличения размера коллекций. В моделях глубокого обучения типа transformer контекст учитывается с помощью механизма внимания, а локальный контекст сочетается с более широким контекстом (BERT [26], GPT-3 [27]).

Важно, что все вышеперечисленные алгоритмы не учитывают естественные структурные единицы текстов (предложения и абзацы), поскольку эти алгоритмы настраиваются на определенный размер контекста, составляющий фиксированное количество слов, в то время как размер предложений и абзацев варьируется.

2. Методы

Задача определения недостающих ссылок формализуется как поиск фрагментов текста, где ссылка отсутствует, но необходима, или, наоборот, присутствует, но не нужна.

Решается задача автоматической классификации с двумя классами (позитивным и негативным). Для каждого фрагмента научной статьи предложенный алгоритм определяет вероятность наличия в нем библиографической ссылки. Набор текстовых документов задается так, что каждый документ состоит из фрагментов. Фрагмент представляет собой последовательность слов (термов) разной длины. Фрагменты могут накладываться друг на друга и различаться по размеру. Каждый фрагмент представляет собой семпл и помечается меткой одного из двух возможных классов: позитивного или негативного. Метка класса соответствует тому, содержит ли данный фрагмент библиографическую ссылку или нет. Задачей данного исследования является поиск такой стратегии построения фрагментов

(семплирования), которая дает наибольшую точность в определении меток класса для заданного классификатора.

Гипотеза исследования заключается в следующем: стратегии семплирования текста, учитывающие контекст, повышают точность классификации предложений, используемой для прогнозирования недостающих библиографических ссылок в научных статьях.

Позитивный семпл состоит из библиографической ссылки, окруженной ее контекстом из исходного текста, а негативный семпл представляет собой фрагмент без библиографической ссылки в нем. Чтобы избежать дублирования образцов, предложение с двумя или более ссылками рассматривается только один раз. Контекст ссылки ограничивается содержанием ее предложением либо расширяется и включает в себя также соседние предложения.

Наилучшим вариантом является ситуация, когда границы контекста ссылки совпадают с границами законченной мысли автора, к которой относится эта ссылка. В этом случае смысловой единицей текста могут быть как одно, так и несколько предложений, что затрудняет определение размера контекста. Тем не менее, чтобы приблизиться к наилучшей ситуации, в предложенном алгоритме в качестве контекста рассматривается фрагмент, размер которого определяется количеством предложений, а не слов (в отличие от алгоритмов нейронной сети). Таким образом, контекст формируется на основе естественных структурных единиц текста.

Пространство признаков создается автоматически на основе статистики словаря в рамках модели «Мешок слов» (Bag of Words). Словарь модели включает в себя слова и все оригинальные знаки препинания и служебные символы. В качестве дополнительных признаков рассматриваются именованные сущности.

3. Алгоритм

Алгоритм состоит из следующих этапов.

- Предварительная обработка текста.
- Очистка текста: удаление служебных символов (табуляция, перевод строки и т. д.), слов (названия журналов, ISBN и т.д.) и разделов (информация о финансировании, список литературы);
- Токенизация:
 - разбиение текста на предложения;
 - нормализация термов.
- Разметка данных.
- Для каждого документа (статьи) добавляются метки начала и конца;
- Для каждого предложения:
 - если предложение содержит библиографическую ссылку (обозначение цитирования), оно помечается как относящееся к классу "Со ссылками";
 - если в предложении нет обозначения цитирования, оно получает метку класса "Без ссылок".
- После разметки предложений обозначения цитирования удаляются.
- Обработка именованных сущностей.
- Обнаружение именованных сущностей в тексте;
- Замена именованных сущностей специальными метками.
- Конструирование семплов. Семплы строятся по-разному в зависимости от класса (позитивного или негативного):
 - Позитивный семпл строится из одного предложения "Со ссылками", к которому добавляется n предыдущих предложений и m последующих предложений; все

предложения берутся в исходном порядке.

- Негативный семпл составляется из k предложений "Без ссылок", идущих подряд в исходном тексте (смежных предложений), где $k = n + 1 + m$.

Структура семплов показана на рис. 1.

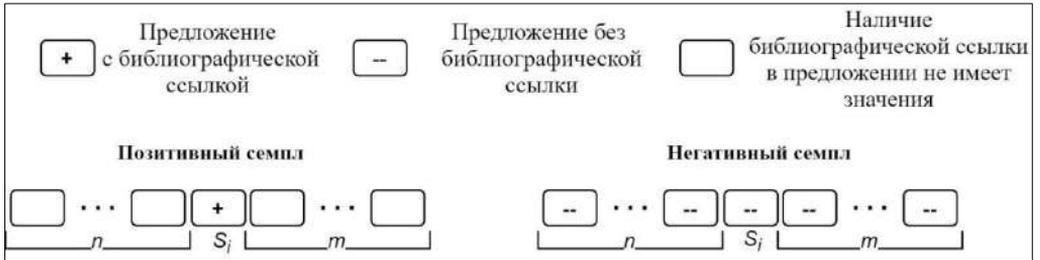


Рис. 1. Построение семплов
Fig. 1. Construction of samples

- Классификация семплов.
- Балансировка распределения классов в обучающей выборке выполняется методом случайного сокращения мажоритарного класса (random undersampling).
- Для каждого семпла строится векторная модель с использованием счетчика слов (как наиболее быстрого и эффективного с вычислительной точки зрения представления текста).
- Векторизованный набор семплов обрабатывается с помощью классификатора.
- Определение оптимальной стратегии конструирования семплов.

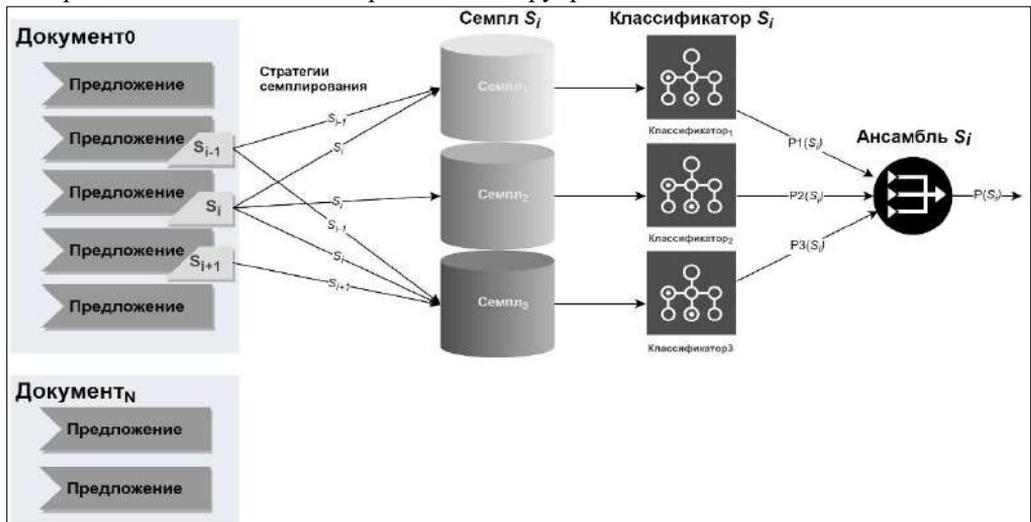


Рис. 2. Общая схема алгоритма классификации предложений на классы "Со ссылками" и "Без ссылок" с использованием различных стратегий семплирования (для $n = m = 1$)

Fig. 2. The algorithm flowchart for classifying sentences into the classes "With links" and "Without links" using various sampling strategies (for $n = m = 1$)

Далее осуществляется построение ансамбля моделей классификации с целью автоматического определения оптимальной стратегии семплирования, а именно: одни и те же данные, собранные в семплы разными способами, обрабатываются однотипными классификаторами, после чего реализуется процедура голосования.

Общая схема алгоритма показана на рис. 2. Каждый семпл для предложения S_i соответствует одной стратегии семплирования, и для каждой стратегии семплирования выполняется своя процедура классификации. При этом все классификаторы используют один и тот же метод, но в качестве входных данных используют разные типы семплов.

4. Эксперимент

Экспериментальная проверка гипотезы была проведена на коллекции научных статей по естественнонаучной и инженерной тематике (STEM) из базы arXiv.org [28]. Документы этого набора данных содержат только тексты, рисунки и таблицы удалены. Математические формулы и обозначения цитирования заменены специальными токенами – $@xmath<число>$ и $@xcite$ (метки цитирования). Документы содержат только разделы до «Заклучения» включительно, все последующие разделы удалены.

Размер набора данных следующий: количество документов – 215 тыс., средняя длина документа – 4938 слов, средняя длина аннотации – 220 слов.

Файлы представлены в формате jsonlines, где каждая строка представляет собой объект json, соответствующий одной научной статье. Каждая строка содержит аннотацию, перечень наименований разделов и основную часть статьи, где весь текст разделен на предложения.

В эксперименте рассматриваются предложения длиной более 30 слов. С учетом этого ограничения набор данных состоит в общей сложности из 458774 предложений.

Предложения, содержащие специальные метки цитирования $@xcite$, относятся к позитивному классу ("Со ссылками"), после чего метки цитирования удаляются. Предложения без меток цитирования классифицируются как негативные ("Без ссылок"). Соотношение классов следующее: 24% – предложения из позитивного класса, 76% – предложения из негативного класса. Такая разметка предложений считается нулевой стратегией семплирования (№0), и результат классификации данных, семплированных таким образом, рассматривается как базовый уровень: точность классификации со стратегией семплирования № 0, измеренная с помощью метрики F1, составляет 0,7866.

После установления базового уровня точности были протестированы различные стратегии семплирования данных с целью повышения точности классификации. Основная идея семплирования состоит в том, чтобы учитывать некоторый контекст предложений со ссылкой. Различные стратегии семплирования предполагают различные направления, позиции и размер контекста, определяемый числом соседних предложений. Каждое предложение $[i]$ в различных стратегиях семплирования включается в разные типы (варианты) семплов.

В эксперименте тестировались 10 стратегий со следующими параметрами: n : [0, 1, 2 3 4 5], m : [0, 1, 2, 3, 4], k : [1, 3]. Все типы семплов, соответствующие выбранным стратегиям семплирования, представлены в табл. 1.

Табл. 1. Стратегии семплирования, протестированные в эксперименте
Table 1. Sampling strategies tested in the experiment

№	Стратегия семплирования (алгоритм построение семплов)	Количество предложений в семпле	
		Позитивный семпл	Негативный семпл
0	Предложение $[i]$	2640	12352
1	Предложение $[i: i+2]$	2640	10639
2	Предложение $[i-1: i+1]$	2640	10639
3	Предложение $[i-1: i+2]$	2640	9376
4	Предложение $[i-2: i+2]$	2640	8384
5	Предложение $[i-3: i+2]$	2640	7574
6	Предложение $[i-3: i+3]$	2640	6880
7	Предложение $[i-4: i+3]$	2640	6287
8	Предложение $[i-4: i+4]$	2640	5776

9	Предложение[$i-5:i+4$]	2640	5326
---	--------------------------	------	------

Распределение длины (количества слов) в позитивных и негативных семплах разных типов показано на рис. 3.

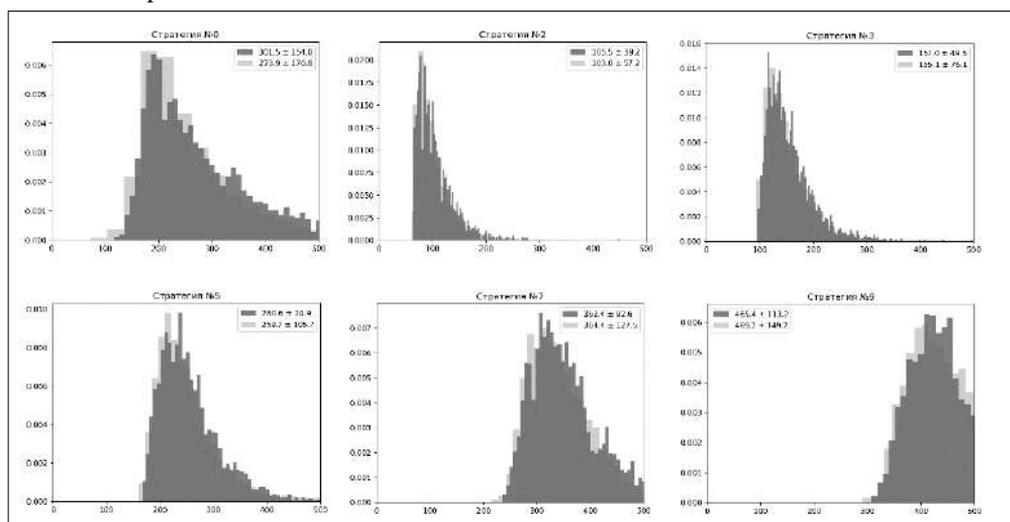


Рис. 3. Распределение количества слов в позитивных и негативных семплах разных типов (темно-серым цветом показаны позитивные семплы, светло-серым – негативные)

Fig. 3. The distribution of the length as a number of words in positive and negative samples of different types ('dark grey' color refers to the positive class, 'light grey' to the negative one)

После балансировки классов методом random undersampling данные были поделены на обучающую и тестовую выборки с параметром test_size=0.33.

Векторное представление строится с использованием метода CountVectorizer библиотеки Scikit-learn. Словарь включает в себя униграммы и биграммы и ограничен по частоте с параметрами min_df=3 (минимальная частота слова для исключения из словаря), max_df=0,7 (порог для исключения из словаря частовстречающихся слов по относительной частоте)

Для классификации используется многослойный перцептрон (метод MLPClassifier библиотеки Scikit-learn). Эффективность классификации в зависимости от используемой стратегии семплирования представлена в табл. 2.

Для каждого предложения сравниваются результаты классификации, полученные при различных стратегиях семплирования, и дополнительно улучшаются с помощью процедуры голосования. Тестировались поочередно методы мягкого голосования (soft voting) и жесткого голосования (hard voting). При мягком голосовании для средней предсказанной вероятности было установлено пороговое значение 0,5. При жестком голосовании сумма всех предсказанных значений вероятности сравнивалась с пороговым значением 3. Было протестировано разное количество классификаторов, участвующих в голосовании. Комбинации классификаторов, соответствующих различным стратегиям семплирования, формировались начиная с группы из трех классификаторов: № 7, № 8 и № 9, а затем к ним добавлялись еще классификаторы – один за другим в обратном порядке. Результаты голосования в зависимости от количества рассмотренных классификаторов показаны на рис. 4.

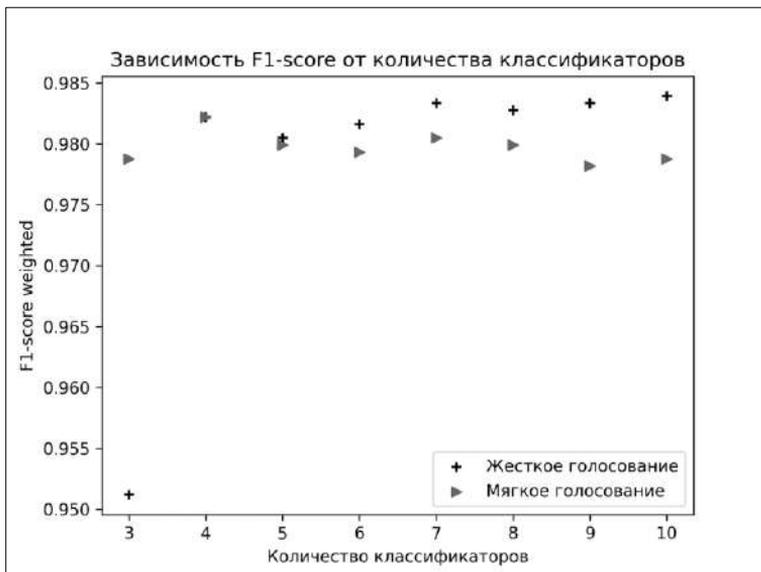


Рис. 4. Результат классификации с жестким и мягким голосованием в зависимости от количества участвующих классификаторов

Fig. 4. The result of classification with hard and soft voting depending on the number of estimators included

3. Результаты и обсуждение

Сформулированная исследовательская гипотеза была подтверждена экспериментально. Исследование показало, что выбор стратегии семплирования влияет на результат классификации текста.

Базовый уровень точности был установлен при использовании стратегии семплирования №0. В этом случае эффективность классификации, измеренная с помощью F1-меры, составляет всего 79%, что не является достаточным для практического использования в прикладных информационных системах.

Табл. 2. Результат классификации предложений классификатором MLP в зависимости от стратегии семплирования (средневзвешенное значение)

Table 2. The result of the classification of sentences by the MLP classifier depending on the sampling strategy (weighted average)

Стратегия семплирования	F1	Precision	Recall
0	0.7866	0.7866	0.7866
1	0.8882	0.8881	0.8881
2	0.8884	0.8881	0.8881
3	0.9214	0.9214	0.9214
4	0.9444	0.9443	0.9443
5	0.9410	0.9409	0.9409
6	0.9601	0.9598	0.9598
7	0.9640	0.9639	0.9639
8	0.9593	0.9593	0.9593
9	0.9581	0.9581	0.9581

Различные стратегии семплирования повышают эффективность классификации. В табл. 2 показано, что наилучший результат (оценка F1 96%) достигается при использовании стратегии семплирования № 7. Последующее увеличение количества предложений в семпле существенно не повышает точность, поскольку она стремится к асимптоте.

Дальнейшее улучшение достигается за счет стратегии семплирования данных, которая предполагает автоматическое определение оптимального типа семпла. Это обеспечивается

применением процедуры голосования к решениям, принятым различными классификаторами.

На рис. 4 показано, что процедура голосования дополнительно улучшает результаты классификации и увеличивает оценку F1 на 1,5%. При всех протестированных комбинациях классификаторов был получен стабильно высокий результат, но наилучшие значения достигались при жестком голосовании 7, 8 или 10 классификаторов, обрабатывающих длинные семплы.

Предложенный алгоритм показывает точность 98% (F1-мера), что сопоставимо с современными результатами для задачи NER с использованием автоматической классификации и других задач классификации текста [29]. Важно, что предложенный алгоритм обеспечивает высокую точность, но не требует для реализации огромных вычислительных ресурсов.

5. Заключение

В статье предлагается новый метод определения вероятности наличия библиографической ссылки во фрагментах научной статьи. Подход предполагает классификацию предложений с применением процедуры голосования, в котором различные стратегии семплирования данных используются классификаторами, реализующими один и тот же метод классификации. Постановка проблемы, сделанная авторами, близка к хорошо изученным задачам NER и анализа тональности, но является новой с точки зрения реального применения.

Основным новшеством предлагаемого метода является нахождение контекста ссылки, который максимально влияет на вероятность обнаружения недостающей библиографической ссылки в предложении. В предлагаемом алгоритме наилучший размер и положение контекста определяются автоматически. Размер определяется границами семантических единиц текста и измеряется количеством предложений, а не слов, таким образом используется тот факт, что предложение является более семантически емкой (значимой) единицей, чем слово. В большинстве существующих методов классификации текстов не предполагается, что контекст фрагмента имеет существенное значение, но данное исследование показывает критическую важность его рассмотрения. Значительное влияние контекста на эффективность классификации демонстрирует, что семантика, связанная с библиографической ссылкой, может быть локализована во фрагментах разной длины.

Точность предложенного алгоритма достигает 98% (оценка F1). Важно отметить высокую вычислительную эффективность описанного метода по сравнению со сверточными искусственными нейронными сетями. Это преимущество достигается за счет большего размера семплов. Исследуемый подход к анализу текста расширяет принцип механизма внимания, направленного на обучение языковой модели пониманию влияния глобального и локального контекстов. Автоматическое определение границ контекста коррелирует с идеей автоматического выбора значимых признаков в искусственных нейронных сетях.

Предлагаемый способ может быть использован в рекомендательных механизмах в прикладных интеллектуальных информационных системах, включая помощь в создании документов и составлении текстов с возможными ссылками на другие документы или помощь в проверке правильности документа. Такие функции полезны во многих областях, например, в науке, юриспруденции или журналистике, где документы содержат утверждения, которые должны быть подтверждены ссылками на правовые акты или другие источники.

Список литературы / References

- [1] Merton R.K. The sociology of science: Theoretical and empirical investigations. University of Chicago press, 1973, 605 p.

- [2] Москалева О.В., Акоев М.А. Наукометрия: немного истории и современные российские реалии. Управление наукой: теория и практика, том 1, no. 1, 2019 г., стр. 135-148 / Moskaleva O.V., Akoev M.A. Scientometrics: a little bit of history and modern Russian realities. *Science Management: Theory and Practice*, vol. 1, no. 1, pp. 135-148 (in Russian).
- [3] Зеленков Ю.А., Анисичкина Е.А. Динамика исследований в области интеллектуального анализа данных: тематический анализ публикаций за 20 лет. Бизнес-информатика, том 15, no. 1, 2021 г., стр. 30-46 / Zelenkov Yu.A., Anisichkina E.A. Trends in data mining research: A two-decade review using topic analysis. *Business Informatics*, vol. 15, no 1, 2021, pp. 30-46 (in Russian).
- [4] Emerson L., Rees M. T., MacKay B. Scaffolding academic integrity: Creating a learning context for teaching referencing skills. *Journal of university teaching & learning practice*, vol. 2, issue 3, 2005, pp. 17-30.
- [5] Gray K., Thompson C. et al. Web 2.0 authorship: Issues of referencing and citation for academic integrity. *Internet and Higher Education*. vol. 11, issue 2, 2008, pp. 112-118.
- [6] Pears R., Shields G. *Cite them right: the essential reference guide*. Palgrave Macmillan, 8th edition, 2010, 112 p.
- [7] Arsyad S., Ramadhan S., Maisarah I. The rhetorical problems experienced by Indonesian lecturers in social sciences and humanities in writing research articles for international journals. *The Asian Journal of Applied Linguistics*, vol. 7, issue 1, 2020, pp. 116-129.
- [8] Aljuaid H., Iftikhar R. et al. Important citation identification using sentiment analysis of in-text citations. *Telematics and Informatics*, vol. 56, 2021, article no. 101492.
- [9] Prester J., Wagner G. et al. Classifying the ideational impact of information systems review articles: A content-enriched deep learning approach. *Decision Support Systems*, vol. 140, 2021, article no. 113432.
- [10] Varanasi K.K., Ghosal T. et al. Iitp-cuni@ 3c: Supervised approaches for citation classification (task a) and citation significance detection (task b). In *Proc. of the Second Workshop on Scholarly Document Processing*, 2021, pp. 140-145.
- [11] Färber M., Sampath A. Determining how citations are used in citation contexts. *Lecture Notes in Computer Science*, vol. 11799, 2019, pp. 380-383.
- [12] Fu J., Huang X., Liu P. Spanner: Named entity re-/recognition as span prediction. arXiv.2106.00641, 2021, 13 p.
- [13] Ziyadi M., Sun Y. et al. Example-based named entity recognition. arXiv.2008.10570, 2020, 15 p.
- [14] Li B. Named entity recognition in the style of object detection. arXiv.2101.11122, 2021, 9 p.
- [15] Wang X., Jiang Y. et al. Improving named entity recognition by external context retrieving and cooperative learning, arXiv.2105.03654, 2021, 13 p.
- [16] Fiok K., Karwowski W. et al. Comparing the quality and speed of sentence classification with modern language models. *Applied Sciences*, vol. 10, issue 10, 2020, article no. 3386.
- [17] Глазкова А.В. Тематическая классификация текстовых фрагментов с учетом их ближайшего контекста. Автоматика и телемеханика, вып. 12, 2020 г., стр. 153-172 / Glazkova A. V. Topical classification of text fragments accounting for their nearest context. *Automation and Remote Control*, vol. 81, issue 12, pp. 2262-2276.
- [18] John M., Jayasudha J.S. Enhancing Performance of Deep Learning Based Text Summarizer. *International Journal of Applied Engineering Research*, vol. 12, no. 24, 2017, pp. 15986-15993.
- [19] Akkasi A., Varoğlu E., Dimililer N. Balanced undersampling: a novel sentence-based undersampling method to improve recognition of named entities in chemical and biomedical text. *Applied Intelligence*, vol. 48, issue 8, 2018, pp. 1965-1978.
- [20] Luo Y., Feng H. et al. A novel oversampling method based on SeqGAN for imbalanced text classification. In *Proc. of the 2019 IEEE International Conference on Big Data (Big Data)*, 2019, pp. 2891-2894.
- [21] Li Y., Guo H. et al. Imbalanced text sentiment classification using universal and domain-specific knowledge. *Knowledge-Based Systems*, vol. 160, 2018, pp. 1-15.
- [22] Chawla N.V., Bowyer K.W. et al. SMOTE: synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, vol. 16, 2002, pp. 321-357.
- [23] Taha A.Y., Tiun S. et al. Multilabel Over-sampling and Under-sampling with Class Alignment for Imbalanced Multilabel Text Classification. *Journal of Information and Communication Technology*, vol. 20, issue 3, pp. 423-456.
- [24] Gallant S.I. A practical approach for representing context and for performing word sense disambiguation using neural networks. *Neural Computation*, vol. 3, issue 3, 1991, pp. 293-309.

- [25] Huang E.H., Socher R. et al. Improving word representations via global context and multiple word prototypes. In Proc. of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2012, pp. 873-882.
- [26] Devlin J., Chang M.W. et al. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv.1810.04805, 2019, 16 p.
- [27] Brown T.B., Mann B. et al. Language models are few-shot learners. arXiv.2005.14165, 2020, 75 p.
- [28] Cohan A., Dernoncourt F. A discourse-aware attention model for abstractive summarization of long documents. arXiv.1804.05685, 2018, 7 p.
- [29] ExplainaBoard – Named Entity Recognition URL: <http://explainaBoard.nlpedia.ai/leaderboard/task-ner/>, accessed 16.05.2022.

Информация об авторах / Information about authors

Федор Владимирович КРАСНОВ – доктор технических наук, эксперт департамента информационных технологий управления. Область научных интересов: интеллектуальная аналитика текстов.

Fedor Vladimirovich KRASNOV – Doctor of Technical Sciences, expert of the Department of Information Technologies of Management. Research interests: intellectual analytics of texts.

Ирина Сергеевна СМАЗНЕВИЧ – бизнес-аналитик, департамент семантических систем. Область научных интересов: применение интеллектуальных алгоритмов в прикладных информационных системах.

Irina Sergeevna SMAZNEVICH – Business Analyst, Department of Semantic Systems. Research interests: application of intelligent algorithms in applied information systems.

Елена Николаевна БАСКАКОВА – ведущий системный аналитик, департамент семантических систем. Область научных интересов: применение интеллектуальных алгоритмов в прикладных информационных системах.

Elena Nikolaevna BASKAKOVA – Leading Systems Analyst, Semantic Systems Department. Research interests: application of intelligent algorithms in applied information systems.



Функции потерь для обучения моделей сегментации изображений документов

^{1,2} А.И. Перминов, ORCID: 0000-0001-8047-0114 <perminov@ispras.ru>

^{1,2} Д.Ю. Турдаков, ORCID: 0000-0001-8745-0984 <turdakov@ispras.ru>

¹ О.В. Беляева, ORCID: 0000-0002-6008-9671 <belyaeva@ispras.ru>

¹ Институт системного программирования им. В.П. Иванникова РАН,
109004, Россия, г. Москва, ул. А. Солженицына, д. 25

² Московский государственный университет имени М.В. Ломоносова,
119991, Россия, Москва, Ленинские горы, д. 1

Аннотация. Работа посвящена повышению качества результатов сегментации изображений документов различных научных статей и нормативно-правовых актов нейросетевыми моделями путём обучения с использованием модифицированных функций потерь, учитывающих особенности изображений выбранной предметной области. Проводится анализ существующих функций потерь, а также разработка новых функций, оперирующих, как только координатами ограничивающих прямоугольников, так и использующих информацию о пикселях входного изображения. Для оценки качества выполняется обучение нейросетевой модели сегментации с модифицированными функциями потерь, а также проводится теоретическая оценка с помощью симуляционного эксперимента, показывающего скорость сходимости и ошибку сегментации. В результате исследования созданы быстро сходящиеся функции потерь, улучшающие качество сегментации изображений документов с использованием дополнительной информации о входных данных.

Ключевые слова: сегментация изображений документов; функции потерь; модификация функции потерь

Для цитирования: Перминов А.И., Турдаков Д. Ю., Беляева О.В. Функции потерь для обучения моделей сегментации изображений документов. Труды ИСП РАН, том 34, вып. 2, 2022 г., стр. 89-110. DOI: 10.15514/ISPRAS-2022-34(2)-8

Loss functions for train document image segmentation models

^{1,2} A.I. Perminov, ORCID: 0000-0001-8047-0114 <perminov@ispras.ru>

^{1,2} D.Y. Turdakov, ORCID: 0000-0001-8745-0984 <turdakov@ispras.ru>

¹ O.V. Belyaeva, ORCID: 0000-0002-6008-9671 <belyaeva@ispras.ru>

¹ Ivannikov Institute for System Programming of the Russian Academy of Sciences,
25, Alexander Solzhenitsyn st., Moscow, 109004, Russia

² Lomonosov Moscow State University,
GSP-1, Leninskie Gory, Moscow, 119991, Russia

Abstract. The work is devoted to improving the quality of the results of image segmentation of documents of various scientific articles and legal acts by neural network models by learning using modified loss functions that take into account the features of images of the selected subject area. The analysis of existing loss functions is carried out, as well as the development of new functions that operate both with the coordinates of the bounding boxes and using information about the pixels of the input image. To assess the quality, a neural network segmentation model with modified loss functions is trained, and a theoretical assessment is carried out using a simulation experiment showing the convergence rate and segmentation error. As a result of the study,

rapidly converging loss functions were created that improve the quality of document image segmentation using additional information about the input data.

Keywords: document image segmentation; loss functions; loss function modifications

For citation: Perminov A.I., Turdakov D. Y., Belyaeva O.V. Loss functions for train document image segmentation models. *Trudy ISP RAN/Proc. ISP RAS*, vol. 34, issue 2, 2022, pp. 89-110 (in Russian). DOI: 10.15514/ISPRAS-2022-34(2)-8

1. Введение

Огромное количество различных документов хранятся в электронном виде. Некоторые из них содержат копируемый текст и предоставляют возможность навигации по нему с помощью оглавления и ссылок. Однако есть документы (например, книги, отчеты, квитанции и т.д.), которые представляют собой сканированные копии бумажных документов. Такие документы не имеют текстового слоя. Таким образом, документы не упорядочены, в них сложно искать информацию и невозможно напрямую извлечь из них текст.

В связи с тем, что существует необходимость систематизировать и анализировать информацию в документах, предполагается, что документы имеют определенную структуру. Это означает, что существуют определенные шаблоны, по которым создаются документы. Это предположение позволяет говорить о возможности автоматического анализа электронного документа и извлечения логической структуры.

Практически для всех документов общим шаблоном можно считать геометрическую организацию – шрифты, отступы, линии, более крупные блоки – абзацы текста, заголовки, изображения и т.д. Для выделения таких признаков используется сегментация страницы документа.

В настоящее время для сегментации используются нейронные сети, поскольку они позволяют производить высококачественную обработку изображений. Для обучения нейросетей используются алгоритмы минимизации некоторой функции, способной оценить качество сети. Такие функции называются функциями потерь. Обычно, к ним предъявляют требование на дифференцируемость и непрерывность в некоторой области, однако существуют функции, для которых это свойство не выполняется в некоторых точках. В силу работы нейронных сетей с компьютерными числами, не являющимися алгебраически точным математическим представлением, это оказывается вполне допустимым.

При разработке функций потерь зачастую преследуется цель получить функцию, оценивающую качество работы оптимизируемой модели, однако далеко не всегда уделяется должное внимание анализу прочих свойств получаемых функций, как, например, градиенты, гладкость, специфические свойства данных предметной области и т.д. Функция потерь может иметь высокую скорость сходимости и показывать близкие к нулю значения после оптимизации, однако в результате оптимизации модель может работать недостаточно хорошо из-за того, что не были учтены какие-то специфические особенности входных данных. Разработка и исследование функций, специфических для конкретной задачи может существенно повысить качество получаемой модели. При этом, возможно, и существующие функции могут помочь достичь такого же качества, но, например, за большее время, из-за чего целесообразнее будет применить модифицированный вариант.

Для сегментации изображений документов в большинстве решений используются те же нейросетевые модели, что и для сегментации естественных изображений - объектов реального мира. Поскольку естественные изображения лишены общей структуры, то и модели сегментации не учитывают дополнительных свойств, присущих изображениям текстовых документов. Помимо этого, такие модели плохо и долго обучаются, а результаты их обучения на изображениях документов приводят к необходимости использования дополнительной обработки.

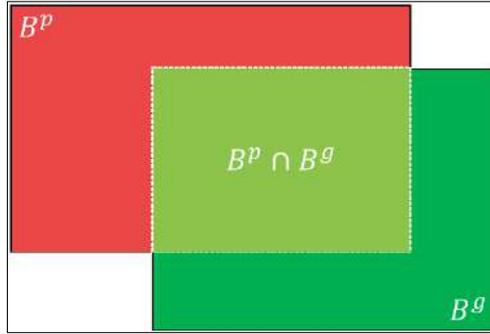


Рис. 1. Пример двух ограничивающих прямоугольников
 Fig. 1. Two bounding boxes example

Модели сегментации оперируют координатами двух ограничивающих прямоугольников (рис. 1): $B^g = (x_1^g, y_1^g, x_2^g, y_2^g)$ – ожидаемым и $B^p = (x_1^p, y_1^p, x_2^p, y_2^p)$ – предсказанным, с помощью которых находятся площади самих прямоугольников и их пересечения. Основной задачей сегментации является регрессия координат предсказанной области к целевой. Оценка качества сегментации выполняется путём анализа значений получаемых площадей. Для сегментации произвольных изображений этого вполне достаточно, однако сегментация изображений документов накладывает такие ограничения, как отсутствие обрезки текста (рис. 2 (в)) или наложение на другой блок текста (рис. 2 (б)). Ситуация, в которой текст выделен с учётом этих ограничений, но при этом взято больше фона (рис. 2 (а)), также считается хорошей, в отличие от распространённых функций потерь. В то же время координатные функции вынуждают модель сегментации подстраиваться под геометрию того домена, на котором производится обучение. Из-за этого обработка документов с другим расположением элементов (но всё ещё являющихся манхэттенскими) приводит к серьёзной потере качества вплоть до нулевого.

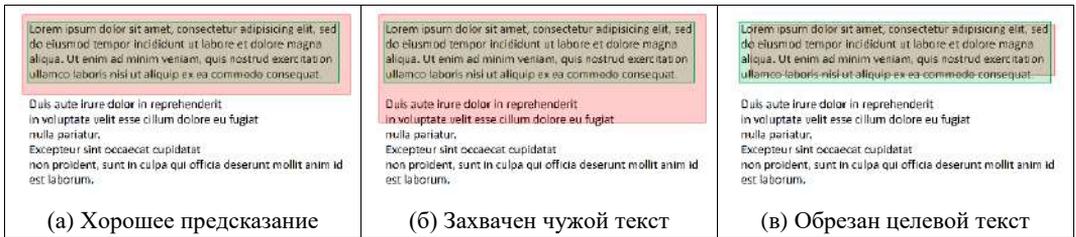


Рис. 2. Пример предсказаний с учётом перечисленных свойств
 Fig. 2. An example of predictions taking into account the listed properties

По этой причине далее разрабатываются функции потерь, учитывающие не только геометрию областей, но и особенности сегментации изображений документов – возможность захватить больше фона, запрет на обрезку текста и захват чужого текста.

2. Существующие решения

В работах [1], [2], [3] к функции потерь добавляют коэффициенты, дающие ненулевые градиенты при отсутствии пересечения, чтобы избежать основной проблемы распространённой IoU функции.

В работе [3] предлагается новая функция потерь SCA для замены классической функции потерь – IoU. Разработанная функция, вместо оценки отношения площади пересечения прямоугольников к площади объединения, как это делает IoU, независимо оценивает отношения размеров пересечения и минимального покрывающего прямоугольников. Исследователи обучают различные нейросетевые модели с её использованием на нескольких

наборах данных изображений общего назначения и сравнивают результаты на метриках усреднённой точности (mean average precision) с различными порогами. По результатам оценки качества большинство моделей показывают более высокие результаты с модифицированной функцией потерь.

Исследователи в [4] активно исследуют проблемы IoU функции и скорость её сходимости и в результате предлагают семейство функций, называемое " α -IoU". Привычная IoU функция возводится в степень α и в результате проводимых экспериментов демонстрируется более высокая сходимость при определённых значениях подбора введённого коэффициента.

В работе [5] авторы исследуют гладкие версии функции IoU, используют совместно с перекрёстной энтропией с целью повышения качества классификации объекта. Анализируя градиенты получаемой функции, исследователям удаётся уменьшить количество выбросов и разрывов, а также повысить точность сегментации на классических наборах данных.

Авторы работы [6] анализируют градиенты модификаций IoU и, жертвуя инвариантностью относительно масштабирования и используя разность площадей двух прямоугольников, получают функцию, способную хорошо различать очень похожие и расположенные близко ограничивающие прямоугольники. При этом градиенты такой функции получаются более гладкими, что приводит к высокой скорости сходимости моделей.

Описанные выше модификации функций потерь опираются лишь на геометрические признаки или математические свойства. Для учёта особенностей входных данных их использование невозможно, а потому необходимо разработать собственные функции потерь.

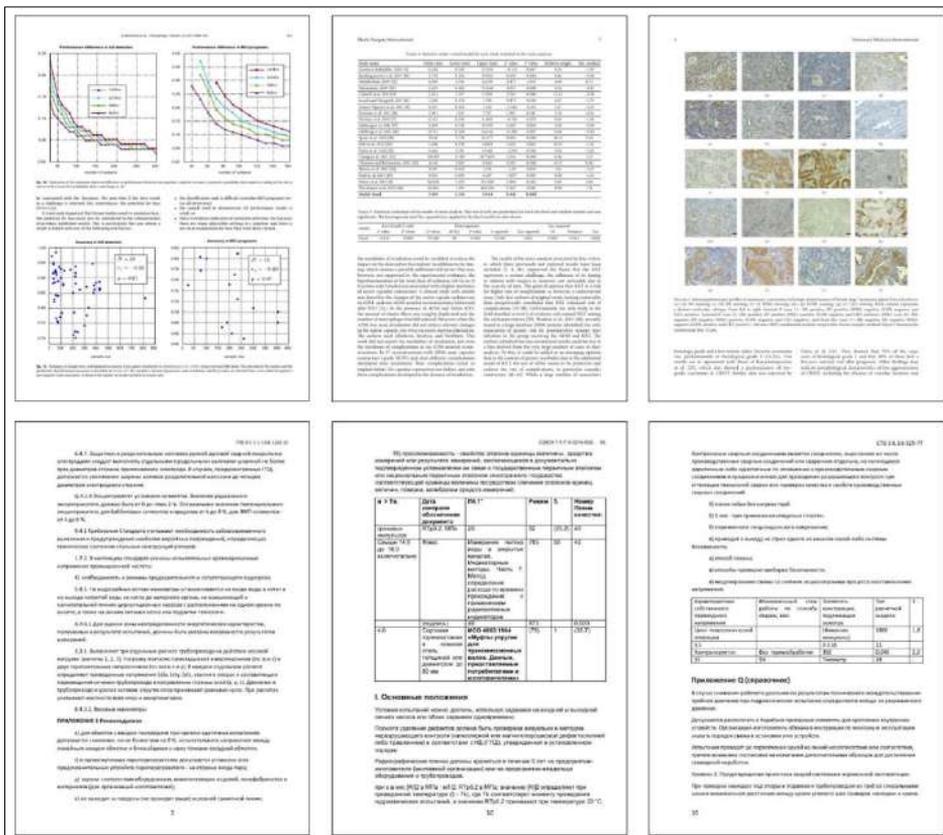


Рис. 3. Примеры реальных документов
Fig. 3. Real documents examples

3. Изображения документов для сегментации

В качестве входных данных рассматриваются отсканированные изображения различных статей и нормативно-правовых документов с тёмным текстом и светлым однородным фоном (рис. 3).

Сканированные документы, как правило, характеризуются высоким качеством символов, белым фоном, низким уровнем шума и манхэттенским стилем оформления. Манхэттенский документ – это документ, в котором границы всех блоков прямые (каждый блок или прямоугольный, или представляет собой несколько прямоугольников, у которых некоторые вершины и части сторон общие).

4. Исследование функций потерь

В настоящее время наиболее качественной моделью сегментации изображений является архитектура YOLO [7] пятой версии. Данная модель использует комбинированную функцию потерь – для регрессии используется CIoU [1], а для классификации используется перекрёстная энтропия. Каждую из составляющих легко модифицировать для собственных экспериментов, чего нельзя сказать о большинстве других используемых моделей. Также эта модель сегментации довольно быстро обучается. По вышеперечисленным причинам была выбрана модель архитектуры YOLO.

4.1 Разработка метрик, учитывающих особенности документов

Свойства метрик, учитывающие особенности изображений документов:

- содержимое целевой области должно полностью находиться внутри предсказанной области – обрезка текста должна быть сведена к минимуму;
- содержимое, относящееся к другому объекту не должно входить в выделенную область – чужой текст не должен быть захвачен;
- размер предсказанной области может быть, как меньше, так и больше определённой, если выполнены указанные выше правила – это позволяет выделять больше фоновых пикселей, что не является критичным, и в то же время позволяет выделять область ровно по контенту, если фоновых пикселей было слишком много в целевой области;
- в случае полного совпадения областей значение метрики равно единице, во всех остальных случаях значение не превышает единицы.

4.1.1 PIoU – Pixel intersection over union

В качестве самой простой функции потерь, удовлетворяющей всем условиям можно взять следующую:

$$PIoU = \frac{Black(img, |B^g \cap B^p|)}{Black(img, |B^g \cup B^p|)}. \quad (1)$$

$Black(img, v)$ – количество тёмных пикселей в области v изображения img . Под тёмным пикселем будем считать тот, чья яркость меньше некоторого порога $threshold$. Поскольку сегментация производится на изображениях в RGB пространстве, то в качестве яркости можно использовать усреднение по трём каналам:

$$Brightness(pixel) = \frac{r + g + b}{3} \quad (2)$$

Областью будем считать прямоугольник, определяемый координатами левого верхнего и правого нижнего углов – (x_1, y_1, x_2, y_2) .

Таким образом, функция подсчёта количества тёмных пикселей может быть описана следующим образом:

$$Black = \sum_{x=x_1}^{x_2} \sum_{y=y_1}^{y_2} \delta(x, y), \tag{3}$$

$$\delta(x, y) = \begin{cases} 1, \text{ если } Brightness(x, y) < threshold \\ 0, \text{ иначе} \end{cases} \tag{4}$$

4.1.2 BWIoU – Black and White intersection over union

Полезным свойством PIoU (1) является выполнение всех перечисленных условий, однако при исследовании её свойств было установлено (табл. 1), что значения метрики при небольшом наложении на чужой текст или малое обрезание целевой области лишь немного уменьшает значение метрики. Поэтому было поставлено дополнительное свойство – более резкое изменение значения при описанных ситуациях и была придумана новая метрика, учитывающая не только соотношение тёмных пикселей, но и светлых (не тёмных), а также штрафующая за тёмные пиксели вне пересечения:

$$BWIoU = \frac{w \cdot Black_{\cap} + (1 - w) \cdot White_{\cap}}{w \cdot Black_{\cup} + (1 - w) \cdot White_{\cup} + \lambda(Black_{\cup} - Black_{\cap})} \tag{5}$$

В качестве значений весовых коэффициентов выбраны $w = 0.7, \lambda = 10$. $Black_{\cap}$ и $Black_{\cup}$ – количество тёмных пикселей в пересечении и объединении областей соответственно. Аналогично, $White_{\cap}$ и $White_{\cup}$ – количество светлых пикселей в пересечении и объединении областей (рис. 4).

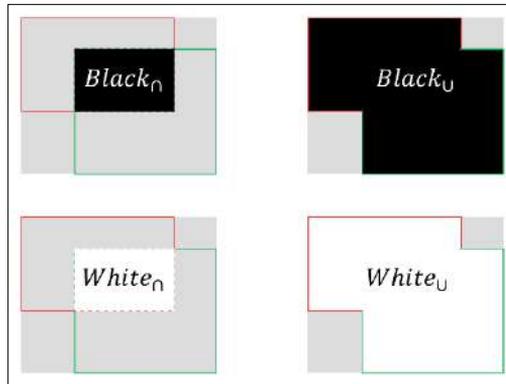


Рис. 4. Визуализация пиксельных метрик
Fig. 4. Pixel metrics visualization

4.1.3 Weighted BWIoU

Обычная функция BWIoU (5) значительно лучше учитывает обрезку текста и наложение на другой текст, однако при увеличении фона довольно сильно уменьшает своё значение (табл. 1). Это происходит в основном из-за подобранных значений коэффициента w . Величину весового коэффициента стоит выбирать адаптивно в зависимости от параметров выделенных и ожидаемых областей, а потому была получена новая метрика:

$$Weighted\ BWIoU = \frac{w_{\cap} \cdot Black_{\cap} + (1 - w_{\cap}) \cdot White_{\cap}}{w_{\cup} \cdot Black_{\cup} + (1 - w_{\cup}) \cdot White_{\cup} + \lambda(Black_{\cup} - Black_{\cap})} \tag{6}$$

В этой метрике весовые коэффициенты для пересечения и для объединения вычисляются следующим образом:

$$w_{\cap} = \frac{White_{\cap}}{Area_{\cap}}, w_{\cup} = \frac{White_{\cup}}{Area_{\cup}} \tag{7}$$

Использование адаптивных коэффициентов позволило практически полностью не реагировать на изменения области лишь в фоновых пикселях, а также более резко реагировать на обрезку или наложение.

Табл. 1. Пример пиксельных метрик при различных положениях областей
 Table 1. Example of a pixel metric for various positions of bounding boxes

<p>for metastasis, the patient is offered big radical lymph node dissection. Department receives referrals regarding melanoma from GPs, dermatology clinics, or other specialists for treatment. If a biopsy is made prior to an offered further treatment in the histological examination of the skin lesions to our department in that a relatively large proportion with skin lesions other than improving diagnostic accuracy in section. Therefore, the objective is to analyze the referral patterns in the FFRS of suspected for to clarify from which health care and how many of the referred patients (MDS), invasive cutaneous melanoma skin cancer (NMIBC), or</p> <p>2.4. Statistical Method. All data were tested for distribution of normality and treated statistically accordingly. Comparison between groups was tested by two-tailed t-test and in the case of skewness or low numbers among groups the Mann-Whitney U-test for nonparametric data was performed. Sta-</p> <p>IoU = 0.7906 PIoU = 0.9299 BWIoU = 0.7083 Weighted BWIoU = 0.6675</p>	<p>for metastasis, the patient is offered big radical lymph node dissection. Department receives referrals regarding melanoma from GPs, dermatology clinics, or other specialists for treatment. If a biopsy is made prior to an offered further treatment in the histological examination of the skin lesions to our department in that a relatively large proportion with skin lesions other than improving diagnostic accuracy in section. Therefore, the objective is to analyze the referral patterns in the FFRS of suspected for to clarify from which health care and how many of the referred patients (MDS), invasive cutaneous melanoma skin cancer (NMIBC), or</p> <p>2.4. Statistical Method. All data were tested for distribution of normality and treated statistically accordingly. Compar-</p> <p>IoU = 0.7403 PIoU = 1 BWIoU = 0.7581 Weighted BWIoU = 0.9792</p>	<p>for metastasis, the patient is offered big radical lymph node dissection. Department receives referrals regarding melanoma from GPs, dermatology clinics, or other specialists for treatment. If a biopsy is made prior to an offered further treatment in the histological examination of the skin lesions to our department in that a relatively large proportion with skin lesions other than improving diagnostic accuracy in section. Therefore, the objective is to analyze the referral patterns in the FFRS of suspected for to clarify from which health care and how many of the referred patients (MDS), invasive cutaneous melanoma skin cancer (NMIBC), or</p> <p>2.4. Statistical Method. All data were tested for distribution of normality and treated statistically accordingly. Compar-</p> <p>IoU = 0.8177 PIoU = 0.9329 BWIoU = 0.7274 Weighted BWIoU = 0.6793</p>
---	---	--

4.2 Метрика, основанная на оптическом распознавании символов

Поскольку одной из основных целей сегментации изображений документов является анализ содержимого, который практически невозможен без средств оптического распознавания символов (OCR), необходимо выполнять сегментацию таким образом, чтобы результат получения текста средствами OCR максимально соответствовал тексту на анализируемом изображении. Для оценки качества можно получить текст, находящийся в целевом и предсказанном ограничивающих прямоугольниках, сопоставить области друг с другом и сравнить похожесть текстов с помощью редакционного расстояния, например, с помощью расстояния Левенштейна:

$$ratio(str_1, str_2) = 1 - \frac{D_{Levenshtein}(str_1, str_2)}{len(str_1) + len(str_2)} \quad (8)$$

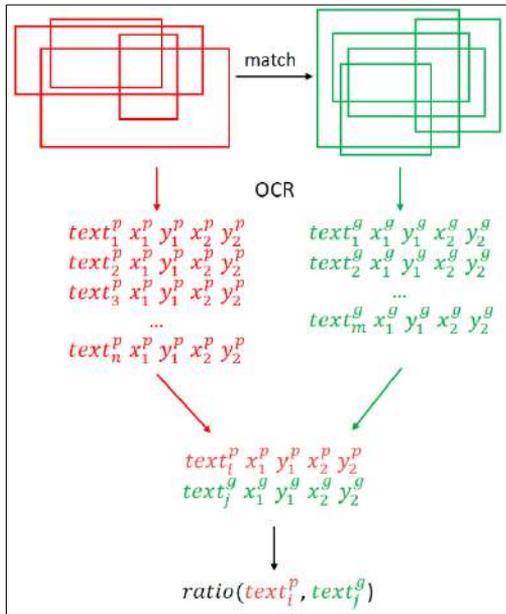


Рис. 5. Схема работы метрики OCR
 Fig. 5. OCR metric flow diagram

В качестве метрики для набора данных выбирается среднее значение *ratio* (формула 8):

$$Metric_{OCR} = \frac{1}{N_p} \cdot \sum_{i=1}^{N_p} ratio(OCR(Nearest(bboxes_g, bbox_i^p), OCR(bbox_i^p))) \quad (9)$$

Здесь $OCR(bbox)$ - получение текста средствами оптического распознавания символов в области, соответствующей прямоугольнику $bbox$. $Nearest(bboxes_g, bbox_i^p)$ – наиболее близкий прямоугольник среди целевых $bboxes_g$ к предсказанному прямоугольнику $bbox_i^p$. В качестве меры близости выбирается прямоугольник, имеющий наибольшую площадь пересечения. N_p – количество предсказанных прямоугольников. Схематично процесс вычисления данной метрики приведён на рис. 5.

4.3 Использование метрик в качестве функций потерь

Чтобы метрику можно было использовать как функцию потерь, необходимо иметь возможность вычислить градиент по входным параметрам такой функции. Входными параметрами являются координаты ограничивающих прямоугольников. Для оптимизации модели сегментации необходимо вычислять градиент по координатам предсказанной области. К сожалению, предложенные функции не имеют градиента по входным параметрам, поскольку для получения оценки качества вместо значений координат, используется взятие подобласти изображения – срез – не дифференцируемая операция, поскольку координаты ограничивающих прямоугольников являются вещественными числами (хранятся нормализованные координаты относительно размеров изображения), а для получения области на изображении необходимо использовать лишь целое число.

В результате метрика даёт некоторое число без градиента – константу, характеризующую предсказанную область. Поскольку получаемое число для каждой области своё, то его можно использовать как весовой коэффициент к дифференцируемой функции, например, к IoU. Таким образом, итоговая функция потерь будет выглядеть следующим образом:

$$L_{pixel} = 1 - f_{diff} \cdot f_{pixel} \quad (10)$$

где f_{diff} – дифференцируемая координатная метрика, f_{pixel} – значение пиксельной метрики. Таким образом, пиксельный коэффициент будет являться регуляризатором – для областей без особенностей практически ничего не будет меняться, а для областей с наложением или обрезкой значение функции потерь будет значительно больше.

4.4 Оценка качества функций потерь

Для оценки качества работы функций потерь до непосредственно обучения модели сегментации исследователи проводят эксперимент-симуляцию (алгоритм 1): выбирается некоторое количество якорей (ограничивающих рамок, относительно которых будет проводиться оптимизация смещения) и сетка на области единичного квадрата. Случайным образом создаются целевые ограничивающие прямоугольники. Аналогично случайным образом для каждого якоря создаются случайные предсказанные прямоугольники в каждой точке выбранной сетки и запускается итерационный процесс: для каждой пары предсказанного и целевого прямоугольников вычисляется значение функции потерь, градиенты по координатам предсказанного прямоугольника и ошибка регрессии (например, сумма модулей разности координат или пиксельная метрика, описанная выше), после чего координаты обновляются методом градиентного спуска и процесс запускается сначала.

Результатом работы такой симуляции является оценка скорости сходимости функции потерь, по которой можно сделать выводы о целесообразности применения такой функции в обучении реальной модели. Основным достоинством такого эксперимента, что он не занимает столько времени как обучение полноценной модели на (обычно) большом наборе данных.

Input:

N_p – количество предсказанных прямоугольников,

N_g – количество целевых прямоугольников,

$B^p = \{B_i^p\}_{i=1}^{N_p}$ – предсказанные прямоугольники,

$B^g = \{B_j^g\}_{j=1}^{N_g}$ – целевые прямоугольники,

T – количество итераций повторения,

η – скорость градиентного спуска, по умолчанию 0.01

Output: средняя ошибка регрессии всех предсказанных прямоугольников ко всем целевым

```

1: for t= 1...T do
2:   for i= 1...Np do
3:     for j= 1...Ng do
4:       Loss = L(Bi,tp, Bjg)
5:       ∇Bi,tp = ∂Loss / ∂Bi,tp
6:       Bi,t+1p = Bi,tp - η∇Bi,tp
7:       RegressionErrort = |Bip - Bjg|
8:     end for
9:   end for
10: end for
11: return mean(RegressionErrort)
    
```

Алгоритм 1. Симуляционный эксперимент
Algorithm 1. Simulation experiment

Для построенных функций был проведён описанный эксперимент. В качестве дифференцируемой функции потерь была использована модификация IoU – CIoU [1].

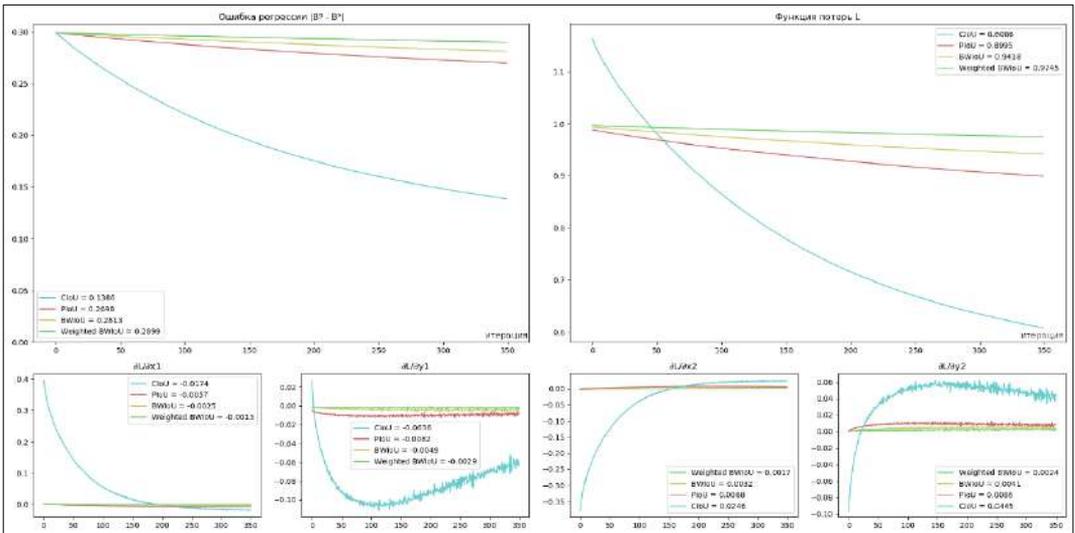


Рис. 6. Результаты симуляционного эксперимента пиксельных функций
Fig. 6. Results of simulation experiment of pixel functions

Симуляция для построенных функций (рис. 6) показала очень медленную сходимость, из-за чего, несмотря на качество оценки, в данном виде использовать их невыгодно. При этом нет никакого смысла использовать пиксельную оценку в то время, когда целевая и предсказанная области находятся далеко друг от друга – имеют маленькое пересечение. Решением данной

проблемы являлась бы функция, которая включала пиксельную метрику лишь ближе к концу, а в остальное время работала бы обычная дифференцируемая функция без изменений.

4.5 Усовершенствованная пиксельная функция потерь

Чтобы избежать проблемы медленной сходимости, но оставить преимущества пиксельного коэффициента, была введена новая функция потерь:

$$L_{fast} = 1 - f_{diff} \cdot (f_{pixel} + 1 - f_{diff}^*). \tag{11}$$

Здесь f_{diff}^* – значение функции f_{diff} , не имеющее градиента. Это позволяет не менять направление градиента, а лишь его амплитуду, аналогично изначальному варианту. Когда два ограничивающих прямоугольника далеко друг от друга или же практически не пересекаются, коэффициент $f_{pixel} + 1 - f_{diff}^*$ практически равен 1, поскольку значение f_{pixel} в такой области либо равно нулю либо ничтожно мало, как и f_{diff}^* . Когда же области находятся близко друг к другу, $1 - f_{diff}^*$ стремится к нулю, благодаря чему основной вклад вносит f_{pixel} . Таким образом большую часть времени сходится обычная функция потерь и лишь под конец пиксельная. Результаты сходимости модифицированных функций потерь представлены на рис.7.

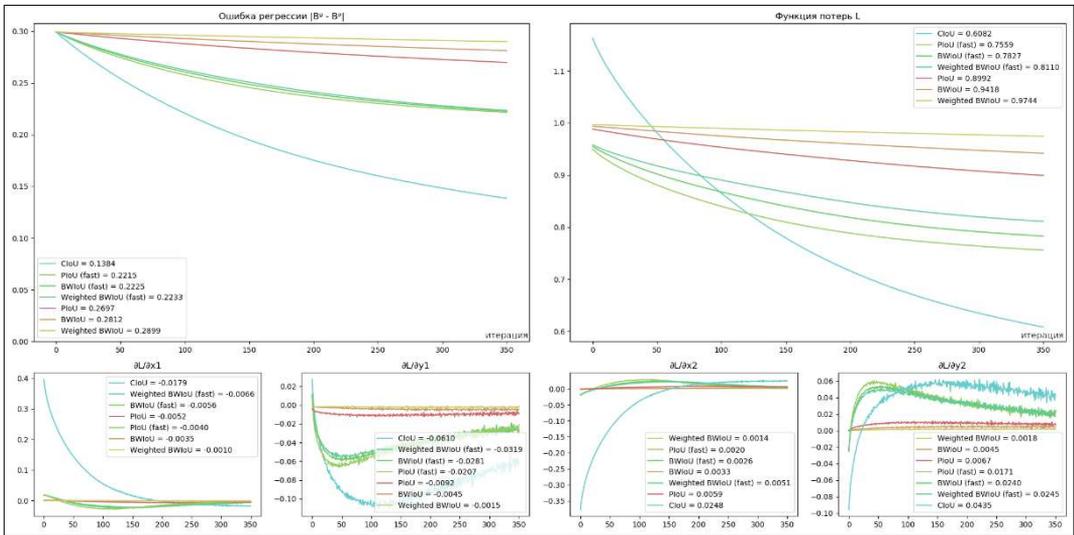


Рис. 7. Результаты симуляционного эксперимента усовершенствованных пиксельных функций
 Fig. 7. Results of the simulation experiment of advanced pixel functions

Скорость сходимости модифицированных функций потерь ($PIoU (fast)$, $BWIoU (fast)$ и $Weighted BWIoU (fast)$) выше по сравнению с исходными версиями ($PIoU (1)$, $BWIoU (5)$ и $Weighted BWIoU (6)$), однако всё ещё довольно низкая. Обучение сегментационной модели показало, что всё ещё нужно довольно много итераций обучения для достижения качественных результатов.

4.6 Покоординатная пиксельная функция потерь

Одним из главных недостатков описанных выше пиксельных функций является независимость коэффициента масштабирования от координат. Если одна из координат уже находится в оптимальном, с точки зрения пиксельного окружения, положении, то, вероятнее всего, её уже не нужно менять. Чтобы этого избежать, но при этом всё ещё иметь возможность вычислять градиент по координатам, можно добавить к координатной функции потерь следующую функцию:

$$L_{pixel} = k_{x_1} \cdot x_1^p + k_{y_1} \cdot y_1^p + k_{x_2} \cdot x_2^p + k_{y_2} \cdot y_2^p. \quad (12)$$

Несложно убедиться в следующих равенствах:

$$\frac{\partial L_{pixel}}{\partial x_1^p} = k_{x_1}, \quad \frac{\partial L_{pixel}}{\partial y_1^p} = k_{y_1}, \quad \frac{\partial L_{pixel}}{\partial x_2^p} = k_{x_2}, \quad \frac{\partial L_{pixel}}{\partial y_2^p} = k_{y_2}. \quad (13)$$

В качестве значений коэффициентов k_{x_1} , k_{y_1} , k_{x_2} и k_{y_2} можно вычислять отношение количества тёмных пикселей в областях, получающихся при сдвиге координат на Δ пикселей в ту или иную сторону, например, так:

$$x_{1_{left}} = black(x_1 - \Delta, y_1, x_2, y_2), \quad (14) \quad y_{1_{top}} = black(x_1, y_1 - \Delta, x_2, y_2), \quad (15)$$

$$x_{1_{right}} = black(x_1 + \Delta, y_1, x_2, y_2), \quad (16) \quad y_{1_{bottom}} = black(x_1, y_1 + \Delta, x_2, y_2), \quad (17)$$

$$x_{2_{left}} = black(x_1, y_1, x_2 - \Delta, y_2), \quad (18) \quad y_{2_{top}} = black(x_1, y_1, x_2, y_2 - \Delta), \quad (19)$$

$$x_{2_{right}} = black(x_1, y_1, x_2 + \Delta, y_2), \quad (20) \quad y_{2_{bottom}} = black(x_1, y_1, x_2, y_2 + \Delta), \quad (21)$$

$$k_{x_1} = k \cdot \frac{x_{1_{left}} - x_{1_{right}}}{black(x_1, y_1, x_2, y_2)}, \quad (22) \quad k_{y_1} = k \cdot \frac{y_{1_{top}} - y_{1_{bottom}}}{black(x_1, y_1, x_2, y_2)}, \quad (23)$$

$$k_{x_2} = k \cdot \frac{x_{2_{left}} - x_{2_{right}}}{black(x_1, y_1, x_2, y_2)}, \quad (24) \quad k_{y_2} = k \cdot \frac{y_{2_{top}} - y_{2_{bottom}}}{black(x_1, y_1, x_2, y_2)}. \quad (25)$$

В качестве значений Δ берётся небольшое значение – примерно 3-7 пикселей. Коэффициент k позволяет ограничивать влияние пиксельной составляющей. Эксперименты показали, что для документов оптимальнее всего использовать область в 3 пикселя и $k = 0.5$. Схематично данная функция представлена на рис. 8.

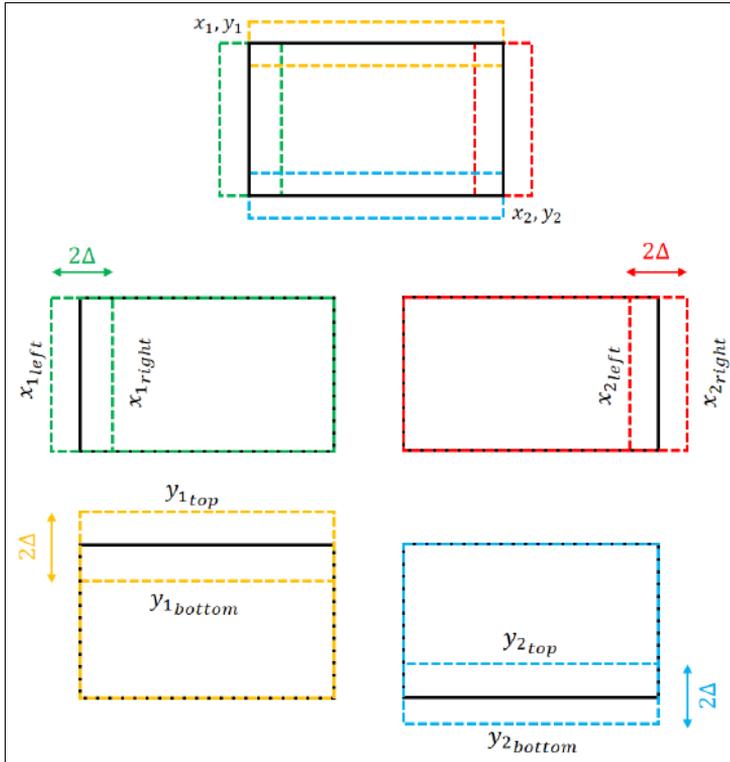


Рис. 8. Схема работы покоординатной пиксельной функции потерь
Fig. 8. Scheme of operation of the coordinate pixel loss function

4.7 Анализ координатных функций потерь

При анализе таких координатных функций потерь, как GIoU [3], DIoU [1], SCA [5] и т.д., было обнаружено, что в подавляющем большинстве случаев в первую очередь выполняется сходимость одного из углов ограничивающего прямоугольника (то есть координаты x_1, y_1 или x_2, y_2) и лишь затем начинает сходиться другой угол. Данный эффект объясняется малыми значениями градиента анализируемых функций потерь и негативно сказывается на сходимости в целом. При этом у функции SCA [5] данный эффект выражен менее сильно, что служит сигналом к более подробному её анализу.

Также большинство функций потерь имеет наибольшее значение градиента при очень близком расположении прямоугольников, из-за чего выбор большей скорости обучения приводит к осцилляциям градиента и не даёт сойтись областям.

Немало проблем дополнительно доставляет ненулевой градиент для уже сошедшихся координат – если прямоугольник совпал по каким-то (но не всем) координатам, градиент анализируемых функций потерь по этим координатам отличен от нуля, причём зачастую довольно сильно, что также приводит к осцилляциям и негативно сказывается на сходимости. Проведённый анализ приводит к поискам оптимальной координатной функции потерь (рис. 9), отвечающей следующим свойствам:

- максимальная амплитуда градиента при большом расстоянии между областями;
- нелинейно уменьшающаяся амплитуда градиента по мере приближения к целевой области;
- максимально близкий к нулю или равный ему градиент для совпадающих координат;
- максимально похожая форма (с точки зрения размеров).

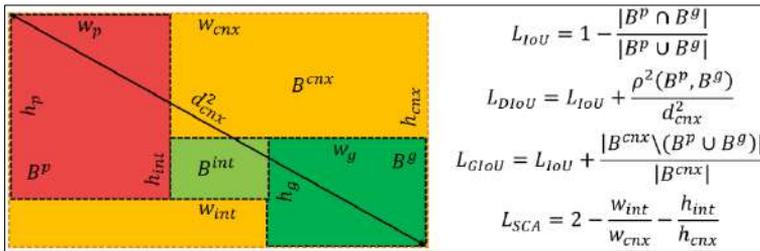


Рис. 9. Используемые координатные функции
 Fig. 9. Used coordinate functions

4.8 Усовершенствованные координатные функции потерь

Были разработаны две функции потерь, основанные на SCA. Общим для них является добавление нормализованного квадрата расстояния между центрами, как это сделано в DIoU [1]:

$$L_{center} = \frac{((x_c^g - x_c^p)^2 + (y_c^g - y_c^p)^2)}{d_{cnx}^2} \tag{26}$$

$(x_c^g, y_c^g), (x_c^p, y_c^p)$ – координаты центров целевого и предсказанного прямоугольника, а d_{cnx}^2 – квадрат диагонали наименьшего покрывающего прямоугольника для целевого и предсказанного прямоугольников. Данный коэффициент позволяет быстрее сдвигать все 4 координаты предсказанного прямоугольника к целевому.

4.8.1 Степенная функция потерь отношения размеров

В SCA используется слагаемое $L_{S0} = 2 - S0$, показывающее сумму отношений ширин и высот прямоугольника пересечения и покрывающим прямоугольником. Возведение данного

коэффициента в степень больше единицы значительно увеличивает сходимость получающейся функции. Таким образом получается функция ISCA:

$$L_{ISCA} = L_{SO}^n + L_{center}. \quad (27)$$

4.8.2 Функция потерь, оптимизирующая форму

Проблема сходимости в первую очередь лишь одного из углов решается подстраиванием формы предсказанной области к целевой и дальнейшее её перемещение. Для получения такой же формы, как и у целевого прямоугольника, необходимо сравнивать ширины и высоты между собой. Однако необходимо получать значения от 0 до 1 для нормировки, из-за чего деление предсказанных размеров на ожидаемые невозможно, так как получающиеся значения могут быть гораздо больше единицы. Этой проблемы можно избежать, если делить минимальный из размеров на максимальный. Таким образом для сохранения формы можно ввести такой коэффициент:

$$k_{form} = \frac{\min(w^g, w^p)}{\max(w^g, w^p)} + \frac{\min(h^g, h^p)}{\max(h^g, h^p)}, \quad (28)$$

w^g, w^p – ширины целевого и предсказанного прямоугольников, а h^g, h^p – высоты соответственно.

Каждое из слагаемых гарантированно не превышает единицы, а потому их сумма не превышает двух. Таким образом для получения функции потерь из него достаточно вычесть его из двух:

$$L_{form} = 2 - k_{form} = 2 - \frac{\min(w^g, w^p)}{\max(w^g, w^p)} - \frac{\min(h^g, h^p)}{\max(h^g, h^p)}. \quad (29)$$

Итоговая функция потерь примет следующий вид:

$$L_{FM} = L_{SO} + L_{form} + L_{center}. \quad (30)$$

4.8.3 Анализ сходимости полученных координатных функций

Для полученных функций потерь был проведён описанный выше симуляционный эксперимент (рис. 10), показавший эффективность новых функций

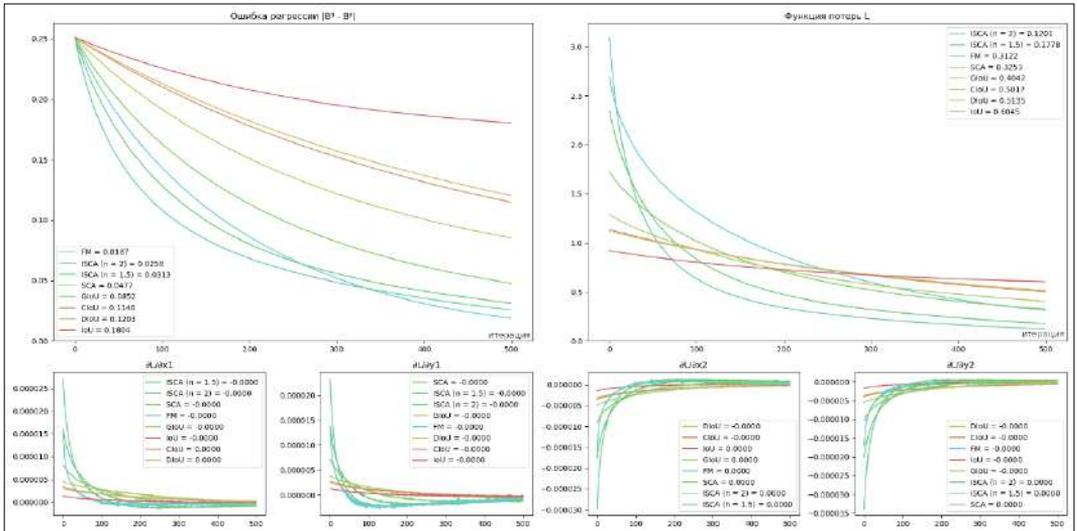


Рис. 10. Анализ сходимости новых координатных функций
 Fig. 10. Analysis of the convergence of new coordinate functions

При большом расстоянии между областями градиенты полученных функций (*ISCA* (формула 27), *FM* (формула 30)) по-прежнему не максимальны и изменяются нелинейно по мере приближения к целевой области, а также имеют ненулевой градиент при совпадении координат, однако в теории позволяют выполнять регрессию двух ограничивающих прямоугольников заметно быстрее и при этом позволяют сохранять форму (функция L_{FM}) по сравнению с *SCA*, *GIoU* [3], *CIoU* [1], *DIoU* [1] и *IoU*.

4.9 Анализ градиентов существующих координатных функций потерь

Для поиска функции потерь, способной удовлетворять описанным требованиям к градиентам, следует проанализировать форму градиентов имеющихся функций потерь. Для этого продифференцируем по x_1^p , y_1^p , x_2^p , y_2^p переменным составляющие следующих функций:

- $L_{IoU} = 1 - IoU$;
- $L_{SCA} = L_{SO} + \alpha \cdot L_{cd}$;
- $L_{FM} = L_{SO} + L_{form} + L_{center}$.

4.9.1 Градиенты L_{IoU}

$$L_{IoU} = 1 - \frac{S_{\cap}}{S_{\cup}}; \quad (31)$$

$$\frac{\partial L_{IoU}}{\partial x_1^p} = -\frac{1}{S_{\cup}} \left(h_p \cdot IoU - \frac{1 - \text{sgn}(x_1^g - x_1^p)}{2} \cdot H(w_{\cap}) \cdot \max(h_{\cap}, 0) \cdot (1 + IoU) \right); \quad (32)$$

$$\frac{\partial L_{IoU}}{\partial y_1^p} = -\frac{1}{S_{\cup}} \left(w_p \cdot IoU - \frac{1 - \text{sgn}(y_1^g - y_1^p)}{2} \cdot H(h_{\cap}) \cdot \max(w_{\cap}, 0) \cdot (1 + IoU) \right); \quad (33)$$

$$\frac{\partial L_{IoU}}{\partial x_2^p} = \frac{1}{S_{\cup}} \left(h_p \cdot IoU - \frac{1 + \text{sgn}(x_2^g - x_2^p)}{2} \cdot H(w_{\cap}) \cdot \max(h_{\cap}, 0) \cdot (1 + IoU) \right); \quad (34)$$

$$\frac{\partial L_{IoU}}{\partial y_2^p} = \frac{1}{S_{\cup}} \left(w_p \cdot IoU - \frac{1 + \text{sgn}(y_2^g - y_2^p)}{2} \cdot H(h_{\cap}) \cdot \max(w_{\cap}, 0) \cdot (1 + IoU) \right). \quad (35)$$

4.9.2 Градиенты L_{SO}

$$L_{SO} = 2 - \frac{w_{\cap}}{w_{cnx}} - \frac{h_{\cap}}{h_{cnx}}; \quad (36)$$

$$\frac{\partial L_{SO}}{\partial x_1^p} = \frac{1}{w_{cnx}} \left(\frac{1 + \text{sgn}(x_1^g - x_1^p)}{2} \cdot \frac{w_{\cap}}{w_{cnx}} - \frac{1 - \text{sgn}(x_1^g - x_1^p)}{2} \right); \quad (37)$$

$$\frac{\partial L_{SO}}{\partial y_1^p} = \frac{1}{h_{cnx}} \left(\frac{1 + \text{sgn}(y_1^g - y_1^p)}{2} \cdot \frac{h_{\cap}}{h_{cnx}} - \frac{1 - \text{sgn}(y_1^g - y_1^p)}{2} \right); \quad (38)$$

$$\frac{\partial L_{SO}}{\partial x_2^p} = \frac{1}{w_{cnx}} \left(\frac{1 - \text{sgn}(x_2^g - x_2^p)}{2} \cdot \frac{w_{\cap}}{w_{cnx}} - \frac{1 + \text{sgn}(x_2^g - x_2^p)}{2} \right); \quad (39)$$

$$\frac{\partial L_{SO}}{\partial y_2^p} = \frac{1}{h_{cnx}} \left(\frac{1 - \text{sgn}(y_2^g - y_2^p)}{2} \cdot \frac{h_{\cap}}{h_{cnx}} - \frac{1 + \text{sgn}(y_2^g - y_2^p)}{2} \right). \quad (40)$$

4.9.3 Градиенты L_{cd}

$$L_{cd} = \frac{(x_1^g - x_1^p)^2 + (y_1^g - y_1^p)^2 + (x_2^g - x_2^p)^2 + (y_2^g - y_2^p)^2}{d_{cnx}^2}; \quad (41)$$

$$\frac{\partial L_{cd}}{\partial x_1^p} = -\frac{2}{d_{cnx}^2} \left(x_1^g - x_1^p - \frac{1 + \text{sgn}(x_1^g - x_1^p)}{2} \cdot w_{cnx} \cdot L_{cd} \right); \quad (42)$$

$$\frac{\partial L_{cd}}{\partial y_1^p} = -\frac{2}{d_{cnx}^2} \left(y_1^g - y_1^p - \frac{1 + \text{sgn}(y_1^g - y_1^p)}{2} \cdot h_{cnx} \cdot L_{cd} \right); \quad (43)$$

$$\frac{\partial L_{cd}}{\partial x_2^p} = -\frac{2}{d_{cnx}^2} \left(x_2^g - x_2^p + \frac{1 - \text{sgn}(x_2^g - x_2^p)}{2} \cdot w_{cnx} \cdot L_{cd} \right); \quad (44)$$

$$\frac{\partial L_{cd}}{\partial y_2^p} = -\frac{2}{d_{cnx}^2} \left(y_2^g - y_2^p + \frac{1 - \text{sgn}(y_2^g - y_2^p)}{2} \cdot h_{cnx} \cdot L_{cd} \right). \quad (45)$$

4.9.4. Градиенты L_{form}

$$L_{form} = 2 - \frac{\min(w^g, w^p)}{\max(w^g, w^p)} + \frac{\min(h^g, h^p)}{\max(h^g, h^p)}; \quad (46)$$

$$\frac{\partial L_{form}}{\partial x_1^p} = \frac{1}{w_{max}} \left(\frac{1 + \text{sgn}(w_g - w_p)}{2} - \frac{1 - \text{sgn}(w_g - w_p)}{2} \cdot \frac{w_{min}}{w_{max}} \right); \quad (47)$$

$$\frac{\partial L_{form}}{\partial y_1^p} = \frac{1}{h_{max}} \left(\frac{1 + \text{sgn}(h_g - h_p)}{2} - \frac{1 - \text{sgn}(h_g - h_p)}{2} \cdot \frac{h_{min}}{h_{max}} \right); \quad (48)$$

$$\frac{\partial L_{form}}{\partial x_2^p} = -\frac{1}{w_{max}} \left(\frac{1 + \text{sgn}(w_g - w_p)}{2} - \frac{1 - \text{sgn}(w_g - w_p)}{2} \cdot \frac{w_{min}}{w_{max}} \right); \quad (49)$$

$$\frac{\partial L_{form}}{\partial y_2^p} = -\frac{1}{h_{max}} \left(\frac{1 + \text{sgn}(h_g - h_p)}{2} - \frac{1 - \text{sgn}(h_g - h_p)}{2} \cdot \frac{h_{min}}{h_{max}} \right). \quad (50)$$

4.9.5 Выводы

Анализируя градиенты имеющихся функций, можно заметить общую закономерность – градиенты нормируются на некоторое общее число (высоту/ширину/площадь некоторого прямоугольника – наименьшего ограничивающего, наименьшего из двух) и т.д.). При этом в некоторых условиях градиенты зависят только от нормирующего множителя, а не от близости координат.

4.10 Построение координатных функций потерь по градиенту

После проведённого анализа был выполнен поиск функций градиента, которые бы также имели нормирующий множитель, но при этом градиенты всегда зависели от координат и приводили к более высокой сходимости по сравнению с имеющимися. Одной из таких функций стала следующая:

$$\frac{\partial L_{grad}}{\partial x_1^p} = -\frac{\text{sgn}(x_1^g - x_1^p) + x_1^g - x_1^p}{w_g}; \quad (51)$$

$$\frac{\partial L_{grad}}{\partial y_1^p} = -\frac{\text{sgn}(y_1^g - y_1^p) + y_1^g - y_1^p}{h_g}; \quad (52)$$

$$\frac{\partial L_{grad}}{\partial x_2^p} = -\frac{\text{sgn}(x_2^g - x_2^p) + x_2^g - x_2^p}{w_g}; \quad (53)$$

$$\frac{\partial L_{grad}}{\partial y_2^p} = -\frac{sgn(y_2^g - y_2^p) + y_2^g - y_2^p}{h_g}. \tag{54}$$

4.10.1 Получение функции L_{grad}

Для восстановления функции L_{grad} достаточно проинтегрировать каждую из четырёх функций и сложить результат, так как каждое из слагаемых не зависит от переменных других слагаемых. При этом в качестве константы стоит выбрать ту, при которой значение функции при идеальном совпадении предсказанных и целевых координат будет равно нулю.

$$L_{grad} = L_{grad_{x_1}} + L_{grad_{y_1}} + L_{grad_{x_2}} + L_{grad_{y_2}}; \tag{55}$$

$$L_{grad_{x_1}} = -\int \frac{sgn(x_1^g - x_1^p) + x_1^g - x_1^p}{w_g} dx_1^p = \frac{|x_1^g - x_1^p| + 0.5(x_1^g - x_1^p)^2}{w_g}; \tag{56}$$

$$L_{grad_{y_1}} = -\int \frac{sgn(y_1^g - y_1^p) + y_1^g - y_1^p}{h_g} dy_1^p = \frac{|y_1^g - y_1^p| + 0.5(y_1^g - y_1^p)^2}{h_g}; \tag{57}$$

$$L_{grad_{x_2}} = -\int \frac{sgn(x_2^g - x_2^p) + x_2^g - x_2^p}{w_g} dx_2^p = \frac{|x_2^g - x_2^p| + 0.5(x_2^g - x_2^p)^2}{w_g}; \tag{58}$$

$$L_{grad_{y_2}} = -\int \frac{sgn(y_2^g - y_2^p) + y_2^g - y_2^p}{h_g} dy_2^p = \frac{|y_2^g - y_2^p| + 0.5(y_2^g - y_2^p)^2}{h_g}. \tag{59}$$

Достоинства функции градиента:

- градиенты нормализованы относительно наименьшего прямоугольника;
- каждая координата сходится независимо;
- нулевой градиент при совпадении координат;
- с отдалением от целевой координаты градиент увеличивается в обе стороны.

Недостатки:

- имеет неустраиваемый разрыв в окрестности равенства координат

4.10.2 Устранение проблемы разрывности функции градиента

Выражение $f(x) = sgn(x) + x$ имеет разрыв в точке 0. Решение – заменить непрерывной функцией, например, $f(x) = \tanh(kx) + x$ или $f(x) = \frac{2}{\pi} \cdot \text{atan}(kx) + x$ (рис. 11):

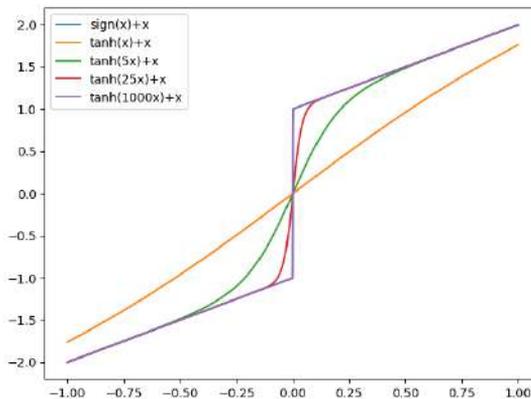


Рис. 11. Сравнение функций
Fig. 11. Functions comparison

Замена разрывной функции $sgn(x)$ на $\tanh(kx) + x$ или $\frac{2}{\pi} \cdot \operatorname{atan}(kx) + x$ даёт две новые функции градиентов (Δv обозначает $v^g - v^p$):

$$\frac{\partial L_{\tanh}}{\partial x_1^p} = -\frac{\tanh(k \cdot \Delta x_1) + \Delta x_1}{w_g}; \quad (60)$$

$$\frac{\partial L_{\operatorname{atan}}}{\partial x_1^p} = -\frac{\left(\frac{2}{\pi} \cdot \tanh(k \cdot \Delta x_1) + \Delta x_1\right)}{w_g}; \quad (61)$$

$$\frac{\partial L_{\tanh}}{\partial y_1^p} = -\frac{\tanh(k \cdot \Delta y_1) + \Delta y_1}{h_g}; \quad (62)$$

$$\frac{\partial L_{\operatorname{atan}}}{\partial y_1^p} = -\frac{\left(\frac{2}{\pi} \cdot \tanh(k \cdot \Delta y_1) + \Delta y_1\right)}{h_g}; \quad (63)$$

$$\frac{\partial L_{\tanh}}{\partial x_2^p} = -\frac{\tanh(k \cdot \Delta x_2) + \Delta x_2}{w_g}; \quad (64)$$

$$\frac{\partial L_{\operatorname{atan}}}{\partial x_2^p} = -\frac{\left(\frac{2}{\pi} \cdot \tanh(k \cdot \Delta x_2) + \Delta x_2\right)}{w_g}; \quad (65)$$

$$\frac{\partial L_{\tanh}}{\partial y_2^p} = -\frac{\tanh(k \cdot \Delta y_2) + \Delta y_2}{h_g}; \quad (66)$$

$$\frac{\partial L_{\operatorname{atan}}}{\partial y_2^p} = -\frac{\left(\frac{2}{\pi} \cdot \tanh(k \cdot \Delta y_2) + \Delta y_2\right)}{h_g}. \quad (67)$$

Интегрирование данных функций градиентов даёт следующие функции:

$$L_{\tanh} = L_{\tanh_{x_1}} + L_{\tanh_{y_1}} + L_{\tanh_{x_2}} + L_{\tanh_{y_2}}; \quad (68)$$

$$L_{\tanh_{x_1}} = \frac{1}{w_g} \left(\frac{\Delta x_1^2}{2} + \frac{\ln(\cosh(k \cdot \Delta x_1))}{k} \right); \quad (69)$$

$$L_{\tanh_{y_1}} = \frac{1}{h_g} \left(\frac{\Delta y_1^2}{2} + \frac{\ln(\cosh(k \cdot \Delta y_1))}{k} \right); \quad (70)$$

$$L_{\tanh_{x_2}} = \frac{1}{w_g} \left(\frac{\Delta x_2^2}{2} + \frac{\ln(\cosh(k \cdot \Delta x_2))}{k} \right); \quad (71)$$

$$L_{\tanh_{y_2}} = \frac{1}{h_g} \left(\frac{\Delta y_2^2}{2} + \frac{\ln(\cosh(k \cdot \Delta y_2))}{k} \right); \quad (72)$$

$$L_{\operatorname{atan}} = L_{\operatorname{atan}_{x_1}} + L_{\operatorname{atan}_{y_1}} + L_{\operatorname{atan}_{x_2}} + L_{\operatorname{atan}_{y_2}}; \quad (73)$$

$$L_{\operatorname{atan}_{x_1}} = \frac{1}{w_g} \left(\frac{\Delta x_1^2}{2} + \frac{2}{\pi} \left(\Delta x_1 \cdot \operatorname{atan}(k \cdot \Delta x_1) - \frac{\ln(1 + k^2 \cdot \Delta x_1^2)}{2k} \right) \right); \quad (74)$$

$$L_{\operatorname{atan}_{y_1}} = \frac{1}{h_g} \left(\frac{\Delta y_1^2}{2} + \frac{2}{\pi} \left(\Delta y_1 \cdot \operatorname{atan}(k \cdot \Delta y_1) - \frac{\ln(1 + k^2 \cdot \Delta y_1^2)}{2k} \right) \right); \quad (75)$$

$$L_{\operatorname{atan}_{x_2}} = \frac{1}{w_g} \left(\frac{\Delta x_2^2}{2} + \frac{2}{\pi} \left(\Delta x_2 \cdot \operatorname{atan}(k \cdot \Delta x_2) - \frac{\ln(1 + k^2 \cdot \Delta x_2^2)}{2k} \right) \right); \quad (76)$$

$$L_{\operatorname{atan}_{y_2}} = \frac{1}{h_g} \left(\frac{\Delta y_2^2}{2} + \frac{2}{\pi} \left(\Delta y_2 \cdot \operatorname{atan}(k \cdot \Delta y_2) - \frac{\ln(1 + k^2 \cdot \Delta y_2^2)}{2k} \right) \right). \quad (77)$$

4.10.3 Сходимость полученных функций

Результаты эксперимента-симуляции представлены на рис. 12. Как и ожидалось, градиенты полученных функций (L_{grad} (формула 52), L_{\tanh} (формула 65) и L_{atan} (формула 70)) схожи как по форме, так и по значению, однако сглаженные функции (L_{\tanh} (формула 65) и L_{atan} (формула 70)) сходятся быстрее L_{grad} (формула 52) и тем более L_{SCA} [5].

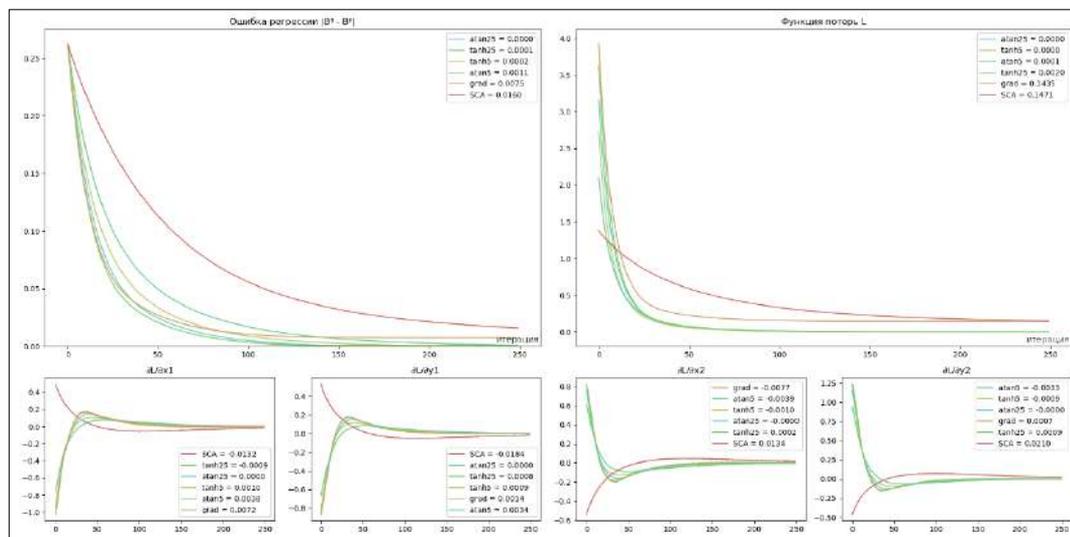


Рис. 12. Результаты симуляционного эксперимента градиентных функций
 Fig. 12. Results of the simulation experiment of gradient functions

5. Выбор набора данных для обучения

Для экспериментов был выбран набор данных PubLayNet [8], представляющий собой коллекцию документов медицинских научных статей. Он содержит 335 тысяч обучающих изображений и 11.5 тысяч проверочных изображений. Поскольку обучение на таком большом наборе требует много времени, а экспериментов с функциями потерь проводить нужно большое количество раз, то для экспериментов была выбрана небольшая часть – 10 тысяч обучающих изображений и 1000 проверочных.

Помимо него также проводились эксперименты на наборе сгенерированных изображений документов различных договоров и технических заданий [9], содержащем 20 тысяч обучающих и 769 проверочных изображений.

6. Оценка качества

Для более полной оценки качества сегментации, в том числе учитывающей особенности сегментации изображений документов, были взяты три типа метрик:

- Стандартные метрики Precision, Recall, mAP (mean average precision) с порогом 0.5, mAP с интервалом порога 0.5...0.95, основанные на IoU функции;
- Перечисленные выше метрики с заменой IoU функции на пиксельную BWIoU (5) функцию;
- OCR метрика (9).

7. Результаты экспериментов

В табл.2 и 3 приведены результаты тестирования выбранной модели сегментации на наборе данных PubLayNet [8] с помощью координатной и пиксельной метрик соответственно. Наилучшие результаты по метрике mAP 0.5 и mAp 0.5...0.95 показывают функции Pixel-delta (12) и PIoU (1).

Табл. 2. Результаты экспериментов на наборе PubLayNet (координатная метрика)

Table 2. Results of experiments on the PubLayNet dataset (coordinate metric)

Функция потерь	Precision	Recall	mAP 0.5	mAP 0.5...0.95
Pixel-delta	0.931	0.906	0.934	0.817
IoU	0.928	0.901	0.923	0.801
SCA	0.94	0.895	0.926	0.803
PIoU	0.932	0.909	0.934	0.817
BWIoU	0.936	0.894	0.929	0.804
Weigted BWIoU	0.939	0.9	0.928	0.812
FM	0.953	0.899	0.932	0.813
atan	0.947	0.884	0.925	0.799
tanh	0.953	0.886	0.926	0.799

Табл. 3. Результаты экспериментов на наборе PubLayNet (пиксельная метрика)

Table 3. Results of experiments on the PubLayNet dataset (pixel metric)

Функция потерь	Precision	Recall	mAP 0.5	mAP 0.5...0.95
Pixel-delta	0.936	0.898	0.924	0.742
IoU	0.927	0.895	0.916	0.721
SCA	0.931	0.888	0.919	0.723
PIoU	0.925	0.902	0.924	0.727
BWIoU	0.931	0.884	0.913	0.716
Weigted BWIoU	0.93	0.891	0.915	0.711
FM	0.947	0.893	0.922	0.733
atan	0.938	0.876	0.913	0.714
tanh	0.941	0.879	0.913	0.714

В табл. 4 и 5 приведены результаты тестирования выбранной модели сегментации на наборе синтетических документов с помощью координатной и пиксельной метрик соответственно. Функция потерь Pixel-delta показывает наилучшее значения в метрике mAp 0.5...0.95, а по метрике mAp 0.5 наилучший результат показывает чисто пиксельная функция PIoU.

Табл. 4. Результаты экспериментов на наборе Generated (координатная метрика)

Table 4. Results of experiments on the Generated dataset (coordinate metric)

Функция потерь	Precision	Recall	mAP 0.5	mAP 0.5...0.95
Pixel-delta	0.992	0.988	0.994	0.874
IoU	0.992	0.983	0.986	0.858
SCA	0.987	0.981	0.987	0.861
PIoU	0.992	0.988	0.994	0.862
BWIoU	0.99	0.982	0.993	0.861
Weigted BWIoU	0.99	0.982	0.991	0.858
FM	0.984	0.972	0.991	0.81
atan	0.989	0.969	0.986	0.856
tanh	0.989	0.975	0.989	0.844

Табл. 5. Результаты экспериментов на наборе Generated (пиксельная метрика)

Table 5. Results of experiments on the Generated dataset (pixel metric)

Функция потерь	Precision	Recall	mAP 0.5	mAP 0.5...0.95
Pixel-delta	0.989	0.982	0.989	0.736

IoU	0.982	0.973	0.98	0.708
SCA	0.978	0.972	0.98	0.67
PIoU	0.989	0.977	0.983	0.697
BWIoU	0.983	0.975	0.987	0.72
Weigted BWIoU	0.98	0.973	0.981	0.733
FM	0.981	0.955	0.977	0.586
atan	0.98	0.96	0.979	0.718
tanh	0.978	0.964	0.978	0.689

В табл. 6 приведены результаты тестирования модели сегментации на наборах PubLayNet и Generated с помощью OCR метрики. Функция Pixel-delta для всех наборов данных показывает наилучшее качество, что говорит о наиболее точном выделении целевых областей.

Табл. 6. Результаты экспериментов (метрика OCR)

Table 6. Results of experiments (OCR metric)

Функция потерь	OCR metric (PubLayNet)	OCR metric (Generated)
Pixel-delta	0.903	0.914
IoU	0.891	0.903
SCA	0.892	0.904
PIoU	0.899	0.912
BWIoU	0.898	0.909
Weigted BWIoU	0.89	0.91
FM	0.891	0.902
atan	0.893	0.903
tanh	0.895	0.901

Использование функций потерь, учитывающих особенности изображений документов (PIoU, BWIoU, Weigted BWIoU), демонстрируют более высокое качество сегментации по сравнению со стандартными функциями. Разработанные координатные функции потерь (FM, tanh, atan) показывают качество на уровне существующих функций, но имеют более высокую скорость сходимости. Функции, комбинирующие координатный и пиксельный подходы (Pixel-delta), имеют более высокое качество сегментации и скорость сходимости по сравнению с функциями с одним подходом. Разработанные функции потерь, учитывающие особенности изображений документов, улучшают качество сегментации на 3-5%. Разработанные координатные функции потерь повышают скорость сходимости.

7. Выводы

В работе была выбрана модель сегментации и разработаны функции потерь, как учитывающие особенности изображений документов, так и использующие классический чисто координатный подход.

Исходя из полученных результатов, можно утверждать, что использование функций потерь, использующих особенности сегментации изображений документов позволяет достичь высокой точности сегментации, а использование координатных функций позволяет быстрее обучать модель. Комбинируя подходы, можно получать функции потерь, сочетающие в себе оба свойства.

Список литературы / References

- [1] Zheng Z., Wang P. et al. Distance-IoU loss: Faster and better learning for bounding box regression. *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 07, 2020, pp. 12993-13000.

- [2] Rezatofighi H., Tsoi N. et al. Generalized intersection over union: A metric and a loss for bounding box regression. In Proc. of the IEEE/CVF conference on computer vision and pattern recognition, 2019, pp. 658-666.
- [3] Zheng T., Zhao S. et al. SCALoss: Side and Corner Aligned Loss for Bounding Box Regression. arXiv preprint arXiv:2104.00462, 2021, 9 p.
- [4] He J., Erfani S. et al. α -IoU: A Family of Power Intersection over Union Losses for Bounding Box Regression. *Advances in Neural Information Processing Systems*, vol. 34, 2021, 13 p.
- [5] Wu S., Yang J. et al. Iou-balanced loss functions for single-stage object detection. *Pattern Recognition Letters*, vol. 156, 2022, pp. 96-103.
- [6] Du S., Zhang B., Zhang P. Scale-Sensitive IOU Loss: An Improved Regression Loss Function in Remote Sensing Object Detection. *IEEE Access*, vol. 9, 2021, pp. 141258-141272.
- [7] Redmon J., Farhadi A. Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767, 2018, 6 p.
- [8] Zhong X., Tang J., Yepes A.J. Publaynet: largest dataset ever for document layout analysis. In Proc. of the 2019 International Conference on Document Analysis and Recognition (ICDAR), 2019, pp. 1015-1022.
- [9] Беляева О.В., Перминов А.И., Козлов И.С. Использование синтетических данных для тонкой настройки моделей сегментации документов. *Труды ИСП РАН*, том 32, вып. 4, 2020 г., стр. 189-202. DOI: 10.15514/ISPRAS-2020-32(4)-14 / Belyaeva O.V., Perminov A.I., Kozlov I.S. Synthetic data usage for document segmentation models fine-tuning. *Trudy ISP RAN/Proc. ISP RAS*, vol. 32, issue 4, 2020. pp. 189-202 (in Russian).

Информация об авторах / Information about authors

Андрей Игоревич ПЕРМИНОВ является студентом магистратуры кафедры системного программирования. Научные интересы: цифровая обработка сигналов, нейросетевая обработка данных, создание искусственных данных, цифровая обработка изображений.

Andrey Igorevich PERMINOV – master’s student of the Department of System Programming. Research interests: digital signal processing, neural network data processing, generation of artificial data, digital image processing.

Денис Юрьевич ТУРДАКОВ – кандидат физико-математических наук, заведующий отделом ИСП РАН, доцент кафедры системного программирования ф-та ВМК МГУ. Научные интересы: анализ естественного языка, извлечение информации, обработка больших данных, анализ социальных сетей.

Denis Yurievich TURDAKOV – PhD, Head of Department at ISP RAS, associate professor of the Department of System Programming at MSU. Research interests: natural language processing, information extraction, big data analysis, social network analysis.

Оксана Владимировна БЕЛЯЕВА – аспирант. Научные интересы: распознавание структуры документов, цифровая обработка изображений, нейросетевая обработка данных, распознавание образов.

Oksana Vladimirovna BELYAEVA – PhD Student. Research interests: document layout analysis, digital image processing, neural network data processing, image pattern recognition.

DOI: 10.15514/ISPRAS-2022-34(2)-9



Система управления заданиями автоматизированного сбора данных из сети Интернет

^{1,2} В.А. Лазарев, ORCID: 0000-0002-4074-5424 <vlazarew@ispras.ru>

¹ М.И. Варламов, ORCID: 0000-0002-1083-6210 <varlamov@ispras.ru>

¹ А.К. Яцков, ORCID: 0000-0002-1312-1675 <yatskov@ispras.ru>

¹ *Институт системного программирования им. В.П. Иванникова РАН, 109004, Россия, г. Москва, ул. А. Солженицына, д. 25*

² *Московский государственный университет им. М.В. Ломоносова, 119991, Россия, Москва, Ленинские горы, д. 1*

Аннотация. Работа посвящена исследованию и разработке системы управления заданиями автоматизированного сбора данных из сети Интернет. Статья содержит описание реализованных методологий и повествует о созданных приемах взаимодействия с контейнерами, содержащими в себе приложения для сбора данных. В ходе работы были изучены и представлены существующие различные сервисы автоматизированного сбора данных из сети Интернет: готовые решения с открытым исходным кодом, облачные сервисы с обширным функционалом, а также собственное решение под управлением Kubernetes. В результате работы реализована и внедрена в платформу для анализа данных Talisman система управления заданиями, которая обеспечивает горизонтальную масштабируемость, изолированность окружения сборщиков и независимость от технологии их разработки.

Ключевые слова: сбор данных; система управления заданиями; виртуализация; Kubernetes

Для цитирования: Лазарев В.А., Варламов М.И., Яцков А.К. Система управления заданиями автоматизированного сбора данных из сети Интернет. Труды ИСП РАН, том 34, вып. 2, 2022 г., стр. 111-122. DOI: 10.15514/ISPRAS-2022-34(2)-9

Job management system for automated data collection from the Internet

^{1,2} V.A. Lazarev, ORCID: 0000-0002-4074-5424 <vlazarew@ispras.ru>

¹ M.I. Varlamov, ORCID: 0000-0002-1083-6210 <varlamov@ispras.ru>

¹ A.K. Yatskov, ORCID: 0000-0002-1312-1675 <yatskov@ispras.ru>

¹ *Ivannikov Institute for System Programming of the Russian Academy of Sciences, 25, Alexander Solzhenitsyn st., Moscow, 109004, Russia*

² *Lomonosov Moscow State University, GSP-1, Leninskie Gory, Moscow, 119991, Russia*

Abstract. This work is devoted to the research and development of a task management system for automated data collection from the Internet. This article contains a description of the implemented methodologies and tells about the techniques created by interacting with containers containing data collection applications. In the course of the work, various existing services for automated data collection from the Internet were studied and presented: ready-made open source solutions, cloud services with extensive functionality, as well as our own solution running Kubernetes. As a result of the work, a task management system was implemented for Talisman data analysis platform, which provides horizontal scalability, isolation of the crawler environment and independence from the technology of their development.

Keywords: data collection; task management system; virtualization; Kubernetes

For citation: Lazarev V.A., Varlamov M.I., Yatskov A.K. Job management system for automated data collection from the Internet. *Trudy ISP RAN/Proc. ISP RAS*, vol. 34, issue 2, 2022, pp. 111-122 (in Russian). DOI: 10.15514/ISPRAS-2022-34(2)-9

1. Введение

В настоящее время активно развиваются приложения и сервисы, позволяющие собирать, накапливать и анализировать информацию для уменьшения затрат на временные ресурсы пользователей и предоставления актуальной информации и контента.

Возможных направлений, в которых бы пригодился сбор и анализ данных, достаточно много. Базовым инструментом в каждом из них является сборщик данных – приложение, которое принимает на вход некоторые аргументы и настройки, отправляет HTTP-запросы на указанные web-страницы и на основе своей спецификации формирует и возвращает сериализуемый ответ. Для получения обширной и полной картины по какому-либо объекту необходимо обрабатывать несколько источников одновременно. Из этого вытекает, что для системы, которая хочет агрегировать в себе всевозможные источники данных для предоставления качественного анализа, предстоит использование большого количества приложений для сбора информации.

Система управления заданиями автоматизированного сбора данных из сети Интернет должна позволять загружать новые сборщики, производить запуск и отмену сбора данных для конкретной версии сборщика, получать собираемые сборщиками документы, а также отслеживать метрики сбора, отправляемые запросы и логи заданий. При этом такая система должна удовлетворять ряду требований.

- Горизонтальная масштабируемость. Количество одновременных задач сбора данных должно быть ограничено аппаратными возможностями сервера, но никак не на уровне приложения.
- Изолированность зависимостей сборщиков. Каждый сборщик должен содержать только необходимые ему зависимости от библиотек. Необходимо для разрешения конфликтов между зависимостями различных приложений для сбора данных
- Независимость от технологии разработки сборщиков. Система должна позволять загружать и запускать сборщики, написанные с помощью различных языков программирования и фреймворков.

В работе описывается методология разработки системы управления заданиями сбора данных, отвечающей поставленным требованиям, и ее реализация в рамках платформы анализа данных Talisman [1].

2. Существующие решения

Рассматриваемые инструменты можно разбить на две категории: локальные системы, которые можно установить, использовать и дорабатывать, с открытым исходным кодом; облачные платформы, предоставляющие доступ к информации, анализу за определенную плату.

2.1 Локальные системы управления заданиями

Авторы статьи [2] предложили свой подход, но он основывается на реализации одного параллельного сборщика. В нашем же случае необходимо осуществлять параллельную работу различных сборщиков данных.

Scrapy Cluster [3] – проект Scrapy для создания распределенного кластера для сбора данных по запросу. Поддерживает динамические сборщики, работающие по запросу и позволяющие произвольно собирать любую web-страницу. Обеспечиваются масштабирование

экземпляров; параллельное выполнение и механизм приоритетов. Итогом сбора является HTML-код страницы, из которого далее необходимо структурировать информацию.

Scrapy [4] – инструмент, позволяющий управлять несколькими проектами сбора данных. Сборщики загружаются в систему как составные части проекта, причем сам файл проекта имеет удобное для Python расширение .egg. Обладает всеми преимуществами Scrapy Cluster, а также поддерживает сборщики, написанные при помощи фреймворка scrapy, что позволяет получать сразу структурированные данные. Однако инструмент обеспечивает только вертикальную масштабируемость и не поддерживает изолированность зависимостей между сборщиками.

Scrapyrt (Scrapy realtime) [5] – это HTTP-сервер, который предоставляет API для настройки расписаний запусков Scrapy сборщиков и получения результатов сбора данных. Обладает довольно скудным функционалом, невозможно останавливать сбор или ставить его на паузу, работает только со scrapy-сборщиками и не рекомендуется для использования с длительным сбором данных.

Ferrit [6] – HTTP-сервер, предоставляющий API для создания сборщиков, их запусков и мониторинга задач сбора данных, написан на языке программирования Scala. Однако созданные сборщики получаются весьма примитивными, без возможности детально настроить поведение и извлекаемую информацию, равно как и указать формат выхода сообщений. Загрузку каких-либо сторонних сборщиков не поддерживает.

2.2 Облачные сервисы сбора данных

Zyte Scrapy Cloud [7] – это сервис для сбора данных. Платформа предоставляет интерфейс для мониторинга собранных данных. Есть возможность фокусированного сбора данных, основываясь на целевом бренде или товаре. Также поддерживается загрузка сборщиков на любом языке программирования, поскольку есть возможность загрузить сборщики как Dockerfile, из которого соберется необходимый образ. Сервис обладает средствами мониторинга и тестирования, высокой масштабируемостью, поддерживает сборщики на основе любых технологий, если Dockerfile соответствует описанному контракту. Платформа доступна только по подписке.

Web Scraper Cloud [8] – сервис, использующий плагин Web Scraper и предоставляющий надстройки для обеспечения автоматизации, масштабируемости и управления задачами сбора. Сервис обладает встроенным прокси, планировщиком задач, средствами мониторинга и тестирования, возможностью планирования задач и их параллельного запуска. Однако используется собственный JSON-формат описания сборщиков, загрузка других сборщиков не поддерживается, и платформа доступна только по подписке без возможности доработок под собственные нужды.

Ostoparse [9] – сервис для извлечения данных. Не требует знаний в программировании, поскольку перед сбором предоставляется возможность выделения областей с требуемой информацией для извлечения. Предоставляет возможность производить каждый последующий запрос с нового IP-адреса. Отсутствует мониторинг задач, требуется платная подписка на сервис, и нет возможности загружать и использовать сборщики на любом фреймворке.

DataOx [10] – сервис для получения данных из сети Интернет в большом количестве. Отличительной чертой является то, что имеется возможность рассылки обновленной информации по указанным ресурсам. Сервис предлагает сбор для сайтов любой сложности, планировщик задач, параллельный запуск задач сбора данных, средства мониторинга и тестирования. Отсутствует возможности загружать и использовать сборщики на любом фреймворке, требуется платная подписка на сервис.

В результате исследования существующих систем управления заданиями автоматизированного сбора данных из сети Интернет можно сформулировать следующие выводы:

- бесплатные сервисы не полностью удовлетворяют наши требования, зачастую поддерживают только один фреймворк и сильно ограничены в масштабируемости;
- облачные сервисы удобны в использовании, но все они не предоставляют исходный код для доработок, и для использования платформ необходимо платить.

Поскольку рассмотренные системы не полностью удовлетворяют требованиям, сформулированным в разд. 1, и, кроме того, необходима изолированность приложений и высокая горизонтальная масштабируемость, следует рассмотреть системы оркестрации контейнеризированных приложений.

3. Исследование и выбор средства контейнерной оркестрации

Основная функциональность инструментов контейнерной оркестрации – развертывание приложений, упакованных в контейнеры, отслеживание статуса жизнеспособности запущенных контейнеров, поддержание необходимого числа реплик контейнеров, мониторинг состояния контейнеров, обновление их версий, и уничтожение развернутых приложений.

Типичный сборщик – это запуск некоторого приложения внутри отдельного контейнера. Используя контейнеры, можно достичь изолированности зависимостей приложений для сбора, горизонтально масштабироваться за счёт существующих средств контейнерной оркестрации, а также поддерживать сборщики на любых технологиях их разработки.

Kubernetes [11] – это открытый проект для автоматизации развертывания, масштабирования и управления контейнерными приложениями. Поддерживает работу с Docker и предоставляет средства для управления контейнерами. Модель делится на два типа узлов: управляющие и рабочие. Рабочие узлы объединены в группы, каждую из групп контролирует свой управляющий узел, который распределяет задачи создания и управления контейнерами, а также отслеживания их состояния. Kubernetes поддерживает работу с Containerd-контейнерами и предоставляет механизмы для развертывания, управления, масштабируемости и мониторинга работы контейнеров.

Удобным инструментом для создания модулей в Kubernetes является Job (задание). При помощи задания можно создать один или несколько модулей, причем если какой-либо из модулей не смог развернуться, то задание повторит попытку указанное в конфигурационном файле количество раз, либо до тех пор, пока все связанные модули успешно не завершат работу. Задание отслеживает состояние всех модулей, которые оно породило. При удалении задания все порожденные модули также будут удалены. Зачастую задания в Kubernetes используются для параллельного запуска нескольких модулей.

Docker Compose [12] – инструмент для настройки и запуска многоконтейнерных приложений Docker. Для конфигурации приложения используется файл `docker-compose.yml`, в котором указываются необходимые для развертывания приложения контейнеры, взаимосвязь и взаимозависимость между контейнерами, настройки и параметры окружающей среды для каждого контейнера. Это очень удобное средство для небольших или локальных приложений, поскольку обладает простой настройкой и конфигурацией. Но присутствуют функциональные ограничения и недостатки по сравнению с другими платформами.

Docker Swarm [13] – специальный режим управления в Docker Engine. Однако в актуальных версиях Docker режим swarm поддерживается изначально без дополнительных настроек. Этот инструмент отлично подходит для небольших кластеров с малыми вычислениями, где требуется по большей части простота в настройке и конфигурации. Главный недостаток инструмента – поддержка контейнеров Docker и только.

Apache Mesos [14] – создан как менеджер кластера нового поколения. Это программное обеспечение с открытым исходным кодом, которое обеспечивает эффективную изоляцию ресурсов и их совместное использование в распределенных приложениях или платформах. Mesos – это ядро распределенной системы с полным API для программирования непосредственно в центре обработки данных. Архитектура Mesos состоит из основного процесса, который управляет подчиненными демонами, работающими на каждом узле кластера, и фреймворков, выполняющих задачи на этих подчиненных устройствах. Mesos лучше всего подходит для больших систем и обеспечивает максимальную избыточность. Mesos рекомендуется, если у вас есть существующие рабочие нагрузки, такие как Hadoop, Kafka и т.д. Он дает платформу, которая позволяет чередовать эти рабочие нагрузки друг с другом. Это самая стабильная платформа, но слишком сложная для небольших систем до 10-20 узлов. Основная сила Mesos заключается в больших данных и аналитике. Оркестровка контейнеров не совсем его дело. Mesos обладает чрезмерно большим функционалом и слишком универсален.

Приведенный анализ систем контейнерной оркестрации показывает, что наиболее взвешенным решением, на базе которого стоит реализовать требуемую систему, является Kubernetes, поскольку система достаточно универсальна, соответствует необходимым функциональным требованиям и не перегружена излишними возможностями.

Кстати, в статье [15] авторы исследовали архитектуру и производительность различных систем контейнерной оркестрации. Их выводы совпали с нашими: они считают Kubernetes лучшим решением на текущий момент.

4. Жизненный цикл абстрактного сборщика данных

Чтобы составить спецификацию для работы со сборщиками как приложениями внутри контейнеров, необходимо рассмотреть этапы жизненного цикла произвольного сборщика в системе.

Жизненный цикл любого сборщика подразумевает под собой все этапы от его загрузки в систему до выполнения запусков на основе сборщика. При желании сборщик можно удалить из системы, после чего доступ к нему теряется навсегда.

- I. Загрузка сборщика в систему управления заданиями автоматизированного сбора данных.
- II. После загрузки сборщик может быть запущен. На его основе создается задача сбора, которая может находиться в нескольких состояниях:
 - a. В ожидании – задача создана только в рамках базы данных, ожидает готовности окружения для начала работы.
 - b. В работе – задача непосредственно находится в процессе выполнения. Есть возможность остановить процесс по необходимости. Если задача выполняется длительное время, по полученным сообщениям видно, что коэффициент полезного действия стремится к нулю, но при этом место в пуле запусков занято.
 - c. Завершена – задача сбора данных завершена, место в пуле запусков освобождено, все собранные сообщения, логи, запросы сохранены в соответствующих хранилищах. Задача может быть завершена с несколькими статусами:
 - i. успешно – ошибок в ходе сбора не обнаружено;
 - ii. с ошибками – в ходе сбора были обнаружены и обработаны ошибки сбора, некоторые сообщения утеряны;
 - iii. критическая ошибка – в ходе сбора была выявлена критическая ошибка, вследствие которой процесс сбора данных был экстренно завершен;

- iv. отменена – при отмене задачи запуска пользователем. При этом та информация, которая успела собраться, будет представлена для хранения и анализа.

III. Когда сборщик становится неактуальным, необходимо иметь возможность удалить его полностью или только его версию. При удалении версии она помечается как удаленная. Создавать задачи сбора по сборщику с удаленной версией невозможно.

Абстрактный сборщик имеет возможность принимать на вход аргументы и настройки. Эти параметры необходимы для более гибкого сбора данных и возможности переиспользовать один и тот же сборщик для различных целей. Настройки влияют на конфигурационные параметры сборщика, аргументы – определяют поведение сборщика в целом.

Результат работы сборщика данных – множество сообщений, логов, метрик и запросов, которые являются, не ограничивая общности, Json-объектами. И поскольку методов сбора данных из сети Интернет огромное количество, а обрабатывающая система одна, имеется необходимость в стандартизации выходного формата сообщений сборщиков данных.

Представленная модель абстрактного сборщика частично совпадает с моделью жизненного цикла сборщика, реализованного на Scrapy. Разница состоит в том, что Scrapy подразумевает загрузку в систему проектов сбора данных – набор сборщиков, объединенных общим конфигурационным файлом. Как следствие и версионирование осуществляется в рамках проекта сбора данных.

5. Спецификация устройства контейнера со сборщиками

Спецификация устройства контейнеров, содержащих приложения для сбора данных, включает спецификации для загрузки сборщиков в систему и получения результатов их работы.

5.1 Спецификация загрузки сборщиков

Для корректной работы внутри системы поставщик сборщиков должен обеспечить следующее:

- Dockerfile для корректного создания контейнера (может использоваться как простое копирование jar-файла, так и установка специального окружения с разархивацией исходных файлов);
- скрипт-entrypoint – входная точка для Dockerfile, в котором должны быть реализованы команды:
 - list – извлечение списка сборщиков в командную строку;
 - schedule – запуск конкретного сборщика с переданными аргументами командной строки.
- архив с исходным кодом.

5.2 Спецификация формата сообщений сборщиков данных

Для обработки сообщений сборщиков была разработана спецификация, которой следует придерживаться для оптимальной работы и корректного разбора сообщений по финальным хранилищам.

Сборщики должны выводить в поток стандартного вывода сообщения в виде сериализованного Json-объекта в формате {"MessageType" : value}.

Поддерживаемые типы сообщений:

- Item – элемент собранных данных;
- Log – лог с уровнем (DEBUG, INFO, WARNING, ERROR, CRITICAL);

- Request – элемент с описанием выполненного запроса;
- Stats – текущие метрики (статистика) сборщика;
- Finish – элемент с описанием статуса завершения сбора.

Описание сообщения типа Item:

- Item – {"Item" : itemContent}, где itemContent – json-объект с допустимыми полями:
 - _url – url-адрес страницы, откуда извлечены данные;
 - _timestamp – время сбора;
 - _attachments – вложения:
 - path – путь в файловом хранилище;
 - filename – имя файла;
 - checksum – хеш файла;
 - value – собранный элемент.

Остальные типы сообщений схожи по структуре с незначительными смысловыми изменениями.

6. Архитектура системы

Общая архитектура системы представлена на рис. 1.

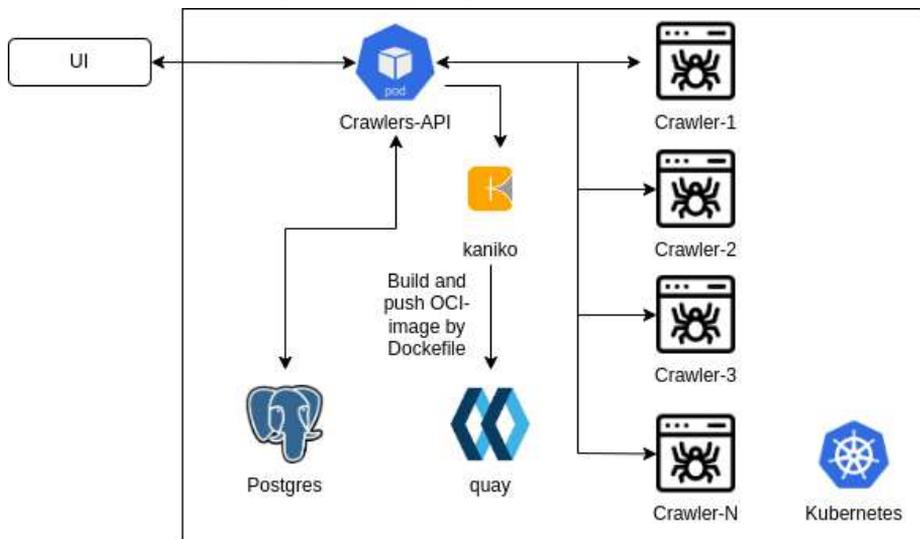


Рис. 1. Диаграмма основных компонентов системы
Fig. 1. Diagram of the main components of the system

Для реализации системы использовался язык программирования Scala.

skuber предоставляет собой полнофункциональный, высокоуровневый и строго типизированный Scala API для управления ресурсами кластера Kubernetes (такими как Pods, Services, Deployments, ReplicaSets, Ingresses и т. д.) через сервер Kubernetes REST API. Был выбран именно skuber, поскольку, по сравнению с его аналогами, является наиболее легковесным и распространённым, с хорошей документацией.

Quay – это реестр образов контейнеров, который позволяет создавать, организовывать, распространять и развертывать контейнеры. В текущем случае необходим такой репозиторий, поскольку есть потребность в хранении собранных OCI-образов проектов, с которыми далее будет вестись работа. Таким образом, каждый проект будет сохраняться отдельно от любого другого, а значит, не будет библиотек, требуемых для других проектов.

Рассматривая работу Kubernetes и API как основную часть, можно выделить общие закономерности:

- в начале каждого из реализованных методов собирается собственный шаблон задания из Kubernetes с указанием контейнера, его переменных, параметров запуска и непосредственно команды;
- каждый шаблон является основой для задания Kubernetes, которое при помощи фреймворка skuber удается собрать;
- каждое задание генерирует модуль, внутри которого разворачивается docker-контейнер и выполняются необходимые процессы;
- цикл жизни каждого пода отслеживается, и в случае его успешного завершения из Kubernetes удаляются выполненные задания, модули, а также конфигурационные файлы модулей.

Ключевой список методов, необходимый для реализации и полноценного взаимодействия с системой управления заданиями:

- `createProjectVersion` – создание новой версии проекта;
- `getCrawlers` – извлечение списка сборщиков проекта;
- `cancelJob` – отмена задачи сбора данных;
- `scheduleJob` – постановка в очередь задачи сбора данных;
- `deleteProject` – удаление проекта;
- `deleteProjectVersion` – удаление версии проекта;
- `listJobs` – список всех задач по проекту в разрезе статуса задачи (ожидание, в работе, выполнено).

6.1 Особенности реализации контракта загрузки проекта

Загрузка проекта происходит в несколько этапов. Сначала с пользовательского интерфейса на сервер поступает файл с проектом, который сохраняется в базе данных. Далее выполняется разархивирование проекта, извлечение необходимых файлов во временную папку. На основе путей к извлеченным данным и шаблонных Docker-файлов составляется конфигурационный файл модуля и сохраняется в Kubernetes. Далее создается контейнер со всеми необходимыми переменными, образом которого является `kaniko` – инструмент для сборки ОСИ-образов внутри Kubernetes на основе `Dockerfile`.

С использованием вышеперечисленных компонентов формируется задача сбора образа и размещения его в `quay` – репозитории собранных образов. Запуск задачи генерирует под, в котором можно отслеживать ход работы, его статус отслеживается для передачи информации о завершении работы пода. Поскольку мы не можем знать заранее, успешно ли соберется образ проекта, сможет ли последний попасть в хранилище, то данный метод реализован на `Promise`, которые возвращают результат только в том случае, если под получил статус «Завершен» или «Ошибка».

6.2 Загрузка проекта через постоянные тома

Второй реализованный метод работает несколько иначе, дольше, но при этом универсальнее и не имеет лимита на объем загружаемых сборщиков. Изначально создается временная директория, в которую складываются шаблонные параметры сборки ОСИ-образа. Затем в нее же распаковывается загружаемый архив с флагом замены в случае, если какой-либо файл уже существует. После подготовительных действий создается постоянный том (`Persistence Volume Claims`) с правами чтения и записи единожды и заданным размером. Создается специальный временный модуль, к которому как том добавляется ранее созданный

постоянный том. После создания постоянного тома и модуля выполняется команда, которая копирует содержимое из временной папки на сервере, который монтируется по пути /mnt. Также был разработан специфичный для фреймворка Scrapy sh-скрипт, который используется по завершению сборки образа для проверки корректности и содержит команды:

- `activate` – опциональное поле, в случае если надо выполнить какие-то дополнительные действия при загрузке проекта. Вызов этой команды должен быть выполнен в рамках `Dockerfile`;
- `list` – обязательное поле, должен возвращать список загруженных сборщиков;
- `schedule` – обязательное поле, должен выполнять запуск сборщика, а также передавать в качестве аргументов имя сборщика, его аргументы и настройки.

6.3 Извлечение списка сборщиков

Для данного метода уже необходимо знать, по какому пути в хранилище лежит собранный несколько шагов назад образ проекта. На основе сохраненного образа строится контейнер, которому подается аргумент `list` для скрипта-входной точки, получающего список сборщиков. В случае, если загруженный проект не имеет `sh`-скрипта, будет использован шаблонный, нацеленный на `scrapy` проект. Поэтому `scrapy`-проекты можно загружать без `sh`-скриптов, но для остальных технологий сбора данных необходимо указывать, как именно выполнять команды в созданном контейнере. Составленное задание Kubernetes автоматически скачает указанную версию проекта и применит к последнему команду. Результатом считается модуль в статусе «Выполнено» или «Ошибка». Список сборщиков извлекается из потока стандартного вывода модуля.

6.4 Создание запуска

Поскольку процесс сбора сообщений может занимать продолжительное время, для пользователя успешным созданием задачи сбора данных является успешно созданное задание Kubernetes, а сама работа и обработка сообщений выполняются в фоновом режиме, незаметно для пользователя. Для удобства мониторинга за запусками все созданные задачи имеют следующий шаблон:

`crawl-pid-«id проекта»-«модифицированное имя сборщика»-«номер версии»-jobid-«id задачи запуска»`

При этом переданные настройки и аргументы преобразуются из пары «ключ-значение» в строку, которую может принять командная строка. Создается контейнер, которому передается аргумент `schedule` для скрипта-входной точки с именем сборщика, настройками и аргументами. На основе этого контейнера создается задание Kubernetes, которое впоследствии выполняется.

Во время работы запуска выполняется команда, которая складывает все записи из потока стандартного вывода модуля во временный файл, с которым будет вестись работа далее. Записанные в файл логи считываются и проверяются по их типу. Если записано сообщение в формате, отличному от указанного в контракте, то оно игнорируется. Если формат корректный, то сообщение дополнительно насыщается необходимой информацией и распределяется:

- `Item` – отправляется в Postgres для хранения;
- `Log` – отправляется в Elasticsearch с индексом `LogsIndex`;
- `Request` – отправляется в Elasticsearch с индексом `RequestIndex`;
- `Stats` – отправляется в Elasticsearch с индексом `MetricsIndex`;
- `Finish` – отправляется в Elasticsearch с индексом `LogsIndex` и пометкой как сообщение

окончания работы сборщика.

6.5 Отмена запуска

Функция отмены запуска реализована как установка значения «Отменено» в поле «Статус сбора» в базе данных. Все запуски, которым проставлено данное значение, перестают отслеживаться системой.

6.6 Список запусков

Список запусков получается путем запроса в Kubernetes тех заданий, у которых имеется пометка с ключом *projectId* и значением из списка запрашиваемых.

Правило установки статуса запускам:

- в ожидании – если у задания Kubernetes отсутствует статус, или сформированные задачей модули Kubernetes не имеют статусов «Завершено» или «Ошибка»;
- выполняется – если у задания Kubernetes есть статус и сформированные модули имеют статус «Активно»;
- завершено – если у задания Kubernetes есть статус и сформированные модули имеют статус «Завершено» или «Ошибка».

7. Экспериментальное тестирование

Прежде всего проверялась корректность работы системы манипуляциями со *scrapy*-сборщиками, поскольку они являются приоритетными и требование по обратной совместимости одно из важнейших. Но поскольку реализация подразумевает поддержку работоспособности сборщиков на прочих фреймворках и языках программирования, то был реализован тестовый сборщик на языке программирования Scala.

При помощи тестового сборщика¹ можно полностью промоделировать поведение любого другого сборщика. Достигается этот эффект за счет возможности детальной настройки ожидаемого результата при помощи аргументов.

Как результат, предложенное решение удовлетворяет каждому поставленному требованию.

За счет выбора архитектурного решения и системы оркестрации контейнерных приложений обеспечивается горизонтальная масштабируемость и изолированность зависимостей сборщиков.

Также были реализованы все необходимые взаимодействия в рамках жизненного цикла сборщика, продемонстрирована реализация и тесты, подтверждающие возможность работы с приложения сбора на основе других фреймворков.

8. Заключение

В работе создана методология разработки системы управления заданиями сбора данных из сети Интернет.

В процессе выполнения проекта было проведено исследование, которое включало анализ литературы и существующих решений для управления заданиями сбора данных из сети Интернет. Обзор показал, что текущие решения с открытым исходным кодом зачастую завязаны на один язык программирования и фреймворк разработки сборщиков, имеют проблемы с обеспечением изолированности и ограниченную степень масштабирования, так что они применимы только для небольших проектов по сбору данных. Платные решения лишены части проблем, но не обладают исходным кодом для дальнейших модификаций. Мы обосновали необходимость использования инструментов контейнерной оркестрации и

¹ https://github.com/vlazarew/scala_test_crawler

провели анализ существующих систем, по итогам которого для разработки выбран Kubernetes.

Для создания системы управления заданиями автоматизированного сбора данных мы определили сущности, с которыми такая система должна работать, и выделили основные элементы жизненного цикла абстрактного сборщика. Затем была описана методология для работы со сборщиками, реализованными над контейнерами, включающая спецификации для загрузки и сборки образов проектов сбора, запуска сбора и получения результатов запуска.

На основе описанной методологии реализована и внедрена в Talisman система управления заданиями автоматизированного сбора данных из сети Интернет, которая удовлетворяет требованиям горизонтальной масштабируемости, полной изолированности зависимостей сборщиков и независимости от технологий их разработки.

Список литературы / References

- [1] ИСП РАН. Talisman: платформа для обработки данных. Доступно по ссылке: <https://www.ispras.ru/technologies/talisman/> ISP RAS. Talisman: a data processing framework. Available at: <https://www.ispras.ru/en/technologies/talisman/>
- [2] Anand V. Saurkar, Kedar G. Pathare, Shweta A. Gode. An Overview on Web Scraping Techniques and Tools. International Journal on Future Revolution in Computer Science & Communication Engineering, vol. 4, no. 4, 2018, pp. 363 - 367
- [3] IST Research. Scrapy Cluster 1.3 Documentation. Available at: <https://scrapy-cluster.readthedocs.io/en/dev/>.
- [4] Scrapy group. Scrapyd. Available at: <https://scrapyd.readthedocs.io/en/stable/>.
- [5] ScrapyRT (Scrapy Realtime). Available at: <https://github.com/scrapinghub/scrapyrt>.
- [6] Ferrit. Available at: <https://github.com/reggoodwin/ferrit>.
- [7] Zyte. Web Scraping Cloud Hosting Data Extraction - Zyte. Available: <https://www.zyte.com/scrapy-cloud/>.
- [8] Web Scraper Cloud. Web Scraper Cloud | Web Scraper documentation. Available: <https://webscraper.io/documentation/web-scraper-cloud>.
- [9] Octopus Data Inc. Web Scraping Tool & Free Web Crawlers | Octoparse. Available at: <https://www.octoparse.com/>, 2022
- [10] data-ox.com. Web Data Scraping Company | DataOx. Available at: <https://data-ox.com/>.
- [11] The Kubernetes Authors. What is Kubernetes? Available at: <https://kubernetes.io/docs/concepts/overview/what-is-kubernetes>.
- [12] Docker Inc. Overview of Docker Compose. Available at: <https://docs.docker.com/compose/>.
- [13] Docker Inc. Swarm mode overview. Available at: <https://docs.docker.com/engine/swarm/>.
- [14] The Apache Mesos Software Foundation. Mesos Architecture. Available at: <https://mesos.apache.org/documentation/latest/architecture/>.
- [15] Isam Mashhour Al Jawarneh, Paolo Bellavista et al. Container Orchestration Engines: A Thorough Functional and Performance Comparison. In Proc. of the IEEE International Conference on Communications (ICC), 2019, pp. 1-6

Информация об авторах / Information about authors

Владимир Александрович ЛАЗАРЕВ является студентом магистратуры кафедры системного программирования МГУ и работает в ИСП РАН. Его научные интересы включают сбор данных из открытых источников, исследование и разработку систем управления заданиями сбора данных.

Vladimir Alexandrovich LAZAREV is a master's student of the System Programming Department of MSU and also is an employee of ISP RAS. His research interests include open source data collection, research and development of data collection task management systems.

Максим Игоревич ВАРЛАМОВ – научный сотрудник ИСП РАН. Сфера научных интересов: автоматизация сбора данных из веб-ресурсов, машинное обучение для извлечения информации из Веба.

Maksim Igorevich VARLAMOV – researcher at ISP RAS. Research interests: automation of data collection from web resources, machine learning for web data extraction.

Александр Константинович ЯЦКОВ – аспирант ИСП РАН. Сфера научных интересов: сбор данных из Веба, автоматизация процесса сбора данных, фокусированный сбор данных, извлечение информации, машинное обучение.

Alexander Konstantinovich YATSKOV – PhD Student at ISP RAS. Research interests: web crawling, web data extraction, focused crawling, information extraction, machine learning.

DOI: 10.15514/ISPRAS-2022-34(2)-10



Automated Object Storage Management Approach with Operator SDK and Custom Resource Definition

K.M. Stonozhenko, ORCID: 0000-0001-7463-9083 <stonozhenko.km@edu.spbstu.ru>

I.V. Nikiforov, ORCID: 0000-0003-0198-1886 <nikiforov_iv@spbstu.ru>

S.M. Ustinov, ORCID: 0000-0003-4088-4798 <usm50@yandex.ru>

*Peter the Great St. Petersburg Polytechnic University,
29, Polytechnicheskaya st., St. Petersburg, 195251, Russia*

Abstract. The work is devoted to the study of automation tools for managing stateful applications in the Kubernetes environment, particularly object storage systems. A review of existing management tools capable solving the set tasks is made. A comparative description of the considered tools based on review is given and a tool is selected that meets the introduced criteria: popularity, support form Kubernetes, reactivity of developed operator, additional features, and others. An approach to automatic object storage management using the Operator SDK and Custom Resource Definition is suggested. As a result of comprehensive comparative analysis of tools Kubebuilder, Juju, Metacontroller, Kudo and Operator SDK, the last one was chosen as a base of approach implementation. The architecture of the system for managing a containerized version of storage systems based on the Kubernetes platform and integrating the operator with a user monitoring system is proposed. The described approach is implemented in a software tool – an operator of the object data storage system resource. The paper describes the details of software implementation, the structure of the storage custom resource descriptor, and methods for testing the end system. As a result, an object storage management system based on the Kubernetes platform was created, which made it possible to reduce both labor costs for supporting and maintaining the system, and it's cost by reducing dependence on hardware. Moreover, described approach corresponds to such features of modern object storages as multi-tier, erasure coding support, geo-replication, cluster topology that is quite innovative among existing automated storage management approaches on Kubernetes platforms.

Keywords: Kubernetes; object storage; data storage system; operator; custom resource definition; container; orchestration

For citation: Stonozhenko K.M., Nikiforov I.V., Ustinov S.M. Automated Object Storage Management Approach with Operator SDK and Custom Resource Definition. Trudy ISP RAN/Proc. ISP RAS, vol. 34, issue 2, 2022, pp. 123-134. DOI: 10.15514/ISPRAS–2022–34(2)–10

Подход автоматического управления объектным хранилищем с помощью Operator SDK и Custom Resource Definition

К.М. Стоноженко, ORCID: 0000-0001-7463-9083 <stonozhenko.km@edu.spbstu.ru>

И.В. Никифоров, ORCID: 0000-0003-0198-1886 <nikiforov_iv@spbstu.ru>

С.М. Устинов, ORCID: 0000-0003-4088-4798 <usm50@yandex.ru>

*Санкт-Петербургский политехнический университет Петра Великого,
195251, Россия, Санкт-Петербург, Политехническая улица, д. 29*

Аннотация. Данная работа посвящена исследованию средств автоматизации управления приложениями с сохранением состояния в среде Kubernetes, в частности объектными системами хранения данных. Произведён обзор существующих инструментов управления, способных решить поставленные задачи, на основе которого дана сравнительная характеристика рассмотренных средств и выбран инструмент, удовлетворяющий введённым критериям. По результатам сравнительного

анализа представлен подход автоматического управления объектным хранилищем с помощью Operator SDK и Custom Resource Definition. Предложена архитектура системы управления контейнеризированной версией СХД на платформе Kubernetes и интеграции оператора с пользовательской системой мониторинга. Описанный подход реализован в программном средстве - операторе ресурса объектной системы хранения данных. В работе описаны детали реализации программного обеспечения, структура дескриптора пользовательского ресурса хранилища и методы тестирования конечной системы. В результате была получена система управления объектной СХД на платформе Kubernetes, что позволило сократить как трудозатраты по содержанию и обслуживанию системы, так и её стоимость за счёт уменьшения зависимости от аппаратного обеспечения.

Ключевые слова: Kubernetes; объектное хранилище; система хранения данными; оператор; custom resource definition; контейнер; оркестрация

Для цитирования: Стоноженко К.М., Никифоров И.В., Устинов С.М. Подход автоматического управления объектным хранилищем с помощью Operator SDK и Custom Resource Definition. Труды ИСП РАН, том 34, вып. 2, 2022 г., стр. 123-134. DOI: 10.15514/ISPRAS-2022-34(2)-10

1. Introduction

The amount of information produced by humanity and requiring storage is increasing every day - a study [1] conducted by the consulting firm IDC (International Data Corporation), specializing in information technology [2], showed that approximately 1.7 Mb of data is generated every second by single human, which in absolute terms is approximately 40 zettabytes per second.

Modern personal computers allow storing up to several terabytes of data, which, on the one hand, is a sufficient amount of memory for the average computer user, but on the other hand, a very modest value for large companies that produce and process petabytes and even exabytes of data. Social networks, file hosting, web applications and other programs that communicate via the Internet can exchange even a larger amount of information that require considerable memory resources for storage.

In addition to information carriers, modern data storage complexes should also contain computing devices capable of performing multi-threaded continuous operations of writing, reading, deleting, and modifying data. Along with this, data storage systems are also responsible for the security and integrity of stored information – access to data is provided only to those users who have the appropriate rights, and if one or more hardware storage nodes fail, the data must be restored using the remaining copies.

All these requirements are implemented in object storages [3], a distinctive feature of which is the storage of all data in the form of objects located at the same logical level and defined by a set of metadata. The absence of a hierarchy in such sort of storages makes this solution infinitely scalable, which allows deploying system on a group of independent nodes called a cluster. The object-oriented nature of the storage system indicates lack of data unit structure, so any data format can act as a storage object and makes this type of storage universal.

However, the above specifics of object storage systems as a software solution dictates certain requirements for the hardware, which increases the cost of equipment and, as a result, reduces the potential customer base of the manufactured complex.

Reducing the dependence between the hardware platform and the software of the system could be solved by the well-known virtualization technique [4] – creating an additional software abstraction layer between the user application and the hardware. However, until recently, existing solutions for deploying virtual machines on a cluster were poorly aligned with the idea of horizontal scaling and required a significant pool of resources.

Everything has been changed with the advent of containerization [4-7] - a way to allocate a lightweight set of system resources to programs – and container orchestration systems [8-10] – software that monitors the execution of containers on a provided set of resources and is responsible for maintaining a certain state of the cluster.

One of the most popular orchestration systems – Kubernetes [11-13]. It is designed to run a huge number of programs wrapped in containers on a set of nodes united in one network, which allows scaling the cluster vertically and horizontally without limits. One of the distinguishing features of Kubernetes is the presence of a replication controller [14] responsible for maintaining multiple copies of stateless containers [11, 15]. Stateless programs can be web clients that form user requests based only on input data, and simple web servers that send responses to a request based only on its content.

Along with stateless applications, Kubernetes terminology also defines stateful applications that are forced to save their state as they execute. Their replication can no longer be performed by sequentially deleting a copy of the program and recreating it in a new location. Work features of each individual stateful application require the creation of a separate custom replication controller, also called an operator [16,17], which monitors the state of its replica set.

Now there are many tools for creating stateful application operators in the Kubernetes environment. Some of them, considered in this paper, have been used by stateful software developers to create their own Kubernetes operators. Such stateful software include databases, metric collectors, machine learning programs, logging systems, and many others.

Transferring an object storage system to the Kubernetes container platform and creating a separate operator for monitoring the state of stateful components is an advanced and relevant direction in the development of object storage, which reduces the cost of storage development, testing, implementation, and maintenance, and also provides the ability to deploy the system on almost any set of hardware resources.

In this paper, we consider an approach to automatic management of object storage based on the use of the Operator SDK framework and Custom Resource Definition.

2. Related works

One of the main components of the developed approach for managing object storage in the Kubernetes environment is an operator – a kind of user resource controller. Developing an operator from a scratch, just like its further maintenance as a product, is a time-consuming and expensive process, which is a consequence of both the complexity of the internal structure of Kubernetes controllers and the rather frequent change of Kubernetes API versions.

In this regard, many tools have been created for your own Kubernetes custom resource operators' implementation. Some of these tools are described below.

Kubebuilder is a command line utility, providing functions for extending the Kubernetes cluster API. The main purpose of the program is to create custom resources [17] and controllers for their monitoring. The utility is implemented and builds controller code in Golang, the native language of the Kubernetes infrastructure. The distinctive features of the Kubebuilder project include the ability to include the configuration and the versioning of the resource being defined.

An alternative to Kubebuilder is the Operator SDK, an open-source toolkit whose main purpose is to build Kubernetes operators. The set includes the operator-sdk utility, which provides a list of commands for generating an operator template for any type of custom resource. The operator itself can be implemented both in Golang (using scripts from the Kubebuilder library) and using Ansible playbooks - configuration management scripts - or Helm charts. In addition, the Operator SDK allows you to connect the developed operator to the Operator Lifecycle Manager - a separate operator lifecycle management program, whose tasks include monitoring the use of system resources, saving metadata about a specific version of the operator, simplifying the deployment, deletion and testing of the custom operator.

Another tool for creating Kubernetes operators is Juju [18]. In Juju terminology, user resource controllers are called «charms» [19]. Python is the preferred language for implementing charms, but other scripting languages can be used that include libraries for interacting with Juju. Along with the «juju» and «charm» command line utilities, the Juju package also downloads the Charmed Operator

Lifecycle Manager, similar to the Operator Lifecycle Manager from the Operator Framework. But unlike the second program, the controller supplied by Juju is the software required to run on the cluster in parallel with user statements, which negatively affects the performance and complexity of system developed.

You can also use Metacontroller [20], a Kubernetes API extension developed by Google, to create custom operators. The tool provides the ability to create bundles of controllers and stateful web applications through a web-hook. These applications implement custom resource management logic and can be written in any language capable of handling JSON-like objects, such as Jsonnet or Python. Google's suggestion is a declarative way to create a Kubernetes operator, which simplifies the development process, but increases the load on the cluster due to the deployment of additional entities.

For a completely declarative way to create Kubernetes operators, you can apply KUDO (Kubernetes Universal Declarative Operator) – kubectl Kubernetes CLI plugin. Statements developed with KUDO manage user resources by defining tasks and plans in statement descriptors in YAML format. The undoubted advantage is the ease of creating controllers, however, the declarative method of setting operators reduces the ability to manage resources - KUDO allows you to deploy complex stateful applications but is not able to respond to changes in the cluster.

A full comparison of the considered tools is presented in Table 1.

Table 1. Comparison of Operator development tools

Criteria \ Tool name	Kubebuilder	Operator SDK	Juju	Metacontroller	KUDO
Popularity (thousands of Google search results)	92	18 100	1 610	6 360	611
Requires additional controller level	-	-	+	+	-
Support from k8s	+	+	-	+	-
Additional opportunities	+	+	-	-	-
Operator reactivity	+	+	+	+	-

During the comparison, the Operator SDK was chosen as the tool for creating the Kubernetes object storage operator, as it meets our requirements more than others.

3. Proposed Approach description

3.1. Approach features

The software of modern object data storage systems is a combination of several services that implement various functions. Among them are software components for accounting for user roles of the storage system, providing access to one or more interfaces of various levels, distributing data between several geographically separated clusters, and many others.

The main service of the system is the storage service, a component responsible directly for encoding and saving user data. Encoding is understood as the splitting of the input stored information into several sets of transformed data, from which it is possible to restore the original information. The number of sets and the type of transformation depend on the selected data encoding scheme. The key feature of this procedure is the guarantee that when the received data sets are placed on different nodes and one or more of them fails, the system is able to restore all information on the remaining sets, after some time recode the data and save them on the working nodes.

The difference in data encoding schemes complicates the process of managing several storage service replicas, therefore, the main difficulty in designing an automated object storage management system is focused on the replication of this component.

Furthermore, none of modern approaches of storage system management on virtualized deployments included extension of Kubernetes replication controlling system in order to maintain erasure coding schemes and geo-replication of stored data. Storage service replication fully demands on orchestrator or simple custom controller reconciliation logic [21, 22].

The main features of the developed approach include:

- Reducing labor costs in the implementation, maintenance and operation of the system;
- Transition to implementation of storage on any provided set of computing resources;
- Abstracting from the hardware component of the system and shifting the responsibility for scaling and replicating storage systems to the Kubernetes platform;
- Use built-in Kubernetes resources to create our own event monitoring system.

The first three features of the approach are related to the transfer of the system to a virtual Kubernetes platform. The system requirements for the hardware are dictated by the very platform, that takes responsibility for connecting the nodes into a single network and deploying Kubernetes on the resulting cluster. Also, the platform can provide a graphical user interface of various levels for managing the network, nodes, Kubernetes resources and interacting with executable programs.

Elastic Cloud Storage (ECS) [23] from Dell Technologies was chosen as the object storage to which the developed automation approach is applied. While vSphere from VMware and OpenShift from RedHat were the Kubernetes platforms that should support deployment of containerized version of object storage.

The fourth specified feature of the approach involves the use of the event resource to monitor the state of storage systems during service procedures - cluster maintenance operations or changing storage system parameters. Examples of service procedures include the following operations:

- Horizontal storage server expansion – increase of service replicas number,
- Vertical storage server expansion – increase of volumes number that bound to each replica,
- Object storage version upgrade,
- Volume or disk replacement,
- Entering maintenance mode of cluster node, etc.

During each of the presented service procedures processing, the system must ensure the correct replication of all components of the object storage and notify the user of all changes in the state of the storage system.

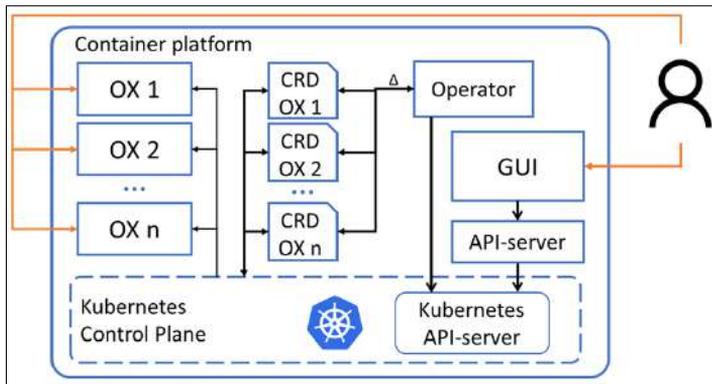


Fig. 1. System architecture

3.2. Architecture

A Kubernetes operator is required [24] to organize automatic management of object storage components when deploying a system or performing service procedures. Its principle of work is shown in Fig. 1.

All software components of the system are executed on the Kubernetes platform, that provides a graphical user interface and a deployed Kubernetes orchestration system. The central element of the orchestration system is the Control Plane. Interaction with it is carried out through the Kubernetes API-server, which accepts requests from any client – the native CLI kubectl [13] or a program using the k8s-client library.

The object storages deployed on the platform, on the one hand, are a set of Kubernetes services, each of which requires the creation of a collection of Kubernetes resources, such as volumes, pods, cluster-roles, replica sets [11] and others. Control plane manages all these resources according to YAML descriptors containing a description of the desired state of the resource.

On the other hand, object stores are resources themselves that are a user-defined extension of the Kubernetes API. Each storage instance, like any resource, is described by its own descriptor. But since the resource is user-defined, its descriptor is called Custom Resource Definition [17], and its management is carried out by a custom controller named operator.

Object storage is managed by the operator upon the occurrence of certain events associated with a change in the state of nodes, volumes, or storage descriptors. These events are triggered by the launch of service procedures that are initiated by the user using the platform's graphical interface. The operator manages the Kubernetes resources of the storage system components by generating requests to the Kubernetes API server [11].

The collaboration of the operator and the Kubernetes control plane maintains the correct operation of the storage system, which is accessed by the user through any provided storage interface (HTTP, S3, etc.)

3.3. Operator and monitoring system interaction

Event Kubernetes-resource [11] allows you to monitor the status of other cluster resources. Generation of events during the operator's work can be useful both for tracking the current state of the storage system by the end user, and for implementing a reaction to some errors that occur during object storage management.

If some error occurs during the execution of any service procedure, it is not enough to inform the user about it. Depending on its severity, the system should give recommendations for its elimination, ask for help from the system administrator or even from the support engineer responsible for the implementation and maintenance of the software product.

To implement the described logic, a Kubernetes event monitoring system is required - an intermediate layer between the events generated by the system components and all the specified actors.

The general scheme of interaction between the operator and the monitoring system is shown in Fig. 2.

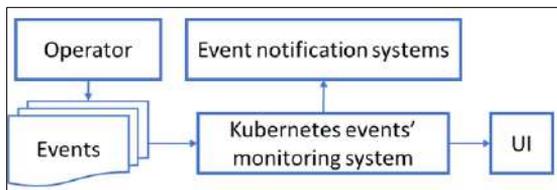


Fig. 2. Operator and monitoring system integration

4. Implementation

4.1. Object storage descriptor design

To organize the processing of all components of the object data storage system by the operator as a single whole, it is necessary to implement a custom resource definition (CRD) containing the settings of all services presented in a containerized form.

Like all Kubernetes descriptors, the storage CRD contains the following sections:

- Used Kubernetes API version (`apiVersion`),
- Kubernetes resource kind,
- Resource metadata,
- Resource specification (`spec`),
- Its current status.

Since the storage descriptor is a custom resource definition, an extension of the standard Kubernetes API, then `apiextensions.k8s.io/v1` acts as the CRD version, and `CustomResourceDefinition` as the resource type. Metadata includes resource name, entity creation time, various labels and annotations. The specification of a resource is its main characteristic since it contains all the desired settings for both the resource itself and all its components. For instance, for the minimal executable unit of a Kubernetes cluster – a pod – the specification contains the launch parameters of all containers, associated volumes, the name of the node on which the pod want to be running, and other information. Object storage, on the other hand, consists of dozens of components, for each of which it is necessary to define a service, a replica set, a set of stateful pods (`statefulset`) [11], or some other resources and their characteristics. Therefore, the storage CRD specification is the largest part of the descriptor.

CRD status contains current information about the state of the resource, that allows you to compare it with the desired state. The distinctive parameters stored in this section of the descriptor of our storage system are:

- Phase that reflects storage current state,
- Components' subsection containing brief information about the current state of all storage components,
- Conditions [25] - a special extension of the resource status that allows you to track the time, cause and status of transitions between resource phases,
- Additional information used by the operator in the course of managing the storage resource, such as information about the nodes that are in maintenance state.

4.2. Reconcile loop implementation

Custom resource processing is implemented in software using the Operator SDK. As mentioned earlier, the package includes the `operator-sdk` utility, which is used to create, build, run and test the custom resource operator.

The basis of the program is the reconcile loop [24], during which the operator reads the descriptor of the controlled resource and performs actions that bring it to a state corresponding to the specification. In fact, this is the main task of the controller. The cycle runs at a certain interval (for example, once a minute) and checks whether the state of the storage components corresponds to the status of the cluster or the ongoing service procedure. Fig. 3 shows the general scheme for processing a user resource in a loop.

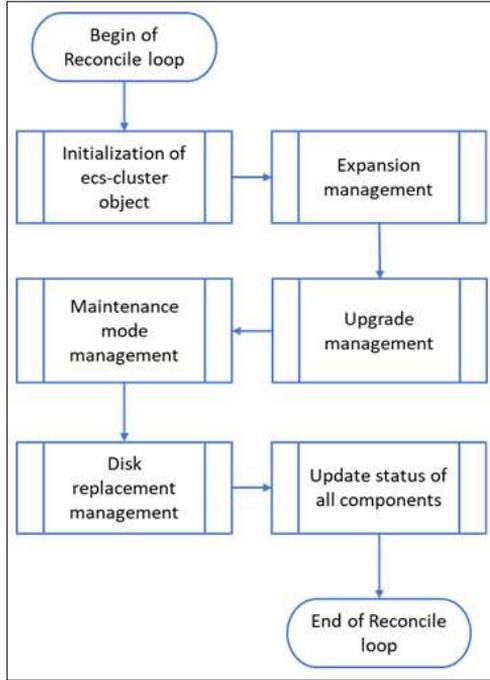


Fig. 3. Storage resource Reconcile loop

The storage processing logic encapsulated in Operator allows us to track the state of the storage system using a phase – a capacious description of the current status of the storage system. The behavior of our user resource can be described by the storage resource state machine, presented on Fig. 4:

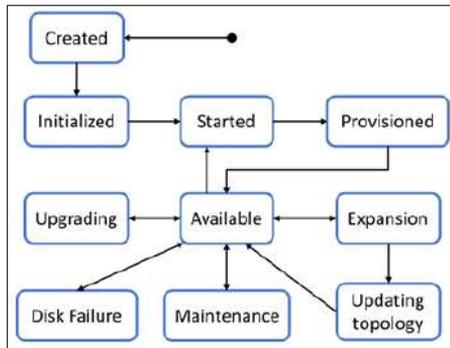


Fig. 4. Storage resource Reconcile loop

4.3. Monitoring system integration

In addition to the current storage phase, the end user can monitor the flow of service procedures and their impact on the operation of object storage components using events. Developed separately from the operator, the event monitoring system allows you to define the severity, the resource and application involved, as well as the cause and message of the event.

If the severity of the event is higher than normal (warning or error), the system generates an issue, which, depending on the settings, can be resolved by the user himself or sent for investigation to the software product support service.

5. Results

5.1. Testing methods

Testing of the implemented system is carried out at different levels - at the level of the operator code and at the level of the entire system.

To test the functional changes of the operator unit and integration tests were created using the Gomega and Ginkgo frameworks, the main scenarios of which include checking the correct processing of the storage descriptor during various service procedures.

Operator testing as part of the operation of the entire object storage management system at the Kubernetes cluster level is performed using the Jenkins automatic CI/CD pipeline, which downloads software dependencies and the Operator SDK, code syntax analysis, executes tests, builds the image and uploads it to the registry. A pipeline is also implemented that allows you to deploy an entire storage management system on a provided virtual or bare-metal cluster. The resulting system is ready for all kinds of end-2-end tests.

To monitor the absence of a decrease in the efficiency of the object storage during system development, a cluster is regularly allocated for load testing: various service procedures are alternately performed for several days. At the same time, object storages constantly take on the load – the main operations are:

- creation and deletion of user with different roles,
- creation and deletion of namespaces,
- creation and deletion of buckets,
- write, read, modify and delete files to/from buckets,
- file metadata search.

To monitor the state of the storage during testing, the Grafana metrics collection and visualization tool is used [26]. The main metrics collected include the execution time of REST requests, the percentage of unsuccessful requests, network bandwidth, etc.

5.2. Comparison to classic approach

The main purpose of the developed object storage management system on the Kubernetes platform is to increase the degree of automation of storage management when changing the configuration of both the physical cluster and the storage system itself. To demonstrate the reduction in labor intensity of storage management, a comparison was made of the time to remove a disk from a cluster between a manual approach and the proposed automated approach.

The list of actions for an engineer when removing a disk as part of a classic deployment of object storage without the use of container technologies includes points, presented below.

1. Simulation of disk outage on HAL level.
2. Waiting for data recovery with rest storage replicas.
3. Logic disk replacement from cluster with fabric CLI.
4. File system deletion.
5. Partition table formatting.
6. Disk wipe.

To remove a disk in the developed system, you must perform the following operations in the UI of the container platform:

1. Annotate unhealthy disk or volume.

2. Wait Service procedure to finish.

The average disk replacement time for the classical approach is ~3 hours, and for the automated one it is approximately 1 hour. In total, we have a gain in time of about three times.

6. Conclusion

Work on operator of the object data storage system helped us to reach following important results. Firstly, an approach of automated object storage management system was introduced. Distinctive feature of the approach is Kubernetes platform. Operator is used as a crucial element of the system as it performs CRUD operations on Kubernetes resources to manage components of storage that is presented by its own Custom Resource Definition.

Secondly, the operator of the object data storage system developed within the framework of this project was successfully implemented into the storage management system on the container platform. The operator not only manages the storage components during service procedures, but also generates a set of Kubernetes events displayed in the platform UI and informing the user about the general state of the storage.

Operator was implemented based on comprehensive analysis of existing solutions: different tools to develop operator of custom Kubernetes resources were studied that helped us to form knowledge about different ways to build custom controllers and to choose Operator SDK as the base of operator development framework according to its benefits.

Finally, resulting system was compared with storage build using classic approach, that let us to see such qualitative changes as possibility to run new solution on wide specter of container platforms, use different tools made in Kubernetes infrastructure to improve storage system and reduce of cost of outcome product as the responsibility of deployment, replication and low-level system management lies on orchestrator.

Manufactured software solution has shown increase in performance and more wide set of functionalities in comparison to classic object storage system, and was applied to real enterprise object storage system.

On the basis of results further directions of project development were suggested:

- new automation pipelines,
- use of new Operator Framework features,
- multiple Service Procedure support,
- introducing new CRD for service procedures.

One of them is the development of new automation pipelines for configurable regression testing. The Robot Framework and Jenkins capabilities allow us to add settings to run a specific set of end-2-end tests for each service procedure. On the one hand, this would reduce the load on the clusters allocated for testing, and on the other hand, it would cover many previously untested use cases.

Operator Framework was also improved and issued several versions, which now allows us to add to the project such tools as collecting metrics for the operator and all resource controllers included in it. Analysis of the measured parameters will help us to find solutions to improve the performance and speed of the software and reduce the consumption of cluster resources.

Moreover, one of the directions of future work on the operator is the support of several parallel processed service procedures. Modifications in the operator code and specification of the storage resource would allow, for example, updating the storage system version and simultaneously deleting one of the cluster disks.

Furthermore, development of additional components of storage management system showed us that they also need some operator-side management during service procedure handling. Introducing new service procedure CRD would simplify monitoring of different service procedures that affect not only storage services, but also components of the management system.

References / Список литературы

- [1]. J. Gantz, D. Reinsel. The Digital Universe in 2020: Big Data, Bigger Digital Shadow s, and Biggest Growth in the Far East. IDC IVIEW. Available: <https://www.cs.princeton.edu/courses/archive/spring13/cos598C/idc-the-digital-universe-in-2020.pdf>, accessed March 24, 2022.
- [2]. International Data Corporation (IDC). Changing the way the world thinks about the impact of technology on business and society. Available: <https://www.idc.com/>, accessed Mar. 24, 2022.
- [3]. V.Yu. Shevtsov, E.S. Abramov. The Analysis of Modern Data Storage Systems. NBI technologies, vol. 13, no. 1, 2019, pp. 25-30 (in Russian) / В.Ю. Шевцов, Е.С. Абрамов. Анализ современных систем хранения данных. НБИ Технологии, том 13, no. 1, 2019 г., стр. 25-30.
- [4]. R. Dua, A.R. Raja, D. Kakadia. Virtualization vs containerization to support PaaS. In Proc. of the 2014 IEEE International Conference on Cloud Engineering (IC2E'14), 2014, pp. 610-614.
- [5]. S.M. Nanyan, T.N. Nichushkina. Virtual Docker containers: purpose and application features. Engineering Bulletin, 2015, no. 2, pp. 554-560 (in Russian) / С.М. Нанян, Т. Н. Ничушкина. Инженерный вестник, no. 2, 2015 г., стр. 554-560.
- [6]. J. Turnbull. The Docker Book, 2014, 338 p. [E-book] Available: <https://dockerbook.com/>, accessed March 24, 2022.
- [7]. D. Silakov. Docker project. Manage virtual environments. System administrator, no. 3, 2015, pp. 10-14 (in Russian) / Д. Силаков. Проект Docker. Управляем виртуальными окружениями. Системный администратор, no. 3, 2015 г., стр. 10-14.
- [8]. D. Silakov. Tools to manage multiple Docker containers. System administrator, no. 5, 2015, pp. 11-15 (in Russian) / Д. Силаков. Инструменты управления множеством контейнеров Docker. Системный администратор, no. 3, 2015 г., стр. 11-15.
- [9]. T. Uphill, J. Arundel et al. DevOps: Puppet, Docker, and Kubernetes. Packt Publishing, 2017, 924 p.
- [10]. Z.A. Orlov. Study of scaling tools for container virtualization systems. Alleya nauki, no. 9(2), 2017, pp. 867-871 (in Russian) / З.А. Орлов. Исследование средств масштабирования систем контейнерной виртуализации. Аллея науки, no. 9(2), 2017 г., стр. 867-871.
- [11]. M. Luksha. Kubernetes in action. Manning, 2018, 624 p.
- [12]. S. Yaremchuk. Get to know Kubernetes. Sistemnyj administrator, no. 1-2, 2017, pp. 41-45 (in Russian) / С. Яремчук. Знакомимся с Kubernetes. Системный администратор, no. 1-2, 2017 г., стр. 41-45.
- [13]. J. Shah, D. Dubaria. Building modern clouds: Using docker, kubernetes google cloud platform. In Proc. of the 2019 IEEE 9th Annual Computing and Communication Workshop and Conference, 2019, pp. 184-189.
- [14]. D. Safronov, K. M. Stonozhenko, I. V. Nikiforov. Automatic load balancing between streaming data processing and cluster internal tasks using Kubernetes. In Proc. of the Conference on Modern Technologies in Programming Theory and Practice, 2020, pp. 165-167 (in Russian) / Д. Сафронов, К.М. Стоноженко, И.В. Никифоров. Автоматическая балансировка нагрузки между потоковой обработкой данных и внутренними задачами кластера с использованием Kubernetes. Сборник материалов конференции «Современные технологии в теории и практике программирования», 2020 г., стр. 165-167.
- [15]. Kubernetes Deployment vs. StatefulSets. Available: <https://www.baeldung.com/ops/kubernetes-deployment-vs-statefulsets>, accessed March 25, 2022.
- [16]. Operator pattern. Available: <https://kubernetes.io/docs/concepts/extend-kubernetes/operator/>, accessed March 25, 2022.
- [17]. P.S.P. Shenoy, S.S. Vishnu et al. Enhancement of observability using Kubernetes operator. Indonesian Journal of Electrical Engineering and Computer Science, vol. 25, no. 1, 2022, pp. 496-503.
- [18]. D. Silakov. JUJU project. Deploy complex applications with one click. System administrator, 2015, no. 10, pp. 4-8 (in Russian) / Д. Силаков. Проект Juju. Развертываем сложные приложения одним кликом. Системный администратор, no. 10, 2015 г., стр. 4-8.
- [19]. Juju SDK Documentation. Available: <https://juju.is/docs/sdk>, accessed Mar. 25, 2022.
- [20]. Introduction to Metacontroller. Available: <https://metacontroller.github.io/metacontroller/intro.html>, accessed March 25, 2022.
- [21]. B. Shrishail, S. Ashish et al. A QOS-aware secure personal cloud storage with ubiquitous access and smart home extension. In Proc. of the 2015 International Conference on Computer, Communication and Control (IC4), 2015, 5 p.

- [22]. C. Wu, V. Sreekanti, J.M. Hellerstein. Eliminating Boundaries in Cloud Storage with Anna. arXiv:1809.00089, 2018, 14 p.
- [23]. Dell EMC ECS: powering a data-driven future. Available: <https://www.dell.com/en-us/dt/learn/data-storage/ecs.htm>, accessed March 25, 2022.
- [24]. A.S. Shemyakinskaya, I.V. Nikiforov. Hard drives monitoring automation approach for Kubernetes container orchestration system. *Trudy ISP RAN/Proc. ISP RAS*, vol. 32, issue 2, 2020, pp. 99-106. DOI: 10.15514/ISPRAS-2020-32(2)-8.
- [25]. Extend the Kubernetes API with CustomResourceDefinitions. Available: <https://kubernetes.io/docs/tasks/extend-kubernetes/custom-resources/custom-resource-definitions/>, accessed March 29, 2022.
- [26]. S. Yaremchuk. Deploy monitoring Prometheus + Grafana. *System administrator*, no. 5, 2017, pp. 36-44 (in Russian) / С. Яремчук. Ставим мониторинг Prometheus + Grafana. *Системный администратор*, no. 5, 2017 г., стр. 36-44.

Information about authors/ Информация об авторах

Kirill Maksimovich STONozHENKO is a master's student at the Higher School of Software Engineering. Research interests: container orchestration systems, data storage systems, distributed systems, big data, machine learning.

Кирилл Максимович СТОНОЖЕНКО – студент магистратуры Высшей школы программной инженерии. Сфера научных интересов: системы оркестрации контейнерами, системы хранения данных, распределённые системы, большие данные, машинное обучение.

Igor Valerievich NIKIFOROV is an Associate Professor at the Higher School of Software Engineering. In 2014 he defended his dissertation for the degree of candidate of technical sciences. Research interests – big data, machine learning, distributed computing, verification.

Игорь Валерьевич НИКИФОРОВ – доцент Высшей школы программной инженерии. В 2014 году защитил диссертацию на соискание ученой степени кандидата технических наук. Область научных интересов – большие данные, машинное обучение, распределенные вычисления, верификация.

Sergey Mikhailovich USTINOV is a Professor of the Higher School of Software Engineering. In 1977 he defended his dissertation for the degree of candidate of technical sciences, and in 1989 – doctor of technical sciences. Research interests - modeling of dynamic systems, numerical methods, mathematical and applied software, control theory, rigid differential systems, stability of modes of large power interconnections.

Сергей Михайлович УСТИНОВ – профессор Высшей школы программной инженерии. В 1977 году защитил диссертацию на соискание ученой степени кандидата технических наук, а в 1989 году – доктора технических наук. Область научных интересов – моделирование динамических систем, численные методы, математическое и прикладное программное обеспечение, теория управления, жесткие дифференциальные системы, устойчивость режимов больших энергообъединений.

DOI: 10.15514/ISPRAS-2022-34(2)-11



Wi-Fi sensing Human Detection with Kolmogorov-Wiener Filter and Gated Recurrent Neural Networks

*P.P. Shibaev, ORCID: 0000-0001-5497-8483 <pshibaev@lvk.cs.msu.ru>
A.A. Chupakhin, ORCID: 0000-0002-3021-1648 <andrewchup@lvk.cs.msu.ru>
Lomonosov Moscow State University,
GSP-1, Leninskie Gory, Moscow, 119991, Russia*

Abstract. Using Received Signal Strength Indicator (RSSI) values to detect human presence is a well-known Wi-Fi sensing technique. In this paper, an overview of existing algorithms solving the problem is considered. Two new techniques based on the discrete Kolmogorov-Wiener filter and the gated recurrent unit neural network are proposed. Human detection experiment results are presented along with algorithms' accuracy analysis.

Key words: Wi-Fi-sensing; human detection; Kolmogorov-Wiener filter; gated recurrent neural network

For citation: Shibaev P.P., Chupakhin A.A. Wi-Fi Sensing Human Detection with Kolmogorov-Wiener Filter and Gated Recurrent Neural Networks. Trudy ISP RAN/Proc. ISP RAS, vol. 34, issue 2, 2022, pp. 135-144. DOI: 10.15514/ISPRAS-2022-34(2)-11

Acknowledgments. This work was partially financially supported by non-profit foundation for the Development of Science and Education «Intellect». Special thanks to Daniil Sliusar and the company "Wimark Systems" LLC for providing network equipment for conducting an experimental study.

Определение присутствия человека с помощью фильтра Колмогорова-Винера и нейронных сетей с управляемыми рекуррентными блоками методом Wi-Fi-сканирования

*П.П. Шубаев, ORCID: 0000-0001-5497-8483 <pshibaev@lvk.cs.msu.ru>
А.А. Чупахин, ORCID: 0000-0002-3021-1648 <andrewchup@lvk.cs.msu.ru>
Московский государственный университет им. М.В. Ломоносова,
119991, Россия, Москва, Ленинские горы, д. 1*

Аннотация. Использование значений индикатора уровня принятого сигнала (RSSI) для обнаружения присутствия человека является хорошо известным методом определения Wi-Fi. В данной статье рассматривается обзор существующих алгоритмов, решающих эту задачу. Предложены два новых метода, основанных на дискретном фильтре Колмогорова-Винера и нейронной сети с управляемыми рекуррентными нейронами. Представлены результаты эксперимента по обнаружению человека вместе с анализом точности алгоритмов.

Ключевые слова: Wi-Fi-сканирование; фильтр Колмогорова-Винера; GRU нейронные сети

Для цитирования: Шубаев П.П., Чупахин А.А. Определение присутствия человека с помощью фильтра Колмогорова-Винера и нейронных сетей с управляемыми рекуррентными блоками методом Wi-Fi-сканирования. Труды ИСП РАН, том 34, вып. 2, 2022 г., стр. 135-144. DOI: 10.15514/ISPRAS-2022-34(2)-11

Благодарности. Исследование выполнено при финансовой поддержке Некоммерческого Фонда развития науки и образования «Интеллект». Отдельная благодарность Даниилу Слюсарю и компании

ООО "Ваймарк Системс" за предоставленное сетевое оборудования для проведения экспериментального исследования.

1. Introduction

Wi-Fi sensing is based on the use of Wi-Fi devices to recognize people's activity by identifying deviations and features in various characteristics of the Wi-Fi signal. The technology is especially relevant on the eve of the new Wi-Fi standard, which is focused on the active development of Wi-Fi networks for the Internet of Things. Wi-Fi sensing technologies can be used in the field of transport, household security, healthcare and other areas of life.

With the development of Wi-Fi technology, in particular, with the improvement of the hardware of access points, it became possible to use Wi-Fi access points as locators. The simplest and most common method of determining the presence of a person near a Wi-Fi access point is based on a detection of changes in the RSSI. Various algorithms have been already proposed to solve the problem of detection changes in the RSSI, but not all of them work with acceptable accuracy.

The relevance of this work is due to the active development of Wi-Fi technologies as a part of the movement towards the latest standard, 802.11 (Wi-Fi-7) [1]. Currently, the Wireless Broadband Alliance has developed a document [2] reflecting the current state and problems of Wi-Fi scanning technologies.

2. Problem Statement

The goal is to develop an algorithm for determining the presence of a person by detecting changes in RSSI values (an indicator of the level of the received signal) received from a Wi-Fi access point. There is a vector of RSSI $r_{train} = [r_1, r_2, \dots, r_n]$ with values collected at regular intervals, and a vector of labels $y_{train} = [y_1, y_2, \dots, y_n]$. The value 1 of the label corresponds to the presence of a person, 0 – absence. It's required to find an algorithm $A(r)$, which should be trained on the data (optionally, this data is cleaned from the noise). The algorithm, receiving new unlabeled data r_{input} tries to determine the presence of a person. The result is a vector y_{out} . It's required to maximize the accuracy score, which is the percentage of samples that have all their labels classified correctly, as well as minimize the number of Type I and Type II errors.

3. Related Works

The following criteria were used for survey of the algorithms:

- algorithm class (for example, a statistical filter or a machine learning algorithm);
- presence of a special noise elimination method in the algorithm/combination of the considered algorithm;
- accuracy of the considered algorithm;
- software and hardware equipment required for the experiment;
- type and dimensions of the room in which the experiment was performed.

3.1 Common Approaches

There are two approaches to Wi-Fi sensing. The first is based on the RSSI, and the second is based on information about the state of the communication channel – Channel State Information (CSI). Both approaches are based on a similar principle. When a person passes between Wi-Fi devices, the RSSI or CSI changes. The RSSI level is a physical quantity that characterizes the full power of the signal received by the Wi-Fi device [3]. The measurement takes place on a logarithmic scale in dBm (decibels relative to milliwatt). This indicator can be obtained from almost all devices that work with Wi-Fi. Each netcard manufacturer makes its own formula for determining RSSI. The general formula is usually written as follows:

$$P_d = P_0 - 10n \log \left(\frac{d}{d_0} \right),$$

where: d is the distance to the signal source in meters, d_0 is the distance in meters from the signal source to the point where the RSSI measurement takes place, n is the coefficient for the middle point in which the measurement takes place, which is a dimensionless quantity, calculated empirically, P_d is the desired RSSI indicator.

A very significant physical factor that makes it possible to detect a person using RSSI is the absorption coefficient of the human body. It is close to the coefficient of water absorption (due to the fact that the human body is mainly composed of water) and differs from the coefficient of, for example, walls and furniture [3]. For reference, an approximate RSSI absorption values are presented in the Table 1.

Table 1. RSSI Absorption Values for Materials

Material	Absorption, dBm
Plasterboard	3–5
Glass door, metal frame	6
Metal door	6-10
Glass window	3
Block wall	6-15
Concrete wall	4-6

For comparison, the absorption value for the human body is approximately 9-30 dBm [4], which approximately corresponds to the absorption value for water 15-20 dBm [5].

The essence of the CSI-based approach is also worth mentioning. To establish a Wi-Fi connection, MIMO technology (Multiple Input Multiple Output) is often used. The essence of MIMO approach lies in increasing the bandwidth of the communication channel due to data transmission via a system of several antennas. In this case, one can set a matrix to describe the connection state in the case when the transmitting device has a composite antenna of M cells, and the receiving device has N cells. When transmitting data, the j -th cell of the transmitter transmits to the i -th cell of the receiver ($k \in 1, \dots, M; j \in 1, \dots, N$). The element of the CSI matrix $H = [h_{kj}] = [a_{kj} + b_{kj}i]$ characterizes the state of data transmission from the j -th cell of the transmitter to the k -th cell of the receiver.

3.2 RSSI-based Approaches

To solve the problem of determining the presence of a person according to RSSI data, statistical algorithms are often used. For example, the Kalman filter with subsequent [6] estimation of variance allows one to achieve accuracy of 95%. In [6] an experiment was conducted in small room 3m x 3m using three Wireless Sensor Network (WSN) nodes. A hybrid statistical approach based on a combination of expectation and variance estimates without noise control was used [7]. The experiment was conducted in the halls of the university. The achieved accuracy exceeded 90%. With the help of special Zigbee modules (used for low-consumption wireless communication) and an algorithm based on moving averages [8], it was possible to achieve 100% accuracy in laboratory conditions. The probabilistic algorithm based on MRF (Markov Random Fields) [9] achieved an accuracy of 86% in a large room with an area of 150 m². The study used a Chipcon 1100 wireless communication board.

It is also possible to use machine learning algorithms. The approach based on the complicated K-Means method allowed to achieve an accuracy of 94%. The experiment was carried out in a room of 18m x 18m using a TelosB wireless communication board.

3.3 Survey Conclusion

Thus, the most frequent approach to solving the problem under consideration is based on the use of various statistical characteristics of the filtered RSSI time series. The best results in this category were shown by an algorithm based on the Kalman filter and an algorithm based on a combination of filters using moving averages. However, there is also an alternative group of approaches based on machine learning algorithms, which is represented in the survey by an algorithm based on *K*-Means. Table 2 presents a comparison of the considered methods according to the selected survey criteria.

Table 2. Algorithms Comparison - in the Living Room

Publication	Algorithm type	Noise Filtering	Accuracy	Hardware	Venue
Wang, H et al. [6], 2020	Kalman filter	Kalman Filter	0.95	3 WSN Nodes	3m × 3m room
Sigg, S et al. [7], 2014	Mean and variance estimate	No	≥0.9	Nexus One smartphone	University hall
Sntiprapan S. et al. [8], 2021	Weighted MA	Built-in	1	Zigbee	Laboratory room
Xu Ch. et al. [9], 2013	MRF	Built-in	0.86	3 Chipcon CC100	Office hall
Yuan Y. et al. [10], 2013	K-Means	Built-in	0.94	3 Telos B	18m × 18m room

It is proposed to consider new approaches from both categories, when choosing the algorithm to solve the considered problem. The Kolmogorov-Wiener filter is presented as a statistical algorithm, and a gated recurrent units (GRU) neural network is presented as a machine learning algorithm.

4. Proposed Algorithms

4.1 Kolmogorov-Wiener Filter

First, it is proposed to consider the Kolmogorov-Wiener filter, which is a statistical approach. The filter is a simpler analogue of the Kalman filter, successfully used to denoise signals.

A discrete version of the Kolmogorov-Wiener filter [11] is considered. The filter works in the following way:

- 1) The measured discrete signal $w[n]$ is fed to the filter input. There is also an unknown useful signal $s[n]$.
- 2) The output signal $x[n] = \sum_{i=0}^N a_i w[n - i]$ is to be obtained, where N is the number of past points (called the filter order), are the filter coefficients a_i , where i is from 0 to N .
- 3) The filter coefficients are obtained by minimizing the mean squared error: $e[n] = x[n] - s[n]$, $a_i = \operatorname{argmin} E[e^2[n]]$.

To detect anomalies corresponding to the presence of a person (assuming that the noise removal was successful), the approach proposed by Frank Hampel [12] is used. The essence of the approach is as follows. Let's consider a data set x_i and a sample $X_N = \{X_{(k)}\}$, $k \in 1, \dots, N - k$ -th order statistics.

As an estimate of the median is often recommended to use $\operatorname{median} \left(X_n = 0.5 \left(X_{\lfloor \frac{N}{2} \rfloor + 1} + X_{\lfloor \frac{N+1}{2} \rfloor} \right) \right)$.

As an estimate of the Mean Absolute Deviation (MAD): $\operatorname{MAD}(X_n) = \operatorname{median}(|x_1 - \operatorname{median}(X_n)|, \dots, |x_n - \operatorname{median}(X_n)|)$. The value of x should be considered an outlier if: $|x - \operatorname{median}(X_n)| \geq g(N, a_N) \operatorname{MAD}(X_N)$, where $g(N, a_N)$ is some function of the sample size N and some parameter a_N .

Thus, based on the Kolmogorov-Wiener filter with an auxiliary approach, which is often called the Hampel filter, it is possible to construct an algorithm for solving the considered problem of human presence detection.

4.2 GRU Neural Network

Recurrent Neural Networks (RNN) are successfully used for processing time series [13]. In particular, networks with Long short-term memory (LSTM) performed well on RSSI data. An alternative to the LSTM approach is architectures based on gated recurrent units (GRU). Although on many tasks GRU and LSTM networks exhibit similar results in signal processing, GRU networks are considered more computationally efficient.

4.3 Decision Tree Ensembles

Gradient boosting classifiers can be used for time series forecasting [14]. These algorithms are based on an ensemble of decision trees. In such a way a strong classifier is built out of weak classifiers. AdaBoosting and Gradient Boosting Machine algorithms are considered in this work.

Random forest algorithm is an ensemble algorithm which consists of the combination of ordinary decision trees. It can also be used for time series forecasting [15].

5. Experimental Stand Design

In practice, the goal is to determine human presence via a system of two Wi-Fi devices (a receiver device – a smartphone, and a Wi-Fi access point). Presence is determined based on RSSI metrics. As an additional condition, absence of significant environmental disturbances (movement of dimensional objects, sudden changes in humidity and temperature) is accepted.

The experiment was conducted using a stand consisting of two Wi-Fi devices: a Samsung A7 2018 mobile phone and a TP-Link TL-MR3020 access point with the Open WRT operating system 07/19/19 firmware. A bash script was written that ran on a Wi-Fi access point and periodically sent RSSI values to the laptop. The experiments were carried out in two rooms. The distance in the living room between the devices is 3 m, and in the office space – 5 m. In both cases, the devices were at a height of about 1.5 m.

6. Implemented Algorithms

In general, pipeline is the following: preprocess/denoise data, apply a prediction algorithm.

6.1 Kolmogorov–Wiener Filter

Kolmogorov-Wiener Filter is implemented in Python3 as a separate class. During the fit phase, RSSI data measured in the human absence should be fed to the algorithm. In such a way, noise level is determined. In the transform phase, any RSSI data is passed to be denoised. After that, the denoised RSSI values outliers are detected via the Hampel filter.

6.2 Data Preprocessing

No additional data preprocessing was done for the Kolmogorov-Wiener filter. However, data was transformed before being fed into the tree-based algorithms and the GRU neural network. Sliding window technique was used. Let the window size be s for raw RSSI values r_1, \dots, r_n . The matrix which can be represented as an array of rows $M = [[r_1, \dots, r_w], [r_2, \dots, r_{w+1}], \dots, [r_{n-w}, \dots, r_n]]$ is composed to be fed into these algorithms.

6.3 GRU Neural Network

The GRU neural network has the following architecture. A GRU layer with 200 units takes M as an input. Hyperbolic tangent (\tanh) is used as an activation function. This layer is followed by two dense layers. The first one has 50 units and $relu$ (rectified linear unit) as an activation function, the second one has one single unit and linear activation. Adam optimizer with a mean squared error loss is used. It takes approximately 36-40 epochs to train such a network. After that, the model becomes overfitted.

7. Experimental research

7.1 Goals

It is required to check the following basic provisions:

- 1) The accuracy of the considered and the implemented algorithms;
- 2) Estimate the reliability of the assembled stand (failure-proof work during the experiment is guaranteed).

7.2 Methodology

Two experiments were conducted. One is Kolmogorov-Wiener filter-specific, the other one is general for all the considered algorithms. In the first case, the person was absent from the room for 200s, and RSSI data corresponding to human absence was collected. After that, the following actions were performed:

- 1) For the first 50 seconds, a person crosses the line between the devices, and is also on the side;
- 2) In the next 150 seconds, the person is removed from the room.

RSSI values are requested every second. This experiment is intended to show the Kolmogorov-Wiener filter in action.

In the second case, for the 2100 seconds, a human leaves the room every five minutes. RSSI values are also requested every second. The goal of this experiment is to collect a dataset to fit and score the considered algorithms. Experiments are conducted in two different rooms: a living room and an office space.

This dataset (matrix M) is divided row-wise into two parts: M_{train} (71.5%) and M_{test} (28.5%). The dataset was split in such a way in order not to split the “active” phase, when a human is moving (the whole phase fits into the test part). Also, it is worth mentioning that cross-validation for hyperparameters tuning was done for tree-based algorithms. Accuracy score, which is the ratio of the number of correct predictions to the number of total predictions, is the key performance metric.

7.3 Experiment results

RSSI raw data and denoised data cleaned with the Kolmogorov-Wiener filter is presented in the Fig. 1 and Fig. 2.

Accuracy, true positive rate (TPR) and true negative rate (TNR) for all algorithms is estimated on the test part. The accuracy score results for Kolmogorov-Wiener filter, GRU neural network and various ensembles of decision trees are presented in the Table 2 and the Table 3. Also it was attempted to train algorithms in the office room and test them in the living room and vice versa. These results for these experiments are presented in the Table 3 and Table 4.

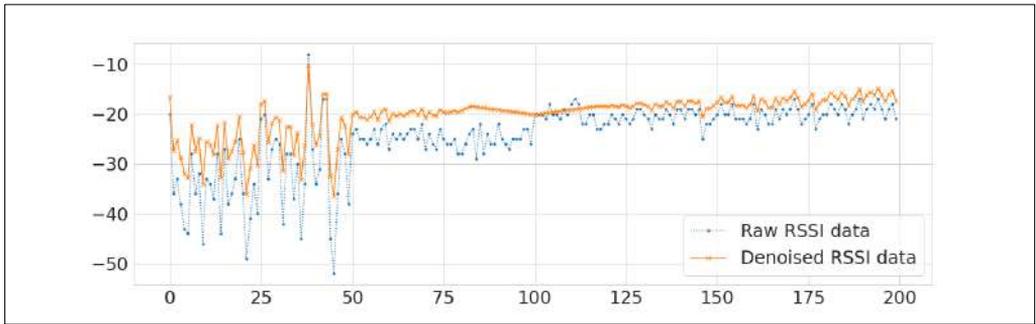


Fig 1. RSSI in the living room

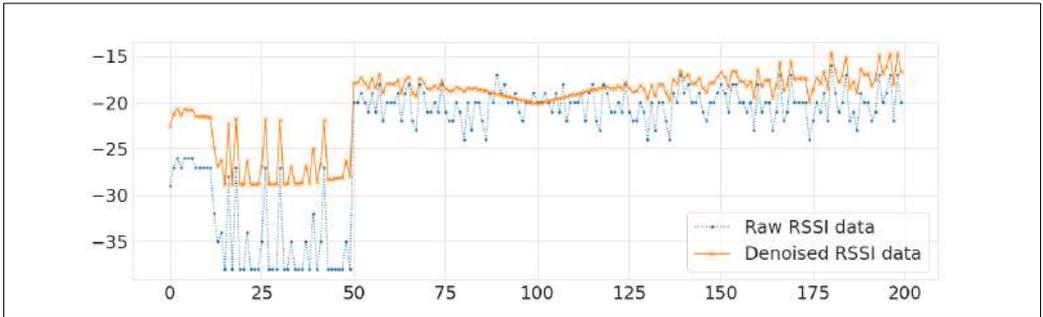


Fig 2. RSSI in the office room

Table 3. Algorithms Comparison - both train and test in the Living Room

Algorithm	Accuracy Score	TPR	TNR
Gradient Boosting	0.99	0.99	0.997
Random Forest	0.99	0.99	0.994
Ada Boost	0.98	0.98	0.997
Wiener Filter	0.90	1.00	0.83
GRU NN	0.98	0.97	0.99

Table 4. Algorithms Comparison - both train and test in the Office Space

Algorithm	Accuracy Score	TPR	TNR
Gradient Boosting	0.99	0.98	0.99
Random Forest	0.98	0.98	0.99
Ada Boost	0.97	0.95	0.99
Wiener Filter	0.94	1.0	0.90
GRU NN	0.97	0.97	0.98

Table 5. Algorithms Comparison — train in the Living Room, test in the Office Space

Algorithm	Accuracy Score	TPR	TNR
Gradient Boosting	0.985	0.99	0.97
Random Forest	0.99	0.99	0.985
Ada Boost	0.98	0.99	0.96
Wiener Filter	0.75	0.97	0.46
GRU NN	0.86	0.83	0.96

Table 6. Algorithms Comparison — train in the Office space, test in the Living room

Algorithm	Accuracy Score	TPR	TNR
Gradient Boosting	0.88	0.82	0.96
Random Forest	0.76	0.99	0.44
Ada Boost	0.76	0.94	0.44
Wiener Filter	0.72	0.94	0.44
GRU NN	0.85	0.81	0.96

8. Conclusion and Future Work

Thus, machine learning algorithms are more accurate than the algorithm based on the Kolmogorov-Wiener filter, which, in addition, requires determining the noise level in the room. Machine learning algorithms are efficient even in the presence of noise. Code of the implemented algorithms and results of experimental studies are available in [16].

Wi-Fi sensing is a very promising area of research, as Wi-Fi hardware is constantly improving. Possible directions of the future work are as follows:

- 1) Develop new algorithms for CSI (channel state information) data;
- 2) Develop an algorithm to solve human detection problem based on both RSSI and CSI values for multiple Wi-Fi access points;
- 3) Develop techniques to integrate WiFi-sensing into existing IoT solutions.

References

- [1] E. Khorov, I. Levitsky, and I. F. Akyildiz. Current Status and Directions of IEEE 802.11be, the Future Wi-Fi 7. *IEEE Access*, vol. 8, 2020, pp. 88664-88688.
- [2] Wi-Fi Sensing, Wireless Broadband Alliance. Available: <https://wballiance.com/resource/wi-fi-sensing/>, accessed: 07-Apr-2022.
- [3] D.D. Coleman and D.A. Westcott. CWNA: Certified Wireless Network Administrator Study Guide: Exam CWNA-107. Sybex, 2018, 1024 p.
- [4] M.A.A. Karunaratna and I.J. Dayawansa. Energy absorption by the human body from RF and microwave emissions in Sri Lanka. *Sri Lankan Journal of Physics*, vol. 7, 2006, pp. 35-47.
- [5] U.M. Qureshi, F.K. Shaikh et al. RF path and absorption loss estimation for underwater wireless sensor networks in different water environments. *Sensors (Basel)*, vol. 16, issue 6, 2016, article no. 890, 15 p.

- [6] H. Wang, F. Zhang, and W. Zhang. Human Detection through RSSI Processing with Packet Dropout in Wireless Sensor Network. *Journal of Sensors*, vol. 2020, article ID 4758103, 9 p.
- [7] S. Sigg, U. Blanke, and G. Troster. The telepathic phone: Frictionless activity recognition from WiFi- RSSI. In *Proc. of the IEEE International Conference on Pervasive Computing and Communications (PerCom)*, 2014, pp. 148-155.
- [8] P. Santiprapan, K. Sengchuai et al. Development of an adaptive device-free human detection system for residential lighting load control. *Computers & Electrical Engineering*, vol. 93, 2021, article ID 107233.
- [9] C. Xu, B. Firner et al. SCPL: Indoor device-free multi-subject counting and localization using radio signal strength. In *Proc. of the 12th International Conference on Information Processing in Sensor Networks (IPSN'13)*, 2013, pp. 79-90.
- [10] Y. Yuan, J. Zhao et al. Estimating crowd density in an RF-based dynamic environment. *IEEE Sensors Journal*, vol. 13, issue 10, 2013, pp. 3837–3845.
- [11] C. Schleicher. Kolmogorov-Wiener filters for finite time series. *SSRN Electronic Journal*, 2004, 31 p.
- [12] F.R. Hampel. The influence curve and its role in robust estimation. *Journal of the American Statistical Association*, vol. 69, no. 346, 1974, pp. 383-393.
- [13] F. A. Gers, D. Eck, and J. Schmidhuber. Applying LSTM to time series predictable through time-window approaches. *Lecture Notes in Computer Science*, vol. 2130, 2002, pp. 669–676.
- [14] T. Cinto, A.L. Gradwohl et al. Solar flare forecasting using time series and extreme gradient boosting ensembles. *Solar Physics*, vol. 295, issue 7, 2020, article no. 93.
- [15] M.J. Kane, N. Price et al. Comparison of Arima and random forest time series models for prediction of avian influenza H5N1 outbreaks. *BMC Bioinformatics*, vol. 15, 2014, article no. 272.
- [16] P. Shibaev. Kolmogorov-Wiener filter implementation, GRU neural network, experimental data. Available: <https://github.com/shibaevff/RSSISensing>, accessed 12.06.2022

Информация об авторах / Information about authors

Павел Павлович ШИБАЕВ – студент кафедры АСВК факультета ВМК МГУ. Научные интересы: распределённые системы.

Pavel Pavlovich SHIBAEV – Bachelor student at the department of Computer Systems and Automation of the Faculty of Computational Mathematics and Cybernetics. Research interests: distributed systems.

Андрей Андреевич ЧУПАХИН – аспирант, математик кафедры АСВК факультета ВМК МГУ. Научные интересы: распределённые системы, интеллектуальный анализ сетевого трафика. and Cybernetics. Research interests: distributed systems, intelligent analysis of network traffic.

Andrey Andreevich CHUPAKHIN – PhD student, mathematician at the department of Computer Systems and Automation of the Faculty of Computational Mathematics and Cybernetics. Research interests: distributed systems, intelligent analysis of network traffic.



Автоматизация процесса экологического мониторинга в особо охраняемых природных территориях Российской Федерации на базе системы управления контентом Directum RX

¹И.Д. Рудько, ORCID: 0000-0003-1068-9166 <ivnrudko@gmail.com>

²Л.Л. Хорошко, ORCID: 0000-0002-7487-8997 <khoroshko@mati.ru>

²П.П. Кейно, ORCID: 0000-0002-1033-0260 <pavel@mati.ru>

¹ ООО "НЛМК-Информационные Технологии",

398040, Россия, г. Липецк, пл. Metallургов, д. 2, помещ. 201

² Московский авиационный институт (национальный исследовательский университет),
125310, Россия, Москва, Волоколамское ш., 4

Аннотация. Для организаций, которые осуществляют контроль особо охраняемых природных территорий Российской Федерации, актуальна задача по консолидации данных о проводимых наблюдениях. Эти данные, называемые летописями природы, долгое время велись в упрощенной, бумажной форме и не имели четкой структуры. Задача автоматизации бизнес-процессов по сбору данных и обменом этими данными между участниками научного сообщества, а также построение аналитических моделей, необходимых для научных подразделений парков и заповедников, является важной и актуальной. В связи с этим актуальна автоматизация процесса экологического мониторинга путём разработки модуля электронного документооборота на базе интеграционной платформы для бизнес-решений Directum RX. Цель исследования заключается в автоматизации процесса экомониторинга на базе платформы Directum RX, которая позволяет построить систему управления корпоративным контентом, а также создать полноценную систему хранения и поиска данных по сбору информации о наблюдениях. В статье описана ролевая модель для работы с системой, архитектура системы и разработанные компоненты модуля «Экомониторинг» на базе платформы Directum RX. Для этого был использован структурный метод путем разбиения задачи на множество независимых этапов, доступных для понимания и решения с помощью иерархического упорядочивания. Разработана интеграция с сервисом Яндекс погода с целью дальнейшего использования в аналитических моделях. Разработано решение по управлению универсальным классификатором животных в соответствие с принятом в мире классификатором биологической систематики. Продемонстрирована автоматизация действий пользователей по сбору и обработке информации о наблюдениях. В результате работы был автоматизирован процесс экологического мониторинга в одном из заповедников Российской Федерации.

Ключевые слова: автоматизация экологического мониторинга; Directum RX; автоматизация бизнес-процессов; архитектура клиент-серверного приложения; разработка модуля электронного документооборота Directum RX

Для цитирования: Рудько И.Д., Хорошко Л.Л., Кейно П.П. Автоматизация процесса экологического мониторинга в особо охраняемых природных территориях российской Федерации на базе системы управления контентом Directum RX. Труды ИСП РАН, том 34, вып. 2, 2022 г., стр. 145-158. DOI: 10.15514/ISPRAS-2022-34(2)-12

Благодарности: Выражаем особую благодарность фонду АНО «Дальневосточные леопарды» за возможность принять участие в амбициозном и перспективном проекте, по результатам которого была написана данная статья.

Automation of the process of environmental monitoring in specially protected natural territories of the Russian Federation based on the content management system Directum RX

¹I.D. Rudko ORCID: 0000-0003-1068-9166 <ivnrudko@gmail.com>
²L.L. Khoroshko ORCID: 0000-0002-7487-8997 <khoroshko@mati.ru>
²P.P. Keyno ORCID: 0000-0002-1033-0260 <pavel@mati.ru>

¹NLMK-Information Technologies,
2, Metallurgov sq., Lipetsk, Russia, 398040
²Moscow Aviation Institute (National Research University)
4, Volokolamskoe sh., Moscow, Russia, 4125310

Abstract. For organizations that execute control over specially protected natural areas of the Russian Federation, the task of consolidating data on ongoing observations is relevant. These data, called the chronicles of nature, for a long time were kept in a simplified, paper form and did not have a clear structure. The task of automating business processes for collecting data and exchanging this data between members of the scientific community, as well as building models necessary for the scientific departments of parks and reserves, is important and relevant. In this connection, we see it relevant to consider automating the process of environmental monitoring by developing an electronic document management module based on the integration platform Directum RX business solutions. The purpose of the research is to automate the eco-monitoring process based on the Directum RX platform, which allows to build a corporate content management system, as well as create a full-fledged data storage and retrieval system for collecting information about observations. The article describes the role model for working with the system, the system architecture and the developed components of the "Ecomonitoring" module based on the Directum RX platform. For this, a structural method was used, by dividing the task into many independent stages available for understanding and solving and hierarchical ordering. Integration with the Yandex weather service has been developed for further use in analytical models. A solution has been developed to manage the universal classifier of animals in accordance with the classifier of biological taxonomy accepted in the world. The automation of user actions for collecting and processing information about observations is demonstrated. As a result, the process of environmental monitoring was automated in one of the reserves of the Russian Federation.

Keywords: environmental monitoring automation; Directum RX; business process automation; client-server application architecture; Directum RX electronic document management module development

For citation: Rudko I.D., Khoroshko L.L., Keyno P.P. Automation of the process of environmental monitoring in specially protected natural territories of the Russian Federation based on the content management system Directum RX. *Trudy ISP RAN/Proc. ISP RAS*, vol. 34, issue 2, 2022, pp. 145-158 (in Russian). DOI: 10.15514/ISPRAS-2022-34(2)-12

Acknowledgements. We express a special gratitude to the ANO Far Eastern Leopards Foundation for the opportunity to take part in an ambitious and promising project, the results of which this article was written.

1. Введение

Разработка средств автоматизации мониторинга особо охраняемых природных территорий (ООПТ) велась в РФ продолжительное время, но материалы и база данных экологических наблюдений являются разрозненными и неструктурированными. Разработка современной платформы сбора, аналитики данных и скорейшая интеграция накопленной информации ООПТ в глобальную систему, а также последующая возможность преобразования результатов исследований, предоставит возможность российской науке выйти на новый уровень по обработке данных и подготовке прогнозных моделей.

Автоматизация сложных бизнес-процессов на базе системы управления контентом Directum RX рассматривалась во многих статьях [1-8]. Решение подобного рода задач начинается со сбора требований у сотрудников заповедника для дальнейшего технического обзора этих требований и описания облика будущей системы. Система будет являться инструментом для

работы сотрудников заповедников и должна включать набор стандартных и уникальных модулей по для автоматизации процесса экологического мониторинга.

Основные цели разработки и внедрения системы:

- фиксация данных наблюдений сотрудниками ООПТ;
- контроль за действиями персонала заповедника;
- перевод данных по наблюдения в электронную форму с быстрым доступом сотрудников ООПТ к этим данным;
- внесение оперативных сведений о состоянии объектов экологического мониторинга;
- централизованный учет и аналитическая отчетность по биологическому мониторингу ООПТ (структура данных привязана к существующему административному устройству ООПТ);
- сбор фото и видео фиксаций в единую базу данных;
- управление регламентами и хранение документов в единой базе данных;
- возможность получения наглядных отчетов.

2. Ролевая модель будущей системы

Ролевая модуль будущей системы построена при помощи разделения ответственного за экологический мониторинг персонала на функциональные группы и разделением обязанностей по работе с системой [9]. Матрица на рис. 1 была составлена при помощи сбора требований у пользователей с учётом будущего масштабирования на другие ООПТ. В заголовках столбцов можно увидеть этапы, а в строках функциональные группы сотрудников. На пересечении, соответственно, находится развернутое описание действий необходимых для каждого этапа проекта от ответственной группы сотрудников [10].

	Разработка методологии	Разработка/добавление типов данных	Разработка процессов сбора данных	Планирование сбора данных	Выполнение сбора данных	Обработка полученных данных
Ученые, научные сотрудники	Подготовка документации и описание методологии сбора данных		Подготовка документации и описание методологии сбора данных	Разработка подходов для выбора маршрутов для проведения экологического мониторинга		
Специалисты АНО «Дальневосточные леопарды»	Адаптация методологии для специфики ООПТ	Рекомендации для новых данных		Подготовка регламентов по сбору данных экологического мониторинга	Контроль управление и оперативная аналитика в Directum RX	Аналитическая обработка данных в Directum RX, экспорт данных, многофакторный анализ
Штатные специалисты ООПТ	Внесение комментариев для расширения функционала сотрудниками		Рекомендации к методикам сбора данных	Добавление маршрутов для проведения мониторинга в Directum RX	Проведение экологического мониторинга, внесение данных в Directum RX	
Разработчики Directum RX		Разработка справочников в Directum RX	Разработка типовых маршрутов в Directum RX	Разработка алгоритмов хранения данных в Directum RX	Разработка интеграционных потоков данных экологического мониторинга в Directum RX	Разработка аналитических отчетов и кубов в Directum RX
Волонтеры / Операторы обработки и ввода					Сбор данных и отправка по цифровым каналам в Directum RX	Ввод данных в Directum RX по инструкции

Рис. 1. Ролевая модель будущей системы

Fig. 1. Role model of the future system

Научные сотрудники осуществляют контроль за текущими методологиями экологического мониторинга, которые будут автоматизированы при помощи системы. Они также отвечают за подготовку документации и методологии сбора данных. Специалисты фонда АНО «Дальневосточные леопарды» адаптируют методологию под задачи конкретного заповедника, подготавливают регламенты по сбору данных экологического мониторинга, осуществляют контроль и управление оперативной аналитикой в системе. Штатные специалисты ООПТ вносят свои комментарии к методологиям и подходам к автоматизации

процесса экомониторинга с учётом особенностей конкретной ООПТ, разрабатывают маршруты для сбора данных экологического мониторинга и занимаются внесением и контролем за данными уже в самой системе. Разработчики системы ответственны за разработку модулей СЭД, построение аналитических отчётов и мониторингом внутреннего состояния системы. Волонтёры, операторы обработки и ввода осуществляют сбор и отправку данных по цифровым каналам в систему [11, 12].

3. Архитектура системы

Архитектура решения для автоматизации процесса экологического мониторинга была построена на базе серверов российского сервиса Яндекс облако. Это обеспечивает импортозамещение, что является очень актуальным в текущих условиях. На рис. 2 представлено описание текущей построенной архитектуры системы. Серыми выделены компоненты, которые используются для обеспечения работоспособности рабочей среды системы. Зелёным цветом выделены элементы, используемые в работе средой для разработки. Синим цветом выделены элементы, которые могут быть развернуты в дальнейшем для улучшения работы и отказоустойчивости системы.

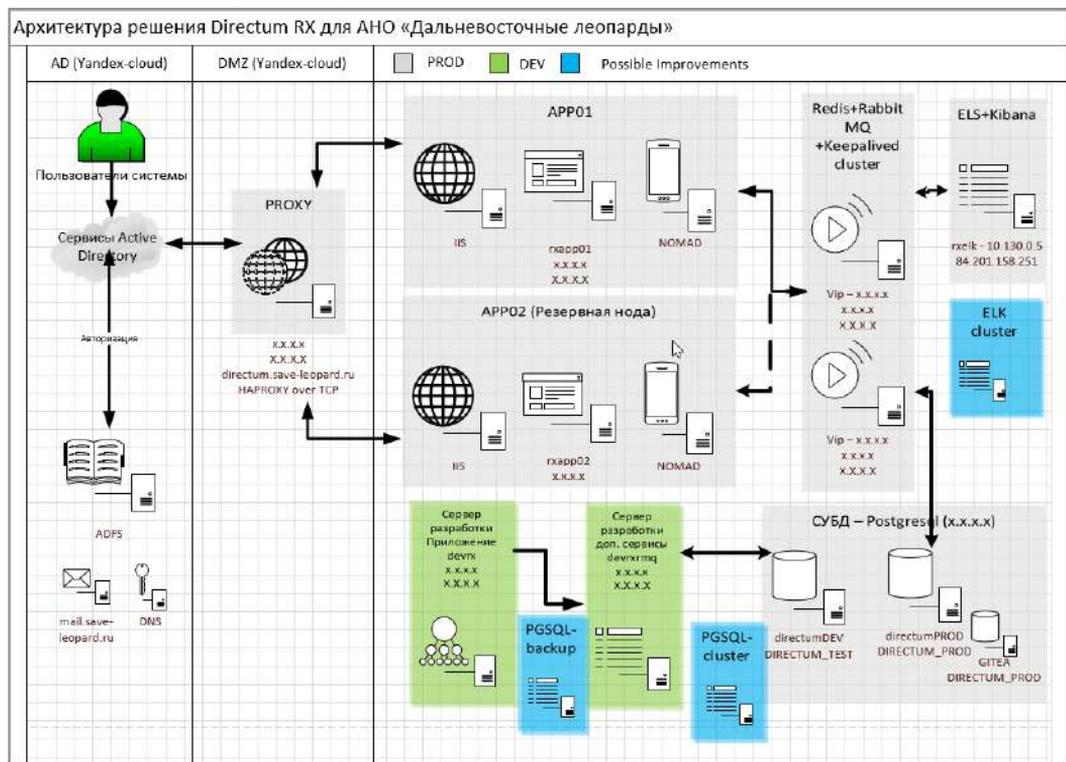


Рис. 2. Архитектура системы

Fig. 2. System architecture

Directum RX позволяет использовать два основных подхода к управлению пользователями системы. Пользователи могут храниться локально в базе данных системы Directum RX или же может быть использовано внешние хранилище учётных записей, такое как LDAP-каталог или Active Directory. Поддерживается авторизация пользователей с использования служб федераций (ADFS). Также система может использовать LDAP-каталоги, разработанные на базе open source решений (например, FreeIPA).

Следующим уровнем защиты является демилитаризованная зона. В демилитаризованной зоне размещается прокси сервер, т. е. сервер, который осуществляет перенаправление трафика на сервера приложений Directum RX. Данный слой необходим для защиты приложения от несанкционированного доступа, а также предотвращает эксплуатацию уязвимостей периметра. На прокси сервере также настроен межсетевой экран, который позволяет гибко настраивать сетевой доступ к элементам приложения.

Далее идёт уровень серверов приложений Directum RX, который включает в себя несколько слоёв. Сервер приложений, который может быть организован, как и на платформе Microsoft Server, так и на платформе Open Source Linux. Использование Фреймворка DOT.net core, который является кроссплатформенным, позволяет организовать работу сервера приложений на любой доступной операционной системе.

Для обеспечения отказоустойчивости работы системы, установки обновлений на операционные системы и на саму систему Directum RX организован кластер серверов приложений [13], который может функционировать, как и в режиме Active-Passive, так и в режиме Active-Active. Поддержка кластеризации осуществляется на уровне операционной системы сервера приложения.

Следующий уровень – сочетания стеков технологий Redis+RabbitMQ. Это система управления очередями, которая используется для работы системой Directum RX. Стек технологий ELS+Kibana используется для индексации информации, которая отображает внутреннее состояние системы. Файловые хранилища – это базы данных системы Directum RX, которые являются наиболее эффективным для хранения документов. Каждый документ в системе реализуется посредством GUID объекта на уровне файловой системы. Directum RX поддерживает управление ссылками на объекты файловых хранилищ. Все метаданные и данные справочников хранятся в базе данных Directum RX, которая является также кроссплатформенной.

Сервер разработки является копией набора серверных компонент системы Directum RX. На сервере разработки развернуты сервер приложения, сервер очередей RabbitMQ, non-SQL СУБД Redis, ELK и Kibana.

В качестве сервера БД в проекте используется система PostgreSQL, на которой размещены продуктивная база данных, база данных системы разработки, а также базы данных системы управления версиями GITEA. Для лучшего управления данными каждая база данных системы расположена в отдельном табличном пространстве, что позволяет гибко управлять быстродействием системы расположив, например, продуктивную систему на быстрых дисках, а системы тестирования и разработки на более медленных, что и было реализовано в рамках проекта.

Для повышения отказоустойчивости СУБД в будущем можно внедрить кластерное решение, которое позволит резервировать вычислительные ноды PostgreSQL: PGSQL-cluster (Active-Active) и PGSQL-backup (Active-Passive). Это позволит повысить отказоустойчивость системы.

4. Разработанные элементы модуля для автоматизации процесса экологического мониторинга в системе Directum RX

Разработка в системе Directum RX осуществляется при помощи языка программирования высокого уровня C#. Это позволяет не только эффективно решать огромное количество прикладных задач, но и использовать уже готовые решения при помощи механизма импорта сторонних библиотек в систему. Разработка компонентов модуля велась согласно методологии разработки модулей системы электронного документооборота Directum RX. Хранение данных в системе обеспечивают справочники, а для хранения документов используются файловые хранилища. Метаданные в виде нескольких взаимосвязанных

компонент позволяют гибко настраивать бизнес-логику системы. Документ в системе реализуется при помощи нескольких сущностей.

- Тип документа позволяет определить документопоток, к которому относится документ. Также от типа зависит набор полей в карточке документа;
- Документопоток – направление движения документов в компании. Различают потоки входящих, исходящих, внутренних и договорных документов. Для каждого документопотока предусматривается свой порядок обработки по разработанным в компании;
- Карточка документа – форма, содержащая набор характеристик, которые описывают документ и могут использоваться для поиска и фильтрации электронных документов в системе. Например, наименование, автор, дата создания, вид и тип документа. Также в карточке документа можно настраивать права доступа на документ, связи с другими документами, создавать версии и просматривать историю работы с документом.

Основными объектами мониторинга являются животные, обитающие на территории ООПТ, следовательно, первой задачей является автоматизация классификация животных и других типов объектов экосистемы. Задача автоматизации каталога таксономии видов животных заключалась в создании набора справочников и их объединения в единую иерархию, а также загрузка исторических данных заповедника в данную структуру системы. Было разработано восемь отдельных справочников, связанных друг с другом полем-ссылкой. Была разработана единая форма создания и редактирования вида животного [15], которая представлена на "рис. 3".

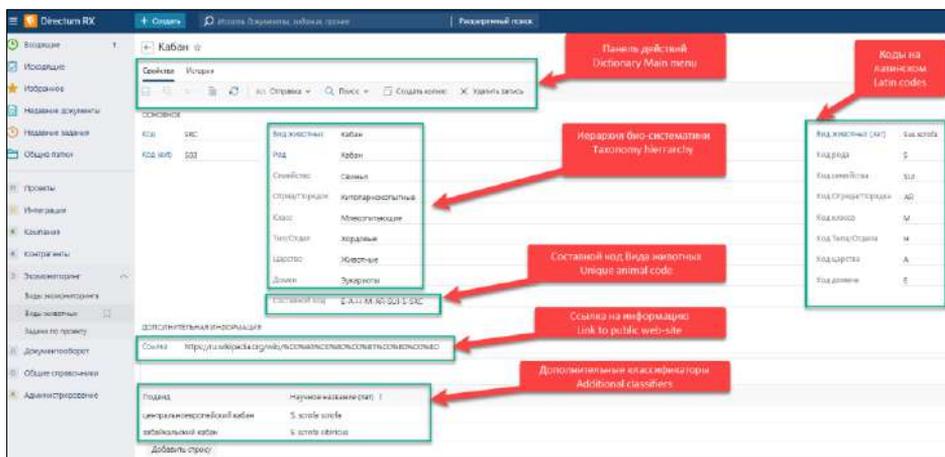


Рис. 3. Карточка справочника «Виды животных»
Fig. 3. «Animal species» directory card

Для сбора информации о наблюдениях были созданы карточки документов. Для каждого документа характерен свой набор реквизитов, в том числе ссылки на справочники, бизнес-процессы и процедуры [16, 17]. Например, для фиксации гибели животного был разработан документ со специальной формой. При загрузке фотографии с места гибели животного она сразу же отображается на карточке документа в соответствующем поле, рис. 4. Функция UploadPhoto, которая позволяет при импортировании файла в версию документа заполнять соответствующий реквизит на форме документа, представлена на листинге 1.

```

/// <summary>
/// Используется для загрузки фотографии на карточку документа из
/// последней версии документа
/// </summary>
public void UploadPhoto()

```

```
{
var version = _obj.Versions.LastOrDefault();
if ( version != null )
{
var body = version.PublicBody != null &&
version.PublicBody.Size > 0
? version.PublicBody : version.Body;
using (var memoryStream = new System.IO.MemoryStream())
{
body.Read().CopyTo(memoryStream);
_obj.Photo = memoryStream.ToArray();
}
_obj.Save();
}
else
{
_obj.Photo = null;
}
}
}
```

Листинг 1, UploadPhoto
Listing 1. UploadPhoto

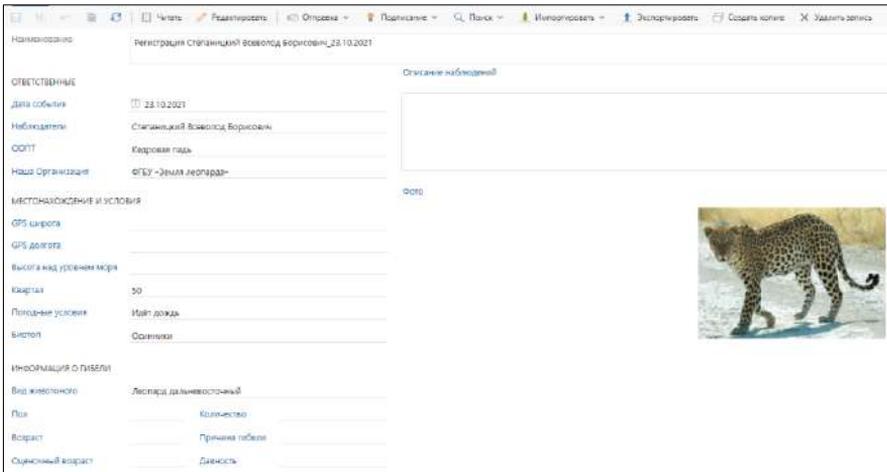


Рис. 4. Карточка справочника «Регистрация гибели животных»
Fig. 4. Card of the reference book «Registration of the death of animals»

Полный набор документов позволяет ответственным за природоохранную деятельность сотрудникам собирать и консолидировать информацию о наблюдениях на ежедневной основе, используя единые справочники и классификаторы [18].

Для сбора данных о метеоусловиях ранее использовался простой справочник погодных условий, без количественных показателей. В рамках проекта были разработаны: интеграционная функция для взаимодействия с сервисом Яндекс погода (представлена листингом функции RequestWeatherForecast), справочники станций мониторинга, фоновое задание, которое ежедневно, каждый час, собирает данные по всем станциям и сохраняет их в системе [19]. На рис. 5 приведён пример записи справочника «Метеоданные».

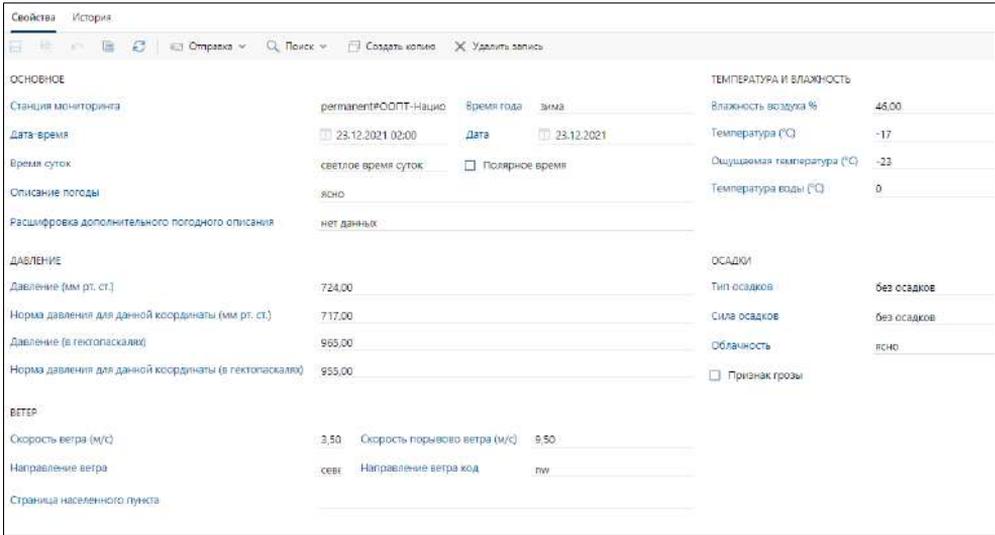


Рис. 5. Карточка справочника «Метеоданные»
 Fig. 5. Card of the reference book «Meteorological data»

```
public static YandexWeatherForecast RequestWeatherForecast
    (string lat, string lon, string lang, int limit,
     Boolean hours, Boolean extra)
{
    SortedDictionary<string, string> urlParamsDictionary
        = new SortedDictionary<string, string>()
    {
        {"lat", lat.Replace(",", ".") },
        {"lon", lon.Replace(",", ".") },
        {"lang", lang},
        {"limit", limit.ToString()},
        {"hours", hours.ToString()},
        {"extra", extra.ToString()}
    };
    string requestURL = "https://api.weather.yandex.ru/v2/forecast?",
        parameterString = "";
    foreach (KeyValuePair<string, string> keyValuePair
        in urlParamsDictionary)
    {
        parameterString += keyValuePair.Key + "="
            + keyValuePair.Value + "&";
    }
    string resultUrl = requestURL + parameterString;
    WebRequest request = WebRequest.Create(resultUrl);
    request.Headers.Add(Constants.Module.YandexWeatherApiKey);
    WebResponse response = request.GetResponse();
    var dataStream = response.GetResponseStream();
    StreamReader reader = new StreamReader(dataStream);
    string responseFromServer = reader.ReadToEnd();
    response.Close();
    return JsonConvert.DeserializeObject<YandexWeatherForecast>
        (responseFromServer);
}
```

Листинг 2. Функция RequestWeatherForecast
 Listing 2. RequestWeatherForecast function

Для автоматизации сбора данных была использована стандартная функциональность платформы Directum RX для выполнения периодических заданий – «Фоновые Задания» [20]. Чтобы обеспечить универсальный подход к реализации интеграционных механизмов было принято решение разработать отдельный модуль «Интеграция», в котором хранятся настройки для внешних сервисов, константы и другая релевантная информация. В рамках модуля было разработано два справочника:

- вендоры – компании и сервисы, с которыми будет выполняться интеграция;
- настройки интеграции – справочник с уникальными настройками для каждого вендора.

Рис. 6. Карточка документа «Регистрация встреч с дикими животными»
Fig. 6. Document card «Registration of encounters with wild animals»

Сравнительная таблица встреч животных и следов их жизнедеятельности в различные сезоны года																				
Обход № _____ с 01.01.20 по 01.06.22 гг. Структурное подразделение _____ Организация _____																				
Наименование вида	Количество встреч с дикими животными										Количество встреч со следами животных									
	Зима		Весна		Лето		Осень		Итого		Зима		Весна		Лето		Осень		Итого	
	Кол-во	%	Кол-во	%	Кол-во	%	Кол-во	%	Кол-во	%	Кол-во	%	Кол-во	%	Кол-во	%	Кол-во	%	Кол-во	%
Амурский ёж	1	9	0	0,00	0	0	10	91	11	100	0	0	0	0	0	0	1	100	1	100
Барсук азиатский	1	20	0	0,00	0	0	4	80	5	100	0	0	0	0	0	0	0	0	0	100
Белка	1	12	0	0,00	0	0	7	88	8	100	0	0	0	0	0	0	1	100	1	100
Ближневосточный рысь	0	0	0	0,00	0	0	6	100	6	100	0	0	0	0	0	0	1	100	1	100
Бронзовый медведь	1	17	0	0,00	0	0	5	83	6	100	0	0	0	0	0	0	0	0	0	100
Бурундук	1	12	0	0,00	0	0	7	88	8	100	1	100	0	0	0	0	0	0	0	100
Бурый медведь	1	20	0	0,00	0	0	4	80	5	100	0	0	0	0	0	0	0	0	0	100
Водяная опояска	3	50	0	0,00	0	0	3	50	6	100	0	0	0	0	0	0	0	0	0	100
Волк	2	29	0	0,00	0	0	5	71	7	100	0	0	0	0	0	0	0	0	0	100
Водяная лисица	0	0	0	0,00	0	0	7	100	7	100	0	0	0	0	0	0	0	0	0	100
Гималайский медведь	1	12	0	0,00	0	0	7	88	8	100	0	0	0	0	0	0	0	0	0	100
Горал	0	0	0	0,00	0	0	6	100	6	100	0	0	0	0	0	0	0	0	0	100
Домашняя кошка	4	33	0	0,00	0	0	8	67	12	100	0	0	0	0	0	0	0	0	0	100

Рис. 7. Сравнительная таблица встреч животных и следов их жизнедеятельности
Fig. 7. Comparative table of animal encounters and traces of their life activity

Сотрудники заповедника регулярно проводят обходы по фиксированным маршрутам. В рамках этой работы фиксируются встречи с дикими животными или следами их жизнедеятельности [21]. Данный процесс является важной частью данных экологического мониторинга. Количество встреч с дикими животными за период является важным критерием наблюдения за популяцией животных и динамикой её увеличения или уменьшения. Фактически, консолидированные данные по обходам позволяют принимать решения по установке фото ловушек и кормушек для животных. Был разработан новый тип документа «Обходы по маршруту». Данный справочник включает в себя информацию о географических координатах проведённых исследований и включает ссылки на справочник «Виды животных», которых удалось идентифицировать в рамках обхода. Карточка документа со всеми реквизитами представлена на рис. 6.

Directum RX позволяет разрабатывать статистические отчёты по данным, которые хранятся в системе. В рамках решения поставленной задачи был реализован отчёт о встречах с дикими животными в разрезе различных сезонов года. Данный отчёт также позволяет сделать выводы об изменении популяции конкретных видов животных. Скриншот отчета приведен на рис. 7.

5. Заключение

В статье была рассмотрена автоматизация процесса экологического мониторинга в одной из ООПТ на базе системы управления контентом Directum RX. Рассмотрена ролевая модель будущих пользователей системы, продемонстрирована техническая архитектура системы и разработанные компоненты модуля. Описаны этапы разработки модуля: разработка справочников и интеграции с сервисом Яндекс погода.

Продемонстрирован новый авторский механизм интеграции с веб-сервисом и создания иерархического 8-ми уровневого справочника таксономии животных, описаны компоненты системы, которые необходимо добавить для корректного функционирования решения. Разработана автоматизация действий пользователей для сбора необходимой информации для выполнения бионаблюдений.

Набор решений, представленный в настоящей работе, демонстрирует механизмы автоматизации действий пользователей системы Directum RX. Механизм интеграции с внешними веб-сервисами является новой разработкой и аналогом платного решения, поэтому представляют научный интерес как образец для применения лучших практик автоматизаций в системе Directum RX подобного рода [22]. Также данное решение может быть тиражировано на другие парки и заповедники, что унифицирует собираемые данные и позволит проводить по этим данным многофакторный анализ, строить модели, получать зависимости и как результат принимать правильные решения по управлению популяцией диких животных. Система на данный момент находится в опытно-промышленной эксплуатации и используется сотрудниками заповедника для работы.

Список литературы / References

- [1] Abdul-Azalova M.Y., Mamatova N.M. Big data technologies in technological and business processes automatization. *Big Data and Advanced Analytics*, no. 7-1, 2021, pp. 50-58.
- [2] Иванов Ф.Ф., Семенов К.Г. Реализация программного робота в системе управления документами Directum. *Вестник КРАУНЦ. Физико-математические науки*, том 30, no. 1, 2020 г., стр. 97-108 / Ivanov F.F., Semenov K.G. Implementation of the software robot in the Directum document management system. *Bulletin KRASEC. Physical and Mathematical Sciences*, vol. 30, no. 1, 2020, pp. 97-108 (in Russian).
- [3] Боровских Н.В., Кипервар Е.А. Электронный кадровый документооборот предприятия: проблемы внедрения и перспективы развития. *Вестник Сибирского института бизнеса и информационных технологий*, no. 3(35), 2020 г., стр. 55-60 / Borovskikh N.V., Kipervar E.A. Electronic personnel

- documentation of the enterprise: implementation problems and development prospects. Herald of Siberian Institute of Business and Information Technologies, no. 3(35), 2929, pp. 55-60 (in Russian).
- [4] Горячева А.П. Технологии регистрации входящих документов с использованием системы Directum RX (на примере департамента образования и науки Курганской области). Сборник докладов научно-практической конференции студентов Курганского государственного университета, 2021 г., стр. 30-34 / Goryacheva A.P. Technologies for registering incoming documents using the Directum RX system (on the example of the Department of Education and Science of the Kurgan Region). In Proc. of the Scientific and Practical Conference of Students of Kurgan State University, 2021, pp. 30-34 (in Russian).
- [5] Дубровина А.И. Организация настройки соединения HTTPS при работе с сервером DirectumRX при использовании сети "ИНТЕРНЕТ", Modern Science, no. 10-2, 2021 г., стр. 386-389 / Dubrovina A.I. Setting up an HTTPS connection when working with the DirectumRX server using the INTERNET network, Modern Science, no. 10-2, 2021, pp. 386-389 (in Russian).
- [6] Живонос Н.В. Анализ модели бизнес-процесса в системе Directum для роботизации маршрутизации документов. Сборник статей VII Всероссийской научно-практической конференции «Молодые учёные России», 2021 г., стр. 85-89 / Zhivonos N.V. Analysis of the business process model in the Directum system for document routing robotics. In Proc. of the VII All-Russian Scientific and Practical Conference «Young Scientists of Russia», 2021, pp. 85-89 (in Russian).
- [7] Златковская Е.М. Интеграция информационных систем с помощью RabbitMQ на примере создания сервиса согласования кадровых приказов. Инновации. Наука. Образование, no. 29, 2021 г., стр. 108-113 / Zlatkovskaya E.M. Integration of information systems using RabbitMQ on the example of creating a service for coordinating personnel orders. Innovation. Science. Education, no. 29, 2021, pp. 108-113 (in Russian).
- [8] Кейно П.П., Хорошко Л.Л., Рудько И.Д. Разработка модуля "Протокол выбора поставщика" на базе системы электронного документооборота Directum. Прикладная информатика, том 15, no. 4(88), 2020 г., стр. 54-63. Кейно П.П., Khoroshko L.L., Rudko I.D. Development of the «supplier selection Protocol» module based on the Directum electronic document management system. Journal of Applied Informatics, vol. 15, no. 4(88), 2020, pp. 54-63 (in Russian).
- [9] Кузнецов А.В. Модернизация системы электронного документооборота в администрации города Сарапула / А. В. Кузнецов, А. М. Чебаков. Материалы 1-й Межвузовской научно-технической конференции с международным участием «Актуальные проблемы науки и техники», 2020 г., стр. 145-148 / Kuznetsov A.V., Chebakov A.M. Modernization of the system of electronic document circulation in the administration of the city of Sarapul. In Proc. of the 1st Interuniversity Scientific and Technical Conference with International Participation on Actual Problems of Science and Technology", 2020, pp. 145-148 (in Russian).
- [10] Ларькин М.С., Гаврилова Н.М., Донкова И.А. Построение регрессионной модели для прогнозирования планового срока выполнения задач в Directum RX. Материалы Всероссийской конференции молодых ученых «Математическое и информационное моделирование», 2020 г., стр. 726-734 / Larkin M.S., Gavrilova N.M., Donkova I.A. Building a regression model for predicting the planned deadline for completing tasks in DIRECTUM RX. In Proc. of the All-Russian Conference of Young Scientists on Mathematical and Information Modeling, 2020, pp. 726-734 (in Russian).
- [11] Ларькин М.С., Гаврилова Н.М. Система расчета планового срока выполнения задач в СЭД Directum RX. Материалы Всероссийской конференции молодых ученых «Математическое и информационное моделирование», 2020 г., стр. 716-725 / Larkin M.S., Gavrilova N.M. The system for calculating the planned deadline for completing tasks in the Directum RX EDMS. In Proc. of the All-Russian Conference of Young Scientists on Mathematical and Information Modeling, 2020, pp. 726-734 (in Russian).
- [12] Никитина П.Н. Использование электронного документооборота для повышения эффективности работы сотрудников образовательной организации в делопроизводстве. Инновации. Наука. Образование, no. 49, 2022 г., стр. 1904-1909 / Nikitina P.N. The use of electronic document management to improve the efficiency of the work of employees of an educational organization in office work. Innovation. Science. Education, no. 49, 2022, pp. 1904-1909 (in Russian).
- [13] Красносельцева И.Е. Сравнительный анализ систем электронного документооборота с помощью методов кластеризации. Скиф. Вопросы студенческой науки, no. 3(43), 2020 г., стр. 221-226 / Krasnoseltseva I.E. Comparative analysis of electronic document treatment systems using clustering methods. Scyth. Issues of Student Science, no. 3(43), 2020, pp. 221-226 (in Russian).

- [14] Новиков А.Ю., Кейно П.П., Хорошко Л.Л. Разработка архитектуры интернет-сервиса организации научных мероприятий с автоматизацией документооборота. Прикладная информатика, том 13, no. 76(4), 2018 г., стр. 73-75 / Novikov A.Yu., Keyno P.P., Khoroshko L.L. Architecture development of internet-service for science conferences organizing with workflow automatization. *Journal of Applied Informatics*, vol. 13, no. 4 (76), 2018, pp. 70–76 (in Russian).
- [15] Салимова А.М. Реализация связывания документов «указания о приемке объекта к учету» с их приложениями на базе системы электронного документооборота Directum. Сборник статей по материалам VI Всероссийской конференции молодых ученых «Наука и инновации XXI века», 2020 г., стр. 16-19 / Salimova A.M. Implementation of the binding of «instructions on the acceptance of an object for accounting» with their applications based on the Directum electronic document management system. In Proc. of the VI All-Russian Conference of Young Scientists on Science and Innovations of the 21st Century, 2020, pp. 16-19 (in Russian).
- [16] Рудько И.Д. Модуль управления задачами TaskM. Свидетельство о государственной регистрации программы для ЭВМ № 2021662615 Российская Федерация, 02.08.2021 / Rudko I.D. TaskM task management module. Certificate of state registration of the computer program No. 2021662615 Russian Federation, 02.08.2021 (in Russian).
- [17] Учебная программа по использованию возможностей базовых модулей системы Directum. Свидетельство о государственной регистрации программы для ЭВМ № 2021662892 Российская Федерация, 06.08.2021 / A tutorial on using the capabilities of the basic modules of the Directum system. Certificate of state registration of the computer program No. 2021662892 Russian Federation, 08/06/2021 (in Russian).
- [18] Галимьянов И.Т., Порядина О.В. и др. Сравнительный анализ систем электронного документооборота. Материалы международной междисциплинарной научной конференции «Человек и общество перед вызовами глобальных трансформаций», том 2, 2020 г., стр. 201-203 / Galimyanov I.T., Poryadina O.V. Comparative analysis of electronic document management systems. In Proc. of the International Interdisciplinary Scientific Conference «Man and Society Facing the Challenges of Global Transformations», vol. 2, 2020, pp. 201-203(in Russian).
- [19] Ульянова В.А. Автоматизация заполнения инструкций по переносу разработки по документам изменений для справочника «Журнал изменения ПО» системы Directum. Сборник статей по материалам VI Всероссийской конференции молодых ученых «Наука и инновации XXI века», 2020 г., стр. 20-24 / Ulyanova V.A. Automation of filling in instructions for transferring development according to change documents for the directory «Software change log» of the Directum system. In Proc. of the VI All-Russian Conference of Young Scientists on Science and Innovations of the 21st Century, 2020, pp. 20-24 (in Russian).
- [20] Евграфов Д.В., Фадеева К.Н. Directum RX для автоматизации процессов на предприятии. Материалы Всероссийской научно-практической конференции «Цифровые технологии и инновации в развитии науки и образования», 2021 г., стр. 130-136 / Evgrafov D.V., Fadeena K.N. Directum RX for enterprise process automation. In Proc. of the All-Russian Scientific and Practical Conference «Digital technologies and innovations in the development of science and education», 2021, pp. 130-136 (in Russian).
- [21] Флакс Д.Б. Оценка технической эффективности интегрированной поддержки жизненного цикла основных фондов. Сборник научных статей 10-й Международной молодежной научной конференции «Поколение будущего: взгляд молодых ученых», 2021 г., стр. 107-110 / Flaks D.B. Evaluation of the technical efficiency of integrated support for the life cycle of fixed assets. In Proc. of the 10th International Youth Scientific Conference «Generation of the Future: View of Young Scientists», 2021, pp. 107-110 (in Russian).
- [22] Читчян Р.К. Оптимизация документооборота на предприятии при помощи облачных технологий и перспективы их развития. Научно-исследовательские публикации, no. 2 (34), 2016 г., стр. 26-36 / Chitchyan R.K. Optimization of the document of a turn at the enterprise by means of a cloud computing and prospect of their development. *Journal of Scientific Research Publications*, no. 2 (34), 2016, pp. 26-36 (in Russian).

Информация об авторах / Information about authors

Иван Дмитриевич РУДЬКО – аспирант, ведущий разработчик. Сфера научных интересов: автоматизация бизнес-процессов, построение математических моделей, компьютерное зрение, нейронные сети.

Ivan Dmitrievich RUDKO – PhD student, leading developer. Research interests: business process automation, building mathematical models, computer vision, neural networks.

Леонид Леонидович ХОРОШКО — к.т.н., доцент, заведующий кафедрой. Сфера научных интересов: системы автоматизированного-проектирования, моделирование пространственных форм, параметрическое моделирование

Leonid Leonidovich KHOROSHKO — Ph.D in engineering, Head of Department. Research interests: computer-aided design systems, modeling of spatial forms, parametric modeling

Павел Петрович КЕЙНО — кандидат технических наук, доцент. Сфера научных интересов: высоконагруженные системы, системы массового обслуживания, системы визуального программирования, теория компиляторов, предметно-ориентированное программирование

Pavel Petrovich KEYNO — Ph.D in engineering, Associate Professor. Research interests: high load systems, queuing systems, visual programming systems, compiler theory, domain-specific languages.

DOI: 10.15514/ISPRAS-2022-34(2)-13



Реализация функций управления задачами и ресурсами высокопроизводительной вычислительной системы в «СПО Супер-ЭВМ»

A.O. Ignatyev, ORCID: 0000-0003-4902-2123 <a.o.ignatyev@mail.ru>

A.A. Kalinin, ORCID: 0000-0001-6152-579X <feycheg@yandex.ru>

C.Yu. Mokshin, ORCID: 0000-0002-7454-6597 <sumo@rambler.ru>

*Всероссийский НИИ технической физики имени академика Е.И. Забабахина,
456770, Россия, г. Снежинск, Челябинская область, ул. Васильева, 13*

Аннотация. В данной работе приводится общее описание программного обеспечения Slurm-ВНИИТФ, разработки ФГУП РФЯЦ-ВНИИТФ им. академ. Е.И. Забабахина, включая его архитектуру и возможности по управлению ресурсами и планированию прохождения задач на высокопроизводительных вычислительных системах, предназначенных для решения задач численного моделирования (ВВС). Проведенные в ходе многолетних работ, связанных с эксплуатацией ВВС, исследования, показывают, что базовых возможностей программного обеспечения Slurm (Simple linux utility for resource management) явно недостаточно для эффективного использования вычислительных ресурсов в крупных вычислительных центрах, поэтому авторами данной публикации предлагается усовершенствованная политика управления задачами и ресурсами, описываются модули расширения (плагины) к Slurm, разработанные в ФГУП РФЯЦ-ВНИИТФ им. академ. Е.И. Забабахина и реализующие эту политику.

Ключевые слова: высокопроизводительная вычислительная система; кластер; подсистема управления задачами и ресурсами; Slurm; Slurm-ВНИИТФ; высокопроизводительные вычисления; моделирование.

Для цитирования: Игнатъев А.О., Калинин А.А., Мокшин С.Ю. Реализация функций управления задачами и ресурсами высокопроизводительной вычислительной системы в «СПО Супер-ЭВМ». Труды ИСП РАН, том 34, вып. 2, 2022 г., стр. 159-178. DOI: 10.15514/ISPRAS-2022-34(2)-13

Task and resources management function in HPC operation system «SPO Super-EVM»

A.O. Ignatyev, ORCID: 0000-0003-4902-2123 <a.o.ignatyev@mail.ru>

A.A. Kalinin, ORCID: 0000-0001-6152-579X <feycheg@yandex.ru>

S.Yu. Mokshin, ORCID: 0000-0002-7454-6597 <sumo@rambler.ru>

*E. I. Zababakhin All-Russian Scientific Research Institute of Technical Physics,
13, Vasilieva street, Chelyabinsk region, Snezhinsk, 456770, Russia*

Abstract. The Slurm-VNIITF software developed by Federal State Unitary Enterprise “Russian Federal Nuclear Center - Zababakhin All-Russian Research Institute of Technical Physics”, its architecture, resource management capabilities and task management for numerical simulation HPC systems described in this paper. During many years usage of the HPC systems researches show that the basic features of the Slurm (Simple linux utility for resource management) software are clearly insufficient for the effective use of computing resources in HPC centers. Therefore, the authors of this paper propose an improved task and resource management policy. Slurm extension modules (plugins) for implementing this policy also described in this paper.

Keywords: high-performance computing system; cluster; computer simulation; resource management; operation system; HPC modeling; HPC

For citation: Ignatyev A.O., Kalinin A.A., Mokshin S.Yu. Task and resources management function in HPC operation system «SPO Super-EVM». *Trudy ISP RAN/Proc. ISP RAS*, vol. 34, issue 2, 2022, pp. 159-178 (in Russian). DOI: 10.15514/ISPRAS–2022–34(2)–13

1. Введение

Многолетний опыт эксплуатации высокопроизводительных вычислительных систем, предназначенных для решения задач численного моделирования (далее по тексту – ВВС) [1] в ФГУП РФЯЦ-ВНИИТФ им. академ. Е.И. Забабахина (далее по тексту РФЯЦ-ВНИИТФ) показал, что возможность эффективного использования вычислительных ресурсов ВВС безусловно связана с реализуемыми подходами по функциональной организации расчетов задач, управления их очередью. В ВВС эти функции возлагаются на специализированную подсистему управления задачами и ресурсами, которая предназначена для запуска и выполнения задач пользователей (прикладных программ) в пакетном и интерактивном режиме на вычислительном поле ВВС. В 2015 году в РФЯЦ-ВНИИТФ была разработана предназначенная для использования на ВВС операционная система «СПО Супер-ЭВМ» [2]. Одним из компонентов «СПО Супер-ЭВМ» является программная часть этой подсистемы управления задачами и ресурсами, основанная на программном обеспечении (далее по тексту ПО) Slurm (Simple linux utility for resource management) [3], разработанном в Ливерморской национальной лаборатории им. Лоуренса (Lawrence Livermore National Laboratory) и функционально дополненным в РФЯЦ-ВНИИТФ. Проведенные в ходе многолетних работ, связанных с эксплуатацией ВВС, исследования, показывают, что базовых возможностей ПО Slurm явно недостаточно для эффективного использования вычислительных ресурсов в крупных вычислительных центрах, поэтому авторами данной публикации предлагается усовершенствованная политика управления задачами и ресурсами, описываются модули расширения (плагины) к Slurm, разработанные в РФЯЦ-ВНИИТФ и реализующие эту политику.

В тексте документа используются следующие понятия:

- «базовое ПО Slurm» и «базовые возможности» по отношению к свободно распространяемой версии ПО Slurm;
- «расширенные возможности» применительно к функциям, реализованным в плагинах, разработанных в РФЯЦ-ВНИИТФ (ПО Slurm-ВНИИТФ);
- «задача» (в терминологии Slurm job) – единица планирования выделения ресурсов для проведения определенной вычислительной работы;
- «задание» – описание задачи, подготовленное пользователем для ввода ее в Slurm, содержащее требуемые для задачи ресурсы и выполняемые действия (сценарий выполнения задачи);
- процесс (в терминологии Slurm task) – часть параллельной задачи, запускаемыми компонентами ПО Slurm на вычислительных узлах ВВС.

2. Описание структуры базовой версии SLURM

Для четкого понимания возможностей и пределов модификации базового ПО Slurm приведем описание его структуры, оценим реализованные функции по управлению ресурсами, управлению планированием запуска и запуском задач.

ПО Slurm является высокомасштабируемой программной системой с открытым исходным кодом, предназначенной для управления ресурсами вычислительного комплекса, планирования и запуска вычислительных задач. ПО Slurm выполняет три ключевые функции:

- определяет и захватывает ресурсы (вычислительные узлы, оперативная память,

процессоры, ядра) для пользователей в необходимом количестве на определённое время;

- предоставляет средства для запуска и мониторинга задач на выделенных узлах;
- организует очередь задач, выполняет планирование запуска задач согласно настроенным правилам, предотвращает конфликты при захвате ресурсов.

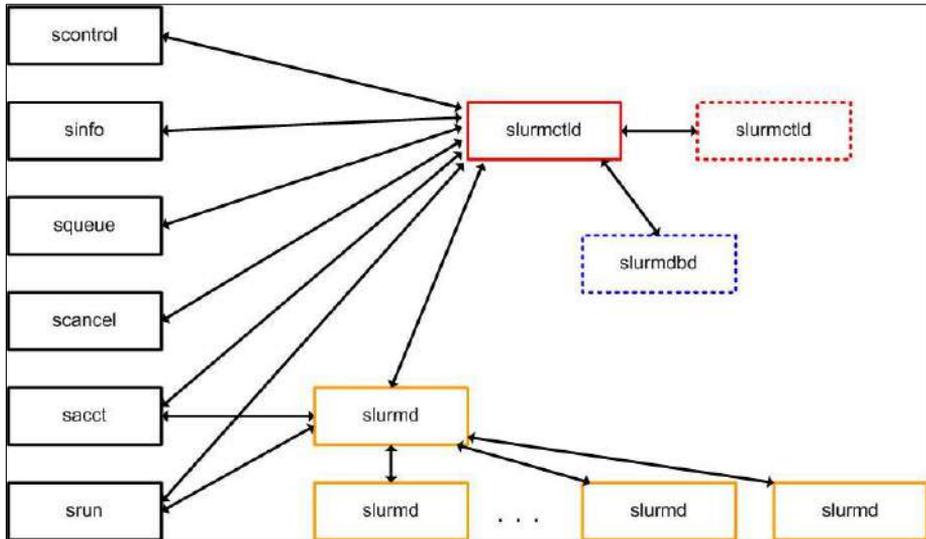


Рис. 1. Состав ПО Slurm и его компонентов
Fig. 1. Slurm structure

ПО Slurm состоит из следующих основных компонентов (см. рис. 1).

- Центральный демон slurmctld (контроллер) – управляет работой всей подсистемы и обрабатывает запросы пользовательских команд. Как правило, для ВВС в целях отказоустойчивости дополнительно может функционировать резервный контроллер, запущенный на другом физическом или виртуальном сервере. Резервный контроллер получает управление в случае отказа основного сервера. Контроллер выполняет функции менеджера ресурсов и планировщика задач.
- Клиентские демоны slurmd, работающие на каждом вычислительном узле. Все клиентские демоны взаимодействуют друг с другом и другими компонентами через сеть, образуя отказоустойчивую иерархическую структуру.
- Утилиты (команды), предназначенные для взаимодействия пользователя (или администратора) с Slurm;
- Необязательный демон slurmdbd, который может использоваться для сохранения статистической и учётной информации по задачам, пользователям и ресурсам в базу данных. Один демон slurmdbd может использоваться для нескольких кластеров и хранить информацию в специально предназначенной для этого базе данных.

Логические объекты, управляемые ПО Slurm, включают следующие сущности:

- узлы (nodes) – вычислительные ресурсы;
- разделы (partitions) – узлы, логически объединённые в одно множество;
- резервации (reservation) – выделенные для специальных целей узлы, логически объединённые в одно множество;
- задачи (задания, jobs) – заданная пользователем вычислительная работа, под которую требуется выделить ресурсы ВВС на определенное время;
- шаги задания (job steps) – множества процессов (экземпляров программы) внутри задачи, обрабатываемые один за другим.

Разделы и резервации можно рассматривать как целевой ресурс планирования очереди задач, каждая из которых имеет набор ограничений, таких как: число выделяемых для задачи узлов, разрешённые пользователи и т.д. Узлы внутри раздела выделяются для задач до тех пор, пока достаточно ресурсов (узлов, процессоров, памяти и т.д.). После того, как под задачу выделен набор узлов, пользователь может инициировать шаги задания (запускать параллельные программы) в любой конфигурации на выделенных ресурсах. Например, один шаг может использовать сразу все узлы, выделенные для задачи, или несколько пунктов могут независимо использовать только часть узлов.

При постановке задачи пользователя в очередь на исполнение ПО Slurm запоминает идентификатор пользователя, поставившего задачу, и связанный с ним идентификатор группы. В дальнейшем, после выделения вычислительных ресурсов, ПО Slurm осуществляет запуск процессов выполняемых в задаче программ и назначает им ранее сохраненные идентификаторы пользователя и группы. Таким образом, процессы на выделенных задаче узлах имеют те же права доступа к данным, что и пользователь, запустивший задачу.

3. Возможности базовой версии SLURM по управлению ресурсами

Большая часть задаваемых при постановке задачи в очередь параметров описывает требуемые для задачи ресурсы. В первую очередь это имя требуемого раздела или резервации. Если они не указаны, используется раздел, заданный администратором как основной. Под разделом подразумевается множество узлов, объединенных общими свойствами (архитектура, характеристики, назначение). Одна задача не может занимать узлы из разных разделов, хотя одни и те же узлы могут входить в разные разделы. Разделы создаются администратором при конфигурировании ПО Slurm на BBS.

Резервацией также является множество узлов, но, в отличие от разделов, резервации создаются для выполнения специальных действий, таких как настройка узлов, или выполнение задач в заданный промежуток времени. Резервации также создаются администратором, но, по мере необходимости, могут быть созданы или удалены в командном режиме без внесения изменений в конфигурационные файлы ПО Slurm. В отличие от разделов, резервации снабжаются списком доступа, т.е. не все пользователи по умолчанию допускаются к конкретной резервации.

Следующая группа параметров включает в себя информацию о количестве требуемых для задачи вычислительных узлов, а также о максимальном количестве запускаемых в задаче одновременно параллельных процессов. Поскольку основное назначение ПО Slurm – запуск параллельных программ, использующих технологию MPI [4] для взаимодействия процессов параллельных программ, то в ПО Slurm используется для этого понятие «task» из MPI, хотя ПО Slurm также способно учитывать потоки OpenMP («threads») [5].

В эту же группу входят дополнительные параметры, уточняющие требуемые характеристики узлов: число сокетов для процессоров на узле, число ядер процессора, число потоков, предполагаемых к запуску на процессе, наличие и тип ускорителя, требуемую для процессов задачи оперативную память на узле, специальные возможности (features), назначенные администратором для вычислительных узлов и многое другое. Многие из этих параметров взаимосвязаны, это позволяет пользователю задать требуемую информацию максимально удобным способом.

Планировщик контроллера при просмотре очереди задач обращается к менеджеру ресурсов для подбора множества узлов, которые были бы свободны в настоящее время и удовлетворяли заданным в задаче требованиям. После успешного планирования задачи, т.е. в момент, когда необходимый набор узлов сформирован, менеджер ресурсов закрепляет их за задачами. После завершения процессов задачи менеджер ресурсов освобождает ресурсы, выделенные процессам, делая их доступными для других задач, ожидающих своей очереди.

4. Возможности базовой версии SLURM по управлению планированием запуска и запуском задач

Будучи, как следует из названия, в основном менеджером ресурсов, ПО Slurm, тем не менее, представляет возможности для планирования запуска задач. Таких возможностей три:

- использование одного из двух простых внутренних планировщиков;
- использование планировщика сторонних производителей, интегрируемых с ПО Slurm (например, Maui [6], Moab [7]);
- написание собственного планировщика (для этого предусмотрен уровень интерфейсов).

В состав ПО Slurm входят два типа внутренних планировщиков:

- builtin (встроенный);
- backfill (с обратным заполнением).

В общем виде в процессе планирования в ПО Slurm выполняются следующие действия:

- создается список готовых к запуску задач (т.е. таких, для которых выполнены все условия, например, время начала выполнения, и зависимости от состояния других задач);
- список задач упорядочивается по нескольким критериям, основным из которых является приоритет;
- если количество задач в списке превышает заданный предел, отсекается менее приоритетная часть списка;
- для каждой из задач формируется список ресурсов (разделов или резерваций), на которых допустим запуск этой задачи;
- предпринимается попытка вычисления времени запуска задач из списка.

Встроенный планировщик просто берет самую приоритетную задачу из нижней части списка и пытается подобрать для нее узлы из списка ресурсов, составленных для задачи. Если необходимые узлы находятся, задаче устанавливается текущее время для запуска и рассматривается возможность запуска следующей задачи. В противном случае процесс рассмотрения данного списка задач прекращается.

Планировщик с обратным заполнением пытается составить расписание запуска задач, учитывая возможность одновременного запуска нескольких задач. За основу опять же берется упорядоченный по приоритетам список задач, для которых с учетом заказанного времени счета строится расписание запуска. Затем свободные ресурсы заполняются менее приоритетными задачами. При этом обеспечивается более плотное, по сравнению с встроенным планировщиком, заполнение вычислительного поля ВВС. При составлении расписания учитываются размер задачи (количество требуемых для ее выполнения узлов), заказанное время счета, возможность совмещения разных задач на узлах и многое другое. Приоритеты назначаются задачам динамически, исходя из времени нахождения задачи в состоянии ожидания, размеров задачи, заказанного времени счета.

Готовая к выполнению задача отправляется контроллером к компоненте slurmd на первом из выбранных для задачи узлов. Демон slurmd обеспечивает запуск процесса, на котором исполняется сценарий задачи, собирает и передает контроллеру выводные файлы процессов задачи (stdout и stderr), контролирует использование внутренних ресурсов узла (CPU и оперативной памяти), а также отслеживает завершение задачи и передает это событие контроллеру.

5. Возможности базовой версии SLURM по расширению функций

Программное обеспечение Slurm ориентировано на использование подключаемых модулей (плагинов). Существует несколько типов плагинов и несколько реализаций каждого типа. Наиболее важные из них следующие:

- TaskPlugin – используется компонентой slurmd на узлах ВВС для запуска задачи и контроля за его исполнением;
- JobSubmitPlugin – используется утилитой sbatch (salloc) для проверки правильности заданных параметров и формирования на их основе атрибутов задачи;
- SchedPlugin – используется для расчета времени запуска задач;
- SelectPlugin – используется для подбора подходящих для задачи узлов и, при необходимости, запуска задачи на счет;
- PriorityPlugin – используется для вычисления приоритетов задач;
- JobAcctGatherPlugin – используется для подсчета использованных задачей ресурсов;
- группа плагинов под общим названием AcctGather – используется для подсчета используемых ресурсов ВВС (сетевой трафик, обращение к системе хранения и т.д.);
- PreemptPlugin – используется для определения возможности предоставления ресурсов задаче за счет вытеснения других задач (под приостановкой понимается временное прекращение счета задачи без освобождения занимаемых ей ресурсов оперативной памяти узлов);
- LaunchPlugin – используется для выполнения определенных действий при запуске задачи;
- JobCompPlugin – используется для выполнения определенных действий при завершении задачи.

Использование конкретной реализации плагина позволяет обеспечить более эффективное выполнение требуемых действий.

6. Анализ требуемых для ВВС функциональных особенностей и возможности их реализации в базовой версии SLURM

Базовые возможности ПО Slurm позволяют применять это ПО на самых различных ВВС. В тоже время при проведении массово-параллельных расчетов в научных организациях, в которых количество насчитываемых задач в год может превышать миллионные значения, а количество имеющихся вычислительных ресурсов не соответствует потребностям в вычислениях, остро встает вопрос о повышении эффективности использования вычислительных ресурсов имеющихся в наличии ВВС.

В таких случаях появляется понятие приоритетности задачи, т.е. появляется класс задач, который необходимо решить в первую очередь, даже, вполне возможно, за счет вытеснения с вычислительного поля ВВС уже запущенных в счет задач. Причем приоритетность задачи – величина, во-первых, непостоянная, во-вторых, она может являться результатом весовой функции из некоторого числа переменных. Критериями оценки значимости этих переменных в весовой функции могут быть важность решаемой конкретной задачи или группы задач, важность набора задач определенного пользователя или группы пользователей, объединенных общей научной или производственной направленностью.

Учитывая особенности проведения работ, в принципе в любой научной организации, подобной РФЯЦ-ВНИИТФ, для правильной организации расчетов можно выделить тематические вычислительные направления, связанные либо с исследуемыми физическими объектами, явлениями и процессами, либо с используемыми при расчетах математическими методами. Набор задач, решаемых по каждому из таких направлений, можно очень условно называть «методикой». Авторы публикации не претендуют на правильность и объективность данного термина, но такая терминология исторически была принята в РФЯЦ-ВНИИТФ.

Важность такого деления на наборы задач напрямую связана с понятием их приоритетности. Любая «методика» в какой-то момент времени, когда требуется срочно получить результат расчета, может стать приоритетной и наоборот, по факту получения результатов важность ее снижается. Таким образом, понятие «методика» может определять группу задач, обладающих общим признаком приоритетности и оперируя сочетанием двух понятий: «методика» – «ее приоритетность» в очереди задач планировщика можно более эффективно использовать ресурсы ВВС. В то же время внутри одной методики также, иногда, возникает необходимость выделения отдельных задач по уникальным для них показателям приоритетности.

Такое логическое объединение вычислительных задач на методики потенциально дает возможность задавать квоты по использованию ресурсов ВВС для конкретной группы задач. Что тоже, безусловно, позволяет более эффективно управлять ресурсами ВВС.

Помимо понятия приоритетности при расчетах на ВВС также целесообразно вводить определение «дневного» и «ночного» счета, связанные с активностью пользователей в рабочее время и, как следствие, определенной зависимостью прироста количества задач от времени суток. Смысл такого деления состоит в том, что в «ночном» режиме, например, не требуется немедленного получения результата от запущенной в счет задачи, так как в ночное время этот результат некому обработать. В силу отсутствия пользователей не происходит запуска новых задач, что позволяет планировщику составлять расписание, оптимально использующее вычислительные ресурсы. Режим «ночного» счета целесообразно также применять в выходные и праздничные дни.

В ходе проведения расчетов на ВВС нередко возникают ситуации, когда в вычислительной очереди скапливается множество разнородных задач, как имеющих важное значение для получения каких-то результатов расчета, так и предназначенных для отладки новых программ или алгоритмов. Ценность последних может оказаться небольшой с практической точки зрения, но, тем не менее, необходимость проведения тестовых и отладочных расчетов тоже безусловно существует. Часть из таких задач требуют небольших вычислительных ресурсов и непродолжительное время счета, но, тем не менее, не имеют возможности запуститься на счет в течение длительного времени из-за занятости ресурсов другими расчетными задачами. Более оперативный запуск таких задач за счет временного использования ресурсов программ, находящихся в расчете в течение нескольких суток или даже месяцев и занимающих большое количество вычислительных узлов, не сказался бы существенно на относительном времени их расчета, но способствовал более эффективному и разностороннему использованию вычислительных ресурсов ВВС.

Еще одной ключевой особенностью эффективного использования вычислительных ресурсов ВВС является способность системы управления (планирования запуска задач) менять размер задачи (количество запускаемых процессов и используемых вычислительных узлов) в зависимости от степени загруженности задачами вычислительного поля. В то время, когда необходимо обеспечить минимальное время задержки запуска, следует уменьшить количество вычислительных ресурсов, занятых задачами переменного размера, обеспечивая пространство для вновь запускаемых задач. А в периоды, когда количество активных пользователей вычислительной системы невелико, лучше предоставить максимальный вычислительный ресурс, для их эффективной утилизации. Естественно, при этом задачи сами должны иметь возможность свободной масштабируемости, т.е. относится к так называемому минимаксному типу задач (для них строго определены минимальные и максимальные потребности в ресурсах), и не относится к критичным по времени счета. Базовые возможности ПО Slurm в настоящее время позволяют задавать переменный размер задачи, но не используют его при планировании.

Важно отметить, что практически всегда на практике существует определенный класс задач, который не попадает по своим признакам и возможностям ни в одну из вышеперечисленных категорий. Это так называемые исключительные задачи. Например, задачи с постоянной

приоритетностью, задачи не имеющие функции автоматического сохранения при перезапуске, задачи, выполняемые в среде эмуляторов и т.п. Они малочисленны, но тем не менее имеют право на существование и исключительные права при распределении ресурсов на ВВС. Как правило, такие задачи являются невытесняемыми.

Есть еще один класс задач, на которые следует обратить особое внимание – это задачи, запускаемые администраторами ВВС в процессе обслуживания (диагностики) функциональных подсистем ВВС. Потребность в наиболее срочном прохождении этих задач очевидна. Возможность срочного запуска задач в этом случае достигается двумя способами. В случае, если требуется тестирование и наладка изолированного множества вычислительных узлов, то после их вывода из состояния наладки менеджер ресурсов автоматически распределяет их в резервацию, предназначенную для запуска тестов. В противном случае весь счет на ВВС, как правило, должен быть остановлен, и планировщик должен уметь работать с так называемым «белым списком» задач, которым предоставлено исключительное право запуска на вычислительном поле, тогда как другие пользователи могут продолжать постановку расчетных задач в очередь.

Коллективы пользователей ВВС, обслуживающие расчеты по различным методикам, могут быть достаточно велики, и организационно бывает удобным выделить из их состава администраторов расчета. В круг их возможностей должны входить функции снятия и перезапуска задач, а также управление порядком запуска задач собственной методики.

Функции, определяющие приоритет задачи, а также порядок рассмотрения кандидатов для вытеснения, зависят от многих критериев (занятость вычислительных ресурсов, среднее время счета задачи, характерное время сохранения данных для перезапуска и т.п.) и требуют аккуратной настройки весовых коэффициентов под каждый вычислительный центр и круг решаемых им задач.

Приведенные требования показывают, что базовых возможностей ПО Slurm явно недостаточно, чтобы удовлетворять столь разносторонним потребностям в управлении расчетами. Имеющиеся на рынке и в свободном доступе решения типа Maui или Moab реализуют часть вышеуказанных потребностей, но в основном в сочетании с менеджером ресурсов Torque [8]. Один из них – проект Maui давно прекратил свое развитие, другой, созданный на основе первого – Moab HPC Suite объединил в себе множество инструментов для управления ВВС и является платным продуктом, исходные коды которого закрыты и не позволяют их адаптировать под любые другие условия использования. Для ПО Slurm оба этих планировщика являются внешними, а протокол взаимодействия не обеспечивает достаточное количество событий и параметров. Поэтому в РФЯЦ-ВНИИТФ было принято решение расширить возможности ПО Slurm собственными разработками.

7. Расширенные возможности ПО SLURM-ВНИИТФ

7.1 Расширенная политика планирования задач

На основе проведенного анализа можно сделать выводы о том, какой может быть политика планирования задач для типовой ВВС, предназначенной для решения задач численного моделирования. Прежде всего, следует отметить, что эта политика подразумевает автоматическое управление запуском задач, и не требует участие администратора ВВС в процессе планирования.

Данная политика была реализована специалистами РФЯЦ-ВНИИТФ в ПО Slurm-ВНИИТФ. В ней задачи по значимости разделяются на следующие классы:

- интерактивные – короткоживущие задачи с минимальным требованием к вычислительным ресурсам, необходимых для отладки и визуализации результатов расчета;
- срочные – требующие как можно более быстрого решения, возможно, за счет прочих

задач;

- обычные – средний класс задач, не предъявляющих особых требований по срокам расчета;
- низкоприоритетные (фоновые) – класс ресурсоемких задач, решение которых представляет интерес, но срочность получения результатов от них не велика, поэтому такие задачи могут быть просчитаны после остальных или при появлении свободных вычислительных ресурсов, а будучи уже запущены, могут быть приостановлены (вытеснены из очереди задач) для предоставления ресурсов более приоритетным задачам.

Задача может быть причислена к соответствующему классу как индивидуально, так и по принадлежности к определенной группе задач. Предполагаются следующие типы группировки:

- по темам – в группу попадают все задачи, объединенные общей темой расчетов, например, расчет какой-либо конструкции, физического объекта, его модуля и т.д.
- по методикам – в группу попадают все задачи, проводящие расчеты по одной математической методике или их некоторому набору.

Способы группировки задач могут быть и другими, но в всех случаях для группировки используются данные из атрибутов задачи.

Для решения проблемы обеспечения высокоприоритетных задач вычислительными ресурсами были определены следующие механизмы:

- предоставление зарезервированных ресурсов,
- квотирование групп задач – для этого каждой группе назначаются ограничения на использование вычислительных ресурсов (лимиты), при превышении которых блокируется постановка новых задач той же группы в очередь на исполнение, либо такие задачи причисляются к классу фоновых,
- вытеснение менее приоритетных – в этом случае при появлении в очереди на выполнение срочной задачи, ей предоставляются ресурсы, занимаемые фоновыми задачами; вытесняемые задачи исключаются из счета и могут быть продолжены при появлении достаточного количества вычислительных ресурсов.

Для реализации этой политики были расширены функциональные возможности базовой версии Slurm:

- реализованы дополнительные возможности для задания атрибутов задач, позволяющих определить степень срочности задачи,
- реализованы различные способы группировки задач по их атрибутам,
- усовершенствованы алгоритмы планирования с учетом принадлежности задач к определенным группам, установленным квотам, степени приоритетности задачи, а также возможности вытеснения менее приоритетных задач,
- введена весовая функция расчета приоритетов,
- введена весовая функция пригодности задач к вытеснению,
- реализовано вытеснение фоновых задач для счета срочных задач,
- обеспечена возможность гибкого управления параметрами, определяющими политику планирования.

7.2 Описание структуры расширенной версии ПО SLURM-ВНИИТФ

Для реализации расширенных возможностей специалистами РФЯЦ-ВНИИТФ были разработаны следующие плагины:

- submit – постановка задачи в очередь, изменение параметров;
- priority – назначение приоритетов задачам;
- sched – расчет приоритетов задач;
- preempt – выбор задач для вытеснения;

- gproctrack – передача сигналов процессам задач.

Работа плагинов управляется общим конфигурационным файлом формата YAML [9], разбитым на секции. Каждая секция содержит данные, необходимые для работы одного или нескольких плагинов. Отдельная нить контроллера отслеживает модификацию конфигурационного файла и перечитывает его содержимое, обеспечивая, тем самым, оперативное вмешательство администраторов в работу ПО Slurm-ВНИИТФ.

На рис. 2 приведен пример конфигурационного файла плагинов для ВВС.

```
# режим планирования
sched:
  mode: normal # day или night в зависимости от времени суток и дня недели
  blacklist: false # режим черного списка (запрет запуска избранных)
  use_preemption: true
  use_limits: true
  interval: 30 # интервал планирования
  worktime: "07:00-19:00" # время работы в режиме встроенного планировщика
  workdays: [2,3,4,5,6] # рабочие дни недели (первый день недели ВС)

preempt: # секция вытесняющего планирования
  background_prio: 10
  standard_prio: 10000
  urgent_prio: 1000000

urgent: # срочные задачи, не подчиняющиеся ограничениям
- { name: '*.urgent' }
background: # принудительно фоновые задачи
- { name: '*.Ф*' }
uncancelable: # невытесняемые задачи
- { name: '*.Н*' }
limits: # описание ресурсов и лимитов
- {key: 'Quser1-1', user: User1, use: PMEM, limit: 20}
- {key: 'Quser1-2', user: User1, use: PMEM, limit: 60}
- {key: 'Quser2-1', user: User2, use: PMEM, limit: 0}
- {key: 'Quser2-2', user: User2, use: PMEM, limit: 20}
# постановка в очередь
submit:
  default part: "PSMEM" # раздел для всех задач по умолчанию
  allow: # допустимые разделы (name, user, users, size -> partitions)
  - group: "Grp1" # и группа пользователя Grp1
    partitions: # список допустимых разделов
    - "*" # допустим любой раздел
  - group: "Grp2" # и группа пользователя Grp2
    partitions: # список допустимых разделов
    - "PSMEM" # допустим только раздел PSMEM
# приоритеты
prio:
  # по типу
  - {name: "*.П*", prio: 20}
  - {name: "*.М*", prio: 10}
  - {name: "*.Т*", prio: 8}
# администраторы методик
adm:
  - name: 'МЕТОДИКА1.*'
    admins:
    - {user: User1}
```

Рис. 2. Пример конфигурационного файла плагинов

Fig. 2. Configuration plugin example

Эта конфигурация содержит 4 основных секции (sched, submit, prio и adm), определяющие порядок планирования, постановку задачи в очередь, установку приоритетов и администраторов методик, соответственно. В данном примере задано чередование дневного и ночного режима планирования, определены признаки обнаружения срочных задач (окончание «urgent» в имени задачи), фоновых задач (буква «Ф» в имени задачи), невытесняемых задач (буква «Н» в имени задачи). Кроме того, для пользователей установлены ограничения (квоты) на количество одновременно выделяемых узлов в разделах, установлено ограничение на использование разделов для разных групп пользователей и заданы приоритеты для разных типов задач. Для задач методики

«МЕТОДИКА1» назначен администратор User1, способный управлять задачами данной методики.

7.2.1 Типы задач

Для целей оптимального планирования пользователь должен указать тип задачи. Основными являются:

- производственная – задача по проведению расчетов в рамках плана работ предприятия;
- методическая – задача, связанная с уточнением существующих и отработкой перспективных вычислительных методик;
- тестовая – задача, связанная с тестированием оборудования или вычислительных методик.

В расширенной версии Slurm-ВНИИТФ также введены дополнительные типы задач.

- *Стандартные*: для каждой методики вводится ограничение на общее число вычислительных узлов, занимаемых суммарно всеми стандартными задачами данной методики, находящимися в счете. Сверх этого лимита стандартные задачи становятся фоновыми. Стандартной может быть любая задача с типами производственная, методическая, тестовая.
- *Фоновые*: в эту категорию попадают стандартные задачи, выходящие за пределы квоты. Они могут сниматься с расчета для освобождения вычислительных ресурсов. Объем вычислительных ресурсов, планируемых для расчета фоновых задач каждой методикой, ограничивается лишь количеством свободных вычислительных узлов, удовлетворяющих ограничениям задачи. Для реализации данного режима расчета предполагается, что задачи обязательно должны иметь возможность записи контрольных точек и продолжения счета с контрольной точки. Тип «фоновая» присваивается при постановке задачи в очередь на основании указанных атрибутов, может быть явно указан пользователем. При появлении нужной квоты возможен обратный переход из фоновой в стандартную.
- *Принудительно фоновые*: задачи, которые никогда не смогут стать стандартными. Их приоритет всегда будет находиться в диапазоне фоновых задач.
- *Отладочные и интерактивные*: данный тип предназначен для оперативного запуска задач, требующих небольших вычислительных ресурсов, с целью отладки программного кода в реальном окружении (на вычислительном поле). Данный тип присваивается тем задачам, время выполнения которых ограничено несколькими минутами и небольшим количеством вычислительных узлов.
- *Срочные*: задачи с приоритетом ниже отладочных, но выше чем диапазон стандартных задач. Для их запуска должны: автоматически сниматься фоновые задачи, автоматически сниматься со счета стандартные задачи этой же методики, автоматически прекращаться запуск стандартных задач. С расчета срочные задачи не снимаются никогда. Список срочных задач задается администратором в конфигурационном файле.
- *Невытесняемые*: задачи, которые не должны перезапускаться для запуска срочных и стандартных задач. Указывается пользователем при постановке задачи в очередь.

Плагин Ppriority на основании информации из секции prio конфигурационного файла, а также уточняющей информации из секций sched.preempt и sched.debug, рассчитывает начальное значение приоритетов задач. В вычислении приоритета участвуют тип задачи, ее размер и прочие атрибуты, заданные администратором.

```

sched:
  preempt:
    background_prio: 10
    standard_prio: 10000
    urgent_prio: 1000000
    urgent: # срочные задачи
      - { name: '*.urgent' }
    background: # принудительно фоновые задачи
      - { name: '*.Ф*' }
    uncancelable: # невытесняемые задачи
      - { name: '*.Н*' }
  debug:
    use: true
    limit: 1
    time: 10
    size: 4
    priority: 1001000
  prio:
    - { name: "*.П*", prio: 20 }
    - { name: "*.М*", prio: 10 }
    - { name: "*.Т*", prio: 8 }

```

Рис. 3. Пример конфигурационного файла Priority
Fig. 3. Priority plugin example

Например, конфигурация, представленная на рис. 3, определяет:

- признаки производственных, методических и тестовых задач (буквы «П», «М» и «Т» в имени задачи) и их поправки к приоритетам (в целочисленном выражении: +20, +10 и +8, соответственно);
- признак фоновых задач (буква «Ф» в имени задачи) и базовый приоритет для них (10);
- признак невытесняемых задач (буква «Н» в имени задачи);
- базовые приоритеты для фоновых, стандартных и срочных задач;
- признак срочных задач (окончание «.urgent» в имени задачи) и приоритет их (1000000);
- признак отладочных задач (одна задача на пользователя с заказанным временем 10 минут и запрашивающая не более 4 узлов) и приоритет для них (1001000).

Полученные для задачи по разным критериям приоритеты складываются.

7.2.2 Режимы планирования

В расширенной версии Slurm-ВНИИТФ введены следующие режимы функционирования.

- *Дневной*: основная стратегия минимизирует время ожидания приоритетных задач в очереди. Обеспечивает быстрый запуск интерактивных и отладочных задач. В дневном режиме невозможен запуск задачи с низким приоритетом, пока в очереди находится высокоприоритетная задача.
- *Ночной*: минимизирует простои вычислительных ресурсов, характеризуется малой динамикой, возможностью составления долговременного расписания и более широкими возможностями оптимизации. Использует ресурсы, предоставленные днем для интерактивных и отладочных задач. В ночном режиме возможен запуск низкоприоритетных задач, в случае, если их счет не помешает запуску высокоприоритетных задач в запланированное время. Определение интервала времени для ночного режима задается в конфигурационных файлах Slurm.
- *Останов*: полный останов запуска задач. Используется во время профилактики для запрета запуска задач.
- *Режим белого списка*: запрет запуска задач кроме заданных категорий. Используется во время профилактики для тестирования вычислительной системы.
- *Режим черного списка*: запрет запуска задач заданных категорий. Этот режим совместим со всеми другими режимами планирования. Можно использовать для запуска срочных

задач за счет пространства заданной методики из черного списка.

Режимы планирования задаются в секции sched конфигурационного файла, пример такой конфигурации представлен на рис. 4:

```
sched:
  mode: day # установка дневного режима

sched:
  mode: night # установка ночного режима

sched:
  stop # установка признака прекращения планирования

sched:
  exclusive
exclusive:
  -(user:User1) # установка разрешения на выполнение
  задач только пользователя User1

sched:
  blacklist
ban:
  -(user:User1) # установка на запрет выполнения задач
  пользователя User1;
```

Рис. 4. Варианты выбора режима планирования в конфигурационном файле
Fig. 4. Scheduling mode variants in configuration file

Конфигурация, представленная на рис. 5, устанавливает чередование дневного и ночного режимов, днем считается промежуток времени от 7 до 19 часов в рабочие дни:

```
sched:
  mode: normal
  worktime: "07:00-19:00" #время работы в режиме встроенного
  планировщика (формат ХХ:ХХ-ХХ:ХХ)
  workdays: [2,3,4,5,6] #рабочие дни недели (первый день недели
  ВС)
```

Рис. 5. Пример раздела sched конфигурационного файла Priority
Fig. 5. Example of the sched section in the configuration file Priority

7.2.3 Правила именования задач и задание информации о начислении ресурсов

Имя задачи является одним из дополнительных источников информации и должно соответствовать шаблону МЕТОДИКА.ЗАДАВА.ТИП[.ОПИСАНИЕ].

Поле «МЕТОДИКА» должно содержать имя «методики», известной Slurm.

Поле «ЗАДАВА» должно содержать числовой номер задачи и ее варианта.

Поле «ТИП» может быть одним из представленных ниже:

- П – производственная;
- М – методическая;
- Т – тестовая;
- Ф – фоновая;
- Н – невытесняемая.

Поле «ОПИСАНИЕ» может содержать произвольную дополнительную информацию о задаче (в том числе ту, которую можно дополнительно применять при расчете приоритетов или принятия решения об исключительности задачи).

Информация о начислении ресурсов, (т.е. как их регистрировать в статистике) задается в атрибуте задачи «ACCOUNT» по шаблону ТЕМА:ЗАКАЗЧИК, где «ТЕМА» - идентификатор темы, а «ЗАКАЗЧИК» – идентификатор заказчика. И то, и другое содержится в отдельных файлах.

Указанная информация обрабатывается плагином Submit. По информации, заданной в секции submit конфигурационного файла плагинов, этот плагин проверяет правильность заданной информации и, при необходимости, отвергает задачу или формирует недостающие атрибуты задачи или перепределяет заданные пользователем атрибуты на указанные администратором.

7.2.4 Группы задач

Множество параметров, используемых при планировании, применяются к группам задач, объединенных общими признаками. Для выделения требуемой группы из всего множества задач применяются фильтры. Каждый фильтр сравнивает один из атрибутов задачи с заданным значением. Фильтры применяются последовательно, в порядке их написания в конфигурационном файле.

```
name: '*.*.П.*'  
group: Grp1
```

Рис. 6. Пример фильтра группы задач

Fig.6. Example of a task group filter

Например, фильтр, представленный на рис. 6 выбирает все производственные задачи группы «Grp1», а фильтр, представленный на рис., выбирает все производственные задачи методики «МЕТОДИКА1» пользователя «User1».

```
name: 'МЕТОДИКА1.*.П.*'  
user: User1
```

Рис. 7. Пример фильтра группы задач

Fig. 7. Example of a task group filter

7.2.5 Распределение ресурсов

Для каждой группы задач выделяется квота на использование ресурсов. Основным ресурсом ВВС являются узлы вычислительного поля, поэтому для групп задач указывается максимально допустимое число узлов в каждом разделе. Ресурсы вычислительного комплекса делятся между методиками таким образом, чтобы небольшая часть ресурсов оставалась свободной для отладочных и интерактивных задач. Стандартные задачи могут использовать только выделенную группе задач квоту. Сверх этой квоты могут существовать только срочные или фоновые задачи. Таким образом, у каждой группы задач есть свое выделенное количество серверов, на которые могут претендовать только задачи этой группы (за исключением срочных и фоновых задач).

```
sched:  
  preempt:  
    limits:  
      - {key: 'M1-PL', name: 'Методика1', use: PLMEM, limit: 80}  
      - {key: 'M1-PS', name: 'Методика1', use: PSMEM, limit: 20}  
      - {key: 'M2-PS', name: 'Методика2', use: PSMEM, limit: 60}
```

Рис. 8. Пример задания квот в файле конфигурации

Fig. 8. Example of setting quotas in a configuration file

Квоты задаются в секции sched.preempt.limits конфигурационного файла. Например, то, что представлено на рис. 8, указывает, что задачи методики 1 могут использовать в разделе PLMEM одновременно все 80 узлов, а в разделе PSMEM не более 20; методика 2 к разделу PLMEM вообще не допущена, а в разделе PSMEM может использовать не более 60 узлов. В этом примере используются фильтры по имени задачи и заказанного раздела.

Другой пример, где вместо методик используется фильтр по группам, представлен на рис. 9.

```
sched:
preempt:
limits:
- {key: 'G1-PL', group: 'Grp1', use: PLMEM, limit: 80}
- {key: 'G1-PS', group: 'Grp1', use: PSMEM, limit: 20}
- {key: 'G2-PS', group: 'Grp2', use: PSMEM, limit: 60}
```

Рис. 9. Пример задания квот по группам в файле конфигурации

Fig. 9. Group quota example

Установленные квоты снабжаются уникальными идентификаторами (key). Текущее значение установленных и использованных квот можно в любой момент посмотреть утилитой qinfo, как представлено на рис. 10.

```
$ qinfo
Режим           : ночной (backfill)
Квоты           : включено
Вытеснение      : включено
Свободно ВУ     : 25
Свободно PLMEM  : 20
Свободно PSMEM  : 5
M1-PL           : ограничение 80 использовано 60 доступно 20
M2-PS           : ограничение 20 использовано 20 доступно 0
M2-PS           : ограничение 60 использовано 55 доступно 5
$
```

Рис. 10. Использование утилиты qinfo для просмотра информации о квотах

Fig. 10. Utility qinfo output example

7.2.6 Администрирование групп задач

Базовая версия Slurm не предоставляет пользователем никаких инструментов для управления задачами других пользователей. Пользователь может вмешиваться в работу Slurm только двумя способами:

- удалять или перезапускать свои задачи;
- менять атрибуты своих задач командой scontrol update job, косвенно влияя на приоритет задачи, задавая атрибут «nice» или переопределяя тип задачи в ее имени.

Однако выделение групп задач в отдельную категорию обрабатываемых плагинами данных и появление множества атрибутов, влияющих на порядок запуска задач, позволяет выделить отдельную категорию пользователей, которым можно делегировать модификацию этих атрибутов и частичного управления порядком запуска в пределах своей группы задач.

Таким образом, введено понятие «администратор группы задач» или «администратор методики». Список администраторов задается в секции adm конфигурационного файла. Например, так, как представлено на рис. 11.

```
adm:
- name: 'МЕТОДИКА1.*.*'
  admins:
  - {'User1'}
- name: 'МЕТОДИКА2.*.*'
  admins:
  - {'User2'}
```

Рис. 11. Пример задания функций администратора в конфигурационном файле ПО Slurm-ВНИИТФ

Fig. 11. Administrator function example in the Slurm-VNIITF configuration file

Приведенная на рис. 11 запись показывает, что пользователь «User1» является администратором методики «МЕТОДИКА1», а пользователь «User2» – администратором методики «МЕТОДИКА2». Плагин Submit на основании этой информации разрешает выполнение команды scontrol update job указанным пользователям для всех задач их групп. Таким образом, администратор группы задач может менять тип и приоритет задач в пределах своей группы.

7.2.7 Вытеснение задач

При появлении в очереди задачи, для которой не хватает ресурсов, может быть предпринята попытка освобождения ресурсов для нее путем перезапуска находящихся в счете менее приоритетных задач, необъявленных невытесняемыми.

Существует две стратегии вытеснения задач, в зависимости от типа задачи.

При появлении в очереди срочной задачи кандидаты на вытеснение ищутся в следующем порядке:

- сначала рассматриваются фоновые задачи, входящие в ту же группу, что и срочная,
- затем рассматриваются прочие фоновые задачи,
- в последнюю очередь рассматриваются стандартные задачи из своей группы.

При появлении в очереди стандартной задачи, укладываемой в квоты группы, кандидаты на вытеснение ищутся из числа фоновых задач других групп, превысивших лимиты.

В других случаях вытеснение не производится, и задача ждет освобождения достаточных для ее запуска ресурсов.

После нахождения подходящих задач производится попытка их перезапуска. Выбранные задачи должны сохранить свое состояние и завершиться, перейдя в состояние ожидания запуска. Поскольку срочная задача имеет наивысший приоритет, освободившиеся в результате ресурсы будут предоставлены ей. Позднее перезапущенные задачи будут снова автоматически запущены.

Предположим, что в конфигурационном файле задана следующая информация, представленная на рис. 12.

```
sched:
  preempt:
    urgent:
      - {name: 'МЕТОДИКА1.12345.П*'}
    background:
      - {name: '*.*.Ф*'}
    uncancelable:
      - {name: '*.*.Н*'}
```

Рис. 12. Пример задания фильтра вытеснения задач в конфигурационном файле ПО Slurm-VNIITF
Fig. 12. Task preemption filter example in the Slurm-VNIITF software configuration file

Тогда после появления в очереди задачи с именем «МЕТОДИКА1.12345.П.Расчет...», которая будет распознана как срочная, из списка фоновых задач (содержащих букву «Ф» в третьем слове имени), находящихся в состоянии выполнения, но не являющихся невытесняемыми (т.е. в третьем слове имени нет буквы «Н»), будут выбраны подходящие по размеру задачи. Если такие задачи будут найдены, им посылается сигнал о прекращении работы, получив который, фоновые задачи должны сохранить свое состояние и завершиться. После завершения, фоновые задачи остаются в очереди в состоянии ожидания запуска, а освобожденные ими ресурсы будут предоставлены срочной задаче.

Важно отметить еще один аспект. Иногда, на первый взгляд, достаточно подобрать для вытеснения такое множество фоновых задач, чтобы занимаемые ими ресурсы были минимально достаточными для счета высокоприоритетной задачи. Однако, стоимость возобновления счета фоновых задач разная, и, как правило, заранее известна для конкретной методики. Например, если фоновой задаче до ее завершения осталось времени меньше, чем потребуется на её перезапуск (время сохранения промежуточных данных расчетов и время считывания данных контрольной точки при запуске задачи существенны), то решение о перезапуске может быть нерациональным. В этом случае оптимальным решением будет отложить запуск срочной задачи до завершения расчета фоновой или выбрать другую задачу для вытеснения. Кроме того, фоновые задачи в очереди могут сильно отличаться по размеру запрашиваемых ими ресурсов. В этом случае время ожидания для задач большего размера растет нелинейно, так как вероятность освобождения малого количества вычислительных

ресурсов на ВВС во много раз больше, чем больших. Это значит, что предпочтительней выбирать для вытеснения несколько небольших задач, чем одну задачу, занимающую необходимые вычислительные ресурсы. Множество учитываемых при планировании очереди задач факторов также должно содержать количество перезапусков задачи, близость времени последнего перезапуска, и поправку к приоритету, которая может быть установлена администратором методики, самим пользователем, запускающим задачу или администратором ВВС.

Таким образом, в результате всех исследований, авторами публикации был разработан некий аналитический подход, когда для каждой вытесняющей задачи строится множество вытесняемых задач J_{ptee} , занимающих ресурсы, подходящие вытесняющей, как показано на рис. 13. Из этого множества формируется набор, элементами которого являются подмножества различных сочетаний вытесняемых задач MJ_{ptee} таких, что используемых ими суммарных ресурсов достаточно для запуска вытесняющей. Затем, для каждого элемента набора вычисляется функция стоимости вытеснения этого подмножества задач. Для вытеснения выбирается тот элемент набора, для которого оценочная функция возвращает минимальное значение.

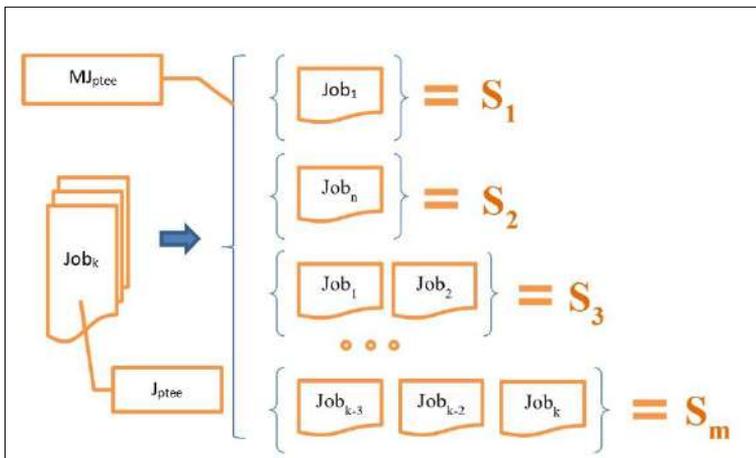


Рис. 13. Схема определения множества вытесняемых задач
Fig. 13. Preemptible task set definition scheme

В результате выбор множества вытесняемых задач для перезапуска сводится к минимизации функции стоимости вытеснения множества задач на интервал планирования Δt :

$$M = \min\{S_1, S_2, \dots, S_m\},$$

Здесь S_m – это вычисляемая на основе набора весовых функций и их коэффициентов функция стоимости вытеснения m -ого набора вытесняемых задач:

$$S_m = \sum_{i=1}^n (k_1 Tend_i + k_2 Tr_i + k_3 Tcalc_i + k_4 N_i + k_5 R_i + k_6 S_i + k_7 A_i),$$

где:

n – количество вытесняемых задач в подмножестве (т.е. рассматриваемом элементе набора);

$k_1 \dots k_7$ – конфигурируемые коэффициенты факторов стоимости;

$Tend_i, Tr_i, Tcalc_i, N_i, R_i, S_i, A_i$ – функции факторов стоимости.

Рассмотрим подробнее каждую из весовых функций факторов стоимости.

$Tend_i$ – весовая функция времени завершения счета задачи:

$$Tend_i = c_1 \left(\frac{t_i - now}{\Delta t} \right),$$

где:

c_1 – переменный показатель степенной функции (0,5...0,6);

t_i – предполагаемое время завершения задачи;

now – текущее время;

Δt – интервал вытеснения.

Tr_i – весовая функция времени последнего перезапуска задачи:

$$Tr_i = c_2 \frac{now - tr_i}{\Delta t},$$

где:

c_2 – переменный показатель степенной функции (0,2...0,3);

tr_i – время последнего перезапуска задачи;

now – текущее время;

Δt – интервал вытеснения.

$Tcalc_i$ – весовая функция полного времени счета задачи:

$$Tcalc_i = c_3 \frac{now - tcalc_i}{\Delta t},$$

где:

c_3 – переменный показатель степенной функции (0,2...0,3);

$tcalc_i$ – предполагаемое полное время счета задачи;

now – текущее время;

Δt – интервал вытеснения.

N_i – весовая функция количества перезапусков задачи:

$$N_i = 1 - \frac{1}{2} r_i^{c_i},$$

где:

r_i – количество перезапусков задачи в процессе счета.

R_i – весовая функция возможности перезапуска задачи:

$$R_i = \begin{cases} r_i = 1 \\ r_i = 0 \end{cases}$$

где:

r_i – возможность перезапуска, определяемая пользователем, администратором методики, администратором ВВС.

S_i – весовая функция размера задачи:

$$S_i = 1 - c_4^{s_i},$$

где:

c_4 – переменный показатель асимптотической функции (0,996...0,997);

s_i – размер задачи в вычислительных узлах.

A_i – весовая функция поправки приоритета.

$$A_i = 1 - c_5^{n_i},$$

где:

c_5 – переменный показатель асимптотической функции (0,993...0,995);

n_i – поправка к приоритету, задаваемая администратором методики или пользователем.

Коэффициенты $k_1 \dots k_7$ так же, как и показатели функций $c_1 \dots c_4$, представляют собой значения, которые определяются эмпирически, на основе большого набора статистических данных и зависят в свою очередь от размерности ВВС, количества просчитываемых на ней ежегодно задач, многообразия типов задач.

8. Заключение

Использование разработанной в РФЯЦ-ВНИИТФ версии ПО Slurm-ВНИИТФ позволило существенно расширить возможности проведения расчетов на ВВС, увеличить эффективность использования имеющихся вычислительных ресурсов. Благодаря ПО Slurm-ВНИИТФ удалось увеличить ежегодное количество просчитываемых задач примерно на 20%. В то же время набор реализованных функций и удобный, интуитивно понятный интерфейс по их настройке, существенно упростили работу администраторов ВВС, позволив им оперативно влиять на очередь задач в зависимости от текущих потребностей в расчетах различных подразделений института.

Результаты данной работы носят, безусловно, достаточно общий и субъективный характер и отражают один из возможных подходов к организации проведения расчетов задач численного моделирования на ВВС. Тем не менее, авторы считают, что опубликованная ими работа может быть полезна специалистам, занимающимся разработкой специализированного ПО для подсистем управления ВВС. Авторы намерены продолжить публикации по данной тематике, с целью представления более развернутой информации по каждой из функциональных подсистем ВВС, основным аспектам и особенностям администрирования ВВС, особенностям разработки прикладного и системного ПО для ВВС, предназначенных для решения задач численного моделирования.

Список литературы / References

- [1] Игнатьев А.О., Мокшин С.Ю. Типовая архитектура высокопроизводительной вычислительной системы для решения задач численного моделирования, Препринт РФЯЦ-ВНИИТФ № 265, Снежинск, 2020 г., 21 с. / Ignatyev A.O., Mokshin S.Yu. Base architecture of the mathematical modelling HPC system, Preprint FSUE «RFNC-VNIITF named after Academ. E.I. Zababakhin» № 265, Snezhinsk, 2020, 21 p. (in Russian).
- [2] «СПО Супер-ЭВМ», Available at: <http://vniitf.ru/article/spo-super-evm>, accessed 01.04.2022 (in Russian).
- [3] Slurm workload manager, Available at: <https://slurm.schedmd.com/documentation.html>, accessed 01.04.2022.
- [4] MPI: The Message Passing Interface. Available at: http://parallel.ru/tech/tech_dev/mpi.html, accessed 01.06.2020.
- [5] The OpenMP API specification for parallel programming. Available at: <https://www.openmp.org/>, accessed 01.04.2022.
- [6] Maui Scheduler, Available at: <https://github.com/TempleHPC/maui-scheduler>, accessed 01.04.2022.
- [7] Moab Cluster Suite, Available at: <https://adaptivecomputing.com/moab-hpc-suite/>, accessed 01.04.2022.
- [8] Torque Resource Manager, Available at: <https://adaptivecomputing.com/cherry-services/torque-resource-manager/>, accessed 01.04.2022.
- [9] YAMML, Available at: <http://yaml.org/>, accessed 01.04.2022.

Информация об авторах / Information about authors

Алексей Олегович ИГНАТЬЕВ – начальник лаборатории. Сфера научных интересов: проектирование вычислительных систем, разработка параллельных программ численного моделирования, разработка операционных систем, методы и средства защиты информации.

Alexey Olegovich IGNATYEV – Head of Laboratory. Research interests: design of supercomputer systems, parallel numerical simulation programs development, operating systems development, methods and means of information security.

Алексей Алексеевич КАЛИНИН – начальник группы. Сфера научных интересов: проектирование вычислительных систем, разработка параллельных программ численного моделирования, разработка операционных систем, разработка компиляторов, методы оптимизации планирования вычислений в параллельных системах, методы формальной верификации алгоритмов.

Alexey Alexeevich KALININ – Head of Research Group. Research interests: design of supercomputer systems, parallel numerical simulation programs development, operating systems development, compilers and debuggers development, task management optimization for HPC system, formal verification of algorithms methods.

Сергей Юрьевич МОКШИН – начальник отдела. Сфера научных интересов: проектирование вычислительных систем, разработка функциональных подсистем для высокопроизводительных вычислительных систем, разработка операционных систем, методы и средства защиты информации.

Sergey Yurievich MOKSHIN – Head of Department. Research interests: design of supercomputer systems, development of functional subsystems for high performance supercomputing systems, operating systems development, methods and means of information security.

DOI: 10.15514/ISPRAS-2022-34(2)-14



A Web Application to Promote Blood Donation in Russia

P.A. Smirnov, ORCID: 0000-0002-1699-2281 <smpashts@gmail.com>
V.V. Malinovskaya, ORCID: 0000-0003-3647-6468 <vera8malinovskaya@gmail.com>
N.V. Voinov, ORCID: 0000-0002-0140-1178 <voinov@ics2.ecd.spbstu.ru>

*Peter the Great St.Petersburg Polytechnic University,
29, Polytechnicheskaya, St Petersburg, 195251, Russia*

Abstract. The paper is devoted to the issue of blood donation and possible ways to promote this activity using modern information technologies. Existing software solutions are analyzed and new Web application is proposed to implement all features required for potential blood donors to make this process clearer and more comfortable.

Keywords: blood donation; software requirements; web application; client-server architecture

For citation: Smirnov P.A, Malinovskaya V.V., Voinov N.V. A Web Application to Promote Blood Donation in Russia. Trudy ISP RAN/Proc. ISP RAS, vol. 34, issue 2, 2022, pp. 179-190. DOI: 10.15514/ISPRAS–2022–34(2)–14

Веб-приложение для продвижения донорства крови в России

П.А. Смирнов, ORCID: 0000-0002-1699-2281 <smpashts@gmail.com>
В.В. Малиновская, ORCID: 0000-0003-3647-6468 <vera8malinovskaya@gmail.com>
Н.В. Воинов, ORCID: 0000-0002-0140-1178 <voinov@ics2.ecd.spbstu.ru>

*Санкт-Петербургский Политехнический университет Петра Великого,
Россия, 195251, Санкт-Петербург, Политехническая ул., 29*

Аннотация. Статья посвящена проблеме донорства крови и возможным путям популяризации этой деятельности с использованием современных информационных технологий. Анализируются существующие программные решения и предлагается новое веб-приложение, реализующее все функции, необходимые потенциальным донорам крови, чтобы сделать этот процесс более понятным и комфортным.

Ключевые слова: донорство крови; требования к программному обеспечению; веб-приложение; клиент-серверная архитектура

Для цитирования: Смирнов П.А., Малиновская В.В., Воинов Н.В. Веб-приложение для продвижения донорства крови в России. Труды ИСП РАН, том 34, вып. 2, 2022 г., стр. 179-190. DOI: 10.15514/ISPRAS–2022–34(2)–14

1. Introduction

Gratuitous blood donation is a very important phenomenon for maintaining public health. Transfusion of blood and its components is required in many situations, such as surgery and emergencies. Many medical interventions would not be possible if voluntary blood donors did not exist.

Today, blood transfusion services around the world face two main challenges. The first of these is a guarantee of the quality and safety of blood. Quality control of donated blood allows medical professionals to be sure that in case of an emergency, the recipient of blood will not be harmed. The second important task is to ensure sufficient blood supplies. In spite of the fact that the first problem has been solved in recent decades, the problem of the shortage of donor blood is still acute in Russia. Today, according to the National Medical Research Center for Hematology [1], only 1.7% of Russians donate blood and its components [2]. However, to ensure a sufficient stock of blood, it is necessary to ensure at least 4% of the population to be donors.

To popularize voluntary blood donation among the population, it is necessary to make this process comfortable and pleasant. While the process of blood sampling itself is the business of medical professionals, it makes sense to provide all the previous stages, for example, searching for a blood donation point, registering for a donation, or filling out questionnaires on contraindications, using online services that allow the donor to quickly go through all these stages.

Another important task is to ensure the regularity of donations from each donor with an optimal interval between two donations of 3-6 months [3]. It is very important that the donor comes for the next donation exactly after this period, then his newly donated blood will be checked for infections therefore the old blood will be removed from quarantine and available for transfusion. To achieve this goal, the integration of the service with SMS or e-mail reminders about the opportunity to donate blood again can help.

Another problem is the suspension of a donor due to temporary contraindications (for example, vaccination, or a recent infectious disease). The solution to this problem could also be automatic reminders via SMS or e-mail after the expiration of the suspension of the donor.

In addition, today in Russia there is no possibility of universal registration at any donor center. To sign up for a donation, you first need to independently find the nearest blood donation point on the Internet, then find its website or a service for registration at this point and fill out an application. Consequently, there is an urgent need to create a single donation registration service.

Outside of Russia, there are applications that can solve some of these problems, but in all applications available on the Russian market, there is no functionality that is critical for comfortable interaction between donors and the blood service.

The main idea of this work is to develop an application that could be used in the Russian market and would enable donors to interact with blood donation points easily and conveniently. The creation of such application, considering successful foreign analogues, could, to varying degrees, solve the above problems, make the process of voluntary blood donation comfortable, and, possibly, attract new volunteers.

2. Current issues in the scope of blood donation

In this section several relevant works are reviewed to identify the most important requirements that an application should meet.

2.1 Global guide towards voluntary blood donation

The problem of an insufficient number of donors is acute in different parts of the world. To help governments move towards solving this problem, the World Health Organization [4], together with the Red Cross, has developed a global action plan to expand blood donation in countries around the world [5]. This publication outlines 20 strategies to help countries promote voluntary blood donation. Some of these strategies are of interest in this research work.

Strategy 2. Establish a national voluntary blood donation program.

Among other things, this strategy recommends raising donor awareness, and one of the steps to achieve this goal is to use communication tools for donors and blood services. Today in Russia there

is no convenient and universal way to communicate with donor points, but the proposed application will create the ability to quickly correspond with any blood donation point using chat.

Also, this strategy proposes to store data on temporary or permanent contraindications of donors, successful donations, and test results in a centralized database, in order to then use this information to alert donors.

Strategy 4. Understand blood donors.

This strategy proposes to explore the factors that positively or negatively influence the motivation of donors, as well as the things that deter people from voluntary blood donation. In these terms it is proposed to create a section in which donors can leave feedback about the donation, as well as about the blood donation point itself.

Strategy 13. Keep in touch with infrequent, inactive, and suspended donors.

This strategy states that donors who donate blood infrequently, or for some reason have stopped donating blood, are one of the target groups to attract donations. It is mentioned that returning former donors to blood donations is much less laborious than attracting new ones. Inactive donors may simply need to be reminded of the importance of their donations and invited to a new donation visit. Therefore, it makes sense to create an e-mail notification system for less active donors, and donors suspended due to temporary contraindications.

The next step is to segment the donor data according to variables such as the length of time since the last donation, the place of donation, and the total number of donations. This will make it possible to identify priority donors who have left; for example, it will be easier to reactivate a donor who last visited a donor site two years ago than one who donated blood ten years ago.

Strategy 14. Retain regular blood donors.

This strategy addresses the various motivations for voluntary blood donation and suggests:

- Give donors access to their donation history, it is important that it is updated every time they donate blood.
- Establish an easy appointment system, encourage donors to plan their next visit, and send a reminder in advance of the appointment using their preferred communication method (e.g., email, calls, or SMS).
- Identify donors with rare blood types and educate them about the importance of their blood type. Also motivate donors to donate blood during holiday periods or in critical situations when the number of donors is expected to be low.
- Conduct donor surveys to assess the quality of the blood transfusion service. Give donors the opportunity to leave feedback and suggestions for improving the blood donation process.

Strategy 16. Make blood donation convenient for donors.

Addressing issues that donors consider important goes a long way in strengthening donor loyalty and encouraging more regular donations. This strategy proposes:

- Optimize the donation process and donor flow by making the most efficient use of time.
- Establish appointments system to prevent delays and ensure a stable and manageable flow of donors throughout the donation process.

2.2 MHealth technology

The authors of the paper claim that only 1.8% of the world's population are blood donors, while the need for health systems can be met when 3 to 5% of people voluntarily donate blood.

In [6] the problem of increasing and maintaining the number of blood donors is considered. The authors of the paper believe that the use of mobile technologies in healthcare can increase the number of blood donors. To meet this need, a qualitative study was conducted to determine the availability of applications that support the activities of the Brazilian blood donation departments. As a result of

this study, a list of requirements was proposed that an application must meet to be effective in recruiting blood donors, the most interesting of which are presented below:

- Registration of donors.
- Ability to schedule a donation.
- Identify contraindications to donation and inform the donor about them.
- Calculate and save the date of the next donation.
- Show the donor the closest blood donation point.
- Make an invitation to the donor to donate blood.
- Allow the donor to send invitations to friends to become a donor.
- Ensure constant communication with the donor via SMS messages.
- Allow posting infomercials and feedbacks in the app.
- Use gamification in the app (points, badges, etc.)
- Provide the donor with blood test results after donation.

Another useful result of this study was anonymous statistics on the popularity of using various functionalities. By collecting anonymous statistics about users of donor applications, it was found that the most popular functions are searching for information about blood donation, determining the nearest blood point, making an appointment to donate blood.

2.3 Russian specifics

Donors' blood scarcity is a major challenge in Russia nowadays. In [7] author investigates peculiarities of donor practices in Russia and considers obstacles to the dissemination of blood donation as well as motivation to donate blood.

It is noted that one of the reasons for the low involvement of Russians in blood donation is the lack of awareness about the process itself. Often, the reluctance to donate blood is associated with concern for health, for example, it is still widely believed that during donation there is a high risk of contracting infections. Therefore, potential donors should have access to accurate information and have at least a basic understanding of the field of blood donation.

Moreover, one of the obstacles is the laboriousness of the donation procedure. The practice of blood donation can be difficult to integrate into everyday life since the process itself is quite time-consuming. So, it is necessary to direct efforts to reduce the time spent by the donor at the transfusion point.

Apart from altruism, the author considers the opportunity to receive material rewards and benefits that are provided to all donors in Russia to be the major incentives towards blood donations. Consequently, it is necessary to inform both potential and active donors about benefits provided for blood donation.

All in all, the features of donor practice in Russia were highlighted, which are important to take into consideration while working towards the promotion of donation.

3. Overview of existing solutions

Available Russian and foreign applications for donors are analyzed and summarized in Table 1 with the following numbering:

- 1) yadonor.ru (Russian blood service website) [8].
- 2) eliz-spb.ru/otdelenie-perelivaniya-krovi (blood transfusion point website) [9].
- 3) RedCrossBlood.org (American Red Cross blood services website) [10].
- 4) donorsearch.org (Russian donor community website) [11].
- 5) Proposed solution.

Table 1. Analysis of existing solutions

Requirement	1	2	3	4	5
Donor registration	+	-	+	+	+
Finding the nearest point (with map)	-	-	+	-	+
Questionnaire with temporary contraindications	-	-	+	-	+
Registration for blood donation	-	+	+	-	+
Reminders for upcoming donations and opportunities to donate blood	+	-	+	+	+
Donor traffic light	+	+	-	+	+
Sorting donor points by donor traffic light ¹	-	-	-	+	+
Feedbacks	+	+	-	-	+
Chat with donor point	-	-	-	-	+
Donor point news	+	-	-	-	+
Viewing your donations	+	-	+	+	+
Countdown to the title of “Honorary Donor”	+	-	-	+	+
Analysis statistics with graphs	-	-	-	-	+
Intuitive interface	-	+	+	+	+
Russian language support	+	+	-	+	+
Registry of donors of this site with smart filtering (function for site administrator)	-	-	-	+	+
Countdown to the title of “Honorary Donor” ²	+	-	-	+	+
Analysis statistics	-	-	-	-	+
Intuitive interface	-	+	+	+	+
Russian language support	+	+	-	+	+



Fig. 1. Donor Traffic Light

3.1 yadonor.ru

The site *yadonor.ru* is the largest website for donors in Russia, it is the official website of the Blood Service. This site provides for the possibility of registering donors, allows you to view the donor traffic lights of different blood donation points, and also makes it possible to leave feedback on

¹ *The Donor Traffic Light* is a system that aims to create an optimal supply of donated blood and alert donors to the need for blood components at a specific blood transfusion point (Fig.1). It is used throughout Russia, as well as in countries of the Commonwealth of Independent State [12].

² *Honorary Donor of Russia* is a badge which expresses recognition by the state of a citizen's contribution to the development of blood donation in Russia. It also allows the donor to receive some privileges and is a way to encourage donors in Russia [15].

donations. There is also a convenient section with donation statistics and an interactive donation counter showing the required number of donations to receive the title of Honorary Donor of Russia. The main advantage of this application, of course, is the support of the Russian language and the ability to receive data on donations from Russian blood donation points using unique donation codes. But a very serious drawback of this web-application is the inability to register for a donation, and as mentioned above, the most popular function of sites for donors is the ability to sign up at any convenient blood donation point. In addition, this site has a rather confusing and non-intuitive interface. However, interface intuitiveness to the end user affects communicability, understandability, user satisfaction and these subsequently affect on system's usability [13].

3.2 Websites of Russian blood transfusion points

The only way to register for blood donation in Russia is through separate websites of blood transfusion points. Since the main functionality of these sites is limited and very similar, only one specific point is considered - the site of the transfusion department of the Elizabethan Hospital in St. Petersburg. This site allows you to view the donor traffic light of this point, find out the working hours, read the news of the change point. The sites of transfusion points do not provide other functionality, they do not allow donors to register, they do not allow receiving reminders of upcoming blood donations, and they do not allow contacting the staff of the blood collection point.

3.3 RedCrossBlood.org

Of all the considered analogs of the developed application, the RedCrossBlood WEB-application of the Red Cross public charitable organization, available in the USA and some other countries of the world, has the most complete functionality. This application allows users to conveniently search for blood donation points by location, allows them to register for a donation at any convenient point, manage upcoming donations, and view donation statistics in a convenient way.

The most interesting and important function of this application is the RapidPass technology [14], which offers donors to answer questions about their health when registering for a donation, thereby either immediately learning about temporary contraindications, or simply reducing the time spent at the blood transfusion point. Another useful feature of this application is the ability to view data on blood tests and thus monitor the dynamics of donors' health.

3.4 donorsearch.org

DonorSearch is another Russian-language website for donors. This web-application is quite different from the previously reviewed analogues and rather resembles a social network for donors.

Just like yadonor it does not provide an opportunity to register for a donation, but it has a more intuitive interface, it also allows donors to independently keep a record of their donations and makes it possible to independently plan future donations. In addition, this site provides access to the donor traffic lights of various donation points and sends you reminders of upcoming donations.

4. Proposed solution

In this work, it is proposed to develop a Web application focused on donors in Russia.

The application users should be able to get information about blood donation points in their city: opening hours, the ability to donate plasma, current news, and a donor traffic light. In addition, the possibility of filtering by distance from the current location and/or donor traffic light should be implemented.

Feedbacks can help donors to choose a blood donation point, on the other hand, point administration can receive feedback and suggestions on how to improve the blood donation process at a particular point and make blood donation as convenient as possible for donors.

The feed section will help transfusion centers to get the important information out to donors, as well as to draw more donors. This section is expected to include information such as the introduction of additional blood donation dates, emergency and critical blood need alerts, and additional donor incentives such as Donor Day, which will help maintain donor loyalty and serve as an additional motivation to donate.

Application user requirements and roles:

1) Donor.

After registration, the donor can specify personal information, including the preferred method of communication.

On the page of a particular donor point, the user will be able to view information about the point, current news, leave a review and read the reviews of other donors. In addition, the user can ask any questions to the employees of the donor center in the chat.

Maps to display blood donation points and find the nearest point will be implemented via 2GIS API [17].

When registering for an appointment, donor is invited to fill out a blood donor questionnaire in order to identify temporary contraindications, as well as speed up the process of donating blood. The introduction of such a system has a number of advantages: it reduces the time that donors spend on donating blood up to 15 minutes; donors who pre-filled the questionnaire online are much more likely to show up to donate blood than donors who do not.

In case the donor is temporarily suspended from donating blood, it will be proposed to set a reminder that will be sent to the mail after the expiration of the contraindications. Also, the user will be able to enable notifications with an invitation to donate blood after the minimum interval between donations.

In the personal account, an authorized user will be able to see the history of his donations, namely the number of donations made with detailed information about each of them. Before each donation, a blood test of the donor is carried out, so the results of the tests will also be available in the personal account. The results will be presented both as a table and as a graph, which will allow donors to track the trends in their health. Also, the personal account will display the number of blood and plasma donations that must be made before receiving the title of “Honorary Donor”. Honorary Donor of Russia is a badge which expresses recognition by the state of a citizen's contribution to the development of blood donation in Russia. It also allows the donor to receive some privileges and is a way to encourage donors in Russia.

2) Administrator.

The administrator should be able to view the registry of donors and filter the data by the time elapsed since the last donation, place of donation, or total number of donations.

In addition to viewing a list of all donors, the administrator can view the personal information of a particular donor and find out his contact details with the specified preferred method of communication.

It is also possible to keep in touch with donors directly on the site: view chat messages and reviews on the page of the blood transfusion point and respond to them. In addition, the administrator can update the donor traffic light, publish, delete, and edit the news of his point.

5. Client-server architecture

To implement the application, a three-tier client-server architecture was chosen, which implies the presence of three types of components in the application: the application server, the database server, and the client application (fig.2).

5.1 Server-side implementation

The Java programming language and the Spring Framework [18] were chosen to create the application server.

The server of the developed product itself is a Spring Boot [19] application that makes it easy to use all the necessary layers of the Spring Framework.

The interaction between the application server and the client application is proposed to be organized using the REST API [20] provided by the server. REST (English Representational State Transfer) is an architectural style of interaction between the components of a distributed application. It usually represents a GET, POST, UPDATE and DELETE HTTP requests. The data exchange format is JSON.

To store all the information, the PostgreSQL database management system [21] was chosen. The database server itself is planned to be launched using the appropriate Docker container for development convenience.

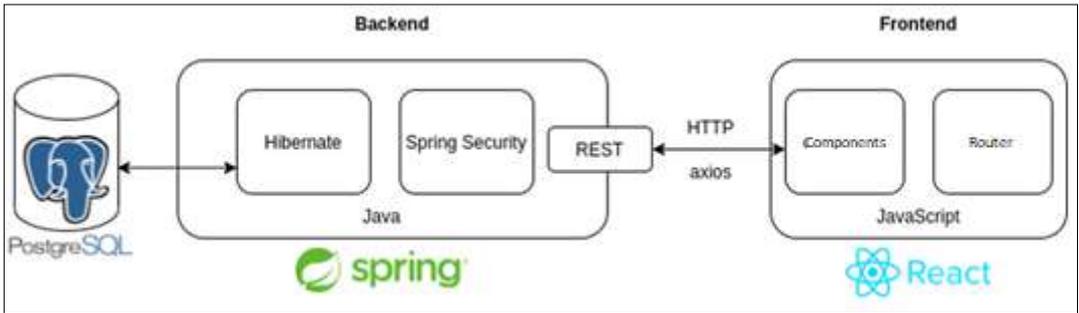


Fig. 2. Software architecture

The four main types of application server components are entities, repositories, services, and controllers. Entities describe the objects stored in the database, and the database itself is accessed using JPA repositories. One of the Spring modules (Spring Data JPA) [22] is used for this. All business logic is described in the service layer, and REST controllers are created to provide a strict REST API.

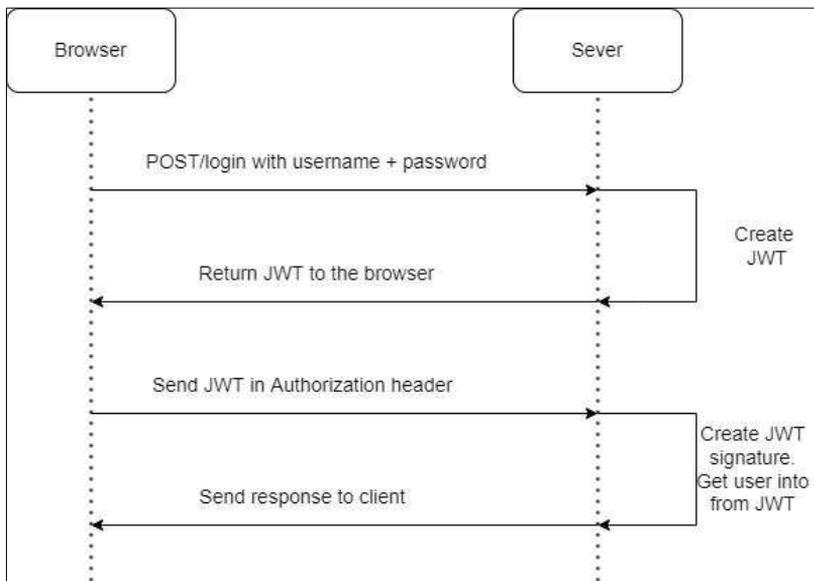


Fig. 3. JWT authorization scheme

Secure access to REST API is organized using another Spring module – Spring Security. JWT (Json Web Token) [23] is used to authorize system users. The scheme for working with JWT consists of the following steps:

- 1) The user contacts the endpoint for authorization.
- 2) The server checks authentication while attempting to login. If the username and password are correct, a JWT is generated, otherwise the user will be denied access.
- 3) The server returns a token in the form of json-a, which is stored in the local storage of the client application.
- 4) When a user tries to access a secure API, a JWT must be sent along with the request.

The server verifies the token, and if it's valid, it opens access to the API endpoint. Chat messaging is planned to be organized using WebSocket [24], a thin and lightweight layer over TCP, and the messaging protocol that is planned to be used is STOMP (Simple Text Oriented Message Protocol) [25].

5.2 Client-side implementation

The site is a responsive single page application (SPA). A single page application is a site that contains a single HTML document that updates dynamically and does not require a full page reload during use.

To develop the client side, the TypeScript [26] programming language and the React library [27] were chosen in combination with the Redux [28] library to save the state of the site when navigating through history. TypeScript extends the power of JavaScript by adding static typing.

As a Redux middleware for working with the REST API and WebSocket it is planned to use the Redux-Thunk module [29]. The react-router [30] library will be responsible for organizing routing to match requests to the application with certain interface components.

6. Conclusion

The problem of shortage of donor blood is still acute both in Russia and around the world. At the moment, this process is quite laborious and takes a lot of time for donors.

This paper explores ways to promote donation, attract new donors, and retain regular donors. Domestic and foreign applications for donors were reviewed, their advantages and disadvantages were analyzed, and a list of requirements that an application must meet was identified.

The proposed solution is designed to facilitate the process of donating blood for donors, as well as to popularize blood donation and ensure the regularity of donations. Such system would allow donors to quickly receive the necessary information, provide convenient registration for donations, as well as the ability to track health indicators. In addition, this system will help speed up the process of donating blood, provide blood banks with a large number of supplies, and, as a result, save many lives.

In the future, the developed system can be improved by expanding the functionality by adding the ability to record bone marrow donors. Also, it is planned to perform load testing of the Web application in real and peak load conditions because our implementation should remain stable with a large number of active users. Moreover, it is considered to implement mobile version of designed application.

References / Список литературы

- [1] National Medical Research Centre for Hematology official website. URL: <http://blood.ru/>.
- [2] National Medical Research Centre for Hematology official website, frequently asked questions. URL: <http://blood.ru/transfuziologiya-idonorstvo-krovi/donoru/chasto-zadavaemye-voprosy.html>.
- [3] D.J. Myers and R.A. Collins. Blood Donation. StatPearls Publishing, 2021. URL: <https://www.ncbi.nlm.nih.gov/books/NBK525967/>.

- [4] World Health Organization official website. URL: <https://www.who.int/>
- [5] World Health Organization & International Federation of Red Cross and Red Crescent Societies, Towards 100% Voluntary Blood Donation: A Global Framework for Action, 2010. URL: <https://www.ncbi.nlm.nih.gov/books/NBK305667/>.
- [6] J. Silva, C. César Praça Brasil et al. MHealth Technology as a Tool to Promote Blood Donation. In Proc. of the 11th International Joint Conference on Biomedical Engineering Systems and Technologies, 2018, pp. 471-477.
- [7] A.G. Orlovetskaya. Blood donation as a social practice in Russia. *The Journal of Social Policy Studies*, vol 15, no 1, 2017, pp. 7-20 (in Russian) / А.Г. Орловецкая. Донорство крови как социальная практика: российская специфика. *Журнал исследований социальной политики*, том 15, no. 1, 2017 г., стр. 7-20.
- [8] Russian blood service Web site. URL: <https://yadonor.ru/en/> / Российский Web-сайт службы крови. URL: <https://yadonor.ru/>.
- [9] St. Petersburg Federal State Budgetary Institution Elizabeth's Hospital web site, blood transfusion unit. URL: <https://elizspb.ru/otdelenie-perelivaniya-krovi> (in Russian) / Сайт Санкт-Петербургского ФГБУ «Елизаветинская больница», отделение переливания крови.
- [10] American Red Cross blood services website. URL: <https://www.redcrossblood.org/>
- [11] Russian donor community website. URL: <https://donorsearch.org/> (in Russian) / Сайт российского сообщества доноров.
- [12] St. Petersburg Blood Transfusion Station official website, donor traffic light: <http://yadonorspb.ru/svetofor/> (in Russian) / Городская станция переливания крови Санкт-Петербурга.
- [13] M. N. Islam. Towards Designing Users' Intuitive Web Interface. In Proc. of the Sixth International Conference on Complex, Intelligent, and Software Intensive Systems, 2012, pp. 513-518.
- [14] American Red Cross, RapidPass. URL: <https://www.redcrossblood.org/donate-blood/manage-my-donations/rapidpass.html>.
- [15] National Medical Research Centre for Hematology official website, Honorary Donor of Russia. URL: <http://blood.ru/transfuziologiya-idonorstvo-krovi/donoru/kak-stat-donorom/pochetnyj-donorossii.html> (in Russian) / Официальный сайт Национального медицинского исследовательского центра гематологии, Почетный донор России.
- [16] Digital dictionary, Rh factor, URL: <https://www.dictionary.com/browse/rh-factor>.
- [17] 2GIS maps API documentation. URL: <https://api.2gis.ru/doc/maps> (in Russian) / Документация по API карт 2ГИС
- [18] Spring Framework official website. URL: <https://spring.io/>.
- [19] Spring Boot project. URL: <https://spring.io/projects/spring-boot>.
- [20] REST APIs. URL: <https://www.ibm.com/cloud/learn/rest-apis>.
- [21] PostgreSQL database official website. URL: <https://www.postgresql.org/>.
- [22] Spring Data JPA. URL: <https://spring.io/projects/spring-data-jpa>.
- [23] Internet Engineering Task Force (IETF), JSON Web Token. URL: <https://datatracker.ietf.org/doc/html/rfc7519>.
- [24] Internet Engineering Task Force (IETF), The WebSocket Protocol URL: <https://datatracker.ietf.org/doc/html/rfc6455>.
- [25] The Simple Text Oriented Messaging Protocol. URL: <https://stomp.github.io/>.
- [26] TypeScript programming language official website. URL: <https://www.typescriptlang.org/>.
- [27] React library official website. URL: <https://reactjs.org/>.
- [28] Redux library official website. URL: <https://redux.js.org/>.
- [29] Redux Thunk middleware source code and documentation. URL: <https://github.com/reduxjs/redux-thunk>.
- [30] React Router official website. URL: <https://reactrouter.com/>.

Information about authors / Информация об авторах

Pavel Alekseevich SMIRNOV – software developer, specialist in software engineering, development and maintenance of web applications, bachelor.

Павел Алексеевич СМИРНОВ – разработчик программного обеспечения, специалист в области программной инженерии, разработки и сопровождения веб-приложений, бакалавр.

Vera Vladimirovna MALINOVSKAYA – software engineer specializing in web design and client-server architecture, bachelor.

Вера Владимировна МАЛИНОВСКАЯ – инженер по разработке программного обеспечения, специализирующийся на веб-дизайне и клиент-серверной архитектуре, бакалавр.

Nikita Vladimirovich VOINOV – PhD, associate professor at Higher School of Software Engineering. Research interests: software engineering, software verification and testing, web design.

Никита Владимирович ВОИНОВ – кандидат технических наук, доцент Высшей школы программной инженерии. Его научные интересы включают программную инженерию, тестирование и верификацию программного обеспечения, веб-дизайн.



Модификация метода расчета полигенных рисков с использованием графа вариации

^{1,2} О.А. Кондратьева, ORCID: 0000-0001-6220-5077 <kondratyeva@ispras.ru>

¹ Е.А. Карпулевич, ORCID: 0000-0002-6771-2163 <karpulevich@ispras.ru>

¹ Институт системного программирования им. В.П. Иванникова РАН,
109004, Россия, г. Москва, ул. А. Солженицына, д. 25

² Московский государственный университет имени М.В. Ломоносова,
119991, Россия, Москва, Ленинские горы, д. 1

Аннотация. Представление последовательности ДНК возможно в различном виде. Граф вариации один из самых точных методов, который позволяет работать с нетипичными участками и учитывать все их разнообразие. На основе этой структуры данных и метода полигенной оценки риска была построена система интерпретации ДНК. В результате был получен коэффициент корреляции между путем в графе, отвечающим за конкретную последовательность ДНК, и признаком. Затем мы сравнили его с коэффициентом, полученным аналогичным методом, но использующим представление последовательности с использованием эталонного генома. Такое сравнение помогло оценить эффективность представления в виде графа. После этого был построен модифицированный метод подсчета полигенной оценки на данных выравнивания инструмента *vg*, который также был сравнен с существующими методами. Модифицированный метод показал улучшение прогноза признака.

Ключевые слова: граф; представление генома; граф вариаций; HISAT2; *vg*; *minimap2*; GGP; геномный граф; PRS; полигенная оценка; полигенная оценка риска.

Для цитирования: Кондратьева О.А., Карпулевич Е.А. Модификация метода расчета полигенных рисков с использованием графа вариации. Труды ИСП РАН, том 34, вып. 2, 2022 г., стр. 191–200. DOI: 10.15514/ISPRAS–2022–34(2)–15

Modification of the Method for Calculating Polygenic Risks with Variation Graph

^{1,2} O.A. Kondratyeva, ORCID: 0000-0001-6220-5077 <kondratyeva@ispras.ru>

¹ E.A. Karpulevich, ORCID: 0000-0002-6771-2163 <karpulevich@ispras.ru>

¹ Ivannikov Institute for System Programming of the Russian Academy of Sciences,
25, Alexander Solzhenitsyn st., Moscow, 109004, Russia

² Lomonosov Moscow State University,
GSP-1, Leninskie Gory, Moscow, 119991, Russia

Abstract. Representation of the DNA sequence is possible in various ways. The variation graph is one of the most accurate methods that allows you to work with atypical areas and take into account all their diversity. Based on this data structure and the polygenic risk assessment method, a DNA interpretation system was built. As a result, a correlation coefficient was obtained between the path in the column responsible for a specific DNA sequence and the feature. We then compared it with a coefficient obtained by a similar method but using sequence representation using a reference genome. Such a comparison helped to evaluate the effectiveness of the representation in the form of a graph. After that, a modified method for calculating the polygenic score on the alignment data of the *vg* tool was built, which was also compared with existing methods. The modified method showed an improvement in the prediction of the trait.

Keywords: graph; genome representation; variation graph; HISAT2; vg; minimap2; GGP; genomic graph pipeline; PRS; polygenic score; polygenic risk score.

For citation: Kondrateva O.A., Karpulevich E.A. Modification of the Method for Calculating Polygenic Risks with Variation Graph. *Trudy ISP RAN/Proc. ISP RAS*, vol. 34, issue 2, 2022, pp. 191-200 (in Russian). DOI: 10.15514/ISPRAS-2022-34(2)-15

1. Введение

У живых организмов их генетическая информация содержится в молекуле дезоксирибонуклеиновой кислоты (ДНК). У всех клеточных организмов, включая человека и растение арабидопсис, геном (совокупность наследственного материала, заключённого в клетке организма) состоит из ДНК. ДНК отличается между разными организмами и определяет наличие различных признаков. Например, у человека это может быть голубой цвет глаз или предрасположенность к какой-нибудь болезни. Такие признаки называются фенотипом организма.

Информация в ДНК хранится в виде кода, состоящего из четырех химических оснований: аденина (А), гуанина (G), цитозина (С) и тимина (Т). Секвенирование ДНК – это процесс чтения нуклеотидных оснований в молекуле ДНК. Прибор, с помощью которого проводится секвенирование – секвенатор считывает только небольшие фрагменты ДНК, называемые короткими чтениями (ридами), например, длиной 120 пар оснований. Процесс повторной сборки всей последовательности ДНК из полученных ридов называется выравниванием. Существует секвенирование парных прочтений, которое позволяет использовать короткие фрагменты и секвенировать и в прямую, и в обратную сторону, что повышает точность дальнейшего выравнивания. Существующие инструменты для осуществления повторной сборки ДНК в одну последовательность можно разделить на использующие эталонную последовательность (или референс) и графовые.

В случае использования референса в процессе выравнивания каждый короткий прочитанный фрагмент сравнивается с эталонным геномом, чтобы найти наилучшее совпадающее место, где он будет соответствовать наименьшему количеству различий. В случае парных прочтений каждое чтение выравнивается отдельно, а информация по обоим парам объединяется и сообщается в одной строке выравнивания. В высоко вариабельных участках последовательности (человеческие лейкоцитарные антигены в т.ч. главный комплекс гистосовместимости) использование эталонного генома затруднительно и приводит к потере информации. Поэтому вместо референсой последовательности можно использовать графовое представление, которое имеет более сложную структуру и позволяет учитывать вариации включая делеции и вставки.

Графы занимают давнее место в анализе биологических последовательностей, где они часто используются для компактного представления множества возможных последовательностей. Как правило, сами последовательности неявно представляются как обходы в графе. Возможно, самым простым представлением графа является ориентированный граф. В контексте сборки генома графы де Брейна являются популярными представлениями ориентированных графов, в которых каждый узел представляет k -мер (уникальную строку длины k), и каждое направленное ребро представляет собой перекрытие $k-1$ оснований между суффиксом узла «от» и префиксом узла «к».

Ориентированные графы не полностью выражают концепцию двух цепочек ДНК. То есть они не различают чтение молекулы ДНК в ее прямой и обратной комплементарной ориентации. Чтобы выразить это свойство, ориентированные графы можно обобщить до двунаправленных графов, в которых каждая конечная точка ребра имеет независимую ориентацию, указывающую, должна ли прямая или обратная дополнительная цепочка присоединенного узла быть посещается при входе в узел через эту конечную точку ребра. Инверсии, обратные тандемные дубликации и произвольно сложные перестановки

выражаются в двунаправленном представлении. Такие сложные варианты не могут быть выражены в версии ориентированного графа без создания независимых узлов прямого и обратного обхода и сохранения дополнительной информации для описания этой взаимодополняемости. Версия двунаправленных графов с ребрами дает эквивалентное представление.

2. Существующие решения

2.1 Minimap2

Существует множество инструментов, которые реализуют выравнивание на эталонную последовательность, такие как BLASR [1], BWA-MEM [2], GraphMap [3], Kart [4] и minimap2 [5]. Minimap2 один из новых инструментов и превосходит другие аналогичные инструменты выравнивания для конкретных предметных областей как по скорости, так и по точности [5]. Поэтому он был выбран для сравнительного анализа с другими.

Minimap2 использует один и тот же базовый алгоритм, но разные наборы параметров в зависимости от типов входных данных. Возможности minimap2 основаны на быстром базовом алгоритме выравнивания последовательности [6] и точном алгоритме выравнивания цепочек [7]. Для этих алгоритмов слабым местом производительности является выравнивание длинных последовательностей, что было непрактично медленным 10 лет назад. Алгоритм Судзуки-Касахары (Hajime Suzuki, Masahiro Kasahara) [8] в значительной степени устраняет эту проблему.

2.2 HISAT2

Продолжением развития подхода с использованием эталонного генома, является сначала создание линейного графа референса, а затем добавление мутаций в качестве альтернативных путей по графу (рис. 1). Такой метод был реализован в инструменте HISAT2 [9].

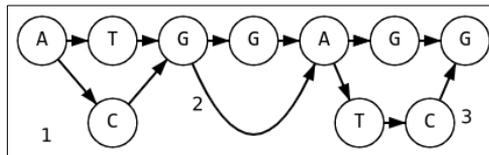


Рис. 1. Представление с использованием графа куска последовательности. Под цифрами указаны различные виды структурных вариаций. 1 - Однонуклеотидный полиморфизм (SNP). 2 - Удаление (делеция). 3 - Вставка

Fig. 1. Representations using the sequence piece graph. Below the numbers are the various types of structural variations. 1 - Single nucleotide polymorphism (SNP). 2 - Deletion (deletion). 3 - Insert

Алгоритм реализует структуру данных на основе графов и использует преобразования Берроуза-Уилера (Michael Burrows, David Wheeler) (BWT) [10]. Вершины представляют собой одно из оснований {A, C, G, T}, а их отношения представлены как ребра. Чтобы обеспечить быстрое нахождение пути в графе, который соответствует конкретному ДНК были сделаны несколько шагов для оптимизации. В отличие от других алгоритмов выравнивания графов, которые используют индексы на основе k-меров, требующих интенсивного использования памяти, таких как vg, HISAT2 использует индекс графа Феррагины-Манзини (Paolo Ferragina, Giovanni Manzini) (ФМ-индекс, GFM) [11].

2.3 Пайплайн геномного графа

В данном инструменте была реализована структура данных генома графа, которая представляет геномные последовательности на ребрах графа. Геномный граф (GGP) строится из популяции геномных последовательностей, так что каждый геном в этой популяции

представлен путем последовательности через граф. Эксперименты, проведенные создателями инструмента [12] по сравнительному анализу, демонстрируют, что использование геномного графа улучшает выравнивание прочтений и поиск вариантов без сопутствующей потери точности.

В настоящее время GGP анализирует образцы по отдельности, и версия, которая выполняет совместный поиск вариантов, находится в стадии разработки. Информация, такая как частоты аллелей каждого варианта и неравновесие по сцеплению между ними, может быть включена в граф, предоставляя дополнительную статистическую информацию для выравнивания считывания и поиска вариантов. В данный момент это реализовано только в графе вариаций.

2.4 Граф вариаций

Граф вариаций [13] представляет собой двунаправленный граф последовательностей ДНК, который представляет генетические вариации в популяции. Эта структура данных обеспечивает сжатое кодирование последовательностей многих геномов. Граф вариаций состоит из:

- вершин, в которых содержится последовательность из {A, C, T, G} и идентификатор;
- ребер, которые соединяют две вершины через любой из их соответствующих концов;
- путей, которые описывают геномы, выровненные последовательностей и аннотации как переходы через вершины, соединенные ребрами.

Чтобы обеспечить сопоставление чтения и другие операции доступа к большим графам последовательностей, инструмент использует краткое представление графа вариаций $vg(xg)$, которое является статическим, но эффективным с точки зрения памяти и времени.

2.5 Сравнение различных представлений

Выравнивание прочтений генома является первым шагом в большинстве рабочих процессов анализа генома. Существует несколько подходов к решению этой задачи, которые можно разделить на две основные категории:

- использование эталонного генома;
- использование графового представления.

Табл. 1. Сравнение инструментов для выравнивания
Table 1. Comparison of alignment tools

Название	Результат	Использование графа	Открытый код
minimap2	sam	Нет	Да
hisat2	sam	Да	Да
GGP	bam	Да	Нет
vg	bam	Да	Да

Хотелось бы отметить, что инструмент GGP, хоть и показал хорошие результаты, к сожалению, не имеет открытого кода. Также для дальнейшей трансформации большинство инструментов биоинформатики принимают и ожидают результатов выравнивания в формате bam, а minimap2 и HISAT2 выдают sam. Это исправляется путем использования инструмента преобразования samtools и возможно, поскольку информация в этих форматах одинаковая, просто bam – это двоичный файл.

3. Задача интерпретирования

По последовательности ДНК возможно определить некоторые признаки организма. Такая операция называется интерпретированием ДНК. Считается, что все признаки организма имеют генетический компонент, однако степень, в которой гены способствуют развитию

фенотипа, варьируется. Существует несколько видов фенотипов - моногенные, хромосомные и полигенные. Моногенные признаки связаны с одним геном. Например, наличие такого заболевания как серповидноклеточная анемия у человека обусловлено мутацией гена HBB. Хромосомные признаки обусловлены изменением количества хромосом. Полигенные признаки же обычно являются результатом сочетания различных генов, каждый из которых увеличивает восприимчивость к этому состоянию. Изучение такого фенотипа всегда представляло сложную задачу. Один из методов интерпретирования называется полигенной оценкой (PRS). Он состоит в том, чтобы на основе данных полногеномного исследования ассоциации (GWAS) и генома организма в формате VCF получить число, которое коррелирует с признаком.

4. Подсчет полигенной оценки

Стандартные оценки полигенного риска обычно строятся на основе взвешенной суммы количества аллелей, то есть:

$$PRS = \sum_i \beta_i x_i$$

Существует несколько подходов для подсчета β_i . Некоторые включенные варианты могут быть ложноположительными, и необработанные оценки размеров эффекта от них могут быть подвержены систематической ошибке отбора [14]. Кроме того, стандартный подход PRS требует тестирования в диапазоне пороговых значений P-value, которые часто выбираются произвольно. Ошибка предсказания, оцениваемая по оптимизированному порогу, также может быть подвержена оптимистическому смещению. Чтобы улучшить предсказание геномного риска, был предложен эмпирические байесовские подходы для восстановления основных размеров эффекта. Этот метод удовлетворительно показал себя при моделировании. Он прост в вычислительном отношении и не требует предположений о распределении размера эффекта. Этот подход заключается в использовании формулу Твиди (Maurice Tweedie) для корректировки оценки β_i :

$$\beta_{Twe,i} = \sqrt{\xi \left(z_i + \frac{f'(z_i)}{f(z_i)} \right)}$$

где ξ – функция преобразования z-статистики в дисперсию объясненной ответственности, описанная в [15], $f(x)$ – ядерная оценка плотности (KDE):

$$f(x) = \frac{1}{nh} \sum_{i=1}^n K \left(\frac{x - X_i}{h} \right)$$

где K является ядром, то есть неотрицательной функцией, а $h > 0$ является сглаживающим параметром, называемым шириной полосы.

Другой подход использует формулу:

$$\beta_{Twe,tdr} = \beta_{Twe}(1 - fdr),$$

где fdr – локальная частота ложных открытий, вероятность нулевого значения с учетом наблюдаемой z-статистики. $(1 - fdr)$, следовательно, является локальной истинной скоростью обнаружения. Этот метод взвешивает каждую оценку величины эффекта по формуле Твиди с вероятностью того, что она не равна нулю. На практике это приведет к дальнейшему уменьшению размеров эффекта до нуля, и часть размеров эффекта станет равной нулю, поскольку локальный fdr равен единице для некоторых маркеров.

Еще один рассматриваемый способ использует:

$$\beta_{tdr} = \beta(1 - fdr),$$

в котором коэффициенты регрессии взвешиваются по локальным истинным показателям обнаружения.

5. Интерпретирование с помощью полигенной оценки

Для анализа генома с помощью оценки полигенного риска для начала требуется выбрать модель для определения вероятности возникновения фенотипа. Поскольку в данном исследовании фенотип представляет собой непрерывную величину, то в таком случае используется линейная регрессия:

$$y = \mu + x\beta + \varepsilon,$$

где μ – коэффициент регрессии.

Линейная регрессия оценивается коэффициентом детерминации R^2 :

$$R^2 = \frac{SS_{reg}}{SS_{tot}}$$

где

$$SS_{tot} = SS_{reg} + SS_{res}$$

$$SS_{reg} = \sum_{i=1}^n (\hat{y}_i - \bar{y})^2$$

$$SS_{res} = \sum_{i=1}^n y_i - \hat{y}_i$$

\hat{y}_i – значение, полученное регрессией, y_i – фактическое значение. Среднее фактических значений:

$$\bar{y} = \frac{1}{n} \sum_{i=1}^n y_i$$

6. Добавление оценки в граф

Геномные графы построены таким образом, что последовательность экземпляра может быть представлена как путь в графе. Чем больше различных вариантов содержит граф, тем точнее будет путь отражать действительную последовательность. Так, при построении графа можно добавить его вершинам оценку, полученную из суммарной статистики (рис. 2), и затем использовать эти данные для подсчета PRS экземпляра.

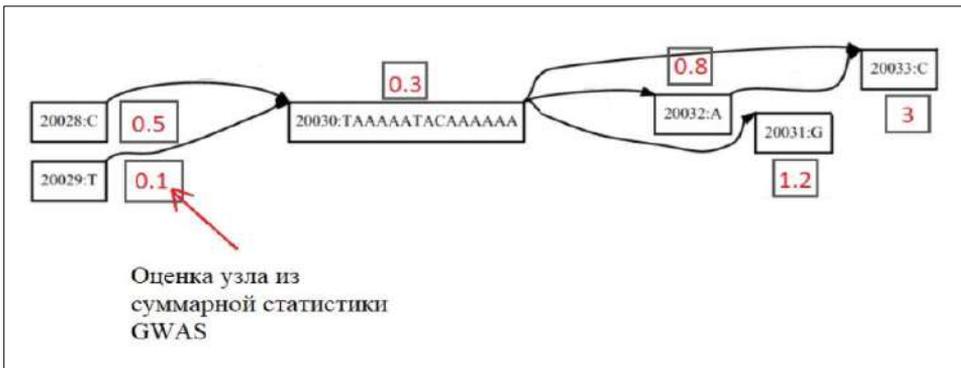


Рис 2. Граф вариаций с оценкой влияния
 Fig. 2. Graph of variations with impact assessment

После построения графа возникает следующий этап – выравнивание данных ДНК на него. Эта операция заключается в поиске пути, который бы соответствовал новой последовательности. Благодаря добавленным ранее данным об определенном признаке, в этот момент можно просуммировав веса всех узлов, входящих в найденный путь, получить полноценную оценку. После этого необходимо построить коэффициент корреляции этой оценки с фенотипом и сравнить результат с исследованием похожего типа.

7. Набор данных

Для оценки эффективности метода были проведены экспериментальные исследования на наборах данных растения *Arabidopsis thaliana*. Доступность естественно инбредных штаммов позволяет повторять фенотипирование одного и того же адаптированного генотипа в различных контролируемых условиях, что делает *Arabidopsis thaliana* подходящим для изучения взаимодействия генотип-среда. Существует проект «1001 геном», который был запущен в начале 2008 г. с целью выявления подробной вариации последовательности всего генома как минимум в 1001 образце [16].

Из доступных в каталоге AgGWAS экспериментов было выбрано несколько исследований с доступной суммарной статистикой. Кроме того, был смоделирован собственный GWAS с помощью инструмента plink на 135 экземплярах. Это было сделано, так как все доступные данные об индивидуальных геномах *Arabidopsis* используются в подсчете. Чтобы проверить, как сильно это влияет, и был построен отдельный GWAS, который был потом протестирован на 44 отдельных экземплярах. В итоге было проведено 2 эксперимента на следующих GWAS.

- M216T665 [17] – признак содержания метаболитов. Профили нецелевых метаболитов ткани листа на основе LC-MS собирали для каждого образца. Характеристика метаболита с отношением массы к заряду 216 и временем удерживания 665 с.
- GWAS, построенный на основе признака в предыдущем пункте.

Собственный GWAS был построен только на 4-й хромосоме, так как исследование M216T665 содержит все значимые варианты только на определенном участке этой хромосомы. Также был построен график (рис. 3), который схож с тем, что показывает каталог для оригинального исследования.

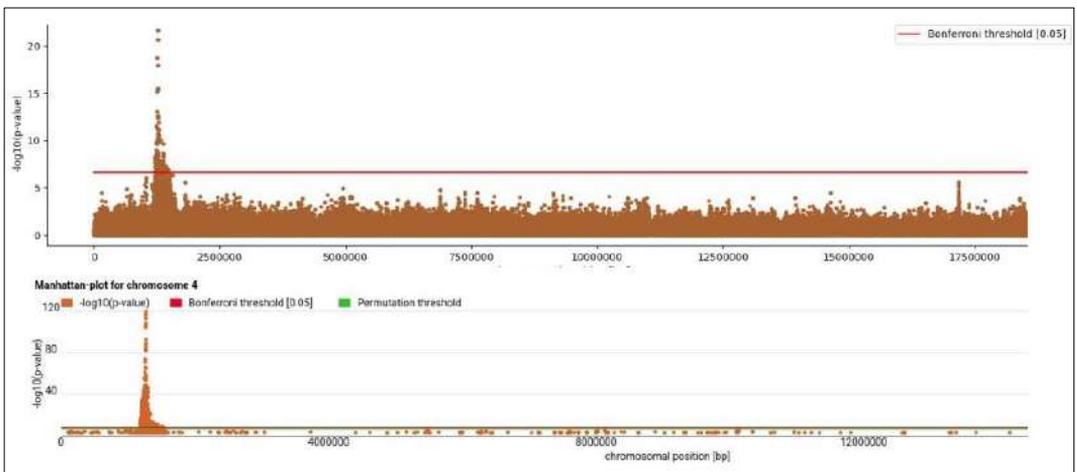


Рис. 3. Манхэттенский график для построенного GWAS на верхнем рисунке и для оригинального GWAS (на нижнем рисунке). Построен для 4-й хромосомы

Fig. 3. Manhattan plot for the constructed GWAS on the top picture and a plot for the original GWAS (on the bottom). Built for chromosome 4

Моделирование данных с секвенаторов было проведено используя данные реального генотипа из проекта 1001 геном. Это исследование содержит данные в формате vcf. С помощью инструмента ART [18] возможно моделирование парных чтений секвенатора Illumina.

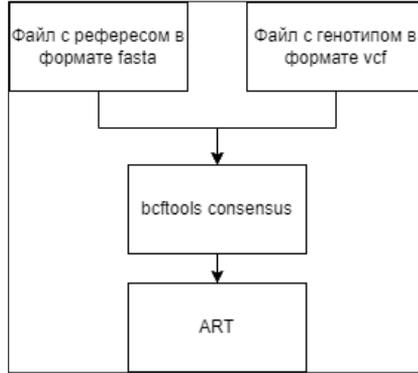


Рис. 4. Моделирование парных чтений секвенатора Illumina
 Fig. 4. Modelling of paired readings of the Illumina sequencer

ART выводит данные чтения в формате fastq, но на вход требует последовательность (без чтений) в формате fasta. Для того, чтобы использовать инструмент ART данный генотипа сначала переводятся из vcf в fasta с помощью bcftools consensus [19]. Он позволяет создать согласованную последовательность для экземпляра, где последовательность включает варианты, выбранные для этого индивидуума. Подробно схема работы для моделирования показана на рис. 4.

8. Результаты

Сначала был проведен анализ использования различных коэффициентов для подсчета PRS. Использовался $\beta_{Twe,tdr}$, β_{tdr} , $\beta_{Twe,i}$, где $i = 207$ равен размеру выборки на которой считался PRS. Анализ проводился для GWAS M216T665. Посчитанный R^2 показан в табл. 2.

Табл. 2. Коэффициент детерминации R^2 для различных коэффициентов при подсчете PRS
 Table 2. Coefficient of determination R^2 for various coefficients when calculating PRS

Коэффициент	$\beta_{Twe,tdr}$	β_{tdr}	$\beta_{Twe,207}$
minimap2	0.6223	0.5531	0.6956
HISAT2	0.6391	0.5776	0.704
GGP	0.5898	0.5153	0.6751
vg	0.5831	0.5059	0.6663
Среднее значение	0.6086	0.538	0.6853

Наилучший результат показал последний коэффициент, поэтому в дальнейшем именно он будет использоваться для подсчета R^2 .

Для тех GWAS, на которых проводились исследования, результаты показаны в табл. 3.

Табл. 3. Коэффициент детерминации R^2 для различных исследований GWAS
 Table 3. Coefficient of determination R^2 for various GWAS studies

Исследование	M216T665	Собственный GWAS (158/44)	Среднее значение
minimap2	0.6956	0.3049(0.3298/0.2908)	0.4053
HISAT2	0.704	0.271(0.3096/0.2897)	0.3936
GGP	0.6751	0.289(0.3362/0.2817)	0.3955

vg	0.6663	0.3095(0.3279/0.285)	0.3972
vg + score	0.6692	0.4583(0.5032/0.4884)	0.5298

9. Выводы

Мы разработали модифицированный метод полигенной оценки, который подсчитывается на результатах выравнивания инструмента *vg*. Он был встроен в существующий инструмент путем добавления опции, которая указывает на необходимость подсчета PRS. Разработанный метод сравнивался с существующими 4 инструментами: HISAT2, minimap2, *vg* (без модификации) и GGP. Для этого были написаны программы подсчета PRS и R^2 .

Для подтверждения результатов было проведено исследование на 3-х наборах GWAS, которые показали одинаковую динамику. Проведение экспериментов было автоматизировано с помощью разработанной программы. Разработанный метод увеличил коэффициент детерминации R^2 для инструмента *vg* с 0.3972 до 0.5298 в среднем, что показывает улучшение предсказательной способности.

Список литературы / References

- [1]. Chaisson M.J., Tesler G. Mapping single molecule sequencing reads using basic local alignment with successive refinement (BLASR): application and theory. *BMC Bioinformatics*, vol. 13, 2012, article no. 238, 17 p.
- [2]. Li H. Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM. arXiv preprint arXiv:1303.3997, 2013, 3 p.
- [3]. Sović I., Šikić M. et al. Fast and sensitive mapping of nanopore sequencing reads with GraphMap. *Nature communications*, vol. 7, issue 1, 2017, article no. 11307, 11 p.
- [4]. Lin H.-N., Hsu W.-L. Kart: a divide-and-conquer algorithm for NGS read alignment. *Bioinformatics*, vol. 33, issue 15, 2017, pp. 2281-2287.
- [5]. Li H. Minimap2: pairwise alignment for nucleotide sequences. *Bioinformatics*, vol. 34, issue 18, 2018, pp. 3094-3100.
- [6]. Polyakov V.O., Roytberg M.A. Tumanyan V.G. Comparative analysis of the quality of a global algorithm and a local algorithm for alignment of two sequences. *Algorithms for Molecular Biology*, vol. 6, 2011, article no. 25, 12 p.
- [7]. Abouelhoda M.I., Ohlebusch E. Chaining algorithms for multiple genome comparison. *Journal of Discrete Algorithms*, vol. 3, issues 2-4, 2005, pp. 321-341.
- [8]. Suzuki H., Kasahara M. Introducing difference recurrence relations for faster semi-global alignment of long sequences. *BMC bioinformatics*, vol. 19, 2018, article no. 45, 14 p.
- [9]. Kim D., Paggi J.M. et al. Graph-based genome alignment and genotyping with HISAT2 and HISAT-genotype. *Nature Biotechnology*, vol. 37, 2019, pp. 907-915.
- [10]. Li H. Fast and accurate short read alignment with Burrows-Wheeler transform. *Bioinformatics*, vol. 25, issue 14, 2009, pp. 1754-1760.
- [11]. Simpson J.T., Durbin R. Efficient construction of an assembly string graph using the FM-index. *Bioinformatics*, vol. 26, issue 12, 2010, pp. i367-i373
- [12]. Rakocevic G., Semenyuk V. et al. Fast and accurate genomic analyses using genome graphs. *Nature genetics*, vol. 51, 2019, pp. 354-362.
- [13]. Garrison E., Sirén J. et al. Variation graph toolkit improves read mapping by representing genetic variation in the reference. *Nature biotechnology*, vol. 36, 2018, pp. 875-879.
- [14]. So H.-C. Improving polygenic risk prediction from summary statistics by an empirical Bayes approach. *Scientific reports*, vol. 7, 2017, article no. 41262, 11 p.
- [15]. So H.-C. Uncovering the total heritability explained by all true susceptibility variants in a genome-wide association study. *Genetic epidemiology*, vol. 35, issue 6, 2011, pp. 447-456.
- [16]. Detlef W., R. The 1001 genomes project for *Arabidopsis thaliana*. *Genome biology*, vol. 10, 2009, article no. 107, 5p.
- [17]. AraGWAS Catalog. 'M216T665' URL: <http://aragwas.1001genomes.org/#/study/144>.
- [18]. Huang W., Li L. et al. ART: a next-generation sequencing read simulator. *Bioinformatics*, vol. 28, issue 4, 2012, pp. 593-594.

- [19]. Li H. A statistical framework for SNP calling, mutation discovery, association mapping and population genetical parameter estimation from sequencing data. *Bioinformatics*, vol. 27, issue 21, 2011, pp. 2987-2993.

Информация об авторах / Information about authors

Олеся Анатольевна КОНДРАТЬЕВА является студентом магистратуры кафедры системного программирования МГУ, работает в ИСП РАН. Ее научные интересы включают в себя графовые представления генома, биоинформатику.

Olesia Anatolevna KONDRATEVA is a master's student of the Department of System Programming of the MSU, also works at ISP RAS. Her scientific interests include graph representations of the genome, bioinformatics.

Евгений Андреевич КАРПУЛЕВИЧ является специалистом отдела информационных систем. Сфера научных интересов: применение алгоритмов анализа данных к биомедицинскому домену, разработку систем распределенного хранения и анализа данных.

Evgeny Andreevich KARPULEVICH is a specialist of the Information Systems Department. Research interests: application of data analysis algorithms to the biomedical domain, development of systems for distributed data storage and analysis.

DOI: 10.15514/ISPRAS-2022-34(2)-16



Approach to Displaying Data to Users of Telemedicine Systems for Early Detection of Diseases by ECG

¹ A.A. Busenkov, ORCID: 0000-0002-2695-2347 <albus.rank@yandex.ru>

¹ E.A. Kholodov, ORCID: 0000-0001-6144-2329 <edosja10@icloud.com>

^{1,2} R.S. Novikov, ORCID: 0000-0002-0656-3475 <rsnovikov@hse.ru>

^{1,2} B.A. Pozin, ORCID: 0000-0002-0012-2230 <bpozin@ec-leasing.ru>

¹ HSE University,

20, Myasnitskaya Ulitsa, Moscow, 101978, Russia

² «EC-leasing» Co.

125 Varshavskoye shosse, Moscow, 117405, Russia

Abstract. With the development of modern technologies in medical organizations, there is an opportunity to modernize existing methods of monitoring public health and detecting diseases. The use of telemedicine can reduce costs and increase the efficiency and accessibility of medical services including monitoring the state of health by remote (outside of medical and preventive institutions) registration and processing of ECG that helps to detect diseases in the initial stages. In this paper, we propose an approach to displaying data to users of telemedicine systems for independent (without medical staff) early detection of diseases by ECG. This approach can be used in the development of a graphical interface for telemedicine systems for the early detection of diseases by ECG.

Keywords: medical systems; user interface; remote screening

For citation: Busenkov A.A., Kholodov E.A., Novikov R.S., Pozin B.A. Approach to Displaying Data to Users of Telemedicine Systems for Early Detection of Diseases by ECG. Trudy ISP RAN/Proc. ISP RAS, vol. 34, issue 2, 2022, pp. 201-208 (in Russian). DOI: 10.15514/ISPRAS-2022-34(2)-16

Подход к отображению данных для пользователей телемедицинских систем раннего обнаружения заболеваний по ЭКГ

¹ А.А. Бусенков, ORCID: 0000-0002-2695-2347 <albus.rank@yandex.ru>

¹ Э.А. Холодов, ORCID: 0000-0001-6144-2329 <edosja10@icloud.com>

^{1,2} Р.С. Новиков, ORCID: 0000-0002-0656-3475 <rsnovikov@hse.ru>

^{1,2} Б.А. Позин, ORCID: 0000-0002-0012-2230 <bpozin@ec-leasing.ru>

¹ Национальный исследовательский университет «Высшая школа экономики»,
101978, Россия, г. Москва, ул. Мясницкая, д. 20²

² ЗАО «ЕС-Лизинг»,

117587, Россия, г. Москва, Варшавское шоссе, д. 125, стр.1

Аннотация. С развитием современных технологий в медицинских организациях появилась возможность модернизировать существующие методы мониторинга здоровья населения и выявления заболеваний. Использование телемедицины позволит сократить расходы и повысить эффективность и доступность медицинских услуг, включая мониторинг состояния здоровья с помощью удаленной (за пределами медицинских и профилактических учреждений) регистрации и обработки ЭКГ, с помощью которой можно обнаружить заболевания на начальных этапах. В настоящей работе предлагается подход к отображению данных пользователям телемедицинских систем раннего обнаружения заболеваний с помощью ЭКГ без участия медицинского персонала. Данный подход может быть использован при разработке графического интерфейса телемедицинских систем раннего обнаружения заболеваний с помощью ЭКГ.

Ключевые слова: медицинские системы; пользовательский интерфейс; дистанционный скрининг

Для цитирования: Бусенков А.А., Холодов Э.А., Новиков Р.С., Позин Б.А. Подход к отображению данных для пользователей телемедицинских систем раннего обнаружения заболеваний по ЭКГ. Труды ИСП РАН, том 34, вып. 2, 2022 г., стр. 201-208. DOI: 10.15514/ISPRAS-2022-34(2)-16

1. Introduction

With the development of modern technologies, it became possible to modernize existing methods for early detection of diseases in the population, including people in remotely located places from medical institutions, using telemedicine. This opportunity allows increasing the coverage of the population with medical services without resorting to an increase in the staff of medical personnel, which makes it possible to reduce the costs of their provision and further treatment of the patient with timely detection of the disease at an early stage [1, 2].

The need for telemedicine technologies manifests itself during planning a mass clinical survey of the population with the aim of early detection of a specific disease - during planning a screening [3]. For example, the screening of the population for the presence of carbohydrate metabolism disorders on a national scale has not yet been carried out in Russia, although, in 2016, the prevalence of undiagnosed type 2 diabetes mellitus (T2DM) among the adult population was higher than the prevalence of diagnosed T2DM [4]. Without the implementation of telemedicine, screening tests recommended in the current standards (blood tests) [5] are too difficult and expensive to carry out, because a large number of medical personnel and laboratory equipment will be required in primary care.

Due to the increased availability of ECG recording devices [6] and the entry into force of the law on telemedicine, since 2018 in Russia [7], a new opportunity has been appeared for the early detection of diseases during the screening - with processing and analyzing ECG obtained from many remote electrocardiographs located outside medical facilities. The patient takes an ECG in a convenient place for himself where the electrocardiograph is located, then the ECG itself is transmitted via the Internet to a data center for processing. After that, the processing results are transmitted back to the

patient's smartphone or e-mail, based on which the patient decides what to do next – to consult the doctor or not.

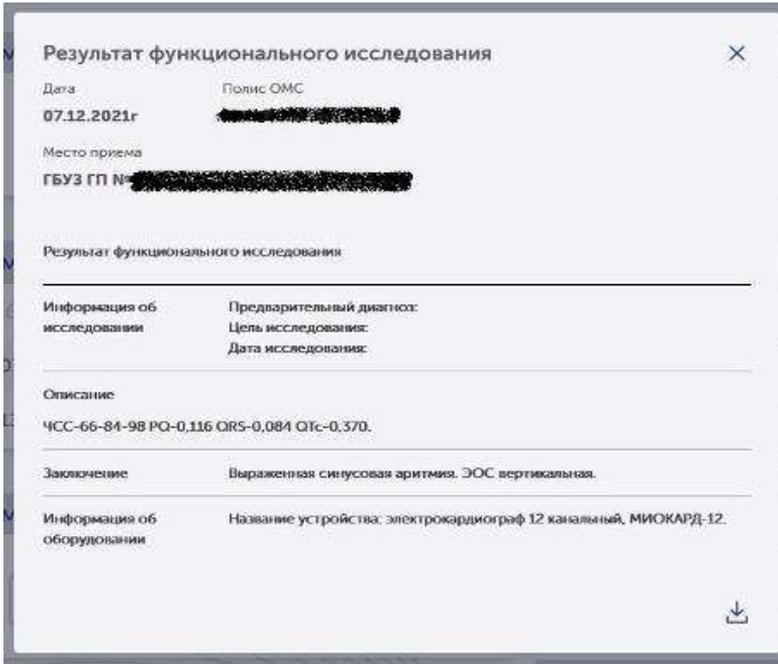


Fig. 1. An example of the results of an ECG study from an electronic medical record EMIAS (text in Russian)

When considering existing medical systems, it was found that they are not suitable for remote ECG screening studies. The main reason is that the systems do not consider the user's qualifications. For this reason, the results of ECG analysis in such systems are provided to the user in terms that are understood by the medical specialist, but incomprehensible to the patient. As a result, the patient will have (if he wants) to find out on his own what to do next after the screening. An example of such results from the EMIAS electronic medical record is shown in Fig. 1.

Thus, when planning remote screening by ECG using telemedicine technologies, it is necessary to consider the qualifications of the system user that will be involved in the screening process.

2. Aims and objectives

The aim of the study is to develop an approach to inform the user about the results of a remote ECG analysis, which the patient will understand on his own, without the assistance of medical personnel. This approach is proposed to be implemented by creating a personal patient's account of the telemedicine system for the early detection of diseases by ECG (hereinafter referred to as the System), in which the data will be displayed in accordance with the qualifications of the patient. The way medical data is presented should be flexible enough to meet the needs of users with different qualifications and experience. Most of the existing medical systems are designed for the fact that the user will not independently decipher the results of your examination, or it will be a patient with experience in the field of medicine. Our approach to displaying data will help to design the user interaction with the System so that the user with limited medical knowledge understands what to do next when passing the remote screening.

In particular, the developed approach helps to design the patient's interaction with the System during remote screening without the participation of a doctor.

To achieve this aim, it is necessary to solve the following tasks:

- Analysis of the patient's role model of the System;

- Determining the requirements for the personal patient's account of the System;
- Developing an example of the personal patient's account of the System.

As an example, it is planned to develop an automated system for carbohydrate metabolism disorders (CMD) screening by ECG together with a personal patient's account.

It is supposed to evaluate the developed personal patient's account by checking for compliance with the stated requirements (verification) and the declared CMD screening process (validation).

3. Overview

When designing a personal patient's account of the System, it is necessary to determine the requirements. To determine the requirements, it is necessary to analyze the role model of the users of the System. The role models of telemedicine systems users affect what information should be displayed at all and in what form. Depending on the functions and purpose of the System, the model may include several roles: Patient, Medical Worker, etc. In the System with a mobile device for recording an ECG and displaying ECG analysis results, the role model includes only the Patient.

The characteristics of the Patient are:

- The Patient does not know medical terminology;
- The Patient may mishandle the electrocardiograph. Although portable electrocardiographs usually have clear and simple instructions, due to various factors (power line interference, unnecessary patient movements that degrade the quality of the ECG, etc.), the resulting ECG may not be suitable for further analysis;
- The Patient does not necessarily record an ECG in a medical facility, so there may not be a qualified medical worker nearby who will check whether the patient is recording the ECG correctly and tell what to do next.

Taking into account the process automated by the system, it is necessary to classify what the Patient needs to know first of all, and what does not need to know at all. Such a description of the user's role model will allow you to answer questions such as:

- What information should be displayed on the main page of the personal account?
- What information may be put on additional pages or displayed when clicking on the button?
- What should not be displayed at all, even at the Patient's request?

For example, during the process of remote ECG screening:

- First of all, it is necessary for the Patient to know the results of the screening and further steps. It is necessary to display information that answers the following questions:
 - "Is he/she all right?" - it should be said whether the Patient has signs of the presence of the studied disease;
 - "Does he/she need to see a doctor?" - a recommendation should be submitted to go to a doctor of a certain specialization when identifying signs of the presence of the disease under study;
 - "Did he/she take an ECG properly?" - if the ECG turned out to be of insufficient quality for further analysis, a recommendation to take the ECG again should be submitted.
- Secondly, if the Patient is interested, it is necessary to let them know additional information. For example, how the result was obtained, other details (the ECG tape or ECG analysis dynamics);
- Finally, the patient must be denied access to data other than the Patient's. For example, the negotiations of doctors at the consultation and data about other patients.

The Patient's needs during remote screening determine the structure of the personal account:

- Priority things need to be displayed on the main page;

- Additional information can be displayed on a separate request to the patient. For example, by clicking a certain button on the main page or expanding a certain list.

Information should be provided in a simplified form for the Patient, such as infographics.

However, prior to displaying data, the Patient must be authorized and authenticated so he wouldn't have access to data not related to the Patient.

4. Implementation

Based on the requirements laid down above, a personal patient's account of an automated system for CMD screening has been developed.

The screening process, based on which the system was developed, is presented in the following algorithm [8]:

- 1) Stage one:
 - a) Registration of the patient during screening;
 - b) ECG recording using a recording device involved in the screening process;
 - c) Checking the quality of the taken ECG. If there are critical interferences on the recorded ECG, the patient will go through steps (b) and (c) again;
 - d) Repeat steps (b) and (c) until enough ECGs are obtained without critical interference;
 - e) Upon successful completion of the ECG quality check, the patient receives a message on the smartphone/e-mail about the presence/absence of CMD signs on the ECG;
- 2) In the second stage, the patient is re-examined for signs of CMD based on national recommendations, with which an appropriate diagnosis is made.

As part of the work, the automation of the first stage is carried out, since the second stage must be performed manually by an endocrinologist in accordance with national recommendations.

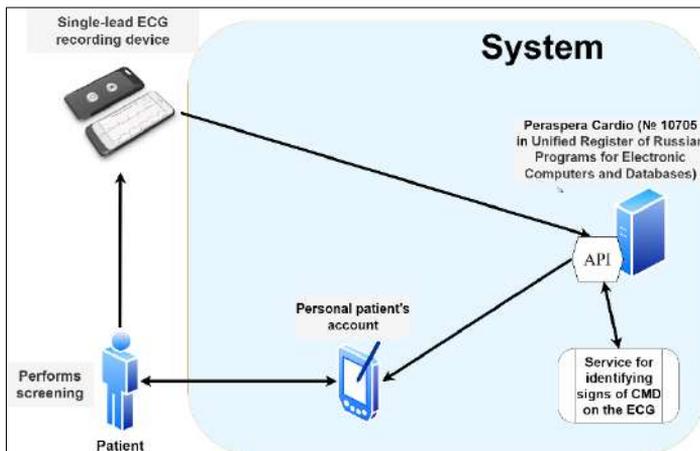


Fig. 2. The overall scheme of the automated system for CMD screening

The developed System consists of the following subsystems/services (Fig. 2):

- Peraspera Cardio software platform [9] is designed to assess the state of the human cardiovascular system based on ECG processing. This platform was chosen as the basis of the System as it accelerates its development since the platform already has the necessary functionality: collection, storage, pre-processing of ECG, performing basic calculations, and providing calculation results via API to related systems;
- Service for identifying signs of CMD on the ECG. The ECG data stored in Peraspera Cardio is transmitted to the API service to perform calculations to detect signs of CMD based on ECG analysis. ECG analysis is carried out according to the mathematical methods described in [8,10].

Calculation results are sent back to Peraspera Cardio;

- Personal patient's account. The main purpose of the personal patient's account is to provide the patient with the opportunity to get acquainted with the results of the ECG analysis and view further recommendations during and after the screening.

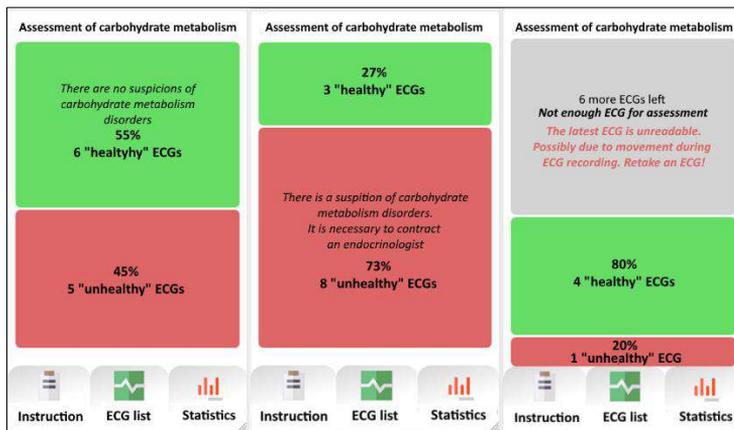


Fig. 3. Examples of displaying the results of ECG analysis on a smartphone in a web application

Access to the personal patient's account is implemented through a web application adapted for smartphones. Fig. 3 shows examples of displaying data for a patient by results or during screening from left to right:

- “The patient has no suspicions of carbohydrate metabolism disorders”;
- “The patient has a suspicion of carbohydrate metabolism disorders, it is necessary to contact an endocrinologist”;
- “The patient needs to retake the ECG due to poor quality. A possible reason is a movement during ECG recording.”

The developed system is compatible with all current operating systems and browsers with JavaScript support. The server part runs under Windows OS or Linux. The server part of the application is implemented using the JavaScript language, PostgreSQL is used as the DBMS.

5. Evaluation

At the moment, all the requirements for displaying data for the patient in the corresponding personal patient's account of the automated screening system for carbohydrate metabolism disorders have been met:

- after each ECG recording, updated values of health indicators are displayed in the personal patient's account, namely, indicators for assessing the state of carbohydrate metabolism (the proportion of ECG with identified signs of CMD);
- the patient is informed in his personal patient's account when it is recommended to take a new ECG to get an accurate answer, whether he needs to consult an endocrinologist;
- if there are enough ECGs for analysis, the patient is informed whether he has a suspicion of CMD and whether he needs to consult an endocrinologist;
- if the current ECG was taken with insufficient quality, the Patient is informed about the need to take the ECG again.
- prior to displaying data, the Patient must log in to his personal account by login and password sent to his smartphone after registration during screening and first ECG recording.

According to the implemented functions, we can say that the patient's personal account corresponds to the declared screening process for CMD.

To evaluate the work of the developed system with the patient's personal account, it is necessary to test the system in practice during the study. The system has not yet been tested, it is required to check in the form of trial operation during the screening study.

6. Conclusion

An approach has been developed to inform the user about the results of a remote examination in telemedicine systems for the early detection of diseases by ECG (Systems), which will be understandable to the patient without the help of medical professionals. The approach is implemented through the development of a user's personal account.

The role model of the patient as the central user of the System has been analyzed. Based on the role model of the patient, the requirements for displaying data in the personal account have been developed - based on the results of processing each ECG, the following information should be provided in the first place:

- whether he has signs of a specific disease;
- if so, which specialty doctor he should go to;
- a recommendation to take the ECG again if the previously obtained ECG has insufficient quality.

An example of a patient's personal account within the System, an automated system for screening for carbohydrate metabolism disorders (CMD) has been developed. The patient's personal account complies with the stated requirements and the CMD screening process. A trial operation of the system is planned for its approbation.

The obtained approach to inform the user about the results of a remote ECG recording can be used later for developing requirements for the graphical interface of the personal user's account for the above-mentioned Systems.

References

- [1]. Ryan P., McGrath C. et al. Enhancing efficiency in a cardiac investigations department by increasing remote patient monitoring. *International Journal for Quality in Health Care*, vol. 31, issue Supplement_1, 2019, pp. 29-34.
- [2]. Kędzierski K., Radziejewska J. et al. Telemedicine in Cardiology: Modern Technologies to Improve Cardiovascular Patients' Outcomes – A Narrative Review. *Medicina*, vol. 58, issue 2, 2022, article no. 210, 12 p.
- [3]. de Moraes E.R.F.L., Cirenza C. et al. Prevalence of atrial fibrillation and stroke risk assessment based on telemedicine screening tools in a primary healthcare setting. *European Journal of Internal Medicine*, vol. 67, 2019, pp. 36-41.
- [4]. Dedov I., Shestakova M. et al. Prevalence of type 2 diabetes mellitus (T2DM) in the adult Russian population (NATION study). *Diabetes Research and Clinical Practice*, vol. 115, 2016, pp. 90-95.
- [5]. Dedov I.I., Shestakova M.V. et al. Standards of specialized diabetes care. 9th edition. *Diabetes mellitus*, vol. 22, issue 1S1, 2-19, pp. 1-144 (in Russian) / Дедов И.И., Шестакова М.В. и др. Алгоритмы специализированной медицинской помощи больным сахарным диабетом. 9-й выпуск. Сахарный диабет, том 22, вып. 1S1, 2019 г., стр. 1-144.
- [6]. Muscovites can check their hearts on cardiac chairs in 25 public service centers. My documents. Available at: <https://md.mos.ru/presscenter/news/detail/10271712.html>, accessed: 30.03.2022 (in Russian). / В 25 центрах госуслуг москвичи могут проверить сердце на кардиокреслах. Мои документы.
- [7]. Federal Law 242-FZ, dated 29.07.2017 «On amendments to certain legislative acts of the Russian federation on the use of information technologies in the field of health care» (in Russian) / Федеральный закон от 29.07.2017 № 242-ФЗ «О внесении изменений в отдельные законодательные акты Российской Федерации по вопросам применения информационных технологий в сфере охраны здоровья».

- [8]. Shmid A.V., Novopashin M.A. et al. Computerized method for non-invasive detection of carbohydrate metabolism disorders by heart rate variability and wearable autonomous device for its implementation. Patent 2751817, Russian Federation, 2021 (in Russian) / Шмид А.В., Новопашин М.А. et al. Компьютеризированный способ неинвазивного выявления нарушений углеводного обмена по вариабельности сердечного ритма и носимое автономное устройство для его реализации. Патент 2751817, Российская Федерация, 2021.
- [9]. Peraspera Cardio. Official Website of the Unified Register of Russian Programs for Electronic Computers and Databases. Available at: https://reestr.digital.gov.ru/reestr/354134/?sphrase_id=364046, accessed: 30.03.2022 (in Russian) / Peraspera Cardio. Официальный сайт единого реестра российских программ для электронных вычислительных машин и баз данных.
- [10]. Shmid A.V., Berezin A.A. et al. Computerized method for non-invasive detection of carbohydrate metabolism disorders by electrocardiogram. Patent 2728869, Russian Federation, 2020 (in Russian) / Шмид А.В., Березин А.А. et al. Компьютеризированный способ неинвазивного выявления нарушений углеводного обмена по электрокардиограмме. Патент 2728869, Российская Федерация, 2020.

Информация об авторах / Information about authors

Aleksei Alexandrovich BUSENKOV – HSE University student. Research interests: cardiology, data analysis.

Алексей Александрович БУСЕНКОВ – магистрант НИУ ВШЭ. Сфера научных интересов: кардиология, анализ данных.

Eduard Andreevich KHOLODOV – HSE University student. Research interests: telemedicine, BigData, information systems.

Эдуард Андреевич ХОЛОДОВ – магистрант НИУ ВШЭ. Сфера научных интересов: телемедицина, «Большие данные», информационные системы.

Roman Sergeevich NOVIKOV – Senior Researcher, Department of Mathematical Cardiology, «EC-leasing» Co., postgraduate student, HSE University. Research interests: cardiology, data analysis.

Роман Сергеевич НОВИКОВ – старший научный сотрудник отдела математической кардиологии ЗАО «ЕС-лизинг», аспирант НИУ ВШЭ. Сфера научных интересов: кардиология, анализ данных.

Boris Aronovich POZIN – Doctor of Technical Sciences, Professor of HSE University, CTO of EC-leasing Co. Expert in Software Engineering field, ESSENCE, Software Systems Performance and Integration Testing, Life Cycle Supporting Systems development, implementation and automation. Author of over 200 publications.

Борис Аронович ПОЗИН – доктор технических наук, профессор НИУ ВШЭ, технический директор ЗАО «ЕС-лизинг». Эксперт в области программной инженерии, ESSENCE, тестирования производительности и интеграции программных систем, разработки, внедрения и автоматизации систем поддержки жизненного цикла. Автор более 200 публикаций.