Институт динамики систем и теории управления имени В.М. Матросова
Сибирского отделения Российской академии наук
Иркутск, Россия

Центр научных исследований и высшего образования
(CICESE Research Center)
Энсенада, Мексика

Материалы VII Международного семинара по информационным, вычислительным и управляющим системам для распределенных сред (ICCS-DE 2025)

7 – 11 июля 2025 г., Иркутск, Россия

Иркутск: Издательство ИДСТУ СО РАН
2025

УДК: 004.7/.9

ББК: 32.96 + 32.97

Редакционная коллегия:

Бычков И.В., Стоянов Ж., Черных А.Н., Феоктистов А.Г.

Материалы VII Международного семинара по информационным, вычислительным и управляющим системам для распределенные сред (ICCS-DE 2025), 7-11 июля, 2025, Иркутск, Россия. Иркутск: Изд-во ИДСТУ СО РАН, 2025. 218 с.

Proceedings of the 7th International Workshop on Information, Computation, and Control Systems for Distributed Environments (ICCS-DE 2025), July 7-11, 2025, Irkutsk, Russia. Irkutsk: ISDCT Publisher, 2025. 218 p.

ISBN 978-5-6041814-8-5

Материалы научного сборника включают избранные статьи и тезисы VII Международного семинара по информационным, вычислительным и управляющим системам для распределенные сред (ICCS-DE 2025), проведенного Институтом динамики систем и теории управления имени В.М. Матросова Сибирского отделения Российской академии наук (Иркутск, Россия) совместно с Центром научных исследований и высшего образования (CICESE Research Center, Энсенада, Мексика) 7-11 июля, 2025 г. в г. Иркутск, Россия. Избранные научные работы посвящены развитию теории и практики разработки информационных, вычислительных и управляющих систем, относящихся к различным предметным областям информационной, технической, экономической и других сфер человеческой деятельности, в частности. Особенность рассматриваемых систем заключается в их ориентации на функционирование в различных распределенных средах. Особое внимание уделяется системам экологического мониторинга Байкальской природной территории, а также технологиям исследования современных систем энергетики.

Официальный сайт семинара ICCS-DE 2025: https://iccs-de.icc.ru/

ISBN 978-5-6041814-8-5

9 785604 181485

© Институт динамики систем и теории управления им. В.М. Матросова СО РАН, 2025

ОРГАНИЗАТОРЫ

Международный программный комитет семинара

Председатель программного комитета

Бычков И.В., Институт динамики систем и теории управления имени В.М. Матросова Сибирского отделения Российской академии наук, г. Иркутск, Россия

Сопредседатели программного комитета

Andrei Tchernykh, CICESE Research Center, Ensenada, Mexico

Zeljko Stojanov, University of Novi Sad, Technical Faculty ”Mihajlo Pupin”, Full professor, Zrenjanin, Serbia

Члены программного комитета

Jorge Mario Cortés-Mendoza, Polytechnic University of Amozoc (UPAM), San Andrés las Vegas, Amozoc, Puebla, Mexico

Dalibor Dobrilovic, University of Novi Sad, Technical Faculty “Mihajlo Pupin”, Zrenjanin, Serbia

Xiwang Guo, Northeastern University, Shenyang, China

Ljubica Kazi, University of Novi Sad, Technical faculty “Mihajlo Pupin”, Zrenjanin, Serbia

Бабенко М.Г., Северо-Кавказский федеральный университет, Ставрополь, Россия

Еделев А.В., Институт систем энергетики им. Л.А. Мелентьева Сибирского отделения Российской академии наук, Иркутск, Россия

Ковтуненко А.С., Уфимский Государственный Авиационный Технический Университет, Уфа, Россия

Костромин Р.О., Cloud.ru, Иркутск, Россия

Легалов А.И., Национальный исследовательский университет "Высшая школа экономики", Москва, Россия

Смагин С.И., Вычислительный Центр Дальневосточного отделения Российской академии наук, Хабаровск, Россия

Федоров Р.К., Институт динамики систем и теории управления имени В.М. Матросова Сибирского отделения Российской академии наук, Иркутск, Россия

Феоктистов А.Г., Институт динамики систем и теории управления имени В.М. Матросова Сибирского отделения Российской академии наук, Иркутск, Россия

Черкашин Е.А., Институт динамики систем и теории управления имени В.М. Матросова Сибирского отделения Российской академии наук, Иркутск, Россия

Шигаров А.О., Институт динамики систем и теории управления имени В.М. Матросова Сибирского отделения Российской академии наук, Иркутск, Россия

Шокин Ю.И. Федеральный исследовательский центр информационных и вычислительных технологий, Новосибирск, Россия

Юрин А.Ю., Институт динамики систем и теории управления имени В.М. Матросова Сибирского отделения Российской академии наук, Иркутск, Россия

Организационный комитет семинара

Председатель организационного комитета

Феоктистов А.Г., Институт динамики систем и теории управления имени В.М. Матросова Сибирского отделения Российской академии наук, г. Иркутск, Россия.

Члены организационного комитета

Бабанин Д.Г., Институт динамики систем и теории управления имени В.М. Матросова Сибирского отделения Российской академии наук, Иркутск, Россия

Баранова К.С., Институт динамики систем и теории управления имени В.М. Матросова Сибирского отделения Российской академии наук, Иркутск, Россия

Башарина О.Ю., Уральский государственный экономический университет, г. Екатеринбург, Россия

Велякина В.Б., Институт динамики систем и теории управления имени В.М. Матросова Сибирского отделения Российской академии наук, Иркутск, Россия

Воскобойников М.Л., Институт динамики систем и теории управления имени В.М. Матросова Сибирского отделения Российской академии наук, Иркутск, Россия

Жарков М.Л., Институт динамики систем и теории управления имени В.М. Матросова Сибирского отделения Российской академии наук, Иркутск, Россия

Игнатьев И.Н., Институт динамики систем и теории управления имени В.М. Матросова Сибирского отделения Российской академии наук, Иркутск, Россия

Кононенко Г.Б., Институт динамики систем и теории управления имени В.М. Матросова Сибирского отделения Российской академии наук, Иркутск, Россия

Кумачев А.А., Институт динамики систем и теории управления имени В.М. Матросова Сибирского отделения Российской академии наук, Иркутск, Россия

Маджара Т.И., Институт динамики систем и теории управления имени В.М. Матросова Сибирского отделения Российской академии наук, Иркутск, Россия

Максимкин Н.Н., Институт динамики систем и теории управления имени В.М. Матросова Сибирского отделения Российской академии наук, Иркутск, Россия

Мелкозеров С.В., Институт динамики систем и теории управления имени В.М. Матросова Сибирского отделения Российской академии наук, Иркутск, Россия

Михайлов А.А., Институт динамики систем и теории управления имени В.М. Матросова Сибирского отделения Российской академии наук, Иркутск, Россия

Михеев М.А., Институт динамики систем и теории управления имени В.М. Матросова Сибирского отделения Российской академии наук, Иркутск, Россия

Новопашин А.П., Институт динамики систем и теории управления имени В.М. Матросова Сибирского отделения Российской академии наук, Иркутск, Россия

Попова А.К., Институт динамики систем и теории управления имени В.М. Матросова Сибирского отделения Российской академии наук, Иркутск, Россия

Столбова Г.Н., Институт динамики систем и теории управления имени В.М. Матросова Сибирского отделения Российской академии наук, Иркутск, Россия
Федоров Р.К., Институт динамики систем и теории управления имени В.М. Матросова Сибирского отделения Российской академии наук, Иркутск, Россия
Фереферов Е.С., Институт динамики систем и теории управления имени В.М. Матросова Сибирского отделения Российской академии наук, Иркутск, Россия
Черкашин Е.А., Институт динамики систем и теории управления имени В.М. Матросова Сибирского отделения Российской академии наук, Иркутск, Россия

СОДЕРЖАНИЕ

ПЛЕНАРНАЯ СЕССИЯ

Z. Stojanov. Graph Clustering Methods in Distributed Socio-Technical Systems..... 8

СЕКЦИОННЫЕ ДОКЛАДЫ

I. Kimyaev, B. Sokolov, V. Zakharov. Methodology of The Manufacturing Companies Viability Evolutionary Management..... 23

P. Kosov, A. Legalov. Adding procedural-parametric polymorphism to the C language..... 31

A. Tolstikhin. Multiple mobile robots formation control using behavioral-based approach 42

I. Vecstejn, Z. Stojanov. Personal experience in the onboarding process: A reflective analysis through autoethnography 47

К. Блинов, И. Курочкин. Подбор гиперпараметров для синхронного метода распределенного глубокого обучения при решении задач классификации изображений 55

Н. Вершков, Н. Кучукова, В. Куценко. Создание искусственных нейронных сетей для работы в распределенных вычислительных средах 61

А. Вихарева, И. Трофимова. Оптимизация функционирования одной цепи поставок на основе эвристического подхода 67

М. Воскобойников, А. Феоктистов. Оценка эффективности управления вычислениями диспетчером научных рабочих процессов..... 74

И. Диомидов. Интегрированная система сопровождения инвентаризации лесов..... 83

Н. Дородных, А. Юрин. Проблемно-ориентированная среда для создания систем поддержки принятия решений на основе таблиц решений 91

В. Мочалов, О. Григорьева. Методика формирования исходных данных для алгоритмов машинного обучения при идентификации элементов ландшафта 97

Ю. Пестова, О. Николайчук, Я. Ветвицкая. Исследование влияния грозовых явлений на лесные пожары на основе методов машинного обучения 102

В. Симонов, М. Хайретдинов. Разработка препроцессора языка программирования python для распределенной обработки данных.....107

И. Скопин. Конструирование программных интерфейсов.....114

А. Столбов. Разработка веб-сервисов для комплексного исследовании окружающей среды на основе открытых стандартов моделирования.....128

Р. Федоров, А. Верхозина. Методика создания композиций сервисов для ботанических исследований 133

А. Феоктистов, М. Воскобойников, И. Бычков, А. Еделев. Разработка сервис-ориентированной среды моделирования природно-технических систем 141

СПЕЦИАЛЬНАЯ СЕКЦИЯ ДЛЯ МОЛОДЫХ УЧЕНЫХ

V. Badenko, A. Kozlov. Application of information and computing systems in a hybrid power grid.....146

N. Kosyanov. On an Improvement to the Next-step Procedure 149

А. Алексеев, А. Михайлов. Обучение представлениям электронных документов 152

М. Боднюк. Развитие системы мониторинга стенда развитие имитационного моделирования микросети 155

А. Булах. Сегментация КЛКТ-снимков на основе U-Net архитектуры 157

Е. Викулова, А. Толстихин. Разработка методов программной симуляции работы LIDAR Development of software simulation methods LIDAR operation.....159

О. Гавенко, Н. Шашок. Обзор методов предобработки мультимодальных данных, содержащих климатическую информацию, при генерации векторных представлений данных.....163

Н. Душкина, А. Толстихин. Методы построения матрицы перспективного преобразования для сшивки изображений 167

М. Загородников, А. Михайлов. Интеграция метода восстановления текста из PDF документов с нарушенной кодировкой в систему Dedoc.....172

И. Игнатьев. Цифровая трансформация научно-организационной деятельности: теоретические подходы к разработке специализированного фреймворка 175

А. Кузнецов, А. Михайлов. Защита от незаметных атак на модели NLP с использованием омоглифов .181

А. Линявский, Н. Нагул. Планирование на основе GOAP в задаче поиска объекта группой мобильных роботов 183

Д. Лукьянов, М. Лапина, В. Лапин, Н. Кучеров. Построение алгоритмов машинного обучения для детектирования сайтов-мошенников 188

Е. Николаев. Автоматизация DevOps в распределенной вычислительной среде применительно к научным приложениям.....192

И. Поддубный, Н. Дородных. Концепция подхода к решению задачи ответа на вопросы потабличной информации с использованием вычислительных графов 196

К. Тобола, Н. Дородных. Адаптация подхода SimCLR к аннотированию столбцов русскоязычных таблиц 199

М. Чекан. Методика применения инструментального комплекса для моделирования взаимодействующих микросетей.....202

Е. Ширяев, М. Бабенко. Модификация метода деления чисел в Системе Остаточных Классов.....205

Е. Юмашев. Методика разработки приложений для получения, обработки и анализа метеоданных 208

Д. Яковлев, А. Феоктистов. Система мониторинга программно-аппаратной инфраструктуры вычислительной среды 211

СПЕЦИАЛЬНАЯ СЕКЦИЯ ДЛЯ ШКОЛЬНИКОВ

В. Лапин, О. Естафиади, А. Смирнов, А. Багаутдинова. Образовательный чат-бот в Telegram с AI-генерацией тестов.....214

Graph Clustering Methods in Distributed Socio-Technical Systems

Zeljko Stojanov[1]

[1]University of Novi Sad, Technical Faculty "Mihajlo Pupin", Zrenjanin, Serbia

Abstract

Socio-technical systems are characterized by dynamic relations between human actors, organizations, and technical components, requiring sophisticated mathematical and engineering methods for their modelling, analysis, and optimization. Graphs have been recognized as an efficient and powerful abstraction for modelling and analysing these systems, while graph clustering plays an important role in analysing and optimizing their structure and characteristics. A method for graph clustering based on attribute similarity of neighbouring nodes is introduced in this paper. The method contains two algorithms, the first one for checking similarity between neighbouring nodes and the second one for merging similar neighbouring nodes into clusters. The implementation of the method for determining the minimal number of air pollution measurement stations dispersed in urban city environments is presented to illustrate a possible scenario of use. Implementation is based on simulation of the method in a simulation environment that creates initial scenarios with measurement stations presented as graph nodes, create data sets for measurement stations, and calculates the minimal number of measurement stations by using the presented graph clustering method.

Keywords

graph clustering, similarity measure, socio-technical systems, air pollution measurement

1. Introduction

Socio-technical systems (STS) are complex and dynamic systems characterized by relationships and interplay between humans, organizational structures, and technical elements, aimed at achieving specific objectives and behaviours [1, 2, 3]. In general, these systems have complex topologies and very heterogeneous structures [4], the understanding of which requires dual focus and diverse and broad knowledge of both technical and social disciplines [5, 6]. These systems emerge in diverse domains such as healthcare, education, transportation, logistics, social networks, cyber security, and collaborative software development. Modelling such systems includes capturing their structural and behavioural aspects. Graph-based models have emerged and are accepted as a powerful abstraction for representing complex networks in STSs [7, 8, 9], where nodes denote both human and technical entities, while edges denote interactions or relations between them. Graph-based models can effectively represent the complexity of systems, enabling easy and efficient analysis by applying various mathematical methods [10].

7th International Workshop on Information, Computation, and Control Systems for Distributed Environments (ICCS-DE 2025), July 7–11, 2025, Irkutsk, Russia

zeljko.stojanov@uns.ac.rs (Z. Stojanov)

0000-0001-6930-5337 (Z. Stojanov)

© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

ICCS-DE 2025 Workshop Proceedings (iccs-de.icc.ru)

DOI: 10.47350/ICCS-DE.2025.01

Clusters in the context of graphs are sets of nodes that have stronger relationships with each other within a cluster than with nodes in other clusters [11, 12]. Within the context of complex STSs, graph clustering methods play a crucial role in uncovering the latent structures of socio-technical networks and in grouping existing entities into specific clusters to improve system performance and scalability, and to enhance resilience [11, 13]. Graph clustering enables detection of functional or social communities and groups, communication bottlenecks and potential structural vulnerabilities. The flexibility of graph clustering methods enables their adaptation to weighted, directed, dynamic, or multilayer networks, making them suitable for analysing a wide range of STSs [14, 15].

The objective of the presented research is to present a method for graph clustering by checking similarity between neighbouring nodes in a sequence of discrete time moments (observations). The nodes that satisfy the similarity criteria for observed period are merged into clusters, leading to the minimal graph structure derived from the initial graph. For the validation of the method, a simulation environment that generates simulation scenarios with initial graphs and air pollution datasets is created. The implementation of the proposed graph clustering method in the case of determining minimal number of air pollution measurement stations in urban environments is presented for the city of Zrenjanin in Serbia.

2. Graph Clustering in Socio-Technical Systems

Clustering is a process of grouping data in an unsupervised manner based on a similarity measure by analysing underlying structure of complex data in a given heterogeneous context [16, 17, 18]. The result of clustering are clusters in which are highly correlated data. Graph clustering relates to grouping nodes in clusters by considering some similarity measure for nodes and edges in the graph [11]. Since the field of graph clustering is very attractive for modern socio-technical systems and new methods and algorithms are emerging [19], the aim of this review is only to gain insight into the field, and not to provide a complete overview of existing methods.

In general, there are two approaches to graph clustering: (1) Classifying nodes into clusters based on the node similarity measures for all nodes in the graph, and (2) Computing a fitness measure over the set of possible clusters to inform the decision on how good or useful clustering is.

For this research, the first group of approaches is of interest because the method proposed in the next section can be classified as node similarity-based clustering method. For this group of methods, the need for grouping nodes in the same cluster is directly related to their similarity. Instead of a similarity measure, a distance measure can be used, which suggests where the cluster boundary should be located, so that including more outside nodes would not drastically increase the intra-cluster distances [11]. Choosing the most suitable similarity or distance measure depends on the implementation domain and stated clustering task, and for that purpose may be used node or edge properties. A variety of graph clustering methods based on similarity measures have been proposed, like a graph-theoretic clustering method based on the Highly Connected Subgraphs (HCS) algorithm that identifies clusters as highly connected subgraphs [20], a new clustering algorithm based on graph connectivity that builds a similarity graph from

data and identifies clusters as connected subgraphs [21], a multi-similarity spectral clustering method for community detection in dynamic networks that creates similarity matrices per time slice based on attributes and topology of the network [22], and a spectral clustering method based on local similarity derived from shared neighbours [23].

Insight into recent literature revealed the use of graph clustering in a variety of domains, such as the reengineering of software architecture [24], software defect prediction [25], smart city resource management [26], cyber-physical power grids [27], or community detection in social networks [28]. Due to the importance of graph clustering in many scientific fields, and its use in a large variety of human life domains, some recent review articles present more detailed reviews of graph clustering methods, such as a review of novel clustering methods [17], a review of fast clustering methods [18], or a review of attributed graphs in which both nodes and edges have attributes describing them [29].

3. Experience

This paper presents a method for minimizing the number of nodes in a graph by merging neighbouring nodes that possess similar characteristics, and model implementation in a simulation environment for minimizing the number of air pollution measurement stations in urban environment. The model involves comparing a predefined set of attributes among adjacent nodes, and when the similarity between them exceeds a specified threshold, the nodes are assigned to the same cluster.

In this section, a graph clustering method and its implementation in a study aimed at determining the minimal number of air pollution measurement stations in urban environments [30] are described. The focus on the article [30] is on detailed presentation of the the implementation of the method in case of the city of Zrenjanin in Serbia, while the focus of this paper is on a detailed presentation of the graph clustering method, especially on algorithms for checking node similarity and merging similar nodes into clusters.

3.1. A Method for Graph Clustering Based on Attribute Similarity of Neighbouring Nodes

The method constructs a clustered graph by determining similarity between neighbouring vertices (nodes) in an unweighted and undirected graph defined as $G = (V, E)$, where V is a set of M vertices and E is a set of edges.

To simplify comparison and merging of neighbouring nodes, a graph is presented as a set of adjacency lists associated to nodes. Adjacency list for node v is defined as $Adj(v) = \{u \in V | \{v, u\} \in E\}$.

Description of the method assumes setting up the basis of the scenario for presenting and manipulating unweighted and undirected graph in N observations in discrete time moments (observations). These basis includes:

- The graph contains M nodes v_i, where $i = 1, ..., M$.
- Each node v_i is presented with a set of K attributes a_h, where $h = 1, ..., K$. The values of each attribute a_h are in the set $[min_h, max_h]$.

- For each node, N observations in discrete time moments of each attribute are considered, so we have a_{hip}, where $p = 1, ..., N$. The naming convention is that for each attribute, the attribute index is first listed, then the node index, and finally the observation index.

The method execution is divided in two phases. In the first phase, similarity between neighbouring nodes is checked and a similarity matrix S is constructed for the graph. In the second phase, merging of similar nodes into clusters is based on values in the similarity matrix S and node similarity threshold T.

3.1.1. Checking the Similarity of Nodes

The similarity check is based on comparing the corresponding attributes of all nodes (e.g., the $h-$th attributes of nodes are compared with each other). For this similarity check, the similarity threshold of the $h-$th attribute is defined as T_h. The corresponding attributes are similar if their absolute difference is less than T_h.

For each individual observation of the graph at a discrete observation moment p, a threshold of nodes similarity K_T is defined based on the number of similar attributes, where $1 < K_T < K$. Nodes are similar if they have at least K_T similar attributes at the observed moment p.

To observe the similarity of nodes at the level of a selected observation moment p, a notation for the similarity of nodes i and j at the observation moment os_{ijp} (*observation similarity*) is introduced. The total similarity of two nodes after comparing in all observation moments is the value s_{ij} and it is entered into the similarity matrix S of the graph G. Pseudo algorithm for creating the similarity matrix S for graph G is presented in Algorithm 1.

By executing Algorithm 1, an absolute similarity matrix S for the graph G is constructed. Each element s_{ij} of the matrix S belongs to the set $\{0, 1, ..., N\}$. The higher the value of s_{ij}, the more similar the nodes v_i and v_j are.

3.1.2. Merging Similar Nodes into Clusters

Merging similar neighbouring nodes in a cluster is based on checking how values from absolute similarity matrix S relate to the overall similarity threshold T. To simplify notation and further calculations, the following terms are introduced:

- Relative similarity matrix RS with elements $\sigma_{ij} \in [0, 1]$, which are calculated as $\sigma_{ij} = \frac{s_{ij}}{N}$.
- Relative threshold $\tau = \frac{T}{N}$, where $\tau \in [0, 1]$. The greater the similarity of nodes, the closer the τ value is to 1.

Further, for the algorithm of merging similar neighbouring nodes into a cluster based on attribute similarity, it is necessary to introduce the concept of *one-dimensional normalized attributes value* A_i for each node v_i. For this purpose, a maximum value $A_{hi} = \max_p A_{hip}$ is introduced for each attribute a_h for node v_i for all N observations.

Further, normalized maxim value for each attribute a_h for node v_i is defined as

$$norA_{hi} = \frac{A_{hi}}{max_h}.$$

Algorithm 1 Create the similarity matrix S for graph G

```
Input: G — Initial graph
Input: M — Number of nodes
Input: K — Number of node attributes
Input: T_h — Similarity threshold of the h/th attribute
Input: K_T — Number of similar attributes for two nodes
Input: N — Number of observations
Output: S — Similarity matrix for graph G
{Check all nodes in graph G}
for i = 1 to M do
  for j = 1 to M do
    s_ij ← 0
    {Check all N observations}
    for p = 1 to N do
      os_i,j,p ← 0
      {Check all K attributes of observed nodes}
      for h = 1 to K do
        if |a_h,i,p − a_h,j,p| < T_h then
          os_i,j,p ← os_i,j,p + 1
        end if
      end for
      {Update the value for observed nodes in similarity matrix}
      if os_i,j,p ≥ K_T then
        s_ij ← s_ij + 1
      end if
    end for
  end for
end for
```

Finally, *one-dimensional normalized attributes value* is determined as a function $A : V \to [0, 1]$. For a node v_i it is defined as

$$A(v_i) = \frac{1}{K} \sum_h norA_{hi}.$$

Based on the the values $A(v_i)$ for initial set of nodes V, an ordered list of nodes is created

$$\overline{V} = (\overline{v}_1, \overline{v}_2, ..., \overline{v}_M), A(\overline{v}_i) \geq A(\overline{v}_{i+1})$$

Since the list of nodes $\overline{V}$ contains initial set of nodes but in new order (with decreasing $A(v_i)$ values), new adjacency lists are created to reflect new ordering of nodes. New adjacency lists are $Adj(\overline{v}_i), i = 1, ..., M$.

Created list of nodes $\overline{V}$ and adjacency lists $Adj(\overline{v}_i)$ are used in the algorithm for merging nodes into clusters by starting from the node with the highest $A(v_i)$, which is $\overline{v}_1$. Algorithm for merging similar neighbouring nodes in clusters is presented in Algorithm 2.

For storing information about clustered graph after merging nodes into clusters, an empty *Clustered Graph* set $CG = \{\}$ is created. The elements of the set CG will be calculated as ordered pairs in the form $(C, Adj(C))$, where C denotes a cluster in the clustered graph, while $Adj(C)$ denotes edges from the cluster C to neighbouring clusters.

Algorithm 2 Create clustered graph by merging similar neighbouring nodes

Input: $\overline{V} = \{\overline{v}_1, \overline{v}_2, \ldots, \overline{v}_M\}$, $Adj(\overline{v}_i)$ — List of nodes and their adjacency lists
Input: $RS = [\sigma_{i,j}]$ — Relative similarity matrix
Input: τ — Relative nodes similarity threshold
Output: CG — Clustered graph
$CG \leftarrow \emptyset$
$Active \leftarrow \overline{v}_1$
{*Pass through the list of nodes starting from the node with the highest* $A(v_i)$}
while $NEXT(Active)$ exists **do**
  $C(Active) \leftarrow \{Active\}$
  for each $\overline{v}_j \in \overline{V} \setminus \{Active\}$ **do**
    Remove $Active$ from $Adj(\overline{v}_j)$
  end for
  {*Check a list of adjacent nodes for the active node*}
  for all $u \in Adj(Active)$ **do**
    if $\sigma_{Active,u} \geq \tau$ **then**
      {*Add adjacency list of node* u *to the adjacency list of the active node*}
      $Adj(Active) \leftarrow Adj(Active) \cup Adj(u)$
      {*Add node* u *to the cluster of the active node*}
      $C(Active) \leftarrow C(Active) \cup \{u\}$
      Remove u from $\overline{V}$
      for each $\overline{v}_j \in \overline{V} \setminus \{Active\}$ **do**
        Remove u from $Adj(\overline{v}_j)$
      end for
    end if
  end for
  {*Add the cluster of the active node to the clustered graph*}
  $CG \leftarrow CG \cup \{(C(Active), Adj(Active))\}$
  $temp \leftarrow Active$
  $Active \leftarrow NEXT(Active)$
  Delete $temp$
end while

3.2. Method Implementation

Presented graph clustering method is implemented in a study aimed at determining the minimum number of air pollution measurement stations in urban city environment [30]. The implementation assumes covering the entire urban area of the city with a large number of

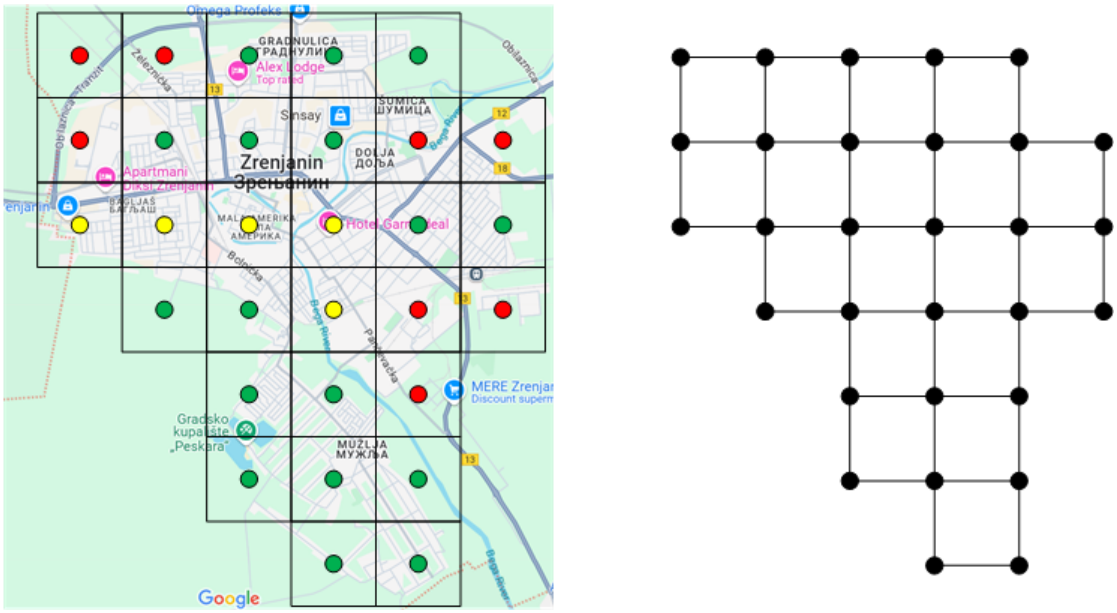


Figure 1: Sample graph with measurement stations for air pollution monitoring in urban settings

densely distributed air pollution measurement stations. The measurement stations are represented as nodes in a graph, each one covering a square region. The edges are constructed to connect neighbouring measurement stations (square regions). A typical scenario for creating a graph with nodes representing air pollution measurement stations placed in the city of Zrenjanin is presented in Figure 1. For the presented scenario measuring stations are 1000 meters apart.

The scenario for distributing measurement stations in the city area assumes considering the zone types in the city. Measurement stations in residential zones are marked with green circles, in commercial zones with yellow circles, and in industrial zones with red circles (see Figure 1). This fact does not affect the graph construction, but it affects the construction of air pollution data sets for specific zones.

The scenario for monitoring air pollution in the city includes the following pollutants: Nitrogen Dioxide (NO_2), Carbon Monoxide (CO), Sulfur Dioxide (SO_2), and Particulate Matter (PM10 and PM2.5). Typical values for residential, commercial and industrial zones in a city, taken from leading international air pollution monitoring organizations (European Environment Agency, World Health Organization) and the World Air Quality Index project are shown in Table 1. These values are used to generate random datasets for the measurement stations in corresponding zones.

3.2.1. Simulation Environment

Method implementation includes the design of a simulation environment for creating scenarios with measurement stations spread over the city area, and data sets for measurement stations based on their locations in different city zones. A simulation environment is created as a

Table 1
Typical values of air pollutants in different city zones used in creating random datasets for measurement stations in the simulation scenarios

	NO_2 (μg/m^3)	**CO (ppm)**	**SO_2 (μg/m^3)**	**PM10 (μg/m^3)**	**PM2.5 (μg/m^3)**
Range for residential zones	0–40	0–3	0–10	0–50	0–30
Range for commercial zones	20–60	2–4	4–20	30–70	20–50
Range for industrial zones	40–200	4–50	20–200	100–500	40–150
Average for residential zones	20	1.50	5	25	15
Average for commercial zones	40	3	12	50	35
Average for industrial zones	120	27	110	300	95
Max value for all zones	200	50	200	500	150
Min value for all zones	0	0	0	0	0

software solution implemented with Jakarta EE platform technologies [31]. The simulation environment enables the creation of the initial graph scenarios and datasets for measurement stations, and implementation of the presented graph clustering method for determining the minimal number of measurement stations that cover the whole city area. Datasets are created as MS Excel files by using Apache POI library [32], which is typical format for air pollution datasets. Data management and exchange between JEE software components is done by using XML [33]. The architecture of the created simulation environment is presented in Figure 2.

The two main modules of the simulation environments are *Scenario Generator* and *Scenario Optimizer*. The first one is used for generating scenarios for the simulation, while the second one analyses created initial scenario and calculates the minimal number of measurement stations for the city area by implementing the proposed graph clustering method. The *Utility* module contains auxiliary functionalities that enable the implementation of scenario creation and manipulation functionalities (graph manipulation utility, data processing utility, and method utility).

Scenario Generator uses external data, including local area maps, information about a city (spread of residential, commercial, and industrial zones), and typical values for air pollution in urban city areas for creating a scenario.

The measurement stations that remain in the graph actually represent clusters that include neighbouring measurement stations with similar characteristics of measured values. Each cluster is represented by one measuring station that has the most critical measured pollutant values. Each cluster is represented by one measuring station that has the most critical measured pollutant values.

3.2.2. Method Adaptation

Measurement stations distributed in the city area are treated as nodes in the graph. For each node, the attributes are measured values for five pollutants: Nitrogen Dioxide (NO_2), Carbon Monoxide (CO), Sulfur Dioxide (SO_2), and Particulate Matter (PM10 and PM2.5).

A unit measurement at a measuring station i is labelled as m_{hip}. Unit measurement m_{hip} corresponds to node attribute a_{hip} in the presented clustering method. For the unit of measurement

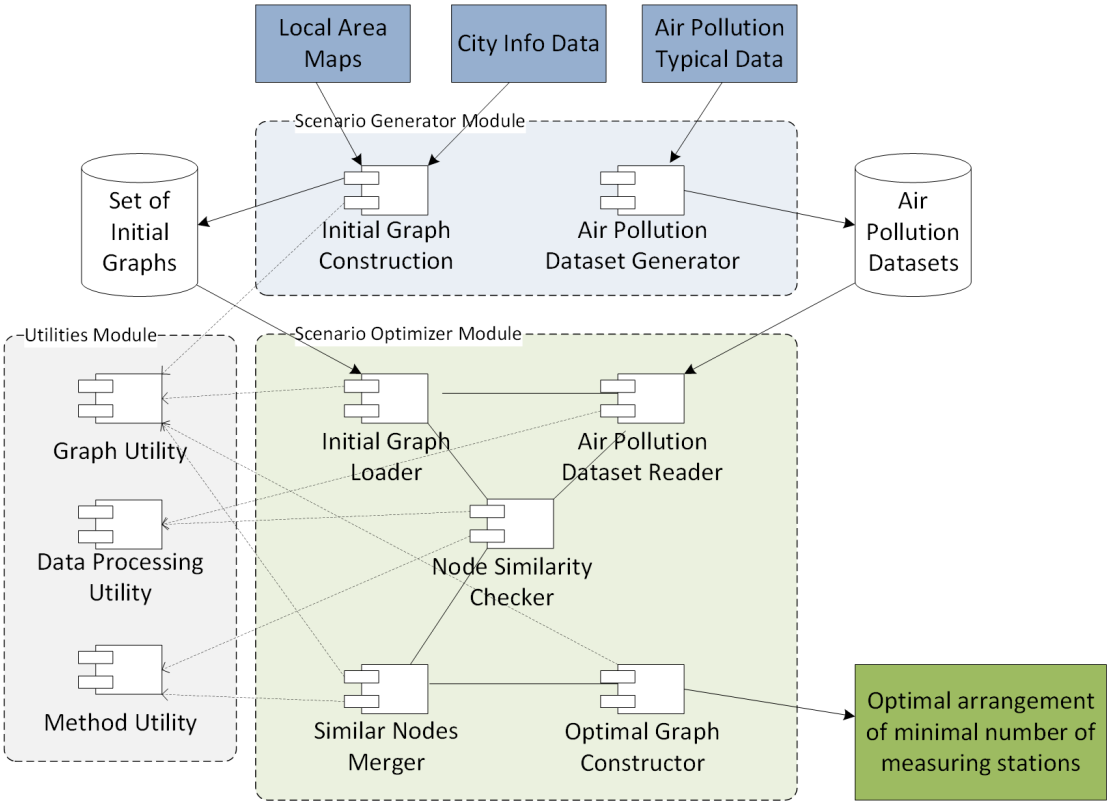


Figure 2: Simulation environment architecture

m_{hip} the indexes are:

- $h = 1, ..., K$ – index representing measurement values for pollutants at each node.
- $i = 1, ..., M$ – index representing the node in the set of nodes in the graph.
- $p = 1, ..., N$ – index representing the ordinal number of a measurement in a set all measurements.

Checking the similarity of two measured values (the same air pollutant at two neighbouring measurement stations), the similarity thresholds T_h are defined for each measured value (node attribute). For each discrete time moment, for all measurement values, the overall similarity threshold K_T is set (how many measured values should be similar on neighbouring nodes), where $1 < K_T \leq K$. The similarity checks for two measured values h of two neighbouring nodes i and j at time p is calculated as

$$|m_{hip} - m_{hjp}| < T_h \times Range_h, \quad \text{where } Range_h = \max_h - \min_h .$$

If at least K_T measured values of two neighbouring measurement stations at a given point p are similar (satisfy the similarity threshold for that value), overall similarity for these two

measurement stations in the similarity matrix S is increased by 1 ($s_{ij} = s_{ij} + 1$). Values T_h, $Range_h$, max_h and min_h are taken from Table 1 for all measured air pollutants.

After checking the similarity between neighbouring nodes for all N measurements (time moments), the similarity matrix S contains similarity measures for all neighbouring nodes in the range from 0 to N. For non-neighboring nodes similarity value is 0. Further, the relative similarity matrix RS with elements σ_{ij} and the relative threshold $\tau = \frac{T}{N}$ are calculated.

The following values were set in the simulation environment to check their influence on the similarity level between neighbouring nodes:

- Distance between measurement stations: $Dist \in \{500m, 1000m\}$
- Total number of measurements at each measurement station (simulation of one measurement per day) $N = 60$.
- $T_h \in \{10\%, 20\%\}$. Threshold T_h is the same for all pollutants to simplify the simulation.
- $K_T \in \{K - 1, K\}$, where $K = 5$
- $\tau \in \{0.8, 0.9, 1.0\}$

3.2.3. Results

Simulation was performed for two global scenarios. The first one has a 500m distance between measurement stations and 92 measurement stations, while the second one has a 1000m distance between measurement stations and 30 dispersed measurement stations. The total number of measurements N in both cases is 60. With proposed variations of parameters T_h, K_T and τ, 12 different sub-scenarios can be identified for both global scenarios, and for each scenario 10 simulations with 10 different datasets were performed.

The construction of a graph with a minimum number of measurement stations by clustering the initial graph is shown in Figure 3. For illustration, an example scenario with 30 measuring stations with the distance of 500m, $T_h = 20$, $K_T = K - 1$, and $\tau = 0.8$ is shown. The resulting minimal graph contains 7 nodes representing the 7 resulting clusters, and represents the scenario with the highest minimization of 77%.

As a result of 10 simulations , the average number for the minimal number of measurement stations is rounded to an integer value. After this initial simulation (the 1st phase minimization), the five control simulations were performed (the 2nd phase minimization) with five created minimal graphs and five new datasets for air pollutants. The results of minimizing the number of measuring stations in percentages for scenarios with 1000m and 500m distances between measuring stations are shown in Tables 2 and 3 respectively. For labelling different scenarios in Tables the following format is used: $(Dist, T_h, K_T, \tau)$.

The results obtained from the simulation indicate the following:

- Greater minimization of the number of measurement stations is obtained when the initial number of stations is 30, i.e., when the initial distance between the measurement stations is larger. This means that increasing the initial number of measurement stations will not lead to a more efficient minimization of the number of stations.
- Simulation results after the control simulations for all scenarios are, in the majority of cases, the same or slightly different than the results after the initial simulation (insignificant minimization of the graph). These results confirm the correctness and efficiency of the presented graph clustering method.

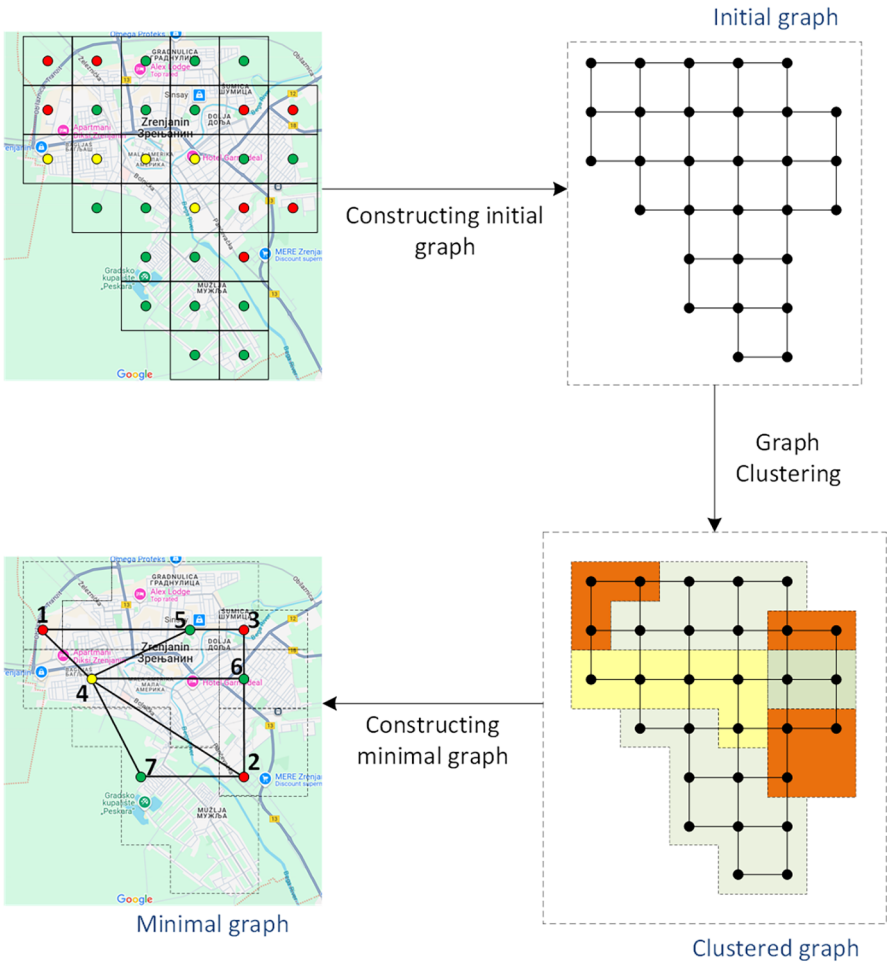


Figure 3: Creating a graph with a minimum number of measurement stations by clustering the initial graph

3.3. Discussion

The main advantages of the presented graph clustering method and presented simulation environment are their adaptability and extensibility.

Method adaptability is reflected in the selection of simulation parameters in the method, such as thresholds for determining the similarity of neighbouring nodes (attribute-related thresholds and node-related thresholds), the number of attributes per node, and the number of time points for observing nodes. In addition, different application scenarios may require adaptation of the similarity checking method at the level of individual or all node attributes, but also at the level of the nodes themselves (e.g., if we consider different social networks, professional or expert networks, distributed algorithms for complex engineering or scientific calculations, urban traffic and logistic networks).

Method extensibility is reflected in the addition of new variables that can affect nodes' attributes

Table 2
Minimization in percentage of initial graphs with measurement stations distance of 1000m

Scenario	1st phase Minimization [%]	2nd phase Minimization [%]
S(1000,10,K-1,0.8)	70	70
S(1000,10,K-1,0.9)	53	57
S(1000,10,K-1,1.0)	3	3
S(1000,10,K,0.8)	60	60
S(1000,10,K,0.9)	43	43
S(1000,10,K,1.0)	0	0
S(1000,20,K-1,0.8)	77	77
S(1000,20,K-1,0.9)	63	63
S(1000,20,K-1,1.0)	10	10
S(1000,20,K,0.8)	53	57
S(1000,20,K,0.9)	47	47
S(1000,20,K,1.0)	3	3

Table 3
Minimization in percentage of initial graphs with measurement stations distance of 500m

Scenario	1st phase Minimization [%]	2nd phase Minimization [%]
S(500,10,K-1,0.8)	66	67
S(500,10,K-1,0.9)	48	48
S(500,10,K-1,1.0)	2	2
S(500,10,K,0.8)	51	51
S(500,10,K,0.9)	34	34
S(500,10,K,1.0)	0	0
S(500,20,K-1,0.8)	68	70
S(500,20,K-1,0.9)	55	55
S(500,20,K-1,1.0)	8	8
S(500,20,K,0.8)	54	54
S(500,20,K,0.9)	38	38
S(500,20,K,1.0)	4	4

in a way that nodes mutually affect each other, leading to the use of directed and weighted graphs (for example, modelling weather conditions in the presented implementation of the clustering method, where wind direction may influence air pollutants in different city zones).

Simulation environment extensibility relates to enabling the support for importing datasets from other sources and in other formats such as XML or JSON, which are commonly used in distributed software systems. This would enable testing the presented method with different datasets and potentially improving its characteristics.

And finally, the use of simulation environments for testing and fine-tuning specific methods is cost and time-effective, especially in the case of their implementation in complex distributed environments. In the presented implementation scenario, it is easier and cheaper to simulate and test measurements of air pollutants in the simulation environment than to install and maintain a set of measurement stations over a long time period to achieve the optimal or minimal number of measurement stations.

4. Conclusions

This paper presents a novel method for graph clustering based on node attribute similarity, together with a simulation environment for validating the method. In addition, method adaptation and implementation for determining the minimal number of air pollution measurement stations in urban settings is presented. Despite the stated advantages and benefits of the presented method and simulation environment, several future research directions are interesting for additional validation and improvements of the presented method.

The first research direction relates to comparing the proposed clustering method with other graph clustering methods on the same datasets. These comparisons will enable positioning this method in the area of graph clustering methods and help in finding potential method improvements.

The second research direction relates to developing more specific similarity measures based on the probability of potential states of nodes and their attributes over a long period, the semantics of the data stored in node attributes, or the correlation among data in node attributes.

And finally, the development and improvement of the simulation environment, and its evolution to a real system used in specific socio-technical systems (social networks, recommendation systems, logistics and transportation, natural eco systems), is an attractive direction, leading to empirical testing of the correctness and usability of the proposed method.

References

[1] G. H. Walker, N. A. Stanton, P. M. Salmon, D. P. Jenkins, A review of sociotechnical systems theory: a classic concept for new command and control paradigms, Theoretical Issues in Ergonomics Science 9 (2008) 479–499. doi:10.1080/14639220701635470.

[2] P. N. Edwards, S. J. Jackson, G. C. Bowker, C. P. Knobel, Understanding infrastructure: Dynamics, tensions, and design. Report of a Workshop on "History & Theory of Infrastructure: Lessons for New Scientific Cyberinfrastructures", resreport, Ann Arbor, MI, US, 2007.

[3] G. Baxter, I. Sommerville, Socio-technical systems: From design methods to systems engineering, Interacting with Computers 23 (2011) 4–17. doi:10.1016/j.intcom.2010.07.003.

[4] A. Barrat, M. Barthelemy, A. Vespignani, Dynamical Processes on Complex Networks, Cambridge University Press, Cambridge, UK, 2008.

[5] W. M. Fox, Sociotechnical system principles and guidelines: Past and present, The Journal of Applied Behavioral Science 31 (1995) 91–105. doi:10.1177/0021886395311009.

[6] Z. Stojanov, J. Stojanov, G. Jotanovic, D. Dobrilovic, Weighted networks in socio-technical systems: concepts and challenges, in: I. Bychkov, A. Tchernykh, A. Feoktistov (Eds.), Proceedings of the 2nd International Workshop on Information, Computation, and Control Systems for Distributed Environments (ICCS-DE 2020), volume 2638, CEUR Workshop Proceedings, Irkutsk, Russia, 2020, pp. 265–276.

[7] J. L. Gross, J. Yellen (Eds.), Handbook of graph theory, Discrete mathematics and its applications, CRC Press, Taylor & Francis Group, Boca Raton, FL, USA, 2003.

[8] K. K. Meng, D. Fengming, , T. E. Gum, Introduction to graph theory, World Scientific Publishing, Singapore, 2007.
[9] A. S. da Mata, Complex networks: a mini-review, Brazilian Journal of Physics 50 (2020) 658–672. doi:10.1007/s13538-020-00772-9.
[10] F. Hu, A. Mostashari, J. Xie (Eds.), Socio-Technical Networks: Science and Engineering Design, CRC Press, Taylor & Francis Group, Boca Raton, FL, USA, 2011.
[11] S. E. Schaeffer, Graph clustering, Computer Science Review 1 (2007) 27–64. doi:10.1016/j.cosrev.2007.05.001.
[12] A. Dasgupta, J. Leskovec, K. J. Lang, M. W. Mahoney, Community Structure in Large Networks: Natural Cluster Sizes and the Absence of Large Well-Defined Clusters, Internet Mathematics 6 (2009) 29–123. doi:10.1080/15427951.2009.10129177.
[13] M. Newman, Networks: An Introduction, Oxford University Press, New York, USA, 2010.
[14] Z. Yang, R. Algesheimer, C. J. Tessone, A comparative analysis of community detection algorithms on artificial networks, Scientific Reports 6 (2016) 30750. doi:10.1038/srep30750.
[15] T. Chakraborty, A. Dalmia, A. Mukherjee, N. Ganguly, Metrics for community analysis: A survey, ACM Computing Surveys 50 (2017). doi:10.1145/3091106.
[16] J. Kleinberg, E. Tardos, Approximation algorithms for classification problems with pairwise relationships: metric labeling and markov random fields, Journal of the ACM 49 (2002) 616–639. doi:10.1145/585265.585268.
[17] M. Hloch, M. Kubek, H. Unger, Graph-based Clustering Algorithms – A Review on Novel Approaches, Advances in Science, Technology and Engineering Systems Journal 6 (2021) 21–28. doi:10.25046/aj060403.
[18] J. Xue, L. Xing, Y. Wang, X. Fan, L. Kong, Q. Zhang, F. Nie, X. Li, A comprehensive survey of fast graph clustering, Vicinagearth 1 (2024) 7. doi:10.1007/s44336-024-00008-3.
[19] V. Estivill-Castro, Why so many clustering algorithms: a position paper, ACM SIGKDD Explorations Newsletter 4 (2002) 65–75. doi:10.1145/568574.568575.
[20] E. Hartuv, R. Shamir, A clustering algorithm based on graph connectivity, Information Processing Letters 76 (2000) 175–181. doi:10.1016/S0020-0190(00)00142-3.
[21] J. Wan, K. Zhang, Z. Guo, D. Miao, A new clustering algorithm based on connectivity, Applied Intelligence 53 (2023) 20272–20292. doi:10.1007/s10489-023-04543-2.
[22] X. Qin, W. Dai, P. Jiao, W. Wang, N. Yuan, A multi-similarity spectral clustering method for community detection in dynamic networks, Scientific Reports 6 (2016) 31454. doi:10.1038/srep31454.
[23] Z. Cao, H. Chen, X. Wang, Spectral clustering based on the local similarity measure of shared neighbors, ETRI Journal 44 (2022) 769–779. doi:10.4218/etrij.2021-0230.
[24] U. Desai, S. Bandyopadhyay, S. Tamilselvam, Graph neural network to dilute outliers for refactoring monolith application, Proceedings of the AAAI Conference on Artificial Intelligence 35 (2021) 72–80. doi:10.1609/aaai.v35i1.16079.
[25] X. Wang, L. Lu, Q. Tian, H. Lin, Ic-graf: An improved clustering with graph-embedding-based features for software defect prediction, IET Software 2024 (2024) 8027037. doi:10.1049/2024/8027037.
[26] A. A. Kapanski, R. V. Klyuev, A. E. Boltrushevich, S. N. Sorokova, E. A. Efremenkov, A. Y. Demin, N. V. Martyushev, Geospatial clustering in smart city resource management: An initial step in the optimisation of complex technical supply systems, Smart Cities 8 (2025).

doi:10.3390/smartcities8010014.

[27] N. Jacobs, S. Hossain-McKenzie, S. Sun, E. Payne, A. Summers, L. Al-Homoud, A. Layton, K. Davis, C. Goes, Leveraging graph clustering techniques for cyber-physical system analysis to enhance disturbance characterisation, IET Cyber-Physical Systems: Theory & Applications 9 (2024) 392–406. doi:https://doi.org/10.1049/cps2.12087.

[28] K. Georgiou, C. Makris, G. Pispirigos, A distributed hybrid community detection methodology for social networks, Algorithms 12 (2019). doi:10.3390/a12080175.

[29] C. Bothorel, J. D. Cruz, M. Magnani, B. Micenkova, Clustering attributed graphs: Models, measures and methods, Network Science 3 (2015) 408–444. doi:10.1017/nws.2015.9.

[30] Z. Stojanov, V. Brtka, G. Jotanovic, G. Jausevac, D. Perakovic, D. Dobrilovic, Graph clustering-based model for optimizing the number of air pollution measurement stations in urban environments, Mobile Networks and Applications (2025). doi:10.1007/s11036-025-02455-8.

[31] Eclipse Foundation AISBL, Jakarta EE | Cloud Native Enterprise Java | Java EE | The Eclipse Foundation, https://jakarta.ee/, 2025. Last accessed: 2025-05-28.

[32] The Apache Software Foundation, Apache POI™ – the Java API for Microsoft Documents, https://poi.apache.org/, 2025. Last accessed: 2025-05-28.

[33] Eclipse Foundation AISBL, Jakarta XML Web Services 4.0 | Jakarta EE | The Eclipse Foundation, https://jakarta.ee/specifications/xml-web-services/4.0/, 2025. Last accessed: 2025-05-28.

Methodology of The Manufacturing Companies Viability Evolutionary Management[1]

Igor Kimyaev [1], Boris Sokolov [2] and Valerii Zakharov [2]

[1] *NorNickel Sputnik LLC, 11 korp. 2, Staropetrovsky Av., Moscow, 125130, Russian Federation*
[2] *St. Petersburg Federal Research Center of the Russian Academy of Sciences (SPC RAS), 39, 14th Line V.O., Saint-Petersburg, 199178, Russian Federation*

Abstract

The paper discusses promising approaches to enhancing the efficiency of vertically integrated business entities through deliberate and systematic enhancement of the complexity of components and the internal structure of information control systems, ensuring the required viability level of these entities. This direction is based on the methodology of complexity control, developed by the authors, and forms the foundation of a methodological approach that describes the systematic and evolutionary replacement of operators and technologists as decision-makers with functionally equivalent software and hardware control systems used in existing and future production facilities.

Keywords

control object, «black box», integrated control system, human factor, viability, complexity control

1. Introduction

A manufacturing company is a complex economic activity object (EAO), consisting of a set of interrelated control objects (CO) and control subjects (CS), united in physical, territorial or functional-logical domains [1-3]. The vitality and viability of CO and EAO as a whole is ensured by material-energy and organizational-information components based on vertical and horizontal flows of resources, including material, information, energy and financial, with multiple feedbacks [4]. Coordination of the work of these components is carried out by the organizational and management apparatus that processes internal and external information flows to form control actions. To support the persons who make, justify and implement decisions (DM, PJD, PID), multilevel integrated information and management systems (IIMS) are developed, partially or fully automating their activities.

In accordance with the principle of necessary diversity, the complexity of an IIMS should correspond to the complexity of EAO, which requires the construction of multidimensional quantitative-qualitative models, including digital twins (DTs) [4]. However, the synthesis of adequate DTs, taking into account all functionality and external influences, is often impossible due to limitations of tools, methodological approaches and resources. Additional complexity is introduced by the presence of legacy operational-technological and information technology (OT/IT) systems that provide local automation [2,3].

The current level of technology does not allow to fully formalize the management processes, which necessitates the involvement of cognitive abilities of the DM [5-6], [7]. Their participation is most significant under conditions of uncertainty, multicriteria, incomplete data and insufficient formalization of the subject area. However, erroneous decisions of operators can lead to a decrease in efficiency or accidents, which actualizes the task of minimizing the human factor.

It is proposed to consider its viability (VAb) — the ability to maintain key production and business indicators (marginality, resource intensity, turnover of funds, etc.) at a given life cycle (LC) interval —

[1] The study was funded by the Russian Science Foundation Grant No. 24-19-00823

7th International Workshop on Information, Computation, and Control Systems for Distributed Environments (ICCS-DE 2025), July 7–11, 2025, Irkutsk, Russia
EMAIL: igor95a@mail.ru (I. Kimyaev); sokolov_boris@inbox.ru (B. Sokolov); valeriov@yandex.ru (V. Zakharov)
ORCID: 0000-0002-5362-4445 (I. Kimyaev); 0000-0002-2295-7570 (B. Sokolov); 0000-0002-2086-2041 (V. Zakharov)
© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).
ICCS-DE 2025 Workshop Proceedings
DOI: 10.47350/ICCS-DE.2025.02

as an integral characteristic of EAO stability (three-component human-machine structure of DM ↔ IIMS ↔ EAO) [4], [14], [13], [15], [17]. At present, there is no generally accepted formalization of VA, which makes it urgent to develop a methodology for the distribution of control functions between DM and OT/IT systems in the evolutionary development of the EAO ↔ IIMS complex.

2. Hierarchy and components of IIMS for complex EAO

It is noted in [5, 6, 8, 10, 11] that the information circulating in the EAO structure and processed by IIMS means can be considered as one of the main types of their resources, allowing in some situations to compensate for the lack of other types of resources. The generally accepted methods of improving the quality of control of EAO and its individual CO as components imply providing them with the necessary sources of quantitative data on the current physical state (Controlled Variable (CV)), acting disturbances (Disturbance Variable (DV)) and required control actions (Manipulated Variable (MV) [1-3,9-11]. At the same time, on the basis of quantitative assessment of vectors CV, DV, MV, using the appropriate mathematical tools, multifactor DT-models of individual CO, technological lines and production domains can be formed and integrated, forming generalized DT [7]. The analysis shows that the more detailed the DTs created for individual CO, as well as for EAO and the corresponding IIMS, the more accurately DTs describe and predict their functioning, and the more efficient and stable the control algorithms work, the less costly and more safe, qualitative and productive will be the functioning of EAO as a whole.

Fig. 1 shows how within the production domain, initially described by the conceptual model as a «Black-Box» (BB) with a set of CO as «nested» functional technological units, quantitative and/or qualitative CV, DV, MV are consistently identified. It is also shown how DT are sequentially (on the basis of incoming information and knowledge of experts and DM) detailed and complicated, which allows the initially opaque BB to «brighten» (formally describe) in terms of understanding the work of its internal mechanisms [4,7,10,11].

For the BB components «brightened» by means of the existing formal methods the necessary models and algorithms are created, on the basis of which OT/IT subsystems are formed as components of IIMS [1], which, in its turn, provides individual CO or entire technological lines (domains) with continuous accident-free operation in automatic (automated) mode with the required performance quality indicators. As a result, there is an evolutionary replacement of human factor in individual IIMS circuits, and the very principle underlying this approach can be formulated as «manage better than the best DM».

In order to develop a methodological basis for correct decomposition of complex CO and technological lines (domains) when building DT and creating IIMS components for processing information flows during the evolutionary replacement of DM, let us consider the existing ways of building the EAO management space in the form of hierarchically related functional subsystems.

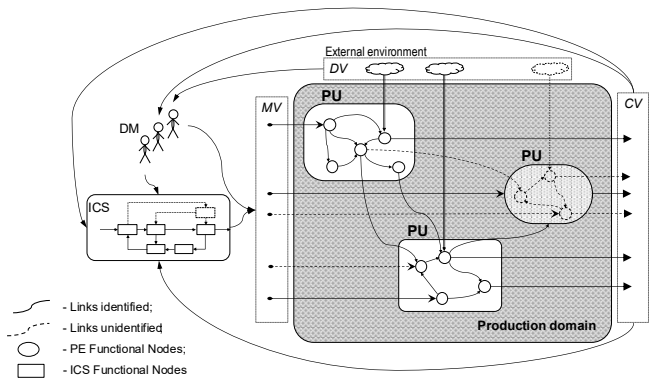


Figure 1: EAO as a «black box», consistently «brightened» by means of complex predictive modeling tools

As an example, Fig. 2 presents a generalized (typical) structure of multilevel IIMS recommended by international standards [2-3], showing the location and rough estimation[2] of the degree of DM involvement in management tasks for a large EAO.

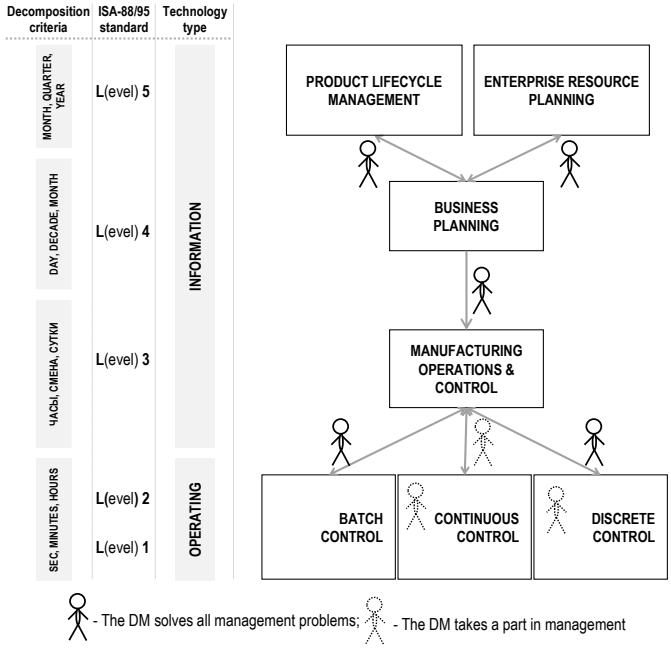


Figure 2: Hierarchical structure of IIMS of a large EAO and proposed criteria for its decomposition by time horizon and management levels

It should be noted that historically the maintenance of EAO in a viable state has been dealt with by a team of DMs of various specializations by processing and analyzing incoming data and information, as well as solving management tasks at all L1...L5 - levels on the basis of acquired knowledge and skills. For example, for L1...L2 - it is maintenance of physical indicators for CO (temperatures, flow rates, pressures, concentrations, etc.) in the established ranges; for L3...L4 - ensuring uniform production utilization by planning the use of its resources and selecting the required indicators of its management, etc.

Summarizing the above, we note that when choosing the place of OT/IT-components to replace DM in the space of control tasks of complex EAO, it is advisable to follow at least two principles: firstly, the quality of information flows processing should be improved due to the creation and use of adequate DT and control models on their basis; secondly, DM should be maximally relieved of the functions of routine processing of the most saturated information flows and regulation by a set of secondary auxiliary subsystems and technological indicators.

3. Methodological approaches to increasing «transparency» of information flow management processes in EAO and IIMS

The development of approaches to increase the "transparency" ("brightening") of the structure of information flow management processes in EAO, which will allow for the formal consideration (describe) external and internal perturbations, will expand the methodological basis for building the IIMS architecture.

Fig. 3 summarizes graphically the information spaces of the three main components of the EAO:

1) «Production domain», consisting of a group of CO1…COn;

[2] The proposed assessment of the degree of DM participation is not exhaustive, but only indicative of the current state of the art for best manufacturing practices

2) «External Environment» as a host of factors, poorly predictable but significant to the effective operation of EAO and its VAb as a whole;

3) «DM» as a set of expert analysts (operators), in specific subject areas, capable of maintaining EAO in VAb state by purposefully compensating (situationally parrying) sets of both external and internal perturbations [5, 6, 10].

In addition, there is a "correction mechanism" by means of which the real production and technological data formed (generated) by CO are compared with the data obtained using Predictive Modeling (PM) tools based on the created DT. Based on the fact of comparison of real and synthesized data, adjustments of characteristics of intermediate DT components are developed for each CO for the purpose of their parametric and structural adaptation and subsequent use in the proactive control of EAO [10].

One of the traditional ways of quantitative assessment of interrelationships of material, energy and information resources and flows within a weakly formalized EAO, aimed at increasing the "transparency" (degree of formal description) of the processes under study, are approaches and methods based on Simulation-stochastic modeling (SSM).

For example, with the help of SSM methods for real technological parameters CV, MV, as indicators of CO and technological lines operation, it is possible in practice to construct response functions in the form of one or a set of analytical polynomials [7, 9]. With the use of these polynomials, it is possible to quantitatively evaluate the relationship between material and information resources and flows for various perturbing influence scenarios with respect to the EAO structure. When applying agent-based logical-dynamic models (LDM), these polynomial dependencies can be replaced by already approximated areas of attainability (A-of-A), the analysis of which allows to identify and predict more complex regularities associated with production activities [7, 10]. At the subsequent steps of EAO and IIMS research, for example, during DT synthesis, the obtained polynomials and A-of-A can be used in the construction and adaptation of DT in the process of functioning of the specified objects and systems.

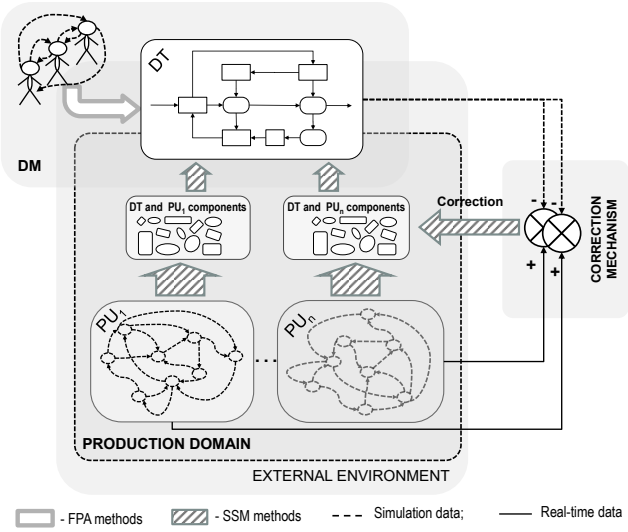


Figure 4: Formation of DT part of EAO by combining SSM and FPM

The analysis shows that one of the most effective approaches to the extraction of expert knowledge and its subsequent representation in a formalized form is the Fuzzy possibility approach (FPA) and the corresponding Fuzzy possibility models (FPM) [9]. Using FPA, it is possible to transform qualitative knowledge of experts (operators-technologists) about internal and external causal relationships in EAO and/or external influences on them into quantitative estimates (e.g., by standard defuzzification means). In this case, the declarative and procedural knowledge of the expert about EAO and IIMS processed by FPA methods, for example, information about DM skills and abilities related to monitoring and control of their components, are also formalized by analytical polynomial(s) reflecting the estimation of real CV, DV when choosing the best MV values. In general, the response functions constructed by the FPA method allow us to obtain quantitative estimates of information flows and processes occurring both in

the internal EAO and IIMS circuits (estimation of MV → CV influence) and outside them (estimation of DV → CV influence).

Below (see Table 1), the calculation models used and the proposed generalized interpretations of the results obtained with their help by SSM and FPM[5] are presented for analysis in a summarized form. The analysis shows that the proposed combined application of SSM and FPM[3] techniques in the processing and analysis of real technological data allows to formally describe and link the structures of physical (material) and information flows $CO_1 ... CO_n$ EAO, to reveal and practically use the declarative and procedural knowledge of DM about the ways and methods of maintaining the required level of VAb EAO [9,10].

Table 1

Analysis of combined application of SSM and FPM techniques

Methodology		Description
Characteristic	Type	
Type of calculation model for output variables	SSM	Calculated value of the output variable with significant coefficients of the approximating polynomial: $$y^{(k)} = b_0 + \sum_{i=1}^{n} b_i^{(k)} \overset{\circ}{x_i} + \sum_{j<u<n}^{n} b_{ju}^{(k)} \overset{\circ}{x_j} \overset{\circ}{x_u} + + b^{(k)}{}_{12...n} \overset{\circ}{x_1} \overset{\circ}{x_2} ... \overset{\circ}{x_n}$$ is formed according to the calculation matrix $\overset{\circ}{\boldsymbol{X}} = \left\| \overset{\circ}{\boldsymbol{X}_{ir}} \right\|_{N\times m}, r = \overline{1,m}$, where $m = n + p$; p – the number of nonlinear coefficients; Significance of the coefficients is determined by Student's *t*-test: $t = \frac{b_m}{s_{b_m}}$, where s_{b_m}- coefficient determination error
	FPM	Estimated value of fuzzy output variable with significant coefficients of approximating polynomial: $$\tilde{y}^{(k)} = \tilde{b}_0 + \sum_{j=1}^{n} \tilde{b}_i^{(k)} \overset{\circ}{\tilde{x}_i} + \sum_{j<u<n}^{n} b_{ju}^{(k)} \overset{\circ}{\tilde{x}_j} \overset{\circ}{\tilde{x}_u} + + \tilde{b}^{(k)}{}_{12...n} \overset{\circ}{\tilde{x}_1} \overset{\circ}{\tilde{x}_2} ... \overset{\circ}{\tilde{x}_n}$$ is formed according to the calculation matrix $\overset{\circ}{\tilde{\mathbf{X}}} = \left\| \overset{\circ}{\tilde{\mathbf{x}}_{ij}} \right\|_{N\times m}$. The model is built based on the formed fuzzy model significance and validity criteria, and a set of polynomial expansion terms approximating the DM implicit knowledge of the EAO component control.
Interpretation of calculation results	SSM	$\mathbf{y} = \mathbf{f}(\mathbf{x}, \mathbf{b}_0, \mathbf{b}_1, ..., \mathbf{b}_n, \mathbf{b}_{12}, ...)$ – coefficients reflecting the specifics of mathematical formalization of the EAO component under study, where $\mathbf{x} = (x_1, x_2, ..., x_r), r = \overline{1,m}$ – real variables specifying the factor space of CO operation as a component of EAO. The approach is to use real process data procedurally.
	FPM	$\tilde{\mathbf{y}} = \tilde{\mathbf{f}}(\tilde{\mathbf{x}}, \tilde{\mathbf{b}}_0, \tilde{\mathbf{b}}_1, ..., \tilde{\mathbf{b}}_n, \tilde{\mathbf{b}}_{12}, ...)$ – coefficients reflecting a quantitative assessment of the DM's knowledge, experience and intuition in relation to the job, where $\tilde{\mathbf{x}} = (\tilde{x}_1, \tilde{x}_2, ..., \tilde{x}_r), r = \overline{1,m}$ – fuzzy variables specifying the factor space of EAO operation based on DM declarative knowledge. The approach is to use the declarative knowledge of the expert.

As a result, the combination of these techniques results in a structurally and quantitatively maximized "clarified" DT component. It should be noted that the DT synthesized in this way will cover all three participants of economic activity, which is practically unattainable when applying SSM and FPM methods separately.

In general, considering the processes of complex predictive modeling (including SSM and FPM) as controlled processes, it is possible to approach the solution of all major problems of evolutionary redistribution of functions in the studied automated (human-machine) EAO control systems at the constructive level. For this purpose, the works [7, 10] proposed specific ways to realize the concept of complexity management based on the system-cybernetic methodology, which implies the use of combined models, methods and algorithms for narrowing the diversity of the external environment and

[3] For a real object, the set of "input" variables **x** and the set of «output» variables y can be either CV or MV, depending on the problem statement; the set DV can be mapped as **x** only.

expanding the diversity of the controlling side on the basis of proactive control of the structural dynamics of the considered man-machine systems [9, 10, 12].

4. Evolution of DM tasks in the creation of model-centered IIMS components

The analysis has shown that the creation of Advanced Process Control (APC) and Real Time Optimizer (RTO) on Real EAO is evolutionary, as a targeted request from business users emerges to improve the efficiency of individual CO and/or process lines [2, 3, 9, 10]. However, in practice, the formation of plans for their implementation is often spontaneous, without a deep methodological justification of the sequence of implementations, component composition of APC/RTO, and extensive coverage of technological objects.

The obligatory technical condition of APC creation is the readiness of components of information CV/DV/MV -flows of L1-level to be included into control models of L2-level, which is actually one of the stages of "brightening" CO as BB and redistribution of control functions between DM and IIMS. The condition of RTO creation is the presence of a group of n CO in the technological line, controlled by their own APCs, the operation of which should be determined by a single production and economic target function. For industrial EAOs, such targets include, for example, maximum productivity, production margin, minimum energy consumption, etc., as well as their combinations.

Fig. 4 presents a generalized map of the evolutionary transformation (or, in other words, development management) of different types of IIMSs depending on the complexity of DT and control models, the saturation of information flows handled by automatic OT/IT subsystems, and the number of EAO components covered.

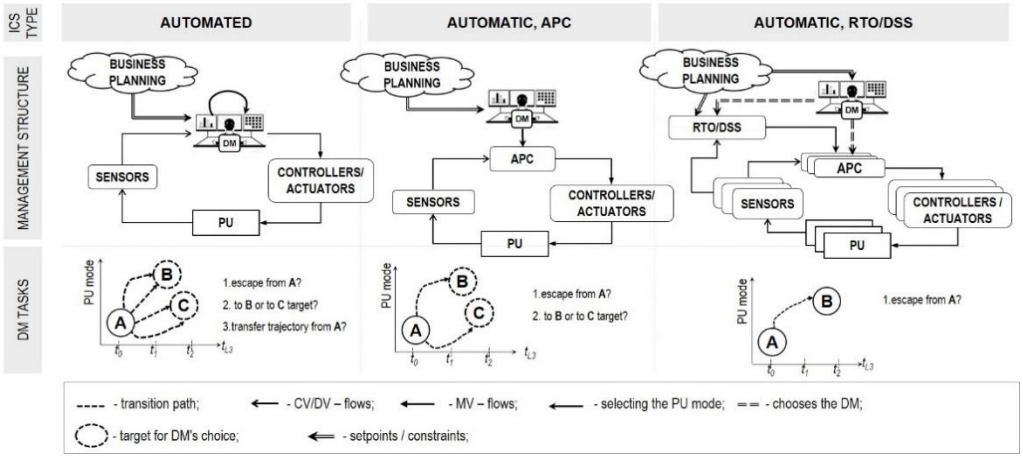


Figure 4: Types of IIMS, structures of their construction, management tasks to be solved, place and role of DM

Fig. 5 shows the qualitative graphs obtained in [9] and reflecting the influence of typical modes of IIMS operation, which are at different stages of evolutionary complexity, on the cost of products:

1) the upper graph shows the modes of CO operation when maintaining the target "Process Indicator", characteristic of the modes "Manual," "Automated," "Automatic, APC," and "Automatic, RTO/DSS."

2) the lower graph shows the specific Cost levels that correspond to the CO operating modes displayed in the upper graph.

Both graphs together illustrate how the dynamics of the evolutionary increase in IIMS complexity from "Automated" to "Automatic, RTO/DSS" increases the so-called "Cost margin".

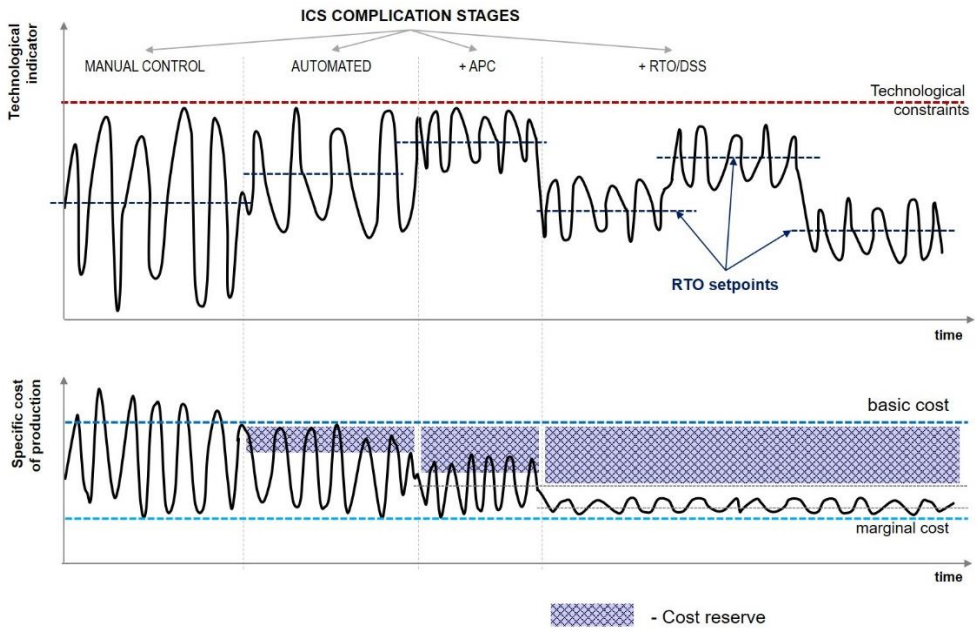


Figure 5: APC (L2) and RTO (L3) are cost management tools as a factor of VAb

The presented correspondence between the IIMS complexity category and the "Cost margin" for a CO (in real production - a group of n CO) implies the possibility to keep its level consistently minimal for the longest possible period of time. This possibility is provided by the IIMS ↔ CO complex, which automatically reconfigures the operating modes of $CO_{1...n}$ to keep Cost at the lowest possible levels following changes in the external conditions for EAO by means of RTO/DSS + APC as tools.

The graphs presented in Fig. 5 show that managing the evolutionary increase in complexity of the IIMS ↔ CO complex allows for the eventual effective implementation of measures to maintain EAO key performance indicators (e.g., Cost) at the required target level. In this case, the generalized EAO performance indicator, as which we consider the VAb indicator, also becomes manageable, since the Cost indicator will be a component of it.

5. Conclusions

The paper presents the results of further development of the complexity management methodology. As promising approaches to solving the problems of multistage redistribution of DM functions of prospective automated control systems for complex technical objects, it is proposed to use the full range of classical and complex predictive modeling. This approach will allow not only to describe at the constructive level difficult to formalize processes of operator control of technological processes, but also to relate the generalized indicator of VAb of EAO and its IIMS to the private performance indicators on the basis of the original fuzzy-possibility convolution [12]. The obtained results can be used in the creation and modernization of advanced monitoring and control systems of complex agro-biotechnical objects. First of all, when redistributing the functions between the agronomist and IIMS.

6. Acknowledgements

The study was funded by a grant from the Russian Science Foundation No. 24-19-00823, https://rscf.ru/en/project/24-19-00823/.

7. References

[1] Lahiri, S. K. **Soft sensors**. In: Multivariable Predictive Control: Applications in Industry. 2017.
[2] **IS/IEC 62264-1** (2003): Enterprise-Control System Integration, Part 1: Models and Terminology.

[3] **ISO 15746-1:2015** Automation systems and integration — Integration of advanced process control and optimization capabilities for manufacturing systems — Part 1: Framework and functional model.
[4] Beer S. **Cybernetics and management**. Management science series. John Wiley & Sons, Inc., New York1959, 214 pp.
[5] Antonov V. V., Konev K. A. **Intelligent decision support method in a typical situation, Design ontology**, 2021, vol. 11, no. 1, pp. 126–136 (in Russian).
[6] Bashlykov A. A., Eremeev A. P. **Expert decision support systems in the energy sector**, Moscow, Publ. MEI, 1994, 213 c (in Russian)
[7] Mikoni S. V., Sokolov B. V., Yusupov R. M. **Qualimetry of models and polymodel complexes**, Moscow, RAN, 2018, 314 p. (in Russian)
[8] Balashov E. P. **Evolutionary synthesis of systems**, Moscow, Radio and communication, 1985, 328 c. (in Russian)
[9] Kimyaev, I.T., Spesivtsev, A.V. (2022). Ontological and Fuzzy-Possibility Approach to the Synthesis of the DM Functional Equivalent for Management of Hierarchical Systems. In: Silhavy, R. (eds) **Artificial Intelligence Trends in Systems**. CSOC 2022. Lecture Notes in Networks and Systems, vol 502. Springer, Cham. https://doi.org/10.1007/978-3-031-09076-9_53.
[10] Okhtilev M. Yu., Sokolov B. V., Yusupov R. M. Intelligent **Technologies for Monitoring and Controlling the Structural Dynamics of Complex Technical Objects**, Moscow, Science, 2006, 410 p. (in Russian)
[11] Trotsky D. V., Gorodetsky V. I. **Scenario-based knowledge model and language for situation assessment and prediction**, SPIIRAS Proceedings, 2009, no. 8, pp. 94–127. (in Russian)
[12] Pavlov A. N., Pavlov D. A., Umarov A. B., Gordeev A. V. **Method of structural-parametric synthesis of configuration multi-mode object**, Informatics and automation, 2022, no. 4(21), pp.812–845. (in Russian)
[13] Ivanov, D, Dolgui, A. (2020). Viability of intertwined supply networks: extending the supply chain resilience angles towards survivability. A position paper motivated by COVID-19 outbreak. International Journal of Production Research, 58(10), 2904-2915.
[14] Ivanov, D, Dolgui, A. (2022). Stress testing supply chains and creating viable ecosystems. Operations Management Research, 15(1-2), 475-486.
[15] Münch, C., & Hartmann, E. (2023). Transforming resilience in the context of a pandemic: results from a cross-industry case study exploring supply chain viability. International Journal of Production Research, 61(8), 2544-2562.
[16] Rostami, O., Tavakoli, M., Tajally, A., & GhanavatiNejad, M. (2023). A goal programming-based fuzzy best–worst method for the viable supplier selection problem: a case study. Soft Computing, 27(6), 2827-2852.
[17] Sarkar, B., & Bhuniya, S. (2022). A sustainable flexible manufacturing–remanufacturing model with improved service and green investment under variable demand. Expert Systems with Applications, 202, 117154.

Adding procedural-parametric polymorphism to the C language

Pavel V. Kosov [1], Alexander I. Legalov [1]

[1] *Higher school of Economics, National research University, 20, Myasnitskaya str., Moscow, 101000, Russia*

Abstract

Program development is often associated with incremental expansion of functionality. Increased reliability and quality in this case can be achieved by minimizing changes to already written code. For instrumental support of evolutionary development, a procedural-parametric paradigm is proposed, expanding the capabilities of the procedural approach. It provides painless extension of both data and functions, using static typing instead of type dereferencing. The work examines the inclusion of procedural-parametric mechanisms in the C language. Additional syntactic structures are proposed that are aimed at supporting the approach. These include: parametric generalizations, specializations of generalizations, generalizing functions, specialization handlers. Their capabilities are being considered. Situations are described in which it is possible to increase the reliability of a program through the use of procedural-parametric polymorphism.

Keywords

programming language, compilation, procedural-parametric programming, polymorphism, evolutionary software development, software reliability

1. Introduction

When developing software systems, various quality criteria must be taken into account. Extending a program without changing previously written code is one of them. It is determined both by incomplete knowledge of the final functionality of programs during its creation and initial operation, and by the need to speed up the product's entry into the market by implementing only a basic set of functions with its subsequent expansion. In such situations, adding new programming constructs, accompanied by changing the code, often leads to errors and unpredictable behavior.

Program extension is largely achieved by dynamically linking objects at runtime. This provides flexible changes to the program structure and allows the use of alternative program objects during calculations. In this case, both explicit and implicit analysis of connected alternatives can be used. The latter defines dynamic polymorphism. It differs from static polymorphism, in which the identification of alternatives is done at compile time.

Initially, the implementation of dynamic polymorphism was proposed for the object-oriented (OO) paradigm. In OO languages with static typing, the key solution was the joint use of inheritance and virtualization mechanisms. Inheritance ensured that the interfaces of parent and child classes were identical, and virtualization made it possible to override methods in child classes. Examples include the languages C++ [1], Java [2] and others.

Support for dynamic polymorphism, based on different principles, began to be included in procedural languages. The Go [3] language implements an interface mechanism that allows you to use functions associated with data structures as alternative handlers. A similar mechanism based on traits is implemented in the Rust [4] programming language. However, the approaches presented above do

[1] 7th International Workshop on Information, Computation, and Control Systems for Distributed Environments (ICCS-DE 2025), July 7–11, 2025, Irkutsk, Russia
EMAIL: pvkosov@hse.ru (A. 1); alegalov@hse.ru (A. 2)
ORCID: 0000-0002-9035-312X (A. 1); 0000-0002-5487-0699 (A. 2)
© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).
ICCS-DE 2025 Workshop Proceedings
DOI: 10.47350/ICCS-DE.2025.03

not directly support the evolutionary expansion of programs in the case of multiple polymorphism associated with the implementation of multimethods. Achieving the required effect is possible only by writing additional code that forms the necessary relationships and is not related to the calculations determined by the problem being solved.

To provide instrumental support for evolutionarily extensible multiple polymorphism, a procedural-parametric (PP) programming paradigm was proposed [5], which made it possible to expand the capabilities of the procedural approach. The solution proposed in it is based on a parametric mechanism for forming relationships between data and the procedures that process them, which can be implemented in various ways [6]. This allows for painless extension of both data and functions in statically typed programming languages. To test the idea, the O2M language was developed, which extends the Oberon-2 [7] programming language. Experiments carried out on its basis made it possible to analyze the capabilities of procedural-parametric programming (PPP). It was shown that the approach can be quite painlessly integrated into existing procedural and functional programming languages, and used in the development of new languages. It is shown that the PP paradigm provides a more flexible evolutionary extension of programs compared to other methods for supporting dynamic polymorphism [8].

Currently, work is underway to include PPP mechanisms in the C programming language, which is one of the five most popular languages according to various ratings. The language is flexible and relatively simple, being widely used in subject areas where the OO approach is not effective. At the same time, many solutions used when developing programs in the C language can lead to errors due to the lack of control over a number of situations during compilation.

This situation often arises because the language supports arbitrary type conversions, often involving dereferencing. Although this often allows for high performance, this approach greatly reduces the reliability of programs. Another problem is that the language does not have a strict relationship between alternative data and their identifying characteristics, which can be arbitrarily entered and used by the programmer. This also leads to errors during program development, which are detected only during execution.

Many of these problems are effectively solved in OO programming languages through the use of OO polymorphism, as well as in the Go and Rust languages. The PP paradigm, along with solving these problems, also allows us to effectively support the evolutionary expansion of multimethods. The basic concepts of the procedural-parametric paradigm, as well as the syntax and semantics of PP mechanisms integrated with the C programming language are presented in [9]. The following describes its main features and also demonstrates how its implementation in the C programming language can improve the reliability of the programs being developed.

2. Basic concepts of the procedural-parametric paradigm

When describing the features of the procedural-parametric paradigm, the following concepts are used:

- basis of specialization;
- generalization specialization;
- parametric generalization;
- instance of a parametric generalization or specialized variable;
- generalizing function;
- parametric specialization handler or specialized function;
- call to a parametric function.

2.1. Basis of specialization

The basis of specialization is understood as any atomic or composite named type, including a previously formed parametric generalization, which in essence is also a composite type. The concept of a basis for specialization does not impose restrictions on the independent use of types in a program

and is introduced only for terminological connection with other concepts. Examples of structures used further as the basis of specializations include the rectangle, triangle, and circle:

```
typedef struct Rectangle {int x, y;} Rectangle;
typedef struct Triangle {int a, b, c;} Triangle;
typedef struct Circle {int r;} Circle;
```

Using `typedef` is not mandatory and was introduced for a more compact presentation of structures in the course of further presentation.

2.2. Generalization specialization

By specialization of a generalization (or «specialization») we mean any of the bases of specialization included as a component in a parametric generalization. Specialization specifies one of the alternatives to generalization. It is semantically close to the alternative field within a union in the C programming language.

2.3. Parametric generalization

A parametric generalization (or «generalization») contains zero to several specializations. It can be extended by new specializations in different compilation units, which distinguishes it from the C language unification. In the proposed C language extension, it is a modification of the structure and is also called a generalized structure, the syntax of which can be described as follows:

GeneralizedStructure = Structure "<" ListOfSpecializations ">" ["const"]
| "<" [TraitPattern] ">".
Feature pattern = ":".
ListOfSpecializations = ListOfSpecializationsByType | ListOfSpecializationsByAttribute.
ListOfSpecializationsByType = TypeName ";" {TypeName ";"}.
ListOfSpecializationsByAttribute = Feature {"," Feature} ":" Type ";"
{Attribute {"," Attribute} ":" Type ";"}.
Feature = identifier.
Type = TypeName | DescriptionOfUnnamedType.

As with the declaration of regular structures, generic structures can also be used in the declaration of `typedef`.

The generalization additionally contains a list of specializations. Uniqueness is determined by a feature that can only be specified by type. In this case, all specializations must be of different types. It is also possible to specify a characteristic with a special identifier, which allows you to have similar specializations. Qualifier `const` prohibits expansion of a generalization by using it as an ordinary union. For example, you can create a generic that specifies the days of the week:

```
struct WeekDay {
    int week_number;
}<
    Monday, Tuesday, Wednesday, Thursday,
    Friday, Saturday, Sunday: void;
> const;
```

Using the `void` type simulates an enumeration. The specialization attributes that precede the type allow you to specify only seven alternatives, each of which can have its own specialization handler associated with it. A common field in the main structure specifies the week number of the year.

In the absence of the `const` qualifier you can simulate an evolutionarily extensible enum type. The appearance of new specializations in the program may not occur immediately, but during the process of its development. In such cases, it is possible to evolutionarily expand the set of specializations. Suppose that at the first stage it is necessary to form a generalization specialization for a rectangle. This can be done by creating the following initial generalized structure:

```
typedef struct Figure{}<Rectangle;> Figure;
```

It is allowed to use the first part of the generalized structure without specifying fields, which indicates the absence of common data for all generated generalizations.

Subsequent expansion of generalization can be carried out either through single or group inclusion of new specializations. The description of their connection is carried out in accordance with the following syntax rules:

AddingSpecializations = LinkToGeneralizedStructure "+" "<" ListOfSpecializations ">".

Then adding a triangle and a circle might look like this:

```
Figure + <Triangle; Circle;>;
```

It is possible that the initial generalization is created empty. In this case, when specifying characteristics of specializations by types, their list will be absent, and its filling will begin with the addition of specializations. For the current example, the initial formation of a rectangle specialization could be implemented as follows:

```
typedef struct Figure {}<> Figure; // There are no specializations
Figure + <Rectangle;>; // Including the rectangle in the generalization
```

Using additional features of specializations to identify allows you to reuse the same type. We use a rectangle to define a rhombus through its diagonals, and to define a segment through its length we use a circle. In this case, with an empty set of specializations, the generalization will look like this:

```
typedef struct Figure2 {}<> Figure2;
```

The extension of the generalization can be described as follows:

```
// Add a rectangle, rhombus, segment
Figure2 + <rect, rhomb: Rectangle; section: Circle;>;
//Add triangle and circle
Figure2 + <trian: Triangle; circ: Circle;>;
```

At the first stage, generalization `Figure2` expanded by a rectangle, a rhombus and a segment of the specified length. The following description adds a triangle and a circle. Generalization extensions can be implemented in different compilation units, each of which will only have its own specializations visible.

2.4. Instances of parametric generalizations

Instances of parametric generalizations are variables formed on the basis of specializations (specialized variables). For each of these variables, the type of specialization is specified. During their description, initial initialization is allowed. It is possible to form both scalar specialized variables and arrays of various dimensions. In addition, you can specify a description of pointers to specialized variables, as well as create arrays of such pointers. Pointers to specializations can only refer to the corresponding specialized variables, in contrast to pointers to generalizations, which can refer to any specializations formed from the corresponding generalization.

DescriptionOfSpecializedVariable = LinkToGeneralizedStructure "." [TypeName | Sign] ListOfSpecializedVariables.

ListOfSpecializedVariables = SpecializedVariable {"," SpecializedVariable}.

SpecializedVariable = {"*"} SpecializedVariableName {"[" Dimension "]"} ["="Initializer].

LinkToGeneralizedStructure = struct GeneralizedStructureName | NameTypedef.

The dimension of the array is specified similarly to the description of the dimension of arrays for ordinary C language variables. Initialization is determined depending on the dimension and type of the variable being formed.

Pointers to generics can refer to any specialization of a given generic, indicating the possibility of dereferencing. The corresponding actions can be carried out during the initial initialization or during the assignment operation.

DescriptionofPointersToGeneralizations = TypeGeneralization ListOfGeneralizedPointers.

ListOfGeneralizedPointers = GeneralizedPointer {"," GeneralizedPointer}.

GeneralizedPointer = "*"{"*"} GeneralizedPointerName {"[" Dimension "]"}["=" Initializer].

Like any C language variables, generic and specialized variables can be global, static, local, or dynamic. Their values can be used as actual parameters when calling functions. As an example, the following options for creating specialized variables and generic pointers can be given:

```
Figure.Triangle t1 = {}<{3,4,5}>, t2;
Figure.Circle c1 = {}<{10}>, *pc1 = &c1, c[10];
Figure2.rect r1, r2[3] = {{}<{1,2}>,{}<{3,4}>,{}<{5,6}>},
```

```
                r3 = <{7,4}>, pr1 = &r1;
Figure *pf1 = &t1, **ppf1 = &pf1;
// Only the week number changes:
struct WeekDay.Monday monday = {35}<>;
```

Features of initializing specializations include specifying the described values inside angle brackets. However, for structure values, you also need to specify additional curly braces, which are not needed if the specialization is a scalar.

2.5. Operations on specialized variables

Over specialized variables, as well as over specialized and generalized pointers, access operations to individual fields for reading data and assignment are possible. It is possible to manipulate individual fields of both the structural part and the specialization. In this case, for the fields of the structural part of a specialized variable, the dot «**.**» acts as a separator, as usual. The generic part field is identified by the sign «@», which may follow the dot «**.**» or the arrow «**->**». For example:

```
t1.@a = 5;
t2 = t1;
pf1 = pc1;
monday.week_number = 24; // Monday of 24th week
```

In cases where specialization fields are structures, they are accessed by indicating the field name, preceded by a dot. The structural part of a specialized variable is accessed in the same way as a regular structural variable.

2.6. Generalized parametric functions

Generic functions provide support for procedural-parametric polymorphism. Their signatures define a common interface for parametric specialization handlers. Each generalization function has from one to several generalized formal parameters. They are pointers to generalizations. When such a function is called, references to specialized variables are passed into it, combinations of which determine a specific specialization handler. The generalizing parametric function is described by the following syntax rules:

GeneralizingFunction = ReturnParameterType NameOfFunction
"<" ListOfGeneralizedParameters ">" "(" [ListOfFormalParameters] ")"
BodyOfGeneralizingFunction.
ListOfGeneralizedParameters = GeneralizedParameter {"," GeneralizedParameter }.
BodyOfGeneralizingFunction = EmptyBody | DefaultHandler.
EmptyBody = "=" "0".
GeneralizedParameter = GeneralizationType "*" GeneralizationName.

For example, a general function for printing a geometric figure can be represented as follows:

```
void PrintFigure<Figure *f>() = 0;
```

Generalized parameters are specified in angle brackets (unlike other parameters, which, as usual, are placed in parentheses when present). The absence of a body in this case indicates that it is necessary to write specialization handlers for all specific figures represented by a parametric generalization. Instead, a default handler can be specified, which is called in cases where the substituted combination of specializations does not have its own handler implemented. An example of the simplest version of such a handler is a call to the program interrupt function:

```
void PrintFigure<Figure *f>() {
    printf("Incorrect Figure!!!\n");
    exit(-1);
}
```

2.7. Parametric specialization handlers

Parametric specialization handlers (or specialized functions) are called when a generic function is called. They provide direct manipulation of specific specializations included in parametric generalizations.

SpecializedFunction = ReturnParameterType NameOfFunction
"<" ListOfSpecializations ">" "(" ListOfFormalParameters ")"
BodySpecializedFunction.
ListOfSpecializations = SpecializedParameter {"," SpecializedParameter }.
SpecializedParameter = SpecializationType "*" SpecializationName.

The handler is selected depending on the values of the parametric arguments substituted instead of the formal parameters. They can be located in different compilation units, being added, for example, as the next specialization is created. For a general shape printing function, specialized functions might look like this:

```
// Output rectangle parameters
void PrintFigure<Figure.Rectangle *r>() {
    printf("Rectangle: x = %d, y = %d\n", r->@x, r->@y);
}
// Output triangle parameters
void PrintFigure<Figure.Triangle *t>() {
    printf("Triangle: a = %d, b = %d, c = %d\n", t->@a, t->@b, t->@c);
}
// Output circle parameters
void PrintFigure<Figure.Circle *c>() {
    printf("Circle: r = %d\n", c->@r);
}
```

2.8. Calls of parametric functions

A function call essentially launches a handler for a specific combination of specializations. It differs from a regular function call only in the presence of a list of additional actual parameters indicating specializations by which a handler is automatically selected, which defines dynamic polymorphism. Calling a parametric function has the following syntax:

CallParametricFunction = NameFunction "<"ListOfPointersToSpecializations">"
"(" [ListOfActualParameters] ")".

Examples include calling the print shape function for the various specialized variables described earlier:

```
PrintFigure<&t1>();   // Print triangle t1
PrintFigure<&c1>();   // Print circle c1
PrintFigure<pc1>();   // Print circle c1 via pointer pc1
PrintFigure<&c[5]>(); // Print circle c[5]
PrintFigure<c+5>();   // Print circle c[5] via offset
PrintFigure<pf1>();   // print triangle t1 via generalized pointer pf1
PrintFigure<*ppf1>(); // print triangle t1 via pointer to pointer *ppf1
```

3. Improving code usability using the procedural-parametric paradigm

The implementation of procedural-parametric polymorphism in the C programming language allows you to take a different look at code development in the following situations, when poorly controlled code is generated in the C language:

- adding new alternatives leads to modification of generalizations containing these alternatives, their attributes, as well as functions that process them, which can lead to incorrect modifications of already written code;

- attributes of alternatives, which are essentially their keys, are not directly related to the alternatives themselves, which is a source of potential errors when a handler for a particular alternative may be accidentally bound to another attribute;
- when connecting alternatives to a common pointer, types are often dereferenced (use of the `void*` pointer), which is very often a source of errors;
- type dereferencing is also often used when passing alternatives to functions, which requires type recovery within these functions and is not always done correctly.

3.1. Adding specializations instead of modifying unions

Using existing C language constructs often involves directly constructing monolithic generalizations based on structures and unions. For example, to create a generalization based on a rectangle and a triangle, you can form a structure in which the union contains alternative specializations, and an additional feature of the specializations is specified by the enumeration:

```
// Key values for each of the shapes
typedef enum Key {
    RECTANGLE, TRIANGLE}
Key;
// Generalized figure
typedef struct Figure {
    Key k; // key
    union { // Alternatives to use
        Rectangle r;
        Triangle t;
    };
} Figure;
```

Processing such a structure, for example printing its contents, is often done using a centralized print function:

```
void PrintFigure(Figure *f) {
  switch(f->k) {
    case RECTANGLE:
      printf("Rectangle: x = %d, y = %d\n", r->x, r->y);
      break;
    case TRIANGLE:
      printf("Triangle: a = %d, b = %d, c = %d\n", t->.@a, t→@b, t->@c);
      break;
    default:
      printf("Incorrect Figure!!!\n");
  }
}
```

In this case, the source of potential errors can be both the use of the wrong handlers with signs of specialization, and the fact that a number of handlers may simply not be taken into account and not added. The likelihood of errors increases with the number of generalization handlers.

The inclusion of new specializations, for example, a circle, is associated with the need to change both the keys that define the characteristics of the figure and the generalization itself, which includes the new specialization.

```
typedef struct Circle{int r;} Circle;
// Key values for each of the shapes
typedef enum Key {
    RECTANGLE, TRIANGLE, CIRCLE
} Key;
// Generalized figure
typedef struct Figure {
    Key k; // key
    union { // Alternatives to use
        Rectangle r;
        Triangle t;
```

```
        Circle c;
    };
} Figure;
```

All handlers for the modified generalization should also change, which can be seen in the example of the same function for displaying information about a figure:

```
void PrintFigure(Figure *f) {
  switch(f->k) {
    case RECTANGLE:
      printf("Rectangle: x = %d, y = %d\n", r->x, r->y);
      break;
    case TRIANGLE:
      printf("Triangle: a = %d, b = %d, c = %d\n", t->@a, t->@b, t->@c);
      break;
    case CIRCLE:
      printf("Circle: r = %d\n", c->@r);
      break;
      default:
        printf("Incorrect Figure!!!\n");
  }
}
```

The above examples demonstrating the addition of new specializations and their handlers in the case of the PP paradigm demonstrate the possibilities of evolutionary expansion of generalizations, generalization handlers and specializations. The direct connection between traits and specializations makes it possible to get rid of centralized processors through the use of PP polymorphism.

3.2. Using instead of dereferenced pointers

The previous solution can also be used in situations where instead of unions, separate structures are used, connected to a common point through pointer dereferencing. This technique is often used to evolve data in the C language. For example, adding new specializations can be achieved by forming them independently of the generalization, or by ignoring the generalization altogether as a specially formed entity. In each specialization, you can put its attribute field in first place and then place the necessary data:

```
typedef struct FigRectangle {Key k; Rectangle r;} FigRectangle;
typedef struct FigTriangle {Key k; Triangle t;} FigTriangle;
typedef struct FigCircle {Key k; Circle c;} FigCircle;
```

Connecting any variable of one of the specified types can be done through a pointer to `void*` with subsequent verification of the key by casting to any of the types.

```
void PrintFigure(void *f) {
  if((FigRectangle*)f->k==RECTANGLE) {
    printf("Rectangle: x = %d, y = %d\n",
          (FigRectangle*)f->r.x, (FigRectangle*)f->r.y);
  } else if((FigTriangle*)f->k==TRIANGLE) {
    printf("Triangle: a = %d, b = %d, c = %d",
        (FigTriangle*)f->t.a, (FigTriangle*)f->t.b,
        (FigTriangle*)f->t.c);
  } else if((FigCircle*)f->k==CIRCLE) {
    printf("Circle: r = %d\n", (FigCircle*)f->c.r);
  } else {
    printf("Incorrect Figure!!!\n");
  }
}
```

This reduces the reliability of the program. Instead of a dereferenced PP pointer, the solution provides a pointer to a generic figure and then detects the specialization both by securely checking its type and using PP polymorphism. The correct connection to these pointers is checked during program compilation, which increases reliability.

3.3. Formation of wrappers around unreliable structures

The use of pointers to void, uncontrolled casting, and type dereferences are common in various libraries written in C. An example is the POSIX Threads library. In particular, a similar approach is used in functions such as thread creation `pthread_create` and its joining `pthread_join`:

```
int pthread_create(pthread_t *thread, const pthread_attr_t *attr,
    void *(*start_routine) (void *),void *arg);

int pthread_join(pthread_t thread, void **retval);
```

Because of this, when using the thread function, you often have to perform unsafe data type conversions. Overcoming these shortcomings in the case of OO programming is carried out by forming a abstract wrapper, which allows you to get rid of the direct use of unreliable constructs.

```
class Thread {
public:
  virtual ~Thread () {}
  virtual void run () = 0;
  int start () {
    return pthread_create(
                  &_ThreadId, nullptr,Thread::thread_func, this );
    }
    int wait () {
        return pthread_join( _ThreadId, nullptr );
    }
protected:
  pthread_t _ThreadId;
  static void* thread_func(void* d) {
    (static_cast <Thread*>(d))->run();
    return nullptr;
  }
};
```

Based on this wrapper, you can create specific threads that solve various tasks. For example, let's create a class that calculates the perimeter of a rectangle:

```
class ThreadRect : public Thread {
public:
  ThreadRect (int xx, int yy): x{xx}, y{yy} {}
  virtual void run () {p = double(2*(x+y));}
  void print(const char* str) {
    std::cout << "Perimeter of " << str << " = " << p << "\n";
  }
protected:
  int x;
  int y;
  double p;
};
```

Using this and other similar classes allows you to create many threads without directly using the functions of the POSIX Threads library:

```
int main () {
  ThreadRect thread1{3, 5};
  ThreadRect thread2{13, 15};
  ThreadRect thread3{4, 10};
  thread1.start(); thread2.start(); thread3.start();
  thread1.wait();  thread2.wait();  thread3.wait();
  thread1.print("Thread1");
  thread2.print("Thread2");
  thread3.print("Thread3");
  return EXIT_SUCCESS;
}
```

A similar opportunity appears in the case of PP programming. One possible approach is to create a generic structure in which specializations can be used to pass arguments to the generated thread and return results from it.

```
typedef struct ThreadData {pthread_t threadId;}<> ThreadData;
```

To work with this structure, you can create a generic function that creates a thread:

```
void RunThread <ThreadData* d>() = 0;
```

The thread function will run the specializations handlers of this generalized function:

```
void* ThreadFunc(void* d) {
    RunThread<(Thread*) d>();
    return NULL;
}
```

This function will be run from a function that is a wrapper over the thread creation function:

```
int StartThread(ThreadData* td) {
    return pthread_create(td->threadId, NULL, ThreadFunc, td);
}
```

You can also create a corresponding wrapper to merge threads:

```
int WaitThread(ThreadData* td) {
    return pthread_join(td->threadId, NULL);
}
```

These functions can be used to describe different threads as specializations of the RunThread generalization. Let's look at the example of calculating the perimeter of a rectangle, the structure of which was determined earlier.

```
Typedef struct Rectangle{int x, y;} Rectangle;
typedef struct RectPerimeter {Rectangle r; double p;} RectPerimeter;
ThreadData + <RectPerimeter;>;
```

We define the specialization handler that calculates the perimeter as follows:

```
void RunThread<ThreadData.RectPerimeter *rp>() {
    rp->p = (double)((rp->r.x+rp.r->y)*2);
}
```

We will also define a function that displays the perimeter of the rectangle, which it takes from the RectPerimeter structure:

```
void PrintRectPerimeter(RectPerimeter* rp, const char* str) {
    printf("Perimeter of %s = %f\n", str, rp->p);
}
```

As an example, here is a main function that starts multiple threads that hide the features of the POSIX Thread library as well as an OO program:

```
int main () {
  ThreadData<RectPerimeter> thread1 = {0}<{3,5,0.0}>;
  ThreadData<RectPerimeter> thread2 = {0}<{7,4,0.0}>;
  ThreadData<RectPerimeter> thread3 = {0}<{6,8,0.0}>;
  StartThread(&thread1); StartThread(&thread2); StartThread(&thread3);
  WaitThread(&thread1);  WaitThread(&thread2);  WaitThread(&thread3);
  PrintRectPerimeter (&(tread1.@), "Thread1");
  PrintRectPerimeter (&(tread2.@), "Thread2");
  PrintRectPerimeter (&(tread3.@), "Thread3");
  return 0;
}
```

4. Conclusion

The proposed extension of the programming language with constructs that provide instrumental support for the procedural-parametric programming paradigm allows the development of more reliable and evolutionarily extensible programs. This ensures the addition of new alternative data, as well as functions that make it possible to painlessly use multiple polymorphism for previously written code. This changes made, implemented in the C programming language within the Linux operating system. The project is being developed under an open source license. The reference to repository

is [10]. The extension is implemented using the clang [11] family of compilers, which allows you to change both the parser and the abstract syntax tree, adapting them to new constructions.

5. References

[1] M. Gregoire. Professional C++. John Wiley & Sons. 2018.
[2] E. Sciore. Java Program Design. Apress Media. 2019.
[3] A. Freeman. Pro Go: The Complete Guide to Programming Reliable and Efficient Software Using Golang. Apress. 2019. p 1105.
[4] J. Blandy, J. Orendorff, L.F. Tindall. Programming Rust. O'Reilly Media. 2021. p 735.
[5] A.I. Legalov. Procedurally-parametric programming paradigm. Is it possible as alternative to the object-oriented style? (dep. hands. number 622-v00 dep. viniti 13.03.2000 ed.). 2000. p 43. (In Russian)
[6] I.A Legalov. 2009. Using of generalized records in procedural-parametric programming language. Scientific Bulletin of the NSTU 28, 3 (2009), 25–37. (In Russian)
[7] A.I. Legalov and D.A. Schvetc. 2003. Procedural language with support for evolutionary design. Scientific Bulletin of the NSTU 15, 2 (2003), 25–38. (In Russian)
[8] A.I Legalov and P.V. Kosov. Evolutionary extension of programs using the procedural-parametric approach. Computational Technologies 21, 3 (2016), 56–69. (In Russian)
[9] A.I Legalov and P.V. Kosov. C Language Extension to Support Procedural-Parametric Polymorphism. Modeling and analysis of information systems 30, 1 (2023), 40–62. (In Russian)
[10] Procedural-parametnric C programming language. URL: https://gitverse.ru/kpdev/llvm-project/content/pp-extension-v2.
[11] Clang: a C language family frontend for LLVM. URL: https://clang.llvm.org.

Multiple mobile robots formation control using behavioral-based approach

Anton A. Tolstikhin[1]

[1] *Matrosov Institute for System Dynamics and Control Theory SB RAS (ISDCT), 134 Lermontova St, Irkutsk, 664033, Russian Federation*

Abstract

This paper proposes a control strategy for a group of autonomous mobile robots to build a stable geometric formation. The main advantage of this approach is its high resistance to robot failures, which allows the formation to be maintained, as well as the absence of the need to predefine the connection graph between them.

Keywords

autonomous mobile robots, formation control, behavioral structure

1. Introduction

Modern trends in robotics development have made it possible to apply robotic systems in a wide range of practical tasks [1, 2]. At the same time, their production technology cost reduction, as well as the improvement of hardware and software components, has made it possible to use coordinated groups of autonomous mobile robots (AMRs) on a massive scale to solve such tasks. When organizing group control, particular attention is paid to achieving and maintaining a given geometric formation by the robots [3].

The currently existing approaches to building formations can be divided into three main categories [4]: "leader-follower", virtual structure, and behavioral approaches (behavioral structure). According to the first approach [5, 6], one or more leaders are chosen among the agents of the group, relative to which the other robots (followers) try to occupy the desired assigned position. Despite its high popularity, mainly due to its simplicity of implementation, this approach has a number of disadvantages. The main one is the low resistance to the failure of individual robots. Thus, if one of the leaders breaks down, the formation may split into independent groups that are unable to solve the task. In addition, it has a fairly high degree of centralization, since follower robots often do not have mechanisms for solving practical tasks, but only maintain a predetermined formation. These disadvantages are partially compensated for by using the virtual structure method [7, 8]. In this case, each robot is assigned to a virtual leader, a position in the formation that the robot must occupy. At the same time, the movement of the virtual formation can be set either centrally or decentrally during collective decision-making, or even, in rare cases, individually by each robot. Obviously, the decentralized option is of the greatest

7th International Workshop on Information, Computation, and Control Systems for Distributed Environments (ICCS-DE 2025), July 7–11, 2025, Irkutsk, Russia

madstayler93@gmail.com (A. A. Tolstikhin)

© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

ICCS-DE 2025 Workshop Proceedings (iccs-de.icc.ru)
DOI: 10.47350/ICCS-DE.2025.04

interest, but its application is associated with a significant increase in the computational load and communications within the group, which can be critical when robots solve a number of practical tasks.

This paper proposes an approach to formation building based on the behavioral structure method [9]. According to this approach, each robot adjusts its position relative to the current coordinates of the other group members, which ultimately allows the desired formation to be achieved. One of the main advantages of behavioral structure is the ability to organize independent control loops (behaviors) aimed at the parallel solution of several subtasks by robots. In addition, this approach allows for complete decentralization while maintaining the coordination of the entire group's actions. Currently, behavioral methods of formation organization are widely used in the work of scientists around the world [1, 10, 11]. However, many of them are designed in such a way that each robot is given a list of neighbors in advance, relative to which its position in the formation is determined. As in the case of the "leader-follower" approach, this can lead to incorrect formations in the event of the failure of several AMRs. The proposed control strategy eliminates this drawback.

2. Formation controll

The paper considers a strategy for controlling a group of AMRs, which are described by the following system of differential equations:

$$\begin{cases} \dot{q}_i(t) = v_i(t), \\ \dot{v}_i(t) = u_i(t), \end{cases} \\ i \in G, \|v_i\| \leq v_{\max}, \|u_i\| \leq u_{\max}, \tag{1}$$

where $q_i \in R^p$, $v_i \in R^p$, and $u_i \in R^p$ are the coordinate, velocity, and control vectors of the i-th robot; p is the dimension of the search space (in this work, $p = 2$ is assumed); v_{max} and u_{max} are the velocity and control constraints; G is the set of available AMRs. Let the velocity and control at the initial time ($t = 0$) be equal to zero, and the coordinates be randomly assigned in some neighborhood of a given point q_0.

According to the proposed approach, the control of each robot is specified by a linear combination of forces, which are responsible for individual behaviors.

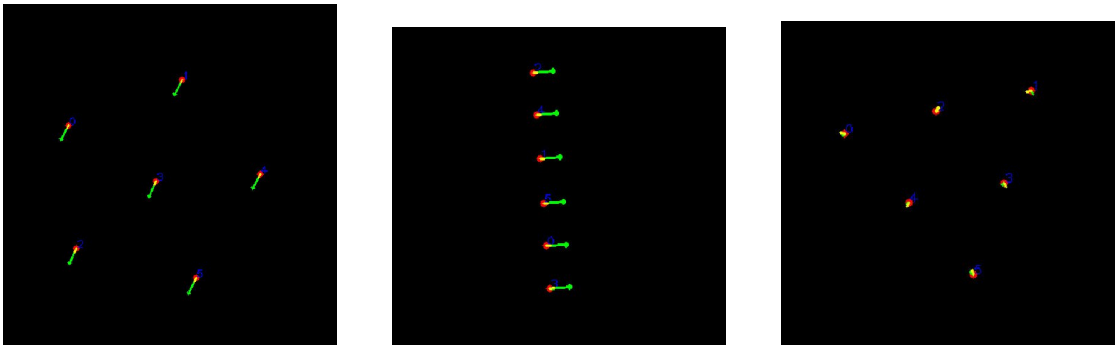
$$u_i = \sum_{k \in B} c_j F_i^k, \tag{2}$$

where $c_j \in R^+$ are controllable weight coefficients, B is the set of behaviors available to the robot.

The simplest formation that can be obtained is specified as follows:

$$F_i^f = -\sum_{j \in G} \left(\nabla_{q_i} U_{ij}^c \left(\|q_{ij}\| \right) \right), \quad i \neq j, \tag{3}$$

$$U_{ij}^c \left(\|q_{ij}\| \right) = \alpha \left(\frac{1}{2} \left(\|q_{ij}\| - d_{ij}^A \right)^2 + \beta \ln \|q_{ij}\| + \beta \frac{d_{ij}^A}{\|q_{ij}\|} \right), \tag{4}$$



(a) "Blob" formation (b) "Line" formation (c) "Triangle" formation

Figure 1: Visualization of the proposed formations.

where $q_{ij} = q_i - q_j$; ∇_{q_i} is the gradient with respect to the vector q_i; $d_{ij}^A > 0$ determines the desired distance between agents; $\alpha, \beta \in R^+$ are control parameters. Here, each robot tries to maintain a given distance from every other robot in the group, that leads to a circular formation. Its distinctive features are compactness and flexibility when robots pass through narrow areas. The “line” formation is achieved through the use of a neighbor selection mechanic. Each AMR selects its neighborhood N_i, which includes one robot closest to the center of mass of the group and one in the opposite direction. At the same time, the second neighbor may be empty, which is typical for robots at the edges of the line. The function that determines this behavior is defined as:

$$F_i^f = a_1 \tan\left(\theta_i - \theta_{offset}\right) - a_2 \sum_{j \in N_i} \left(\nabla_{q_i} U_{ij}^c \left(\|q_{ij}\|\right)\right), \tag{5}$$

where θ_i is the angle between the main normal to the vector emanating from the robot to the center of mass of the group and the vector defining the desired movement direction; $a_1, a_2 \in R^+$ are control parameters. When $\theta_{offset} = 0$, the robots line up perpendicular to the desired movement direction, which can be obtained, for example, from other behaviors of the agent. Changing this parameter allows the group to be oriented at any angle to the movement vector. For example, when $\theta_{offset} = \pi/2$, the group is lined up in a column.

Another way to apply the neighbor selection mechanic is to organize a triangular formation. In this case, the two nearest AMRs in the direction of the center of mass of the group are selected as neighbors, and the agent’s behavior consists of two ideas: maintaining a given distance from neighbors and adjusting its position so that the angle θ between the vectors directed at the neighbors is equal to 600. Thus, the behavior for maintaining the formation is as follows:

$$F_i^f = a_3 \tan(\pi - 3\theta)(-1)^{\left[\frac{3\theta}{\pi}+1\right]} - a_4 \sum_{j \in N_i} \left(\nabla_{q_i} U_{ij}^c \left(\|q_{ij}\|\right)\right), \tag{6}$$

where $a_3, a_4 \in R^+$ are control parameters; [] is the operator for rounding to the nearest integer.

Examples of the above formations for a group of six AMRs are shown in Figure 1.

3. Conclusion

The proposed approaches to formation building based on the behavioral structure method have two important features. First, the control strategy is completely decentralized and does not require additional prior knowledge. Therefore, the failure of one or more robots does not lead to the loss of the formation, but causes its restructuring without involving additional calculations or planners. Second, the approach works correctly in the case of restrictions on the communication range, causing incomplete information about the coordinates of the group agents. Currently, work is underway to find behaviors for organizing a wider range of formations, as well as to develop a mechanism for individual decision-making on whether to change the formation during the task solution process, depending on its current conditions and environmental events.

Acknowledgments

The work was supported by the Ministry of Science and Higher Education of the Russian Federation, project No.121032400051-9.

References

[1] J. Alonso-Mora, S. Baker, D. Rus, Multi-robot formation control and object transport in dynamic environments via constrained optimization, The International Journal of Robotics Research 36 (2017) 1000–1021. doi:10.1177/0278364917719333.

[2] L. Pan, S. Manjanna, M. A. Hsieh, Marlas: Multi agent reinforcement learning for cooperated adaptive sampling, in: J. Bourgeois, J. Paik, B. Piranda, J. Werfel, S. Hauert, A. Pierson, H. Hamann, T. L. Lam, F. Matsuno, N. Mehr, A. Makhoul (Eds.), Distributed Autonomous Robotic Systems, Springer Nature Switzerland, Cham, 2024, pp. 347–362.

[3] W. Guanghua, L. Deyi, G. Wenyan, J. Peng, Study on formation control of multi-robot systems, in: 2013 Third International Conference on Intelligent System Design and Engineering Applications, 2013, pp. 1335–1339. doi:10.1109/ISDEA.2012.316.

[4] S. Kurochkin, A. Tachkov, Methods of formation control for a group of mobile robots (a review), Mekhatronika, Avtomatizatsiya, Upravlenie 22 (2021) 304–312.

[5] Y. Zolotukhin, K. Kotov, A. Maltsev, A. Nesterov, M. Sobolev, M. Philipov, A. Yan, Robust control of mobile objects in a leader-follower group using the structural synthesis method, Avtometria 5 (2015) 81–91.

[6] Z. Yang, S. Zhu, C. Chen, G. Feng, X. Guan, Leader-follower formation control of nonholonomic mobile robots with bearing-only measurements, Journal of the Franklin Institute 357 (2020) 1628–1643. URL: https://www.sciencedirect.com/science/article/pii/S0016003219308014. doi:https://doi.org/10.1016/j.jfranklin.2019.11.025.

[7] X. Chen, F. Huang, Y. Zhang, Z. Chen, S. Liu, Y. Nie, J. Tang, S. Zhu, A novel virtual-structure formation control design for mobile robots with obstacle avoidance, Applied Sciences 10 (2020). URL: https://www.mdpi.com/2076-3417/10/17/5807. doi:10.3390/app10175807.

[8] X. Yan, D. Jiang, R. Miao, Y. Li, Formation control and obstacle avoidance algorithm of a multi-usv system based on virtual structure and artificial potential field, Journal of

Marine Science and Engineering 9 (2021). URL: https://www.mdpi.com/2077-1312/9/2/161. doi:10.3390/jmse9020161.

[9] T. Balch, R. Arkin, Behavior-based formation control for multirobot teams, IEEE Trans. Robot. Autom. 14 (1998) 926–939.

[10] X. Wu, R. Wu, Y. Zhang, J. Peng, Distributed formation control of multi-robot systems with path navigation via complex laplacian, Entropy 25 (2023). URL: https://www.mdpi.com/1099-4300/25/11/1536. doi:10.3390/e25111536.

[11] Y. Zolotukhin, K. Kotov, A. Maltsev, A. Nesterov, M. Sobolev, A. Tsupa, Coordinated control of a group of robots in cargo handling tasks, Vicislitelnie tekhnologii 1 (2016) 70–79.

Personal experience in the onboarding process: A reflective analysis through autoethnography

Igor Vecstejn [1], Zeljko Stojanov[1]

[1] *University of Novi Sad, Technical faculty "Mihajlo Pupin", Djure Djakovica bb, Zrenjanin, Serbia*

Abstract

This paper investigates an onboarding process from the perspective of a new employee, using an autoethnographic approach. Through daily reflection, notes and analysis of personal experience during the period of adaptation in a software company, the first author has identified the key steps, entities and relationships that shape the process of integration into the company's work environment. This study offers a structured representation of onboarding through two visual models: initial and final that show how elements such as mentoring, access to resources and reflection complement each other in adaptation. The reflection not only contributed to the first author's professional development but also provided insights that are very useful for designing an effective onboarding process.

Keywords

Onboarding process, IT Company, Personal experience, Reflection, Autoethnography

1. Introduction

Onboarding represents a key process in the integration of new employees in the company, which includes not only the transfer of technical knowledge but also socialization in the organizational culture. Contemporary research points out that a quality onboarding process significantly affects employee retention. A study from 2025 shows that a well-structured onboarding program, which includes a welcome from managers and colleagues, positively affects the identification of employees with the organization and reduces the intention to leave the position and the job [1]. In the context of digital transformation and remote work, companies are faced with challenges in adapting the onboarding process. A qualitative study conducted in a higher education institution in the Netherlands during the COVID-19 pandemic identified four key factors influencing the adjustment of new employees: relationship building, responsibility for adjustment, managing well-being and sense of belonging, information seeking [2].

Autoethnography is a qualitative research method that combines autobiographical storytelling with ethnographic analysis to explore the cultural and social dimensions of personal experience [3]. This method allows researchers to provide a deeper understanding of organizational phenomena through introspective analysis through their own experience [4]. In the contemporary work environment, autoethnography has been used to examine how individuals navigate inclusion and collaboration, particularly as it relates to virtual or hybrid organizational environments. This approach allows researchers to uncover the emotional and cultural dynamics related to power within an organization through first-person and reflective perspective [3]. Also, works from the field of autoethnography methodology emphasize the importance of ethical aspects of research, especially in situations where the personal narrative includes other actors from the organizational context [5, 6].

Reflection is a key component of professional development that allows individuals to analyze their own actions and experiences with the aim of improving everyday practice [7]. In a business and organizational environment, reflection allows employees to better understand their roles, identify

7th International Workshop on Information, Computation, and Control Systems for Distributed Environments (ICCS-DE 2025), July 7–11, 2025, Irkutsk, Russia
EMAIL: igor.vecstejn@tfzr.rs(A.1); zeljko.stojanov@uns.ac.rs(A.2);
ORCID: 0009-0009-9940-4411(A.1); 0000-0001-6930-5337(A.2);
© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).
ICCS-DE 2025 Workshop Proceedings
DOI: 10.47350/ICCS-DE.2025.05

challenges and develop strategies to overcome obstacles [8]. Reflective practice is increasingly used in technical fields such as software engineering. Barr et al. [9] analyzed the longitudinal development of reflective skills in students in a work-based software engineering program and indicated clear patterns of progress through academic and professional experience.

The aim of this paper is to present the onboarding process through the first author's personal experience and reflection and how specific elements of the new employee integration model are developed through it.

Based on this aim, autoethnographic method was selected as a valid approach for analyzing professional practice. Autoethnography allows deep interpretation of individual experience and supports the structured development of a general onboarding model.

2. Methodology

In this work, a qualitative methodology that relies on an autoethnographic approach was applied, which allows the researcher to use his own experience as a source of data for understanding and modeling the onboarding process. Autoethnography is at the crossroads between autobiography and enography because it interprets the researcher's personal narratives in a broader cultural and organizational context [10].

This approach is particularly suitable in situations where the first author is a direct participant in the process, and the goal is to discover patterns and meanings through reflective practice [3, 4]. According to Ellis [11] and Given [10], autoethnography often includes taking reflective notes that simultaneously represent data and the basis for analysis.

In this research, the onboarding process was documented through daily reflections, notes, internal documentation, and procedural monitoring, and data collection was conducted during the onboarding process, followed by analysis and model development. Analysis process didn’t use formal thematic coding, but it was conducted through iterative reflection and categorization. Written and digital notes were reviewed and grouped according to concepts. Entities, (e.g. Mentor, Manager, Resources) and other key steps were identified through process and visually represented.

3. Description of Experience

The experience that served as the basis for reflective analysis was gained in a software company that develops integration solutions in software tools, electronic stores, in tools related to bookkeeping, design, music, communication, etc. The company specializes in the implementation of soft and hard integrations that are designed to connect with different platforms, applications, and tools that clients use in their business processes.

The first author's work was focused on soft integrations, which were realized using software integration solutions developed within the company. Unlike complex hard integrations that include deeper changes in existing systems, development of new functionalities within existing and new integrations, inclusion of hardware components, soft integrations imply faster and more flexible connection of systems/software, most often available via API. The use of documentation and existing API interfaces enables a simple implementation that does not require complex technical interventions or changes in the infrastructure.

During the period of one year and ten months, when the first author worked as an Integration Specialist, the emphasis was on understanding client requirements, analyzing technical documentation, configuring connections, developing integrations, as well as testing functionality after implementation. The role of a mentor, cooperation with the team and constant exchange of knowledge contributed to the development of professional and social skills, but also to a deeper understanding of the process of inclusion in team dynamics through daily work.

This kind of work environment represented a context for considering and recording the process of personal and professional inclusion in the new work environment, as well as reflecting on the importance of defining goals, mentoring and informal learning within the onboarding phase.

4. Reflective Analysis and Model Development

During onboarding process, first author was engaged with various internal company materials such as documentation, internal tutorials, internal software and other onboarding resources that was granted in period. Also, daily communication with a mentor and attendance at team meetings as a participant/observer occurred. Reflections included technical insights and emotional reactions such as confusion, huge burst moments of clarity and motivation and questions about environment, integrations, APIs, etc. Through whole process of communication and working, first author was thinking about gained experiences, about progress, potential challenges, ignorance and based on that, notes were created both in paper form and in digital form. The notes that were created during that period and the reflections that accompanied first author, represents the basis for the analysis. Notes were recorded on daily basis, both on separate papers, notebooks, and digital tool (MS Word) usually during meeting with Mentor and while working and studying processes of creating integration. Figure 1 shows authentic note of the onboarding process that was illustrated during period of integrating into company. This reflects mapping daily mentor and weekly team meetings, internal software and documentation usage, resource access. These notes were base ground for creating initial onboarding model. The identification of entities and steps is based on the analysis of daily activities and reflective observations during the process. Entities are identified as functional components in onboarding while steps are derived from the logical flow of recorded events.

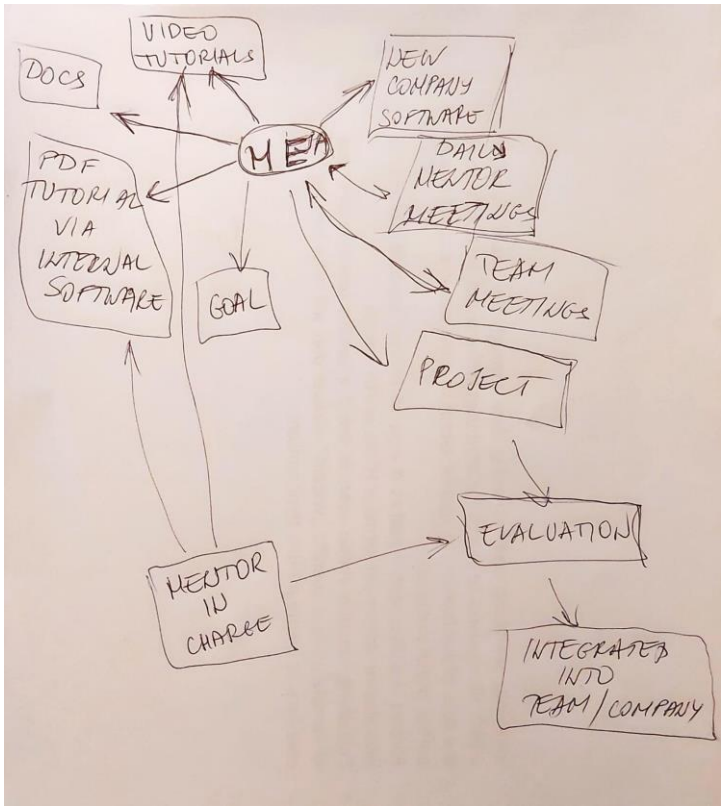


Figure 1: Example of written notes during integrating into company

Based on notes, first author developed a structured initial onboarding model that is shown in Figure 2. This model visualize the most essential entities and relationships between them such as mentor, team, resources, method, learning goal, etc. The main focus of the model version was to interpret personal logic that was gained through experimental learning into visual representation. It emerged from introspective practice of writing and recording daily activities and personal insights (reflection) and serves as first semantic representation of the onboarding process from participant's perspective.

In this model, onboarding process begins with assignment of a role and access to resources that is followed by mentoring and task involvement. This model captures flow of communication, learning and reflection.

After that notes were categorized, grouped, this part is about organizing with aim of creating a structure for further development of model. Through data organization and analysis process, new procedures/steps, processes and entities were detected. After reorganizing data, initial semantic visualization of the entities and their mutual relations was created. Figure 3 presents the result of analytical effort, a final onboarding model. This model introduced more refined visual representation with better and clearer picture with expanded entities such as “Manager“ and “Learning objectives“, and semantic text between entities. Existing ones were linked with semantically more precise relationships and entities such as “Learning objectives“ instead of “Learning goal“. This model still relies on first author’s experience but with this version, it moves toward systematical generalization and usability in different aspects of process. The model shows how onboarding begins, i.e. by communicating with the manager, it then includes intensive communication with the mentor, use of resources and work methods, setting learning goals, as well as evaluation after the completion of a specific task or project. This ensures a clear flow of entry into the organization until full integration into the company's team.

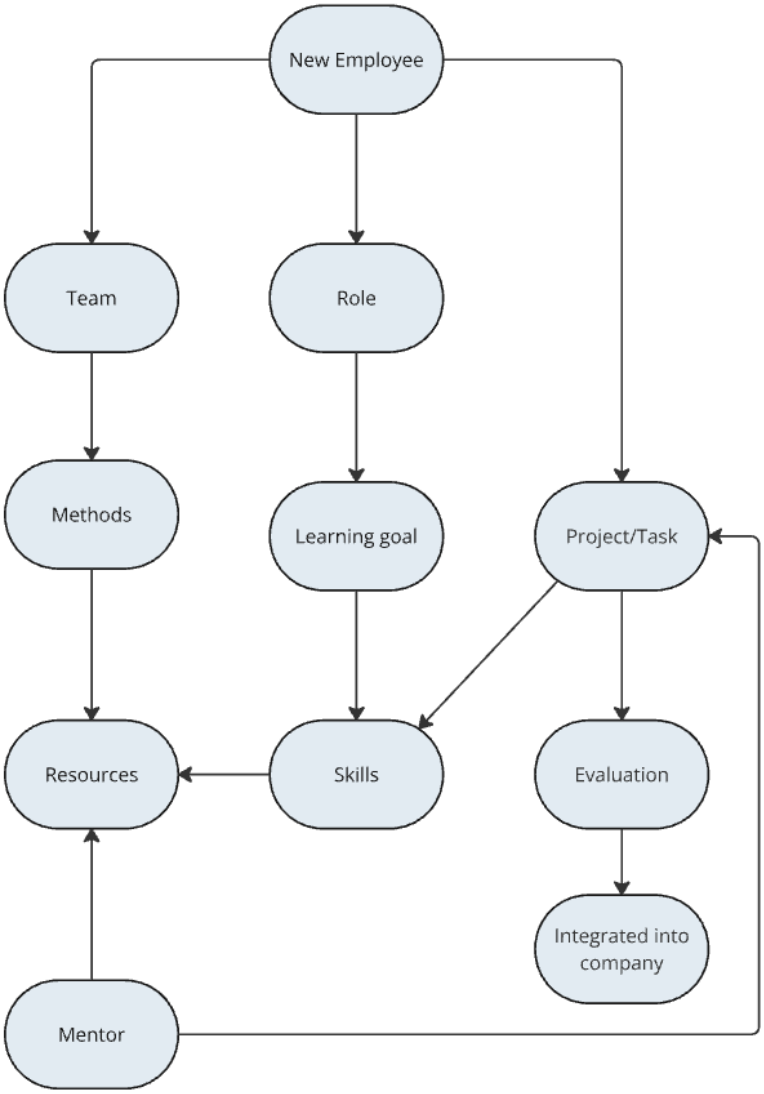


Figure 2: Initial Onboarding model created based on written notes

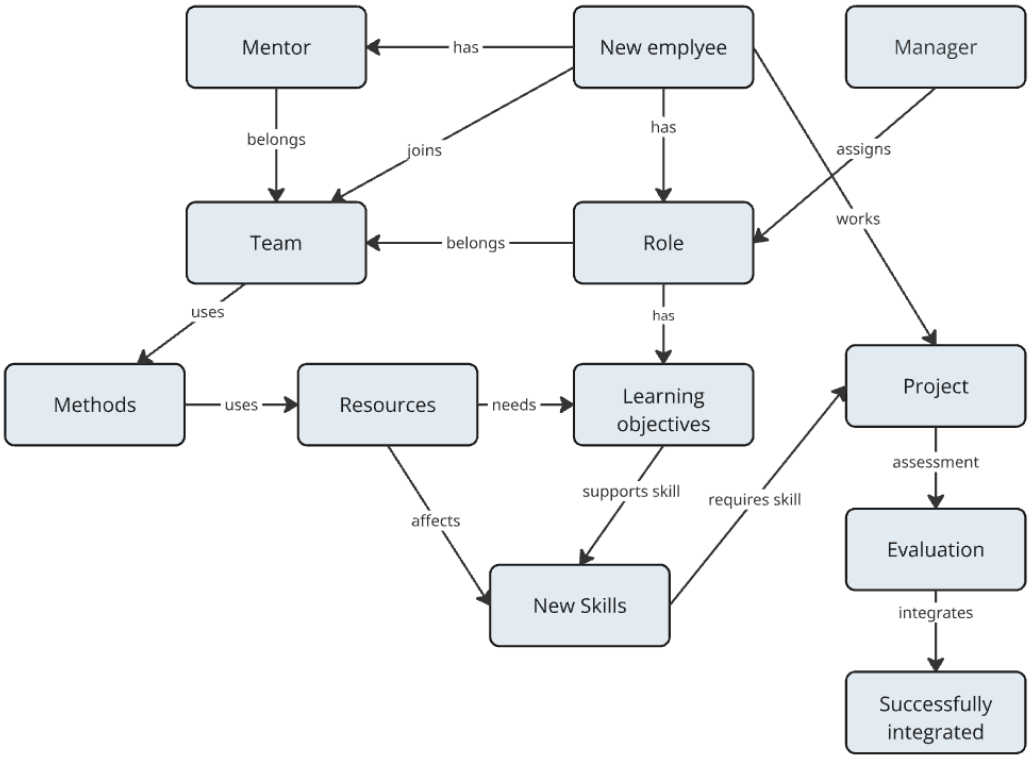


Figure 3: Final onboarding model

Figure 4 and represents the autoethnographic basis of the onboarding process model. It includes core elements that are based on personal experience and introspection, such as daily communication with the mentor and team, use of internal documentation and tutorials and interaction with internal software, but also the process of constructing and developing onboarding model. Through mentioned daily activities, the first author engaged in continuous reflection based on the emotional-cognitive process of personal learning and adaptation. Feelings such as confusion, motivation and moments of complete clarity emerged through continuous questioning such as: (1) "Will I succeed?" (2) "Do I understand this type of integration?" (3) "What specifically should I focus on?", etc.). These reflections were systematically documented in both physical and digital format, capturing personal insights, project goals and critical practical knowledge specific to the use of the software. This documenting process enabled the construction of an initial personal onboarding model that was directly derived from reflective note analysis.

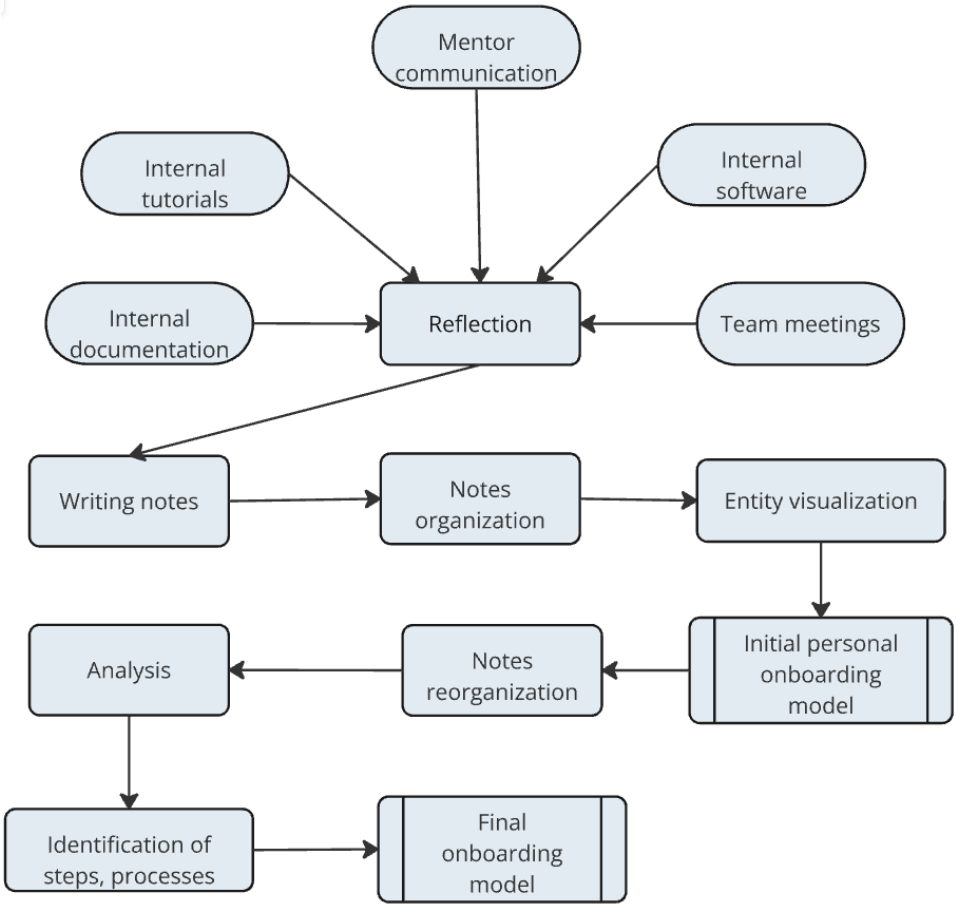


Figure 4: Autoethnographic landmark of onboarding process model

5. Discussion

Reflection is a key element in a person's professional development process, especially during the onboarding period. Through continuous note-taking (paper and digital format) and analysis of his own experience, first author had the opportunity to identify certain patterns of behavior, challenges and success that shaped inclusion in the company. From the introspective analysis, it is possible to recognize reactions to specific situations, which contributes to a deeper understanding of the culture within the company and first author's role within the team. As Schon [12] points out, reflection allows professionals to learn from their experiences, which is important for adaptation in a new work environment.

Using an autoethnographic approach made it possible to analyze and model the onboarding process through personal experience. This approach provided an authentic insight into the organization's processes, but it also carries its own challenges related to subjectivity. According to Ellis [11], autoethnography allows the researcher to connect personal experience with broader cultural, political and social meanings, thereby enriching the understanding of organizational phenomena.

The developed onboarding model is based on personal experience and analysis and can serve as a basis for improving existing practices in organizations that strive for more effective inclusion of new team members in the company. The model identifies the key entities and steps that shape the onboarding experience itself, providing a structure for effective onboarding of new hires. This structure can be especially useful for HR experts, mentors and managers in the IT sector where the speed and efficiency of the onboarding process is essential to the company.

5.1. Validity

In qualitative research, validity is often evaluated through the concept of trustworthiness, which includes four key criteria according to the authors of Lincoln & Guba [13] and Shenton [14]: (1) Credibility (2) Transferability (3) Dependability (4) Confirmability.

Credibility is achieved through detailed documentation of reflective notes and a transparent presentation of the analysis process and daily observations recorded during work in the company. These notes include technical and emotional reflections with certain questions that was in authors' mind, and they are shown in Figure 1 as evidence of an authentic documenting process.

Transferability is enabled through a detailed presentation of the work context. The first author worked as an Integration Specialist in a software company dealing with integrations. Given the ubiquity of very similar positions in the IT industry, the model has the potential to be adapted and applied in similar company structures.

Dependability is ensured through the description of the iterative process of reflection and categorization where, after writing notes, data organization, entity identification and model development were done. Although no thematic coding was used, the analytical procedure was documented and presented in Figure 2 and Figure 3.

Confirmability was achieved through the use of authentic written and digital notes that served as the basis for the creation of the model. One of the notes is presented in Figure 1 and further explained in the section "Reflective Analysis and Model Evolution". Visual models represent the result of the analysis of the notes, which confirms the high transparency of the process.

6. Conclusion

Through an autoethnographic approach, the paper provided a detailed and deep insight into the onboarding process from the perspective of a new employee in the IT sector. With the help of reflection and analysis of personal notes, daily activities and interactions with the mentor and the team, key entities and steps that form the basis for the onboarding process were identified. Based on that, an initial and final model was developed that shows the structure and flow of successful integration of an employee into the organization.

The developed models can serve as a tool for HR experts, mentors, and managers in companies who want to systematize the onboarding process.

Future research can be directed in several directions, such as the improvement of the existing model based on the comparative analysis of a large number of onboarding experiences in different organizations, the theoretical validation of the model through white and gray literature, the development of a software tool that would implement the model in digital form and thus enable its application as part of the internal human resource management system.

7. References

[1] P. Mosquera, M. E. Soares, Onboarding: a key to employee retention and workplace well-being, Review of Managerial Science, 2025, pp. 1–25.

[2] T. Schutte, A. Asatiani, Digital onboarding of newcomers: Insights from an organization navigating onboarding challenges during the COVID-19 pandemic, in: Proceedings of the 14th Scandinavian Conference on Information Systems, 2023, Paper 16.

[3] C. Ellis, T. E. Adams, A. P. Bochner, Autoethnography: An overview, Forum: Qualitative Social Research 12 (1) (2011) 37–49.

[4] H. Chang, Autoethnography as Method, Left Coast Press, Walnut Creek, CA, 2008.

[5] T. E. Adams, S. H. Jones, C. Ellis (Eds.), Handbook of Autoethnography, Routledge, New York, NY, 2016.
[6] S. Stahlke Wall, Toward a moderate autoethnography, Int. J. Qual. Methods 15 (1) (2016) 1609406916674966. doi:10.1177/1609406916674966.
[7] G. E. J. Bolton, Reflective Practice: Writing and Professional Development, 5th ed., SAGE Publications Ltd, London, UK, 2018.
[8] J. R. Bailey, S. Rehman, Don't underestimate the power of self-reflection, Harvard Business Review 4 (2022)
[9] M. Barr, S. W. Nabi, O. Andrei, The development of reflective practice on a work-based software engineering program: A longitudinal study, arXiv preprint arXiv:2504.20956, 2025
[10] L. M. Given, The SAGE Encyclopedia of Qualitative Research Methods, SAGE Publications, Thousand Oaks, CA, 2008.
[11] C. Ellis, The Ethnographic I: A Methodological Novel about Autoethnography, Rowman Altamira, Walnut Creek, CA, 2004.
[12] D. A. Schön, The Reflective Practitioner: How Professionals Think in Action, Routledge, London, UK, 2017.
[13] Y. S. Lincoln, E. G. Guba, Criteria for Assessing Naturalistic Inquiries as Reports, 1988.
[14] A. K. Shenton, Strategies for ensuring trustworthiness in qualitative research projects, Education for Information 22 (2) (2004) 63–75.

Подбор гиперпараметров для синхронного метода распределенного глубокого обучения при решении задач классификации изображений

Кирилл Блинов [1], Илья Курочкин[2]

[1] *Национальный исследовательский технологический университет "МИСИС", Ленинский проспект 4, Москва, 119049, Россия*
[2] *Институт проблем передачи информации имени А. А. Харкевича РАН, Большой Каретный переулок 19 стр. 1, Москва, 127051, Россия*

Аннотация

Для решения задач с помощью глубоких нейронных сетей в случае большого количества данных может применяться распределенное глубокое обучение. Для некоторых распределенных систем количество вычислительных узлов может быть не определено или изменяться динамически. Необходимо найти такие гиперпараметры, для которых при разном количестве узлов метрики качества моделей не сильно изменялись. В статье рассматривается распределенное глубокое обучение с параметрическим сервером и синхронной агрегацией результатов. С использованием архитектуры ResNet-18 на наборе данных CIFAR-10 была проведена серия экспериментов, в которых варьировались число клиентов, скорость обучения и размер батча. Применялась стратегия частой агрегации, при которой обновления моделей синхронизировались после обработки каждого локального батча данных. На основе полученных результатов было подтверждено подбора такого набора гиперпараметров распределенного обучения, при котором значения точности классификации слабо зависят от количества узлов. На основе анализа полученных результатов продемонстрировано, что время распределенного обучения для некоторых наборах гиперпараметров может быть меньше, чем локальное глубокое обучение. А значение размера батча сильно влияет не только на точность классификации, но и на время распределенного обучения.

Ключевые слова

Глубокое обучение, Распределенное обучение, Распределенная вычислительная система, Метод синхронной агрегации результатов, Классификация изображений, Набор гиперпараметров

1. Введение

В последние годы возрастающие количество данных в сочетании с ограничениями по памяти и вычислительными мощностями стимулировали разработку новых подходов к машинному обучению [1]. Одним из решений в этой области стало распределенное обучение (Distributed Learning) — парадигма, предполагающая разделение вычислительной нагрузки по обучению модели между несколькими взаимосвязанными вычислительными узлами [2].

В рамках распределенного обучения существует множество стратегий и архитектур, каждая из которых обладает своими особенностями. Одним из таких подходов является локальное обучение (Local Training), где каждый узел выполняет несколько итераций обучения на своих локальных данных, прежде чем синхронизировать результаты с остальной системой [3].

7th International Workshop on Information, Computation, and Control Systems for Distributed Environments (ICCS-DE 2025), July 7–11, 2025, Irkutsk, Russia
EMAIL: skips0skips@mail.ru (A. 1); kurochkin@iitp.ru (A. 2)
ORCID: 0009-0004-8759-1719 (A. 1); 0000-0002-0399-6208 (A. 2)
© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).
ICCS-DE 2025 Workshop Proceedings
DOI: 10.47350/ICCS-DE.2025.06

В рамках одного раунда обучения узлы (клиенты) обновляют локальные копии глобальной модели, после чего отправляют результаты (например, градиенты или веса) на сервер для агрегации. Обновление глобальной модели может происходить как синхронно, так и асинхронно [4]. Данная работа фокусируется на синхронном подходе, при котором сервер ожидает обновлений от всех выбранных участников раунда, прежде чем вычислить новую версию глобальной модели и разослать ее обратно клиентам [5]. В противоположность этому, асинхронный подход позволяет узлам обновлять глобальную модель независимо друг от друга. Эффективность и скорость сходимости такой системы зависят от множества факторов, однако одним из наиболее критичных является стратегия отбора клиентов для участия в каждом раунде обучения [6].

Проблема отбора клиентов является многоаспектной. Проблему можно разделить на два взаимосвязанных направления: качественное, направленное на выбор наиболее «полезных» клиентов с точки зрения их вклада в модель, и количественное, связанное с определением оптимального числа участников в раунде [7]. Многочисленные исследования подтверждают, что увеличение числа клиентов, участвующих в агрегации, как правило, положительно сказывается на скорости сходимости [8]. Тем не менее, этот эффект нелинеен и подчиняется закону убывающей предельной полезности: прирост производительности значительно снижается после достижения определенного порога участников, как было продемонстрировано в основополагающих работах в этой области [9]. Таким образом, определение оптимального количества клиентов является нетривиальной задачей, поскольку оно зависит от характеристик набора данных, архитектуры и гиперпараметров модели [10].

Основная цель настоящей работы — подбор гиперпараметров для различного количества клиентов в распределенной системе с синхронной агрегацией. Для оценки и сравнения результатов была выбрана эталонная задача классификации изображений с использованием архитектуры ResNet-18 [11]. Чтобы обеспечить высокую скорость итераций исследования использовался набор данных CIFAR-10 [12].

Проведя серию контролируемых экспериментов с варьированием числа клиентов и ключевых гиперпараметров, данное исследование решает следующие задачи: 1) оценить зависимость скорости сходимости от числа активных клиентов; 2) предоставить практические рекомендации гиперпараметров распределенного обучения для распределенных систем с различным числом вычислительных узлов.

2. Материалы и методы

В качестве глубокой нейронной сети использовалась архитектура ResNet-18, которая до сих пор используется в некоторых задачах классификации изображений [11]. В качестве метода распределенного обучения был выбран подход с разделением по данным и с параметрическим сервером (глобальной моделью), а также с синхронной агрегацией результатов. Локальные модели инициализировались путем копирования весов глобальной модели. Использовался датасет CIFAR-10 (60 тыс. цветных изображений 32x32), разделенный на обучающий (80%, 40 тыс. изображений), валидационный (10%, 5 тыс. изображений) и тестовый (10%, 5 тыс. изображений) наборы. Разделение датасета на наборы происходило с сохранением балансов классов [12]. Применялась стандартная нормализация по 3 каналам RGB-изображения.

Ключевой особенностью метода распределенного обучения является динамическое распределение данных: в начале каждой эпохи обучения для каждого клиента случайным образом выбирался локальный набор данных размером N/K (N – общее количество изображений в обучающем наборе, K – количество клиентов) [5]. Это позволило обеспечивать изменение состава данных у каждого клиента от одной эпохи распределенного обучения к другой.

Распределенное обучение проводилось в течение 100 эпох. Каждая эпоха включала распределение данных, локальное обучение (прямой проход, вычисление потерь, обратный проход, шаг оптимизатора) на батчах данных, оценку на валидации и синхронную агрегацию обновлений локальных моделей.

2.1. Локальная синхронная агрегация

В исследовании применялась методология распределенного обучения с интенсивным информационным обменом, при которой агрегация глобальной модели производилась после каждого цикла обучения локальных моделей на одном батче данных. Это обеспечивало быструю адаптацию глобальной модели к изменениям на локальных клиентах. Использовался метод усреднения состояний (State Dict Averaging) [5]. Центральный сервер собирал обновленные веса от клиентов, вычислял поэлементное среднее арифметическое параметров и обновлял глобальную модель по формуле:

$$W_i^{(t+1)} = \frac{1}{N}\sum W_{k,i}^{(t+1)}, \quad (1)$$

где $W_{k,i}^{(t+1)}$ – i-й параметр локальной модели клиента k после раунда обучения t, N – количество клиентов, $W_i^{(t+1)}$ – соответствующий параметр обновленной глобальной модели.

2.2. Определение эпохи обучения и метрики оценки

"Общая эпоха" определялась как завершение каждой локальной моделью полного прохода по всему выделенному ей локальному набору данных. Оценка производительности глобальной модели производилась по завершении каждой "общей эпохи".

Отслеживалась точность (Accuracy) как (Количество верных предсказаний) / (Общее количество предсказаний). Использовалась функция потерь Cross-Entropy Loss [13].

Метрики:

- Global accuracy (точность на валидации в конце каждой эпохи обучения глобальной модели).
- Test accuracy (итоговая точность на тесте глобальной модели).

Также измерялось время выполнения одной полной глобальной эпохи.

3. Эксперименты и результаты

Эксперименты по распределенному обучению для многоклассовой классификации изображений проводились с использованием PyTorch на устройстве с CUDA: процессор Intel Core i9-13900K, 32 ГБ DDR5 RAM, NVIDIA GeForce RTX 3090 Ti, 512 ГБ SSD. Схема эксперимента включала параметрический сервер (глобальная модель) и несколько клиентов-узлов (локальные модели).

Для исследования влияния гиперпараметров распределенного обучения проводилась серия экспериментов (Таблица 1). Варьировались:

- NUM_CLIENTS (K): 1, 2, 4, 6, 8, 10, 15, 20, 30, 40. Влияет на объем данных на клиенте и количество локальных обновлений для агрегации.
- LEARNING_RATE (LR): 0.1, 0.01. Коэффициент обучения контролирует размер шага при обновлении весов.
- BATCH_SIZE: 1024, 256, 64. Влияет на точность оценки градиента и использование памяти.

Изменение этих параметров позволило исследовать динамику распределенного обучения с синхронной агрегацией и влияние на качество обучения модели.

Основным показателем качества обучения является итоговая точность (Test Accuracy) глобальной модели, измеренная на отложенном тестовом наборе данных после завершения 100 эпох обучения. Результаты для всех исследованных комбинаций гиперпараметров сведены в таблицу 1.

Таблица 1

Точность (Test accuracy) глобальной модели на тестовом наборе данных в зависимости от количества клиентов (K), скорости обучения (LR) и размера батча (B)

Кол-во клиентов (K)	LR = 0.1, B = 64	LR = 0.1, B = 256	LR = 0.1, B = 1024	LR = 0.001, B = 64	LR = 0.001, B = 256	LR = 0.001, B = 1024
1	0.7252	0.7280	0.7050	0.6636	0.5474	0.4958
2	0.7288	0.7400	0.7040	0.6204	0.5278	0.4818
4	0.7226	0.7180	0.6752	0.5886	0.5102	0.4742
6	0.7288	0.7122	0.6602	0.5724	0.4972	0.4600
8	0.7218	0.7058	0.6432	0.5560	0.4848	0.4636
10	0.7180	0.6778	0.6610	0.5450	0.5022	0.4472
15	0.7152	0.6788	0.5818	0.5350	0.5042	0.4548
20	0.7090	0.6688	0.5444	0.5330	0.4806	0.3910
30	0.6970	0.6486	0.5912	0.5398	0.4840	0.3830
40	0.6814	0.6324	0.0998	0.5286	0.4624	0.1022
Мат. ожидание	0.7148	0.6910	0.5866	0.5682	0.5001	0.4154
Дисперсия	0.0002	0.0014	0.0305	0.0016	0.0006	0.0152

Эксперименты проводились с различным числом узлов (клиентов) в распределенной системе. Одной из целей данной серии экспериментов являлась проверка возможности подбора таких гиперпараметров, чтобы точность (accuracy) на валидационной и тестовой выборке не сильно зависела от количества узлов и была на достаточно высоком уровне.

Были построены графики динамики точности на валидационном наборе для различных наборов гиперпараметров (Рисунок 1).

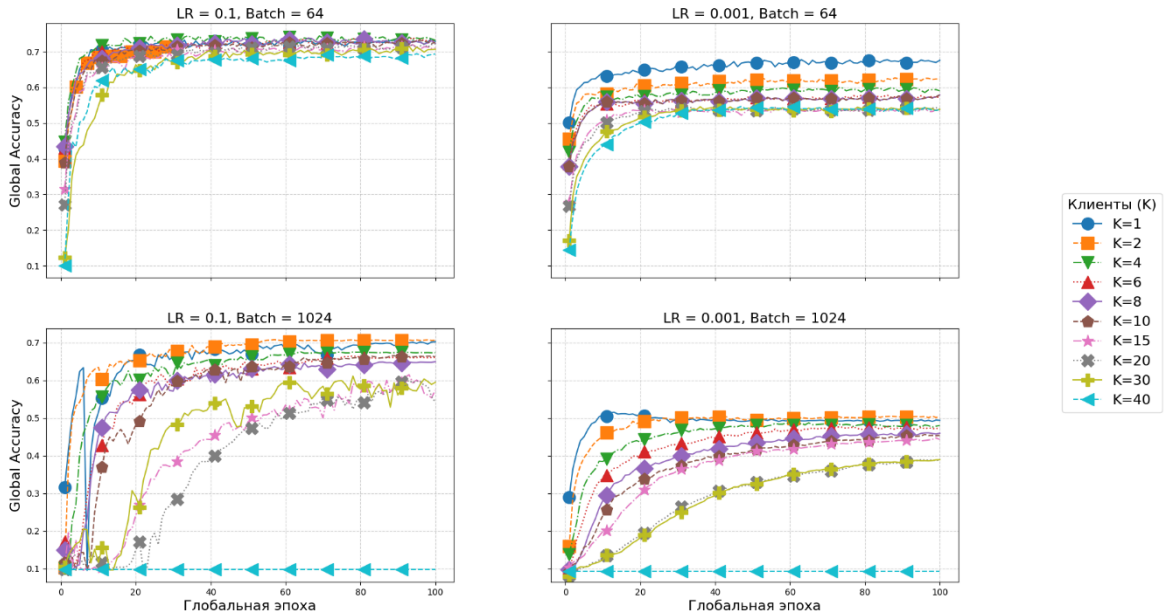


Рисунок 1: Динамика точности на валидационном наборе данных (%) после 100 эпох обучения в зависимости от числа клиентов (K), скорости обучения (LR) и размера пакета (Batch Size)

В таблице 1 и на рисунке 1видно, что при использовании набора гиперпараметров LR=0.1 и Batch Size=64 точность (accuracy) на валидационном наборе данных достаточно высока и значения для разных количеств узлов не сильно различаются.

Уменьшение параметра скорости обучения LR ухудшает результаты для всех вариантов комбинаций значений параметров Batch Size и количества узлов K. При этом следует заметить,

что для больших распределенных систем (с большим количеством узлов) точность падает сильнее.

Несмотря на то, что лучшие результаты по точности accuracy=0.74 (Таблица 1) были достигнуты на другом наборе параметров LR=0.1, Batch Size=256, K=2, дисперсия значений по точности в зависимости от числа узлов была выше, чем для набора LR=0.1 и Batch Size=64.

При этом математическое ожидание точности для набора LR=0.1 и Batch Size=64 выше, чем математическое ожидание для набора LR=0.1, Batch Size=256.

Следовательно, следует заметить, что с ростом числа узлов K объем локального набора данных (N/K) у каждого клиента пропорционально уменьшается. Это приводит к тому, что локальные обновления, вычисленные на малых и потенциально нерепрезентативных выборках, становятся более «шумными». Однако данный недостаток будет нивелирован в случае фиксированного размера локального набора данных, а также при больших датасетах.

На основе анализа динамики точности можно сделать вывод, что для распределенной системы с большим количеством узлов и/или с непостоянным количеством узлов нужно использовать высокие значения скорости обучения и небольшой размер Batch Size.

Помимо точности, важным аспектом производительности распределенной системы является общее время, необходимое для обучения модели. Было измерено полное время выполнения 100 эпох для различных комбинаций гиперпараметров (Рисунок 2).

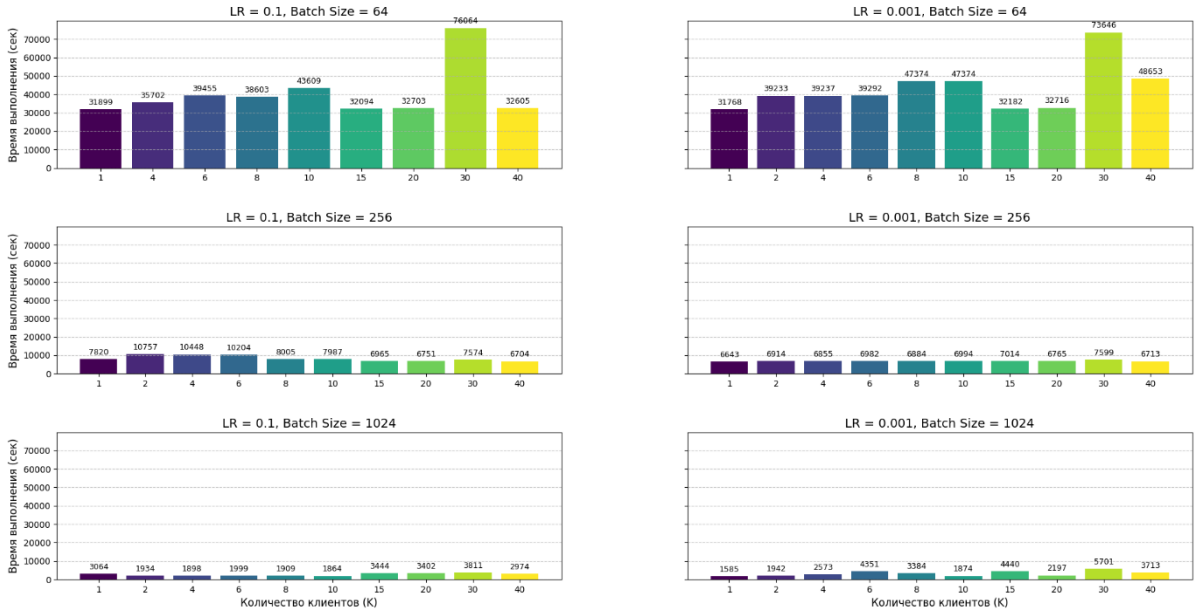


Рисунок 2: Общее время обучения (в секундах) за 100 эпох в зависимости от количества клиентов (K), скорости обучения (LR) и Batch Size

Совместный анализ времени и точности показывает, что не существует единственного набора гиперпараметров. Выбор должен быть компромиссом между скоростью и качеством:

- Использование малых батчей (B=64) является вычислительно неэффективным в исследуемой архитектуре, так как увеличение K приводит к росту временных затрат без гарантированного улучшения точности.
- Использование средних батчей (B=256) демонстрирует наиболее сбалансированное поведение, позволяя найти компромисс между ускорением и итоговой точностью при K в диапазоне 4–15.
- Использование больших батчей (B=1024) обеспечивает максимальное ускорение, однако сопряжен с риском деградации точности и нестабильности обучения при большом числе клиентов.

Стоит отметить, что время обучения для некоторых наборов гиперпараметров меньше для 15-20 узлов, чем для 1 узла. Здесь можно говорить о достаточно эффективном разбиении датасета CIFAR на части между узлами.

Анализ результатов по измерению времени распределенного обучения позволяет сделать вывод о значимости выбора гиперпараметра Batch Size и размера локального набора данных для уменьшения времени распределенного обучения.

4. Заключение

В работе проведено исследование влияния гиперпараметров скорости обучения и размера батча для распределенной системы с различным количеством узлов(клиентов). Был найден набор гиперпараметров, при котором значения точности достаточно высокие для разного количества узлов в распределенной системе. Подтверждено предположение, что время распределенного обучения может быть меньше, чем при локальном глубоком обучении даже на достаточно небольшом датасете. При этом размер батча сильно влияет как на точность распределенного обучения, так и на его скорость. Перспективным направлением дальнейших исследований являются методы распределенного обучения с ограничениями на распространение данных и несбалансированными (non-IID) данными.

Финансовая поддержка: Работа выполнена в рамках государственного задания ИППИ РАН, утвержденного Минобрнауки России.

Список литературы

[1] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (2015) 436–444. doi:10.1038/nature14539.
[2] J. Dean, G. Corrado, R. Monga, K. Chen, M. Devin, Q. V. Le, M. Mao, M. Ranzato, A. Senior, P. Tucker, K. Yang, A. Y. Ng, Large scale distributed deep networks, Advances in Neural Information Processing Systems (NeurIPS) 25 (2012). URL: https://proceedings.neurips.cc/paper/2012/hash/6aca97005c68f1206823815f66102863-Abstract.html.
[3] T. Ben-Nun, T. Hoefler, Demystifying parallel and distributed deep learning: An in-depth concurrency analysis, ACM Computing Surveys (CSUR) 52 (2019) 1–43. doi:10.1145/3320060.
[4] C. Xie, S. Koyejo, I. Gupta, Asynchronous Federated Optimization, in: Proceedings of the 3rd MLSys Conference, Austin, TX, 2020. URL: https://arxiv.org/abs/1903.03934.
[5] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, B. Agüera y Arcas, Communication-efficient learning of deep networks from decentralized data, in: Artificial intelligence and statistics, PMLR, 2017, pp. 1273–1282.
[6] S. Zawad, F. Yan, Hyperparameter tuning for federated learning–systems and practices, in: Q. Yang, L. Wang (Eds.), Federated Learning, Academic Press, 2024, pp. 219–235.
[7] Y. Zhou, P. Ram, T. Salonidis, N. Baracaldo, H. Samulowitz, H. Ludwig, FLoRA: Single-shot hyper-parameter optimization for federated learning, 2021. URL: https://arxiv.org/abs/2112.08524.
[8] S. Reddi, Z. Charles, M. Zaheer, Z. Garrett, K. Rush, J. Konečný, S. Kumar, H. B. McMahan, Adaptive federated optimization, 2020. URL: https://arxiv.org/abs/2003.00295.
[9] H. Zhang, Z. Wu, W.-L. Tam, R. Li, Y.-C. Chang, S. Wang, K. Eykholt, Fedtune: Automatic tuning of federated learning hyper-parameters from system perspective, in: Proceedings of the 2022 IEEE Military Communications Conference (MILCOM), IEEE, 2022, pp. 478–483.
[10] M. Kundroo, T. Kim, Demystifying impact of key hyper-parameters in federated learning: A case study on CIFAR-10 and FashionMNIST, IEEE Access 12 (2024) 23811–23824. doi:10.1109/ACCESS.2024.3364239.
[11] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), IEEE Computer Society, 2016, pp. 770–778.
[12] Krizhevsky, Learning multiple layers of features from tiny images, Technical Report, University of Toronto, Toronto, ON, 2009.
[13] Goodfellow, Y. Bengio, A. Courville, Deep Learning, MIT Press, Cambridge, MA, 2016.

Создание искусственных нейронных сетей для работы в распределенных вычислительных средах

Николай Вершков [1], Наталья Кучукова [1] и Владислав Луценко [1]

[1] *Северо-Кавказский федеральный университет, ул. Пушкина, 1, г. Ставрополь, 355017, Россия*

Аннотация

В статье рассматриваются аспекты обучения и применения по назначению модульных искусственных нейронных сетей с целью их использования в распределенных вычислительных средах. Для достижения поставленной цели предлагается метод проектирования входной информации на пространство ортогональных полиномов с целью получения подпространств, имеющих меньшую размерность и позволяющих осуществлять раздельную обработку с помощью модулей, расположенных на разных вычислительных узлах распределенной среды. Предлагаемый подход имеет преимущество перед методом послойного разделения в объемах информации, передаваемой между модулями и в гибкости вычислений, позволяющей оптимизировать соотношение «объем вычислительных ресурсов/качество».

Ключевые слова

Искусственные нейронные сети, оптимизация нейронных сетей, ортогональные преобразования, модульные нейронные сети, модульное обучение, распределенные вычисления.

1. Введение

Распределенные вычисления становятся все более важной частью современных интеллектуальных технических систем таких как туманные и периферийные вычисления интернета вещей, управление группой беспилотных устройств, группой спутников на орбите и т.п. При этом все больше задач таких как компьютерное зрение и распознавание объектов, работающих в контуре управления объектами, требуют использования искусственных нейронных сетей (ИНС). Высокоточные модели ИНС имеют значительные объемы и требуют больших вычислительных ресурсов. При наличии вычислительных узлов высокой мощности проблем с использованием ИНС нет, вычислительных ресурсов достаточно. Однако смещение акцента на распределенные вычисления часто требует размещения ИНС на узлах, вычислительная мощность которых ограничена. В связи с этим для поддержания модели высокоточной ИНС требуется использовать несколько вычислительных узлов одновременно.

Для решения подобной проблемы наиболее часто используют модели распределенного программирования, например, MapReduce [1]. Такой подход позволяет разделить ИНС на слои (по тем или иным признакам) и выполнять один или несколько слоев на отдельном узле распределенных вычислений [2, 3]. Значительным недостатком этого подхода является большая нагрузка на каналы передачи данных, т.к. объем информации, передаваемый между слоями, очень велик. Другим, также значительным недостатком, является отсутствие гибкости, т.е. приспосабливаемости ИНС, к неопределенностям в наличии необходимых вычислительных ресурсов. Например, в среде туманных вычислений нет достаточного количества свободных узлов, поломка или потеря нескольких беспилотных устройств в группе может привести к отсутствию возможности использования ИНС. В этом случае потребуется

7th International Workshop on Information, Computation, and Control Systems for Distributed Environments (ICCS-DE 2025), July 7–11, 2025, Irkutsk, Russia
EMAIL: vernick61@yandex.ru (A. 1); nkuchukova@ncfu.ru (A. 2); vvlutcenko@ncfu.ru (A. 3)
ORCID: 0000-0001-5756-7612 (A. 1); 0000-0002-8070-0829 (A. 2); 0000-0003-4648-8286 (A. 3)
© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).
ICCS-DE 2025 Workshop Proceedings
DOI: 10.47350/ICCS-DE.2025.07

резервирование вычислительных мощностей, что, в свою очередь, приводит к неэффективному использованию вычислительных узлов распределенной среды. Предлагаемый метод обладает свойством эластичности, т.е. при отсутствии свободных вычислительных узлов ИНС в состоянии выполнять свои функции с помощью такого количества модулей, сколько свободных вычислительных узлов имеется в наличии, хотя и с несколько низким качеством, при этом не требует переобучения. При освобождении вычислительных мощностей в распределенной среде количество модулей может увеличиваться, повышая качество работы ИНС.

2. Реализация модульных ИНС

Теоретической основой ИНС считается доказательство теоремы Колмогорова-Арнольда, которая показала возможность представления непрерывной действительной функции n переменных $f(x_1, x_2, \dots, x_n)$ в виде суперпозиции функций меньшего числа переменных.

В своей работе А.Н. Горбань делает вывод [4] о том, что несмотря на то, что теорема Колмогорова-Арнольда говорит о точном представлении функций многих переменных в классе непрерывных функций, практическое вычисление большинства функций производится приближенно даже при наличии точных формул. А подобное решение уже лежит в плоскости приближения функции $f(x_1, x_2, \dots, x_n)$ на замкнутом ограниченном множестве Q последовательностью полиномов (теоремы Вейерштрасса, Стоуна).

Из обобщения Горбанем теоремы Стоуна следует, что любая нелинейность обладает универсальными аппроксимационными свойствами, т.е. из многочлена p степени $m > 1$ и многочленов первой степени с помощью операций суперпозиции, сложения и умножения на число можно получить все элементы пространства $\boldsymbol{C}[x_1, x_2, \dots, x_n]$. Таким образом, если существуют ИНС S_1, S_2,…, S_k, которые вычисляют функции y_1, y_2, …, y_K от вектора входных сигналов x, то с помощью оператора $f(w_0 + w_1 y_1 + \dots + w_k y_k)$ можно получить результат аппроксимации с заданной точностью. В соответствии с теоремой Стоуна-Горбаня множество функций, вычислимых нейронными сетями с заданной непрерывной нелинейной характеристической функцией, плотно в пространстве непрерывных функций от входных сигналов [4].

Приближенное решение исходной задачи можно получить в виде разложения x по некоторому множеству базисных элементов $\{u_i(t)\}_{i=1}^{k}$

$$x_k = \sum_{i=1}^{k} a_i u_i(t) \quad (1)$$

Базисные функции в (2) могут быть заданы аналитически, алгоритмически или численно. Одними из первых, кто предложил использовать ортогональные преобразования для выделения признаков, являются Н. Ахмед и К.Р. Рао [5]. Используя стандартную архитектуру ИНС авторы [5] показали, что с помощью, например, дискретного косинусного преобразования (ДКП) можно снизить размерность классификатора изображений, что стало первым шагом в оптимизации ИНС, используя приближенное представление входного сигнала.

2.1. Преобразование Фурье как способ приближенного представления входного сигнала ИНС

Наиболее известным и широко используемым оператором в теории передачи информации является преобразование Фурье [6]. Поскольку входными значениями ИНС являются дискретные значения, то можно использовать дискретное преобразование Фурье (ДПФ):

$$Ax(t) = F(\omega) = \sum_{i=0}^{n} x_i e^{-\frac{2\pi j \omega i}{n}}$$

Здесь $x \in L^1(R)$ – входные значения оператора A, $\omega = 2\pi f$ – круговая частота. Для функций, интегрируемых с квадратом на R, можно определить гильбертово пространство

функций $L^2(R)$, на котором преобразование Фурье может быть определено как оператор $A: L^2(R) \to L^2(R)$.

Коэффициент разложения a_i из выражения (1) представляет собой проекцию функции на i-тую ортогональную функцию выбранного базиса на всем промежутке $[0, T]$. Воспользуемся понятием точности приближения для замкнутых систем ортогональных функций [7]. На основании равенства Парсеваля:

$$B = \sum_{i=0}^{\infty} x_i^2 = \sum_{n=0}^{\infty} a_n^2 \quad (2)$$

Тогда относительную интегральную точность приближения с учетом (4) можно оценить как

$$\gamma = \frac{\sum_{n=0}^{k} a_n^2}{B} = \frac{a_0^2}{B} + \frac{a_1^2}{B} + \cdots + \frac{a_n^2}{B} = \frac{B_1}{B} + \frac{B_2}{B} + \cdots + \frac{B_d}{B} \quad (3)$$

Из свойств преобразования Фурье известно, что чем выше степень гладкости функции $x(t)$, тем быстрее убывает ее преобразование Фурье $F(\omega)$. Следовательно, частичные суммы из ряда (2) будут убывать. Тогда интегральная точность приближения (3) не будет значительно убывать при отбрасывании последних членов ряда, а частичные суммы коэффициентов уменьшаются по мере роста номеров гармоник. Таким образом, разделив спектр на две равные части, получим следующие значения интегральной точности приближения для каждой половины спектра:

$$\gamma_1 = \frac{\sum_{n=0}^{k/2} a_n^2}{B} \quad (4)$$

$$\gamma_2 = \frac{\sum_{n=k/2+1}^{k} a_n^2}{B} \quad (5)$$

Поскольку частичная сумма в выражении (5) будет меньше, чем в выражении (4), то точность приближения ИНС, обученной первой частью спектра будет выше, чем второй. В дальнейшем это будет показано экспериментально. В зависимости от решаемой задачи, количество частей, на которые делится спектр входной функции, может быть различным. Кроме того, части спектра не обязательно должны быть одинаковы. Если имеется несколько вычислительных узлов повышенной мощности, для них может быть выделены большая часть спектра с использованием более объемной ИНС.

Преобразование Фурье является эффективным инструментом исследования функций, но имеет существенный недостаток: имея высокую степень частотной локализации обладает очень слабой временной (пространственной) локализацией.

2.2. Использование вейвлет-преобразования для создания модульной структуры входного сигнала ИНС

Если же вместо Фурье-подобных преобразований использовать вейвлет-преобразование [8], то можно совместить операции генерации и отбора признаков и получить выигрыш в количестве значимых признаков без проведения дополнительных исследований [9]. Вейвлеты в общем виде представляют собой систему функций вида

$$\varphi_{a,b}(x) = \sqrt{2^a}\varphi(2^a x - b) \quad (6)$$

Если V_a – пространство, порожденное системой функций (6), то имеют место следующие включения [9] $V_0 \subset V_1 \subset \cdots \subset V_a$. Т.е. получается система вложенных подпространств $V_i \subset L^2(R)$, в каждом из которых выделен ортонормированный базис $\{\varphi_{i,b}(x)\}$. Полученная последовательность подпространств может быть использована для того, чтобы перейти от некоторой функции $f(x)$ из $L^2(R)$ к ее приближению с помощью оператора ортогонального проектирования:

$$P_a: L^2(R) \to V_a, P_a(f) = \sum_{b \in Z} (f, \varphi_{a,b})\varphi_{a,b}(x)$$

Проекции P_i являются приближениями $f(x)$ все более точными с ростом i. Возвращаясь к нейронным сетям, можно провести следующую аналогию. Набор входных векторов $\{f_i(x_j)\}$ в пространстве $L^2(R)$ может быть спроектирован на набор подпространств $V_0 \subset V_1 \subset \cdots \subset V_a$ так,

что каждая проекция P_i является приближением входных данных. Выполнив обучение нейронной сети на проекции P_0 получим максимально грубое приближение ожидаемого результата. Однако, за счет децимации это будет самое «короткое» приближение и для функционирования потребуется минимум вычислительных ресурсов.

Широко известный вейвлет Хаара [5, 10, 11] позволяет разбить пространство $L^2(R)$ на 2 подпространства V_0 и V_1. Используя подпространство V_0 в качестве базового, можно получить модульную архитектуру ИНС, которая позволит для маломощных устройств использовать базовый модуль [11] и, таким образом, получить оптимизированную ИНС для маломощного устройства. Фактически вейвлет Хаара выполняет разбиение пространства коэффициентов на две равные части [10]. Благодаря этому возможно решение задачи, когда базовый модуль позволит решить задачу применения ИНС по назначению на устройствах с невысокой вычислительной мощностью с незначительной потерей точности и без дополнительного обучения. Решение задачи распределенных вычислений требует размещения модулей на узлах вычислительной сети, причем в условиях высокой неопределенности наличия вычислительной мощности изначально может быть использован базовый модуль, а по мере освобождения вычислительных узлов на них могут размещаться дополнительные модули, повышая качество работы ИНС.

При практическом применении предлагаемых теоретических положений нужно учитывать, что подобное преобразование можно выполнить еще раз для каждой половины коэффициентов. В этом случае получится разбить все пространство на 4 части и сформировать 4-хмодульную структуру и двухуровневый вейвлет-спектр, и т.д. Кроме вейвлетов Хаара для решения подобной задачи могут быть использованы вейвлеты Добеши [12], обучение модульных ИНС может вестись одновременно на нескольких уровнях вейвлет-спектра. В зависимости от состава и размера признаков наилучший результат обучения может быть получен на любом уровне вейвлет-спектра.

2.3. Варианты реализации параллельных высокопроизводительных ИНС

Практическая реализация модульных ИНС осуществлялась с помощью библиотеки PyTorch [13] с использованием сета MNIST [14].

При построении модульных ИНС рассматривались 2 основных варианта: 1. в наличии имеется 1 вычислительный узел и на него может быть помещено столько модулей, сколько позволяет его вычислительная мощность; 2. в наличии имеются несколько вычислительных узлов, при этом точное количество узлов и их вычислительные возможности заранее неизвестны.

Задача оптимизации возможностей одного вычислительного узла (вариант 1) решается путем размещения максимально возможного количества модулей с учетом его вычислительных возможностей. В качестве первого примера представляется 4-хмодульная ИНС, построенная с помощью ДКП. ДКП выполняется в 1-м слое прямосвязной ИНС, при этом веса первого слоя установлены с помощью выражения () и запрещено их изменение. Полученные коэффициенты ДКП разделены на 4 равных части и обрабатываются 4-мя отдельными модулями. Модули представляют собой монолитную ИНС, размер которой уменьшен в 4 раза. Результаты обработки всех модулей поступают на вход последнего объединяющего слоя. ИНС была обучена на сете MNIST до достижения качества распознавания 98%.

Таблица 1

Показатели ИНС при подключении нескольких модулей

	1 модуль	2 модуля	3 модуля	4 модуля
Качество распознавания, %	78.8	94	97.2	98
Среднее время 1 цикла, с	0.01	0.02	0.02	0.03

Таблица 1 иллюстрирует качество распознавания цифр при подключении дополнительных модулей. Если в работу включены не все модули, то вместо отсутствующих данных на соответствующий вход объединяющего слоя подается нулевой тензор. Так, 3 модуля дают вполне приемлемый по качеству результат и более эффективное использование вычислительных возможностей узла в плане его быстродействия.

Аналогичный эксперимент был проведен с использованием вейвлет-преобразования Хаара. Первый слой является сверточным, в ядро которого занесены значения фильтра [15]. Значения ядра также зафиксированы – их изменение запрещено. Показатели ИНС с вейвлет-преобразованием приведены в таблице 2.

Таблица 2

Показатели ИНС при подключении нескольких модулей с вейвлет-преобразованием

	1 модуль	2 модуля	3 модуля	4 модуля
Качество распознавания, %	88.1	96.4	97	98.2
Среднее время 1 цикла, с	0.01	0.02	0.02	0.02

Из таблицы 2 видно, что при использовании вейвлет-преобразования качество распознавания модулей несколько выше.

2-й вариант требует несколько иного подхода. Целью проводимого эксперимента было сравнение временных характеристик модульной ИНС с монолитной. Дистрибутивный пакет, включенный в PyTorch [11] позволяет распараллеливать вычисления по процессам и кластерам машины. Для этого он использует семантику передачи сообщений, позволяя каждому процессу передавать данные любому другому процессу. Поддержка параллельных вычислений осуществляется только под управлением Linux-подобных операционных систем. При работе с бэкендом «gloo» вся вычислительная среда разделяется на процессы с возможностью установить необходимое количество потоков в каждом процессе. Т.е. каждый процесс является виртуальной вычислительной средой, в котором выполняется указанный отрезок кода.

Для разделения ИНС на процессы использовался следующий подход:

1. В первом слое выполняется вейвлет-преобразование путем помещения в ядро сверточного слоя фильтра из банка фильтров, полученного с помощью библиотеки PyWavelets [15]. Эксперимент проводился с использованием вейвлета Хаара с размером фильтра 2x2. Вейвлет-преобразование выполняется в отдельном модуле.

2. Полученный результат в виде двух или более последовательностей с размерностью в 2 раза меньшей, чем исходный сигнал поступает на 2 ИНС. Каждая ИНС выполняется в своем процессе размером 16 потоков. Передача данных на входы ИНС и их синхронизация осуществляются сервером данных, который выполняется в отдельном процессе.

3. Результаты обработки модульных ИНС передаются в объединяющий слой, также выполняющийся в отдельном процессе. В процессе обучения используется версия оптимизатора, которая позволяет осуществлять обучение распределенных ИНС.

Временные характеристики монолитной и 2-хмодульной ИНС представлены в таблице 3.

Таблица 3

Временные показатели монолитной и модульной ИНС

	Макс.	Мин.	Среднее
Монолитная ИНС, время выполнения, с	0.536	0.128	0.202
Модульная ИНС, время выполнения, с	1.416	0.065	0.12

Из таблицы 3 видно явное преимущество модульной ИНС по времени выполнения, при том, что у нее на 2 слоя больше (вейвлет-преобразование и объединяющий). Превышение максимального времени выполнения модульной ИНС объясняется тем, что модульная архитектура работает под управлением сервера данных, который управляет запуском и завершением модулей, и передачей данных для обработки.

Таким образом, возможность построения параллельных архитектур для выполнения распределенной обработки данных искусственными нейронными сетями позволяет проектировать высокоточные ИНС на вычислительных узлах в распределенных средах,

создавать модульные структуры, которые превосходят по скорости обработки монолитные структуры, а также имеют возможность адаптироваться к имеющимся свободным вычислительным ресурсам.

3. Благодарности

Исследование выполнено за счет гранта Российского научного фонда № 24-21-00149, https://rscf.ru/project/24-21-00149/.

4. Ссылки

[1] Dean J, Ghemawat S MapReduce: simplified data processing on large clusters. Communications of the ACM 2008; 51(1): 107-113. DOI: 10.1145/1327452.1327492.
[2] Mao J, Chen X, Nixon KW, Krieger C, Chen Y. MoDNN: Local distributed mobile computing system for Deep Neural Network. Design, Automation & Test in Europe Conference & Exhibition (DATE). IEEE 2017; 1396-1401. DOI: 10.23919/DATE.2017.7927211.
[3] Bakhtin VV. Algorithm for decomposition of a monolithic neural network for implemented fog computing in devices based on programmable logic. PNRPU Bulletin. Electrical engineering, information technology, control systems 2022; 41: 123-145. DOI: 10.15593/2224-9397/2022.1.06
[4] Горбань А.Н. Обобщенная аппроксимационная теорема и вычислительные возможности нейронных сетей, Сиб. журн. вычисл. матем., 1:1 (1998), 11–24
[5] Ахмед Н., Рао К.Р. Ортогональные преобразования при обработке цифровых сигналов: Пер. с англ./Под ред. И.Б. Фоменко. – М.: Связь, 1980
[6] А.В. Солодов. Теория информации и ее применение к задачам автоматического управления и контроля. – М.: Издательство «Наука» Главная редакция физико-математической литературы, 1967
[7] Дедус Ф.Ф., Куликова Л.И., Панкратов А.Н., Тетуев Р.К. Классические ортогональные базисы в задачах аналитического описания и обработки информационных сигналов: учеб. пособие к спецкурсу "Обобщ. спектр.-аналит. метод обраб. информ. массивов" // под ред. Ф. Ф. Дедуса; Моск. гос. ун-т им. М. В. Ломоносова, Фак. вычисл. математики и кибернетики (МГУ, ВМК). – М.: Издат. отд. Фак. вычисл. математики и кибернетики МГУ им. М.В. Ломоносова, 2004.
[8] Н. А. Вершков, М. Г. Бабенко, Н. Н. Кучукова, В. А. Кучуков, Н. Н. Кучеров, “Поперечнослойное разделение искусственных нейронных сетей для классификации изображений”, Компьютерная оптика, 48:2 (2024), 312–320.
[9] Смоленцев Н.К. Основы теории вейвлетов. Вейвлеты в MATLAB. – М.: ДМК Пресс, 2019 г.
[10] Haar A. Zur theorie der orthogonalen funktionensysteme. – GeorgAugustUniversitat, Gottingen; 1909.
[11] Vershkov N., Babenko M., Tchernykh A. et al. Optimization of Artificial Neural Networks using Wavelet Transforms. Program Comput Soft 48, 376–384 (2022). https://doi.org/10.1134/S036176882206007X
[12] Добеши И. Десять лекций по вейвлетам. — Ижевск: НИЦ «Регулярная и хаотическая динамика», 2001.
[13] PyTorch (англ.) Дата обращения 29 мая 2025.
[14] Qiao, Yu THE MNIST DATABASE of handwritten digits (2007). Accessed 29.05.2025.
[15] https://pypi.org/project/PyWavelets/. Дата обращения 26 февраля 2024.

Оптимизация функционирования одной цепи поставок на основе эвристического подхода

Александра Вихарева [1] Инна Трофимова [1]

[1]*Санкт-Петербургский государственный университет, Университетская наб., 7--9, Санкт-Петербург, 199034, Российская Федерация*

Аннотация

В данной работе рассмотрена задача оптимизации функционирования цепи поставок, состоящей из поставщика, производителя, дистрибьютора и розничного продавца, на основе алгоритма поиска по дереву Монте-Карло (MCTS). Разработка эффективных методов оптимизации логистических цепей является актуальной задачей, поскольку позволяет минимизировать затраты и регулирует процесс взаимодействий между участниками цепи на разных этапах. В рамках исследования был проведён анализ существующих подходов, выявлены их преимущества и ограничения.

В работе разработана модель на основе алгоритма поиска по дереву Монте-Карло, она исследована и адаптирована для решения поставленной задачи. Она показала высокую адаптивность и эффективность при решении задач управления запасами в условиях ограниченных данных и вычислительных ресурсов. Проведённые экспериментальные исследования, на основе численной реализации данной модели, подтвердили её потенциал в снижении затрат, повышении точности прогнозирования и минимизации негативных эффектов, например, эффекта кнута.

Ключевые слова

Модель цепи поставок, метод поиска по дереву Монте-Карло, оптимизация работы логистической цепи

1. Введение

Современные цепи поставок представляют собой сложные многоуровневые системы, в работе которых необходимо учитывать множество факторов, таких как стохастический спрос, неопределенность времени выполнения заказов и взаимодействие между участниками. Эффективное управление запасами в таких условиях является одной из ключевых задач, поскольку оно напрямую влияет на издержки, уровень обслуживания клиентов и устойчивость функционирования всей цепи. Особенно актуальной является задача оптимизации работы многослойных логистических систем, включающих несколько эшелонов поставок [1]-[3].

Несмотря на существующие подходы к решению подобных задач, многие из них сталкиваются с ограничениями по точности, вычислительной сложности или недостаточной адаптивностью к динамическим условиям. В связи с этим возникает необходимость разработки новых методов и алгоритмов, способных эффективно находить оптимальные решения в условиях высокой сложности и неопределенности.

Использование эвристических подходов позволяет значительно повысить качество решений и снизить затраты времени на их получение. Внедрение таких подходов в управление логистическими системами способствует повышению их эффективности, снижению издержек и улучшению уровня обслуживания клиентов.

7th International Workshop on Information, Computation, and Control Systems for Distributed Environments (ICCS-DE 2025), July 7–11, 2025, Irkutsk, Russia
EMAIL: isolovyeva@mail.ru (A. 1); st087955@student.spbu.ru (A. 2)
ORCID: 0000-0002-0548-9330 (A. 1)
© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).
ICCS-DE 2025 Workshop Proceedings
DOI: 10.47350/ICCS-DE.2025.08

Таким образом, применение методов оптимизации в работе многослойных логистических систем является актуальной задачей, которая существенно повышает эффективность современных бизнес-процессов и обеспечивает конкурентные преимущества на рынке. В последнее время проблему оптимизации цепей поставок все чаще решают с помощью методов искусственного интеллекта. В частности, большое количество научных исследований посвящено применению машинного обучения, нейросетевых моделей и эвристических алгоритмов для решения задач управления запасами и логистики [4]-[7]. Однако, такие подходы требуют значительных затрат времени на обучение моделей и наличия большого объема данных. В данной работе предложено использовать альтернативный подход: метод поиска по дереву Монте-Карло (Monte Carlo Tree Search, MCTS), который не нуждается в больших объемах точных данных для обучения. Этот алгоритм способен быстро адаптироваться к различным условиям и эффективно находить решения даже при ограниченных ресурсах и данных.

В основе MCTS лежит идея последовательного построения дерева решений с использованием случайных симуляций для оценки перспективности различных вариантов. Такой метод позволяет находить стратегии управления запасами, которые приближаются к оптимальным, минимизируя затраты на логистические операции.

Особенность применения MCTS заключается в его способности эффективно работать с большими пространствами решений и динамическими условиями, что особенно важно при моделировании сложных логистических цепей. Также в современных исследованиях отмечается, что интеграция MCTS с методами глубокого обучения (Deep Learning) значительно повышает точность прогнозов и качество принимаемых решений [8], [9]. Глубокие нейросетевые модели позволяют предсказывать вероятности возникновения различных сценариев и оценивать текущие состояния системы, что способствует более точному управлению запасами и снижению негативных эффектов, таких как эффект кнута — ситуация, когда неправильные или запаздывающие решения вызывают цепную реакцию увеличения затрат.

Таким образом, использование MCTS открывает новые возможности для повышения эффективности управления цепями поставок. Эти подходы позволяют не только находить оптимальные стратегии, но и быстро адаптироваться к изменяющимся условиям рынка и внутренним особенностям логистических систем.

2. Математическая модель цепи поставок

Мы рассмотрим последовательную многозвенную цепь поставок, с четырьмя эшелонами: розничный продавец, дистрибьютор, производитель и поставщик (см. Рис. 1), с индексами $i = 1, \dots, 4$ соответственно. В каждом периоде $t = 1, \dots, T$ розничный продавец получает информацию о величине D_t от внешнего клиента (стохастический спрос), после чего пополняет свои запасы, размещая заказ у дистрибьютора. Дистрибьютор принимает решение об объеме запасов, который необходим ему и, соответственно, размещает необходимый заказ у

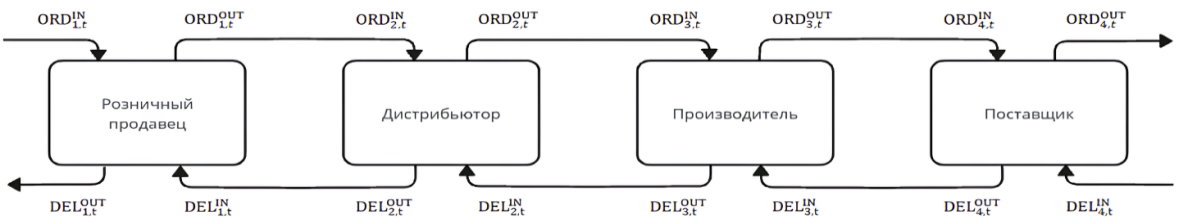


Рисунок 1: Структура цепи поставок

производителя, который, в свою очередь, делает заказ у поставщика. Предполагается, что последний получает пополнения от внешнего источника с неограниченным запасом. Для

упрощения обозначений мы индексируем внешний источник как $i = 5$, а внешнего клиента как $i = 0$.

В каждом периоде каждый участник решает, размещать ли заказ и в каком объеме у соответствующего вышестоящего участника, который в свою очередь обязан выполнить поступивший заказ. Любой заказ, который не может быть выполнен немедленно, откладывается. Таким образом, мы можем рассмотреть 4 варианта развития событий, чтобы определить размер отправленной поставки одного определенного субъекта i в некоторый временной промежуток t:

$$\mathrm{DEL}_{\mathrm{i,t}}^{\mathrm{OUT}} = \begin{cases} \mathrm{ORD}_{\mathrm{i,t}}^{\mathrm{IN}}, & \mathrm{INV}_{\mathrm{i,t}} \geq 0 \cap \mathrm{INV}_{\mathrm{i,t}} + \mathrm{DEL}_{\mathrm{i,t}}^{\mathrm{IN}} \geq \mathrm{ORD}_{\mathrm{i,t}}^{\mathrm{IN}} \\ [\mathrm{INV}_{\mathrm{i,t}}]^{+} + \mathrm{DEL}_{\mathrm{i,t}}^{\mathrm{IN}}, & \mathrm{INV}_{\mathrm{i,t}} \geq 0 \cap \mathrm{INV}_{\mathrm{i,t}} + \mathrm{DEL}_{\mathrm{i,t}}^{\mathrm{IN}} < \mathrm{ORD}_{\mathrm{i,t}}^{\mathrm{IN}} \\ [\mathrm{INV}_{\mathrm{i,t}}]^{-} + \mathrm{ORD}_{\mathrm{i,t}}^{\mathrm{IN}}, & \mathrm{INV}_{\mathrm{i,t}} < 0 \cap \mathrm{DEL}_{\mathrm{i,t}}^{\mathrm{IN}} \geq \mathrm{ORD}_{\mathrm{i,t}}^{\mathrm{IN}} + [\mathrm{INV}_{\mathrm{i,t}}]^{-} \\ \mathrm{DEL}_{\mathrm{i,t}}^{\mathrm{IN}}, & \mathrm{INV}_{\mathrm{i,t}} < 0 \cap \mathrm{DEL}_{\mathrm{i,t}}^{\mathrm{IN}} < \mathrm{ORD}_{\mathrm{i,t}}^{\mathrm{IN}} + [\mathrm{INV}_{\mathrm{i,t}}]^{-} \end{cases} \quad (1)$$

Будем считать, что $[x]^{+} = \max(0, x)$, а $[x]^{-} = |\min(0, x)|$.

Из-за переменных временных задержек, поставки не обязательно прибывают немедленно. Мы предполагаем, что как временные задержки L_t, так и спрос D_t в каждом периоде t являются независимыми и равномерно распределенными со следующими функциями вероятностного распределения:

$$\mathrm{P}(\mathrm{L_t} = \mathrm{l_t}) = \begin{cases} \dfrac{1}{a+1}, & \mathrm{l_t} \in \{0;\ 1;\ \ldots;\ \mathrm{a}\} \\ 0, & \mathrm{l_t} \notin \{0;\ 1;\ \ldots;\ \mathrm{a}\} \end{cases} \quad (2)$$

$$\mathrm{P}(\mathrm{D_t} = \mathrm{d_t}) = \begin{cases} \dfrac{1}{T+1}, & \mathrm{d_t} \in \{0;\ 1; \ldots;\ \mathrm{T}\} \\ 0, & \mathrm{d_t} \notin \{0; 1;\ \ldots;\ \mathrm{T}\} \end{cases}$$

В (2) принято, что время опережения информации равно единице, т.е. заказы, размещенные субъектом, поступают к поставщику выше по цепочке с задержкой в один период. Следовательно,

$$\mathrm{ORD}_{\mathrm{i,t}}^{\mathrm{OUT}} = \mathrm{ORD}_{\mathrm{i+1,t+1}}^{\mathrm{IN}}\ \forall\, \mathrm{i} = 1, \ldots, 4;\ \mathrm{t} = 1, \ldots, \mathrm{T} \quad (3)$$

Поскольку время выполнения заказов является стохастическим и варьируются от 0 до 4, возможно, что несколько поставок поступят одновременно к одному и тому же участнику. Таким образом, входящие поставки $\mathrm{DEL}_{\mathrm{i,t}}^{\mathrm{IN}}$ участника i в период t определяются следующим образом:

$$\mathrm{DEL}_{\mathrm{i,t}}^{\mathrm{IN}} = \sum_{\mathrm{j} \in \mathrm{J}} \mathrm{DEL}_{\mathrm{i+1.t-j}}^{\mathrm{OUT}},\ \mathrm{J} = \{\mathrm{j} | \mathrm{j} = \mathrm{l_{t-j}}\} \quad (4)$$

Обозначим инвентаризационные затраты: пусть $c_{i,+}$ — затраты на хранение единицы продукции субъектом i за один период и $c_{i,-}$ — соответствующие затраты на невыполненный заказ. Общей целью участников является минимизация общих затрат на цепочку поставок в течение горизонта планирования T, т.е. минимизация

$$\sum_{\mathrm{t=1}}^{\mathrm{T}} \sum_{\mathrm{i=1}}^{4} \mathrm{c_{i,+}} [\mathrm{INV_{i,t}}]^{+} + \mathrm{c_{i,-}} [\mathrm{INV_{i,t}}]^{-} \quad (5)$$

В (5) $[\mathrm{INV_{i,t}}]^{+}$— количество товаров, хранящихся в момент времени t у -го участника, $[\mathrm{INV_{i,t}}]^{-}$ — нехватка товаров в момент времени t субъекта под индексом i.

3. Марковские процессы

Формально, Марковский процесс принятия решений представляет собой вектор *(S,A,p,r)*, где *S* — это набор состояний *s*, описывающих окружающую среду, а *A* — это набор возможных действий *a*. Когда действие *a∈A* выбирается в момент времени *t*, среда переходит из состояния *s* в момент времени *t* в состояние *s'* в момент времени *t +1* с вероятностью

перехода $p(s'|s,a)$. Последний компонент $r(s'|s,a)$ — это функция вознаграждения. В нашем случае это сумма затрат на хранение запасов и затрат на невыполненные заказы, после выполнения действия a в состоянии s в момент времени t и последующим переходом в состояние s' в момент времени $t+1$.

Правило принятия решения δ_t — это отображение состояний в действия, т.е. δ_t определяет действие $a \in A$, которое должно быть принято в состоянии s' в момент времени t. Более того, политика $\pi = (\delta_1, \delta_2, \ldots, \delta_{T-1})$ представляет собой последовательность таких правил принятия решений на горизонте планирования T, целью которой является минимизация общих затрат на запасы в цепи поставок на горизонте планирования.

Исходя из всего вышесказанного, определим переменные, подходящие данной структуре цепи поставок. Переменная состояния s, представляющая всю цепочку поставок, задается матрицей (6):

$$\begin{bmatrix} INV_{1,t} + DEL_{1,t}^{IN} - ORD_{1,t}^{IN} & \cdots & INV_{4,t} + DEL_{4,t}^{IN} - ORD_{4,t}^{IN} \\ DEL_{1,t+1}^{IN} & \cdots & DEL_{4,t+1}^{IN} \\ DEL_{1,t+2}^{IN} & \cdots & DEL_{4,t+2}^{IN} \\ DEL_{1,t+3}^{IN} & \cdots & DEL_{4,t+3}^{IN} \\ DEL_{1,t+4}^{IN} & \cdots & DEL_{4,t+4}^{IN} \\ ORD_{1,t}^{IN} & & ORD_{4,t}^{IN} \end{bmatrix}. \quad (6)$$

Хотя MCTS может справиться с чрезвычайно большими пространствами состояний, большие пространства действий часто не поддаются обработке. Поэтому нам нужно сокращенное пространство действий вместо пространства действий, включающего все возможные векторы действий. Средством достижения этой цели является так называемое «$x + y$ правило». Оно гласит, что каждый заказ, размещенный субъектом i, состоит из заказа x, только что полученного от субъекта $i - 1$ плюс дополнительное значение y, которое должно быть определено для каждого субъекта и времени отдельно. Следуя нашим обозначениям и принимая «$x + y$ правило», мы определяем порядок $ORD_{i,t}^{OUT}$ размещено субъектом i в момент времени t следующим образом: $ORD_{i,t}^{OUT} = ORD_{i,t}^{IN} + y_{i,t}$. Следовательно, набор доступных действий в состоянии s в момент времени t запишем в виде (7):

$$A_s = \{a = [y_{1,t}; y_{2,t}; y_{3,t}; y_{4,t}] | y_{i,t} \in \{-3; -2; \ldots; 4; 5\} \bigwedge y_{i,t} \geq -ORD_{i,t}^{IN} \ \forall i\}. \quad (7)$$

После действия $a \in A_s$, окружающая среда переходит из состояния s в состояние s' с вероятностью $p(s'|s,a)$. Таким образом, последующее состояние s' зависит не только от действия a, но и от функции вероятности p и, следовательно, как от спроса клиентов, так и от времени выполнения заказа. Следовательно, суммарное ожидаемое вознаграждение определяется по формуле:

$$\sum_{s' \in S} \left(\sum_{i=1}^{4} c_{i,+} \left[INV_{i,t+1} \right]^{+} + c_{i,-} \left[INV_{i,t+1} \right]^{-} \right) p(s'|s,a). \quad (8)$$

Важно отметить, что в нашем случае вознаграждение на самом деле означает затраты. По этой причине мы стремимся найти политику, которая минимизирует сумму ожидаемых вознаграждений за горизонт планирования.

4. Алгоритм поиска по дереву Монте-Карло

Данный метод исследования, предложенный Прейлом и Краппом [10] в 2022 году, продемонстрировал высокие показатели эффективности и достиг значительных результатов в решении поставленных задач. Однако, несмотря на его перспективность и положительные результаты, данный подход не получил широкого распространения в научной и практической среде. Однако, покажем, что использование данного подхода имеет значительный практический потенциал для разработки эффективных решений в области оптимизации цепочек поставок и управления запасами.

Базовая версия MCTS основана на четырех шагах (см. Рис. 2), которые выполняются последовательно в каждой итерации. Выбор: начиная с корневого узла, политика выбора потомков рекурсивно применяется для спуска по дереву до тех пор, пока не будет достигнут самый срочный расширяемый узел. Узел расширяемый, если он представляет нетерминальное состояние и имеет не посещённых (т.е. неразвернутых) потомков. В контексте нашей задачи, если текущее состояние s находится в наборе расширяемых состояний, мы выбираем действие, которое минимизирует критерий выбора, который мы можем выразить с помощью следующей формулы:

$$Q(s,a) - C\sqrt{\frac{logN(s)}{N(s,a)}}, \tag{9}$$

где $C > 0$ - константа исследования, $N(s, a)$ - общее количество симуляций, в которых действие a было выбрано в состоянии s, а $N(s)$ - общее количество симуляций, запущенных из состояния s. В противном случае мы переходим к шагу расширения. Шаг выбора применяется рекурсивно до тех пор, пока не будет достигнуто состояние, которое еще не было расширено.
Расширение: в дерево поиска добавляются дочерние узлы в соответствии с доступными действиями.

Моделирование: начиная с неразвернутого состояния, запускается моделирование для рекурсивного обхода дерева до тех пор, пока не будет достигнута глубина h. Моделирование выполняется в соответствии с политикой развертывания π_{rollout}, которая в нашем случае полностью случайна. Это позволяет нам рассчитать затраты $r(s,a,s')$, а также сумму q этих затрат на пути из состояния в момент времени t в состояние в момент времени $t + h$.
Обратное распространение: оценочное значение распространяется в обратном направлении от последнего выбранного узла к корневому узлу дерева поиска. Функция затрат $Q(s, a)$

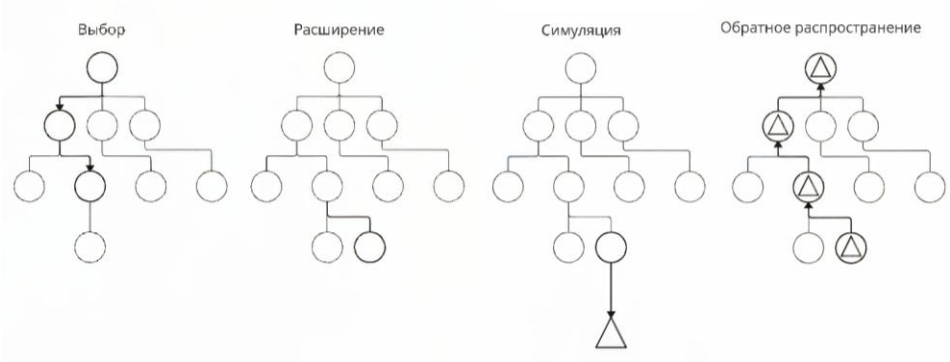


Рисунок 2: Одна итерация алгоритма MCTS
обновляется из состояния, недавно добавленного на шаге расширения, обратно в корневое состояние в момент времени t путем обратного распространения q.

5. Результаты численной реализации

В ходе исследования было проведено изучение влияния параметра C в формуле (9) на распределение посещений дочерних узлов в дереве поиска. При значениях C, меньших единицы, алгоритм преимущественно фокусировался на первом найденном варианте с удовлетворительными результатами, что ограничивало полноту исследования пространства решений. В то же время при увеличении значения C наблюдалось более равномерное распределение посещений между дочерними узлами, что способствовало более глубокому и всестороннему анализу возможных действий.

Мы сравниваем результаты, полученные с помощью подхода MCTS, с результатами из статьи [11]. Для возможности сравнения численных реализаций, применим MCTS к четырем тестовым задачам TP1,…,TP4 статьи [11]. Все исследования используют такие начальные настройки: уровни запасов изначально равны 12 единицам на участника цепочки. Каждое звено

разместило «предварительный заказ» на 4 единицы у соответствующего вышестоящего субъекта до фактического размещения первого «реального» заказа в момент времени. Из-за времени подачи информации эти предварительные заказы принимаются в первом периоде. Затем соответствующие количества помещаются в транзит и прибудут в момент времени $t = 2$. Кроме того, 4 единицы находятся в пути к каждому субъекту и прибывают в момент времени $t = 1$.

Таблица 1

Полученные затраты при различных подходах

	TP1	TP2	TP3	TP4
Генетический алгоритм (GA)	2555	3109	4156	4330
Обучение с подкреплением (RL)	2417	3169	4038	4205
MCTS	2144	1842	2677	2317

Метод MCTS показывает лучшие результаты по сравнению с другими методами во всех тестовых задачах. Это неудивительно, поскольку MCTS обновляет стратегию заказов на основе информации о нескольких ближайших этапах, тогда как подходы GA и RL используют только стратегии заказов на один ближайший этап.

6. Заключение

Метод поиска по дереву Монте-Карло показал высокую эффективность при решении задачи управления запасами в многоуровневой цепочке поставок. Алгоритм успешно балансирует между исследованием новых вариантов и использованием уже известных успешных стратегий, что способствует снижению логистических издержек. Реализация данного алгоритма позволяет автоматизировать процесс принятия решений по формированию заказов и управлению ими, что способствует снижению эффекта кнута и улучшению стабильности работы логистической цепи. При тестировании работы алгоритма удалось выяснить, что параметры алгоритма, такие как количество итераций и коэффициент исследования C существенно влияют на качество и скорость поиска оптимальных решений.

7. Литература

[1] D.Ivanov, B.Sokolov, Adaptive Supply Chain Management, London, Springer, 2010. doi: 10.1007/978-1-84882-952-7

[2] Sokolov B., Trofimova I., Nazarov D., Zakharov V. Modification of Multiple-Model Description and Planning and Update Control Algorithms of Supply Chain, IFAC-PapersOnLine, in: Proceedings of the 9th IFAC Conference on Manufacturing Modelling, Management and Control, MIM 2019, 2019, 1972–1977. doi: 10.1016/j.ifacol.2019.11.492

[3] Ivanov D., Sokolov B., Solovyeva I., Dolgui A., Jie F., Dynamic Recovery Polices for Time-Critical Supply Chains Under Conditions of Ripple Effect, Int. J. of Production Research, 2016, 7245–7258. doi: 10.1080/00207543.2016.1161253

[4] Gayam S. R., Yellu R. R., Thuniki P., Optimizing supply chain management through artificial Intelligence: techniques for predictive maintenance, demand forecasting, and inventory optimization , Journal of AI-Assisted Scientific Discovery, 2021, 129–144. . URL: http:// https://scienceacadpress.com/index.php/jaasd/article/view/58

[5] Kumar P. et al. AI-enhanced inventory and demand forecasting: Using AI to optimize inventory management and predict customer demand , World journal of advanced research and reviews, 2024, 1931–1944. doi:10.30574/wjarr.2024.23.1.2173

[6] Sahu M. K. Advanced AI Techniques for Optimizing Inventory Management and Demand Forecasting in Retail Supply Chains,бJournal of Bioinformatics and Artificial Intelligence, 2021, 190– 224.

[7] Toorajipour R. et al. Artificial intelligence in supply chain management: A systematic literature review, Journal of Business Research, 2021, 502–517. doi:10.1016/j.jbusres.2020.09.009
[8] Saleem R. et al. Explaining deep neural networks: A survey on the global interpretation methods, Neurocomputing, 2022, 165–180. doi:10.1016/j.neucom.2022.09.129
[9] Samek W. et al. Explaining deep neural networks and beyond: A review of methods and applications, in: Proceedings of the IEEE, 2021, 247–278. doi:10.1109/JPROC.2021.3060483
[10] Preil D., Krapp M. Artificial intelligence-based inventory management: a Monte Carlo tree search approach, Annals of Operations Research, 2022, 415–439. doi:10.1007/s10479-021-03935-2
[11] Chaharsooghi S. K., Heydari J., Zegordi S. H. A reinforcement learning model for supply chain ordering management: An application to the beer game, Decision Support Systems, 2008, 949–959. doi:10.1016/j.dss.2008.03.007

Оценка эффективности управления вычислениями диспетчером научных рабочих процессов

Михаил Воскобойников, Александр Феоктистов

Институт динамики систем и теории управления им. В.М. Матросова СО РАН, ул. Лермонтова, д. 134, 664033, Иркутск, Россия

Аннотация

Рассматривается проблема повышения эффективности управления вычислениями при выполнении научных приложений в гетерогенной распределенной вычислительной среде, схемы решения задач в которых реализуются научными рабочими процессами. Одним из ключевых факторов, влияющих на время выполнения рабочих процессов, является время, затрачиваемое на передачу данных между их модулями. В известных системах управления рабочими процессами передача данных между модулями зачастую осуществляется через центральный узел, что увеличивает время выполнения этих процессов. В работе предложены средства управления вычислениями с учетом структуры рабочего процесса и передачи данных напрямую между узлами. Разработан эвристический алгоритм назначения узлов для запуска модулей рабочего процесса на основе оценки времени их выполнения и размера данных, передаваемых между ними. Результаты экспериментов показывают существенное сокращение времени выполнения ряда известных научных рабочих процессов при использовании данного алгоритма относительно применения алгоритмов подобного назначения в системе HTCondor.

Ключевые слова

Научные приложения, рабочие процессы, назначение узлов, передача данных

1. Введение

Управление вычислениями (планирование вычислений и назначение ресурсов) является чрезвычайно актуальной проблемой при проведении крупномасштабных экспериментов в гетерогенной распределенной вычислительной среде (ГРВС). Как правило, в современных научных приложениях план решения задачи реализуется научным рабочим процессом (НРП) – информационно-вычислительной структурой, определяющей бизнес-логику предметной области, расчетные данные, набор используемых прикладных модулей и вычислительные ресурсы, требуемые для проведения экспериментов. НРП может быть абстрактным или конкретным. Абстрактный НРП описывает схему решения задачи без привязки ее операций к конкретному программному обеспечению (ПО) и вычислительным ресурсам. В конкретном НРП операции схемы решения задачи связываются с определенным ПО, которое выполняется на заданных ресурсах.

Как правило, при построении НРП используют процедурную или непроцедурную постановку задачи. В рамках процедурной постановки задачи формируется последовательность операций НРП, а затем осуществляется информационное планирование входных и выходных параметров. При непроцедурной постановке задачи задаются исходные (входные) и целевые (выходные) параметры НРП. Далее на вычислительной модели предметной области (описании ее параметров, вычислительных операций, операций обработки данных, отношений между параметрами и операциями) осуществляется автоматическое планирование последовательности

6th International Workshop on Information, Computation, and Control Systems for Distributed Environments (ICCS-DE 2025), July 07–012, 2025, Irkutsk, Russia
EMAIL: mikev1988@mail.ru (A. 1); agf@icc.ru (A. 2)
ORCID: 0000-0003-3034-4907 (A. 1); 0000-0002-9127-6162 (A. 2)
© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).
ICCS-DE 2025 Workshop Proceedings
DOI: 10.47350/ICCS-DE.2025.09

операций НРП, необходимых для решения поставленной задачи. В системах управления НРП (СУ НРП) автоматическое планирование вычислений осуществляется одним из ее компонентов (например, планировщиком вычислений).

СУ НРП – это программное средство, позволяющее частично автоматизировать процессы создания и выполнения НРП, а также управления им в соответствии с его информационно-вычислительной структурой. В СУ НРП рабочий процесс зачастую задается в виде ориентированного ациклического графа (англ. Directed Acyclic Graph, DAG). При этом вершины DAG соответствуют операциям НРП, реализуемым в дальнейшем прикладными модулями или сервисами, а ребра – потокам данных между ними. На практике хорошо известны, поддерживаются, развиваются и широко используются следующие СУ НРП: UNICORE [1], Galaxy [2], Pegasus [3], DAGMan [4], Workflow-as-a-Service Cloud Platform [5], Apache Airflow [6], HyperFlow [7] и Orlando Tools [8].

Метапланировщик DAGMan осуществляет построение плана решения задачи на основе описания НРП в форме DAG. В качестве системы управления прохождением заданий (СУПЗ) на вычислительном кластере выступает HTCondor. Узлы кластера HTCondor подразделяются по следующим ролям: узел Submit, с которого происходит отправка заданий и исходных данных, узел Execute, предназначенный для выполнения заданий, и узел Master, координирующий работу всех узлов кластера. Перед запуском каждого прикладного модуля происходит перемещение его входных данных с узла Submit на узел Execute. После завершения выполнения прикладного модуля его выходные данные перемещаются обратно на узел Submit. Как правило, НРП имеют сложную структуру, состоящую из большого числа операций, экземпляры которых могут выполняться последовательно и параллельно. В связи с этим число пересылок данных в худшем случае равняется $n * 2m$, где n – число операций НРП, m – число вычислительных узлов. При больших объемах пересылаемых данных это может приводить к существенному увеличению времени выполнения НРП и вычислительного эксперимента в целом.

В работе предложены средства управления вычислениями с учетом структуры рабочего процесса и передачи данных напрямую между узлами, обеспечивающие сокращение время проведения экспериментов в сравнении с DAGMan и HTCondor.

2. Управление вычислениями диспетчером НРП

В состав инструментального комплекса FDE-SWFs [9] входит диспетчер НРП, который осуществляет управление вычислительными процессами и средой их выполнения. В качестве СУПЗ в диспетчере НРП используется HTCondor. Схема управления вычислениями представлена на рис. 1. В HPCondor используется централизованный механизм передачи данных. При запуске НРП на вычислительных узлах создаются временные папки, в которые перемещаются данные и модули. Задание выполняется от имени пользователя nobody. Данный пользователь не имеет разрешений на выполнение модулей вне созданной временной папки и команд на перемещение данных между вычислительными узлами. Организация прямого перемещения данных между вычислительными узлами и выполнения на них модулей включает следующие этапы:

1) задание пользователя для каждого слота HTCondor (слот представляет собой единицу представления ресурсов вычислительного узла) с разрешением на чтение/запись в папки вне временной папки, создаваемой в операционной системе (ОС) при выполнении задания;
2) планирование очередности выполнения модулей для обеспечения минимизации времени выполнения НРП с учетом перемещения данных между узлами;
3) генерация скриптов-оболочек, в которых прописываются команды для выполнения модулей и передачи данных между вычислительными узлами в соответствии с НРП;
4) формирование паспортов заданий, в которых в качестве исполняемых модулей указываются скрипты-оболочки и адреса вычислительных узлов.

На рис. 2 показана схема работы механизма передачи данных через центральный узел. Можно заметить, что при запуске и завершении выполнения каждого прикладного модуля НРП происходит передача входных и выходных данных между вычислительными узлами и

центральным узлом. На рис. 2 изображена схема работы механизма перемещения данных через центральный узел.

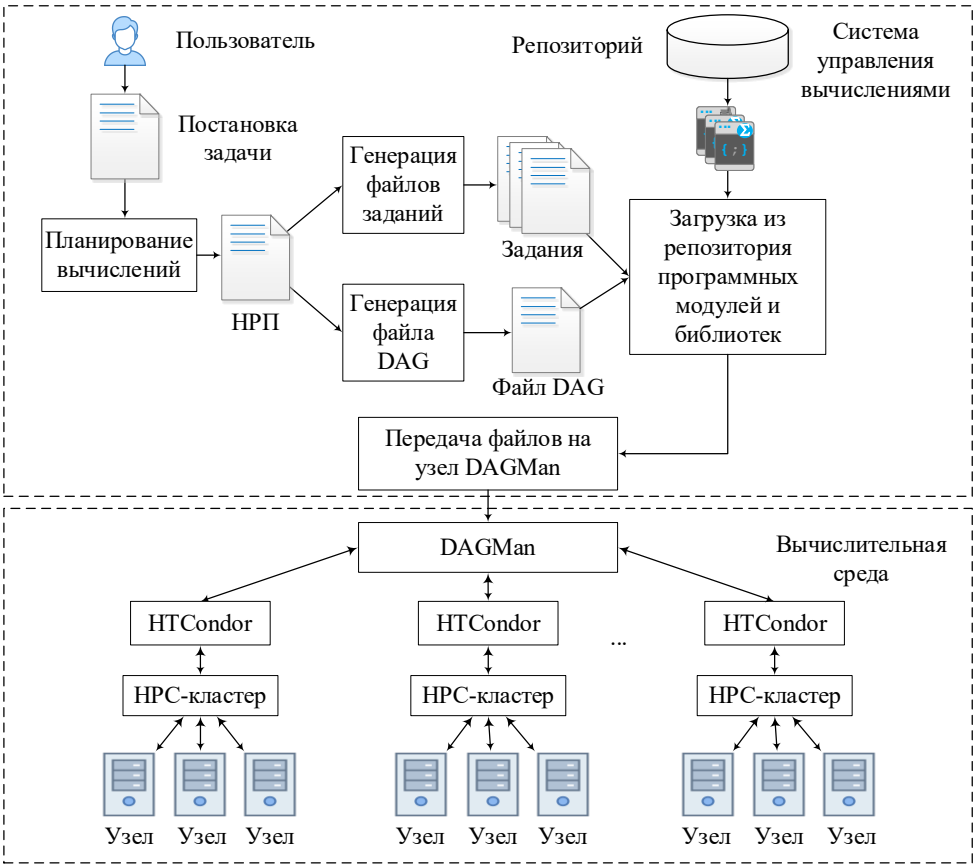


Рисунок 1: Схема взаимодействия диспетчера НРП с DAGMan и HTCondor

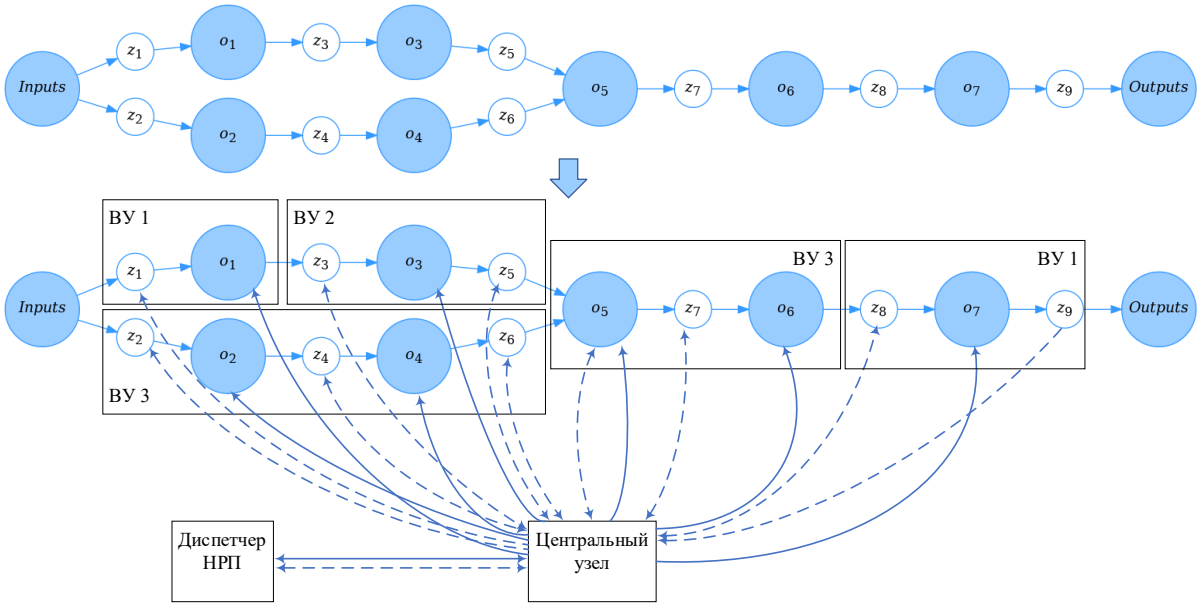


Рисунок 2: Механизм передачи данных через центральный узел

На рис. 3 приведена схема работы механизма передачи данных между вычислительными узлами на основе информации о структуре НРП.

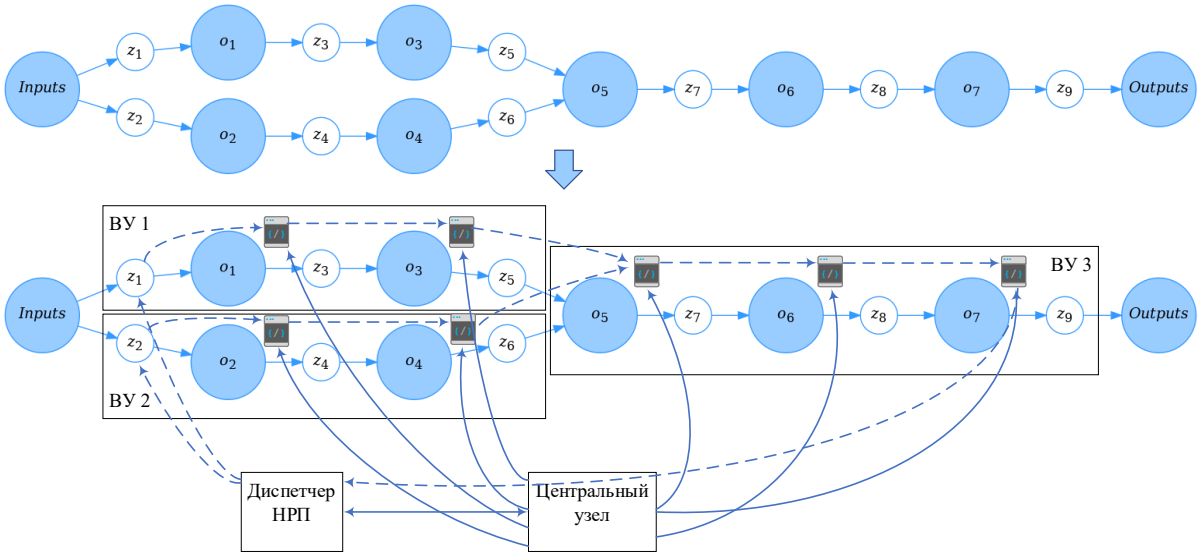


Рисунок 3: Механизм прямой передачи данных между вычислительными узлами

4. Постановка задачи

Пусть имеются m узлов ГРВС (включая центральный узел), на которых должны быть выполнены n операций НРП, где b – индекс основного узла. Введем следующие обозначения:

- t_{il} – время выполнения i-й операции на l-м узле, $i \in \overline{1,n}$, $l \in \overline{1,m}$;
- P – булева матрица размерности $n \times n$, элемент матрицы $p_{ij} = 1$ означает, что i-я операция зависит по входу от j-й операции (j-я операция предшествует i-й операции), $i,j \in \overline{1,n}$;
- D – матрица размерности $m \times m$, отражающая оценки размеров передаваемых данных между узлами, элемент матрицы $d_{lr} > 0$ показывает размер данных, передаваемых между l-м и r-м узлами, $r,l \in \overline{1,m}$;
- X – булева матрица размерности $m \times n$, элемент которой x_{li} показывает, что i-я операция выполняется на l-м узле;
- матрица W размерности $m \times m$ предоставляет информацию о пропускной способности интерконнекта между узлами, элемент матрицы $w_{lr} \geq 0$ демонстрирует пропускную способность интерконнекта между l-м и r-м узлами;
- $0 < \omega_{lr}(t) \leq 1$ – коэффициент снижения пропускной способности интерконнекта в момент времени t между l-м и r-м узлами;
- где τ_{ir} (τ_{jl}) – оценка времени, прошедшего с начала обработки схемы до завершения операции o_i (o_j) на r-м (l-м) узле;
- q_{ir} – оценка времени нахождения модуля, реализующего операцию o_i (в то время, когда все данные, необходимые для его выполнения, готовы) в очереди r-го узла.
- N_l – таблица операций, которые назначены на l-й узел.

Матрица X должна удовлетворять следующим условиям (1) и (2):

$$\bigvee_{i=1}^{n} \bigvee_{l=1}^{m-1} \bigvee_{k=l+1}^{m} (x_{li} \wedge x_{ki}) = 0, \tag{1}$$

$$\bigvee_{i=1}^{n} \bigwedge_{l=1}^{m} \bar{x}_{li} = 0. \tag{2}$$

В случае выполнения НРП средствами СУПЗ HTCondor, где выбор и назначение узлов для запуска операций осуществляется главным менеджером этой системы, а обмен данными между узлами осуществляется через центральный узел, оценка $C(X)$ времени выполнения НРП в асинхронном режиме по готовности данных определяется следующим образом:

$$C(X) = \max_{i=\overline{1,n}} \tau_{il}, \tag{3}$$

$$\tau_{il} = q_{il} + t_{il} + \max_{\forall j \in \overline{1,n}: p_{ij}=1, i \neq j, r \in \overline{1,m}} \left(\tau_{jr} + \frac{d_{rl}}{\omega_{rb}(t) w_{rb}} + \frac{d_{rl}}{\omega_{bl}(t) w_{bl}} \right). \quad (4)$$

Обмен данными между узлами через центральный узел существенно увеличивает общее время выполнения НРП. Поэтому в диспетчере НРП реализован дополнительный механизм передачи данных напрямую с узла выполнения предшествующей операции на узел выполнения последующей операции. В этом случае, если операции выполняются на одном узле, то размер передаваемых данных между ними считается равным нулю. Тогда оценка $Y(X)$ времени выполнения НРП в асинхронном режиме по готовности данных с помощью диспетчера НРП определяется следующим образом:

$$Y(X) = \max_{i=\overline{1,n}} \tau_{il}, \quad (5)$$

$$\tau_{il} = q_{il} + t_{il} + \max_{\forall j \in \overline{1,n}: p_{ij}=1, i \neq j} \left(\tau_{jr} + \frac{s}{\omega_{lr}(t) w_{lr}} \right), \quad (6)$$

$$s = \begin{cases} d_{lr}, & \text{если } l \neq r, \\ 0 & \text{и противном случае.} \end{cases} \quad (7)$$

В (5)-(7) $Y(X)$ является целевой функцией, которую требуется минимизировать:

$$Y(X) \downarrow \min_{x_{li}}. \quad (8)$$

В общем случае задача (8) является NP-трудной задачей [10], оптимальное решение которой сложно получить за время, приемлемое для управления вычислениями в реальном времени. Поэтому в статье предлагается эвристический алгоритм решения этой задачи.

5. Алгоритм назначения операций на вычислительные узлы

Суть алгоритма **A** назначения операций на вычислительные узлы состоит в следующем: каждому узлу вычислительной модели назначаются такие операции из очереди операций, чтобы общее время выполнения НРП было минимальным. Для этого выполняется оценка времени выполнения операции на каждом узле с учетом оценок размера передаваемых данных между операциями и скоростью передачи данных между узлами ГРВС. При этом если на очередном шаге работы алгоритма одному и тому же узлу назначаются две и более операции, то выбирается та операция, которая предшествует большему числу операций НРП.

Алгоритм А. Назначение операций на вычислительные узлы
Inputs: *P, T, D, O, n, m, q, W, ω*
Outputs: $nodes$

```
t = 0
freeNodeList = Array()
for k = 1,m step 1 do
  freeNodeList[k] = True
knownParams = get_known_params(O)
readyOperations = get_ready_operations(O)
subOperList= get_subsequent_operations(P)
nodeTime = Array()
while length(readyOperations) > 0 do
  operation = getOperation(readyOperations)
  currNodes = Array()
  times = Array()
  for l in freeNodeList do
    for r in freeNodeList do
      for i = 1,n step 1 do
        if p[i] == 1 and use[i] == 1 then
          times.add(calcExecTimes(operation, T, D, W, ω, nodeTime))
        end if
```

```
19          end do
20        end do
21      end do
22      currNodes = get_curr_nodes(times)
23      k = getNodeMinTimes(times)
24      if length(currNodes[k]) == 1 then
25         currOperation = currNodes[k]
26         nodeTime[k] = get_operation_time(currOperation, T, D)
27         readyOperation.remove(currOperation)
28         nodes.add(currOperation)
29         freeNodeList[k] = False
30      else if length(currNodes[k]) > 1 then
31         currOperation = get_suitable_operation(currNodes[k], subOperList)
32         nodeTime[k] = set_operation_to_node(currOperation, T, D, W, ω)
33         nodes.add(currOperation)
34         readyOperation.remove(currOperation)
35         freeNodeList[k] = False
36      end if
37      do
38          countExecutedOper = get_executed_operation(nodes)
39          if countExecutedOper == n then
40             exit
41          t = getTimeShift(freeNodeList, nodeTime)
42          for k = 1,m step 1 do
43             if nodeTime[k] == t then
44                operation = get_operation_from_node(nodes[k])
45                freeNodeList[k] = True
46                knownParams = knownParams.union(operation)
47             end if
48          end do
49          readyOperations = get_ready_operations(O)
50       while length(readyOperations) == 0
51    end do
```

6. Вычислительный эксперимент

Для демонстрации сокращения времени выполнения вычислений в ГРВС с помощью разработанных средств были выбраны два известных НРП – Epigenomics и CyberShake [11], структура которых приведена на рис. 4.

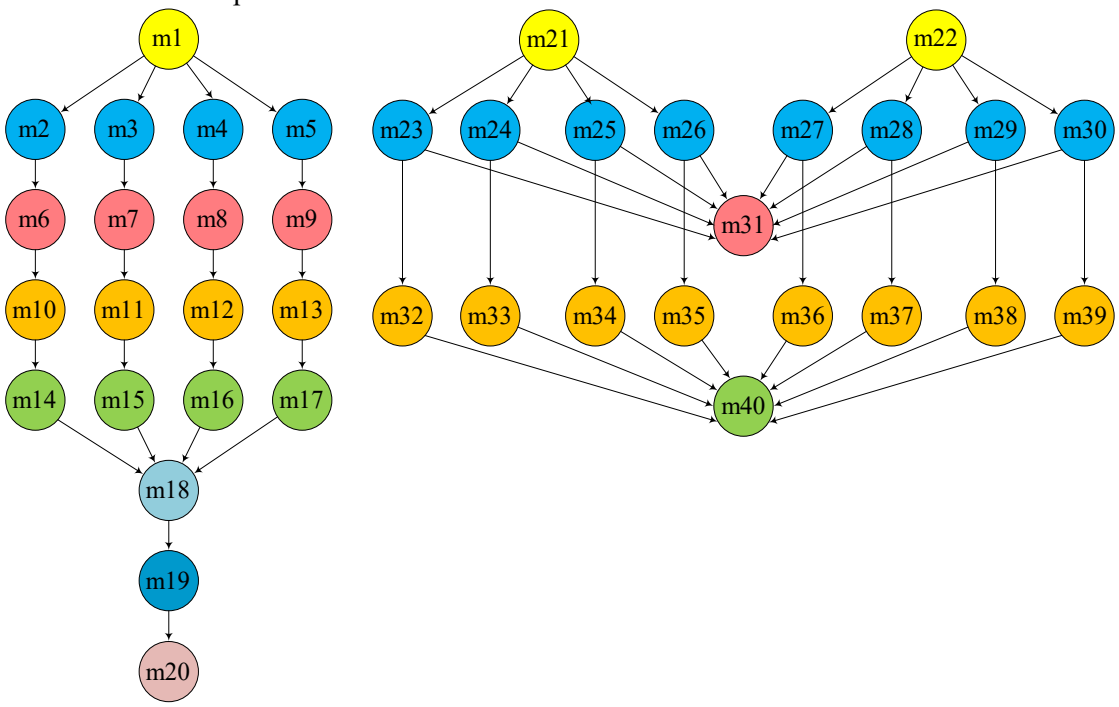
В рамках эксперимента реализованы три схемы выполнения НРП под управлением HTCondor:

1) с централизованной передачей данных;
2) с помощью механизма прямой передачи данных;
3) с учетом структуры и использованием механизма прямой передачи данных.

Вычислительная среда включает четыре узла со следующими характеристиками: 1 CPU, 2 ГБ памяти, жесткий диск размером 30 ГБ, пропускная способности сети – 5 Гбит/с. Расписания времени выполнения модулей Epigenomics и CyberShake приведены соответственно на рис. 5 и 6. Очевидно, что при переходе от схемы 1 к схемам 2 и 3 расписание выполнения НРП становится все более плотным, задержки на передачу данных становятся короче. Диаграмма времени выполнения Epigenomics и CyberShake для трех схем дана на рис. 7.

В случае выполнения Epigenomics схемы 2 и 3 показали преимущество по сокращению времени вычислений в сравнении со схемой 1 более чем на 32% и 40% соответственно. Для

CyberShake это преимущество оказалось еще существеннее. Схемы 2 и 3 позволили сократить время вычислений в сравнении со схемой 1 более чем на 47% и 53% соответственно.



а) б)

Рисунок 4: НРП: Epigenomics (а) и CyberShake (б)

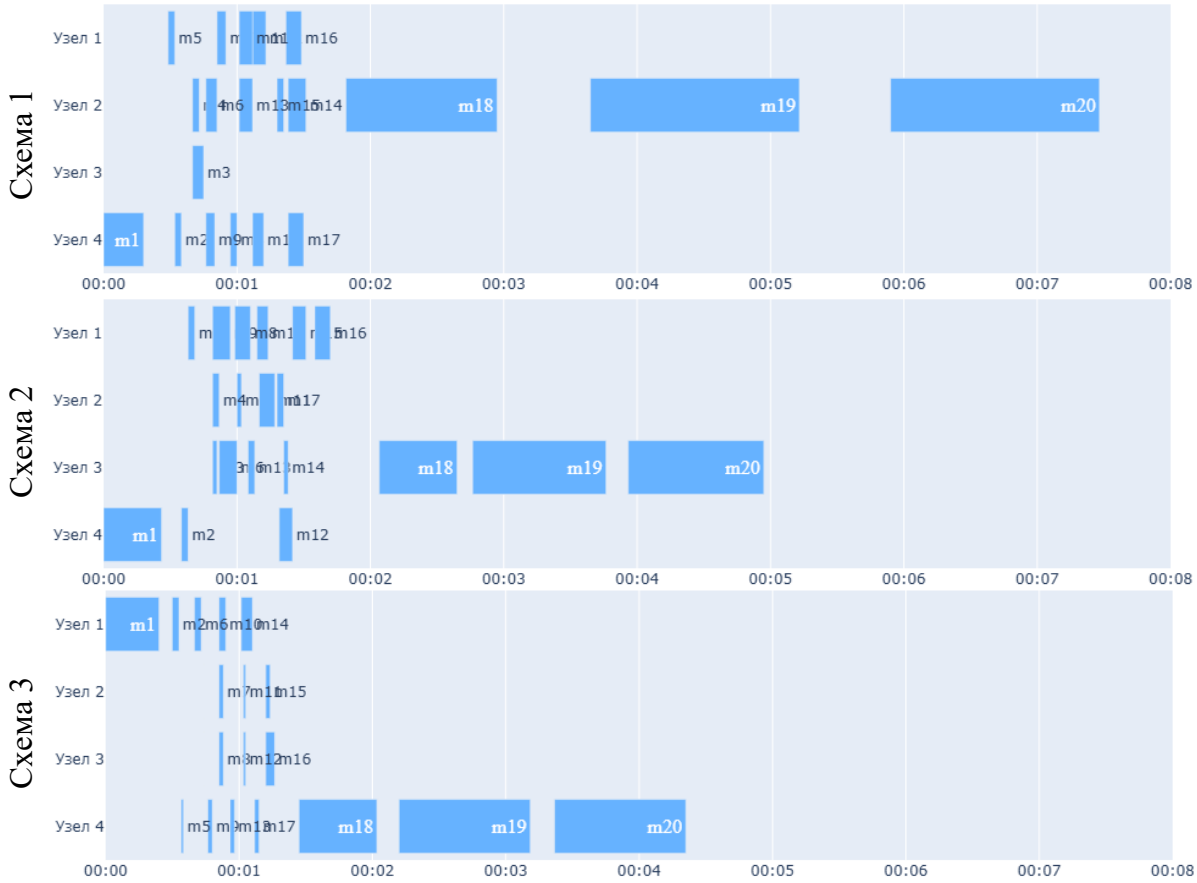


Рисунок 5: Расписания времени выполнения модулей Epigenomics

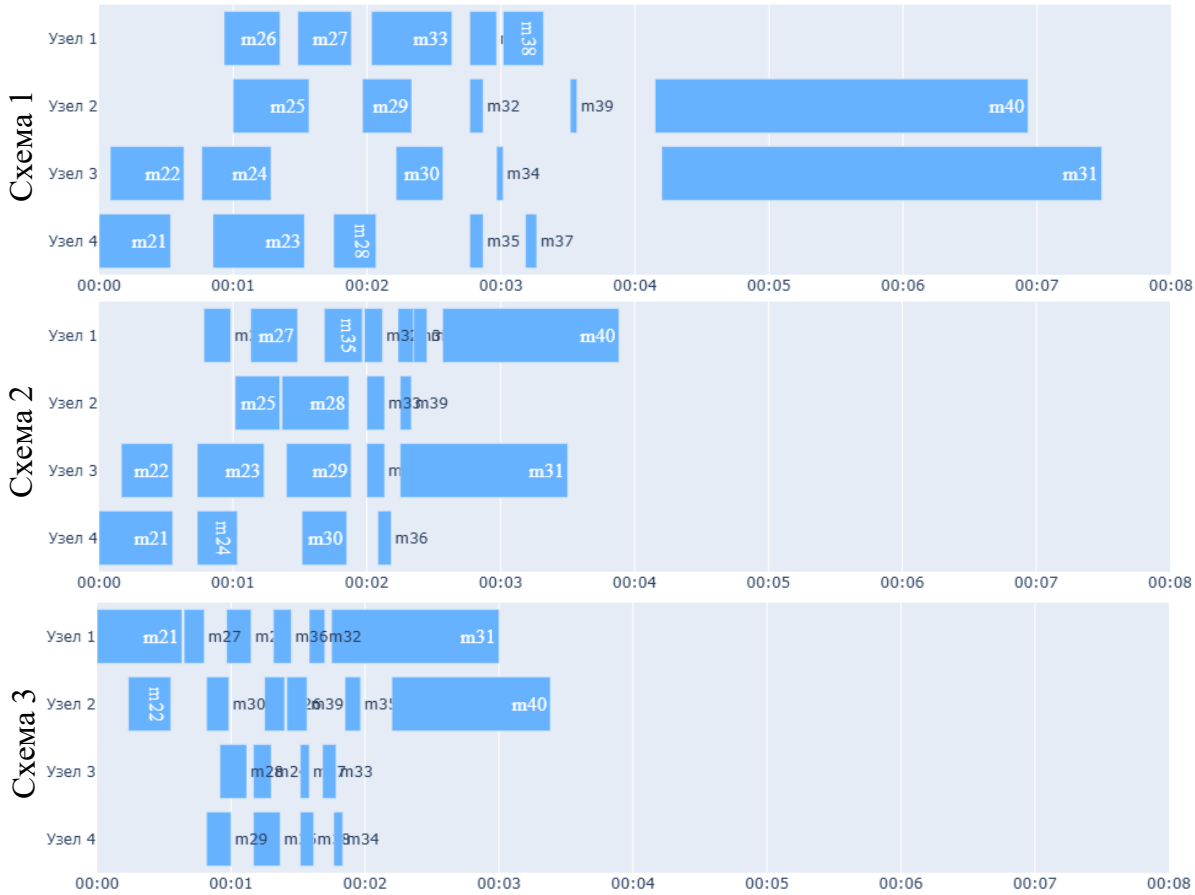


Рисунок 6: Расписания времени выполнения модулей CyberShake

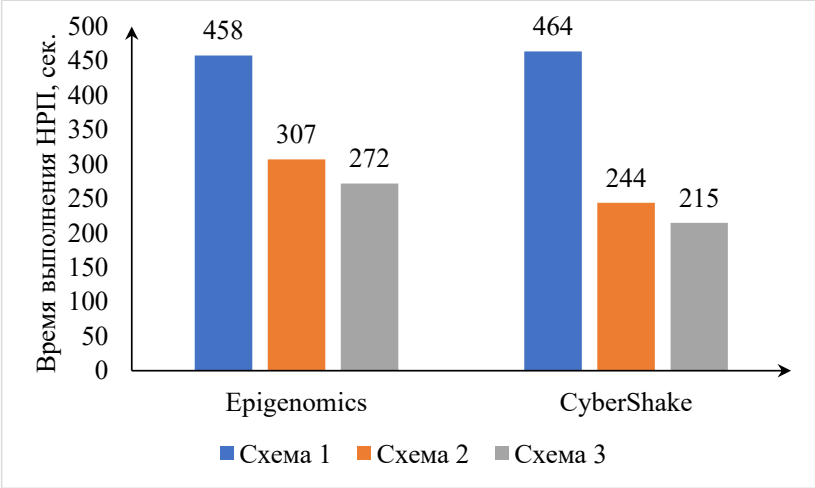


Рисунок 7: Диаграмма времени выполнения НРП

7. Заключение

Предложены новые средства управления для кластера, функционирующего под управлением DAGMan и HTCondor. Экспериментальные исследования показали, что данные средства обеспечивают существенное сокращение времени вычислений при выполнении НРП в сравнении с традиционной схемой управления, используемой в DAGMan и HTCondor. Это сокращение достигается благодаря учету структуры НРП и поддержке механизма прямой передачи данных между узлами кластера.

8. Благодарности

Исследование выполнено при поддержке Министерства науки и высшего образования Российской Федерации, проект № FWEW-2021-0005 «Технологии разработки и анализа предметно-ориентированных интеллектуальных систем группового управления в недетерминированных распределенных средах» (рег. № 121032400051-9).

9. References

[1] UNICORE. URL: https://www.unicore.eu.
[2] J. Goecks, A. Nekrutenko, J. Taylor et al. Galaxy: A comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences, Genome biology, 2010, vol. 11, pp. R86. doi:10.1186/gb-2010-11-8-r86.
[3] E. Deelman, G. Singh, M.H. Su et al. Pegasus: A Framework for Mapping Complex Scientific Workflows onto Distributed Systems, Scientific Programming, 2005, vol. 13, pp. 219–237. doi:10.1155/2005/128026.
[4] HTCondor. URL: https://htcondor.org.
[5] M. H. Hilman, M. A. Rodriguez, R. Buyya. Workflow-as-a-service cloud platform and deployment of bioinformatics workflow applications, Knowledge Management in the Development of Data-Intensive Systems, Auerbach Publications, 2021, pp. 205–226. doi:10.1201/9781003001188-14.
[6] P. Singh, Learn PySpark: Build Python-based Machine Learning and Deep Learning Models, Apress, 2019, pp. 67–84. doi:10.1007/978-1-4842-4961-1.
[7] B. Balis. HyperFlow: A model of computation, programming approach and enactment engine for complex distributed workflows, Future Generation Computer Systems, 2016, vol. 55, pp. 147–162. doi:10.1016/j.future.2015.08.015.
[8] A. Feoktistov, S. Gorsky, I. Sidorov et al. Collaborative Development and Use of Scientific Applications in Orlando Tools: Integration, Delivery, and Deployment, Communications in Computer and Information Science, 2020, vol. 1087, pp. 18–32. doi:10.1007/978-3-030-41005-6_2.
[9] А. Г. Феоктистов, М. Л. Воскобойников, А. Н. Черных, Разработка и применение сервис-ориентированных научных приложений в инструментальном комплексе FDE-SWFs, Труды Института системного программирования РАН, 2024, №36(6), с. 195–214. doi:10.15514/ISPRAS-2024-36(6)-11.
[10] М. Гэри, Д. Джонсон, Вычислительные машины и труднорешаемые задачи, М., Мир, 1982.
[11] G. Juve, A. Chervenak, E. Deelman, S. Bharathi, G. Mehta, and K. Vahi. Characterizing and profiling scientific workflows, Future Generation Computer Systems, 2013, vol. 29, no. 3, pp. 682–692. doi:10.1016/j.future.2012.08.015.

Интегрированная система сопровождения инвентаризации лесов

Иван Андреевич Диомидов [1,2]

[1] *Иркутский Государственный университет путей и сообщения, Россия, 664074, г. Иркутск, ул. Чернышевского, 15*
[2] *Филиал ФГБУ Рослесинфорг "Прибайкаллеспроект", Россия, 664040, г. Иркутск, ул. Розы Люксембург, 150*

Аннотация

В условиях растущей сложности информационных систем эффективное управление данными требует чёткого учёта их происхождения и взаимосвязей. Информация сегодня формируется как результат взаимодействия множества разрозненных источников и сервисов. В рамках данного исследования рассматривается применение модели направленного ациклического графа для описания и отслеживания процессов обработки и трансформации данных в задачах инвентаризации лесных ресурсов. Предложен подход к построению системы сопровождения инвентаризации лесов, основанный на учёте происхождения данных, что позволяет интегрировать существующие разрозненные базы данных и повысить прозрачность всех этапов работы с информацией. Такой подход не только облегчает контроль над качеством данных, но и создаёт основу для масштабируемой цифровой среды лесного хозяйства.

Ключевые слова

Системы учета происхождения данных, интеграция данных, инвентаризация лесов, информационные системы

1. Введение

Леса являются неотъемлемой частью природных экосистем, занимают значительную часть суши и играют ключевую роль в поддержании экологического баланса. Они выполняют важнейшие экономические, климатические и социальные функции, что делает задачу их устойчивого управления критически значимыми.

В рамках лесного хозяйства одной из ключевых функций является инвентаризация лесных ресурсов. Однако её реализация сталкивается с рядом устойчивых системных проблем. Наиболее остро стоят вопросы, связанные с информационным обеспечением: ведомственная фрагментация, задержка в обновлении данных, несогласованность форматов и слабая интеграция существующих программных решений [1,2]. Учитывая масштабы и разнообразие лесных экосистем, а также техническую сложность сбора первичных данных, актуальной задачей становится организация эффективной цифровой среды сопровождения процессов учёта и анализа.

Существующие государственные и коммерческие системы в отрасли, как правило, функционируют изолированно, не предусматривают обмена данными между собой и предъявляют жёсткие требования к структуре входных данных. Кроме того, мало внимания уделяется первичной обработке информации и контролю качества данных, что затрудняет автоматизацию процессов и снижает достоверность получаемых результатов.

7th International Workshop on Information, Computation, and Control Systems for Distributed Environments (ICCS-DE 2025), July 7–11, 2025, Irkutsk, Russia
EMAIL: pilad0hwtts@yandex.ru (A. 1)
© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).
ICCS-DE 2025 Workshop Proceedings
DOI: 10.47350/ICCS-DE.2025.10

Таким образом, проблема не сводится к необходимости создания новых методов инвентаризации, а заключается в обеспечении взаимодействия между существующими решениями и формировании единой цифровой среды сопровождения.

Цель настоящей работы показать, какая интегрированная система сопровождения инвентаризации лесов способна решить описанные проблемы, связанные с фрагментацией данных, различиях между форматами и низким уровнем интеграции существующих программных решений.

2. Система интеграции и учета происхождения данных

Ведомственные системы выступают в разных ролях в рамках единого процесса поддержании общего информационного поля. Без чётко заданной интеграционной схемы происходит искажение данных, потеря информации, дублирование или задержка — как следствие, нарушается логика коммуникации.

В работах [3,4] выполнялось моделирование и анализ процесса интеграции. В первом приближении задача интеграции заключается в сборе информации из разнородных источников, ее преобразовании и сохранении. Для формализации этого процесса используется следующая математическая запись:

$$\langle DS^L, Map(DS^L, DS^I), DS^I \rangle \tag{1}$$

где

- $DS^L = \{\langle D_i, \Sigma_i \rangle | i = 1, \dots, N\}$ — множество исходных наборов данных, Σ_i — схема данных для i-го входного набора, D_i — множество значений;
- $DS^I = \langle D_I, \Sigma_I \rangle$ — выходной набор интегрированных данных, где Σ_I — выходная схема, а D_I множество значений выходного набора;
- $Map(DS^L, DS^I)$ — отображение локальных входных данных в глобальный выходной набор.

Исторически ETL-подход применялся в системах бизнес-аналитики для заполнения хранилищ и витрин. Специализированные инструменты для работы с данными разнородной структуры и процесс включает сопоставление и интеграцию схем, трансформацию, разрешение и слияние сущностей [5-7].

В этом отношении ETL-подход может служить основой для формализации процесса интеграции данных. Он универсален для задач из разных предметных областей [8-10]. При наличии разнородных, но связанных между собой по смыслу наборов данных, возникает необходимость в механизме их объединения и централизованном хранении.

В лесной сфере это может быть представлен как цикл: сбор, передача, обработка и интерпретация информации [11]. Этот цикл можно распределить во времени, что приводит к формированию циклической модели с шагом n:

$$\begin{cases} \langle D^{(n)}_{\text{intermediate}}, \Sigma^{(n)}_{\text{intermediate}} \rangle = \text{Extract}\left(\langle D^{(n)}_{\text{source}}, \Sigma^{(n)}_{\text{source}} \rangle\right) \\ \langle D^{(n)}_{\text{transformed}}, \Sigma^{(n)}_{\text{transformed}} \rangle = \text{Transform}\left(\langle D^{(n)}_{\text{intermediate}}, \Sigma^{(n)}_{\text{intermediate}} \rangle\right) \\ \langle D^{(n)}_{\text{target}}, \Sigma^{(n)}_{\text{target}} \rangle = \text{Update}_{\text{load}}\left(\langle D^{(n-1)}_{\text{target}}, \Sigma^{(n-1)}_{\text{target}} \rangle, \langle D^{(n)}_{\text{transformed}}, \Sigma^{(n)}_{\text{transformed}} \rangle\right) \\ \langle D^{(n+1)}_{\text{source}}, \Sigma^{(n+1)}_{\text{source}} \rangle = \text{Update}_{\text{source}}\left(\langle D^{(n)}_{\text{source}}, \Sigma^{(n)}_{\text{source}} \rangle, \langle D^{(n)}_{\text{transformed}}, \Sigma^{(n)}_{\text{transformed}} \rangle\right) \end{cases} \tag{3}$$

С практической точки зрения, данная модель может быть использована для моделирования требований к программным алгоритмам и их реализации. Важным аспектом этой модели является отражение поэтапного внедрения системы интеграции и равномерности самого процесса интеграции данных (Рисунок 1).

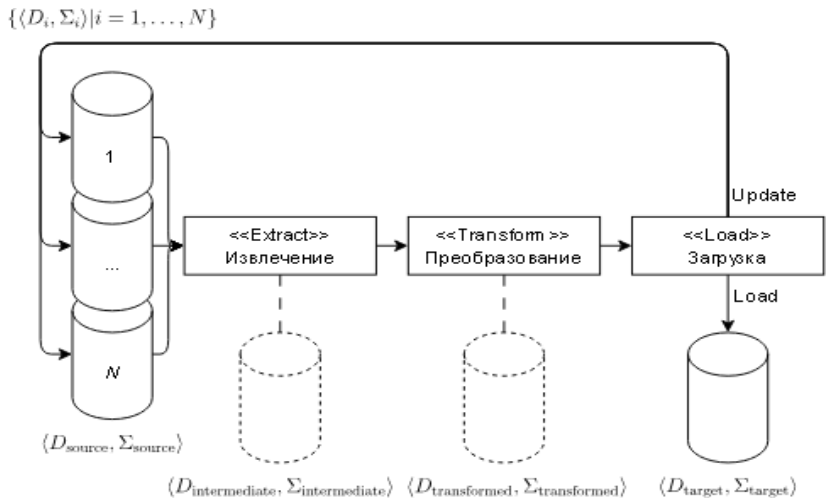


Рисунок 1: Схема интеграции ETL

Каждый шаг N-можно представить можно представить, как определенную версию данных на шаг N. В специфике такой предметной области, как лесное хозяйство, источники данных могут быть независимыми, развиваться асинхронно, использовать разные версии схем (в пределах самих себя же) одномоментно и локальную терминологию. Имеется десинхронизация по времени обновления, полноте и качеству данных. Сами данные могут быть сегментированы на основе организационных и территориальных факторов.

Версионирование данных можно проводить во времени. Их хранение можно организовать в такую структуру как направленный ацикличный граф. Он позволяет обозначать зависимости между данными, и позволяет строить на своей основе инструменты трассировки изменений в данных (ведение последовательности преобразований вперед и назад), сборки данных (интеграции, компиляции) и распределения данных по узлам хранения:

$$\begin{gathered} Branch = \{b_1, b_2, \ldots, b_{bn}\} \\ b_{bi} \in Branch \\ Relation_{b_{bi}} = \{r_{i,j}^{(b_{bi},t)\to(b_{bi},t+1)} \mid bi, i, j \in I; t \in \mathbb{N}\} \\ r_{i,j}^{(b_{bi},t)\to(b_{bi},t+1)} : \langle D_{i,j}^{(b_{bi},t)\to(b_{bi},t+1)}, \Sigma_{i,j}^{(b_{bi},t)\to(b_{bi},t+1)} \rangle \\ Relation = \bigcup_{b \in B} Relation_b \end{gathered} \tag{4}$$

где r - сущность содержащая информацию о преобразовании набора i в j.

Таким образом, система хранения ориентирована не на хранение данных в конкретной схеме, а хранение и организации информационных потоков в ветви, содержащие историю изменений. Вершинами графа является отношение между двумя наборами данных $\langle D_{i,j}^{(b_{bi},t)\to(b_{bi},t+1)}, \Sigma_{i,j}^{(b_{bi},t)\to(b_{bi},t+1)} \rangle$, а ребрами тип операций P над отношением $r_{i,j}^{(b_{bi},t)\to(b_{bi},t+1)}$:

$$P = \left\{ \begin{array}{ll} (b,t) \to (b, t+1) & \text{— Commit} \\ (b,t) \to (b_{\text{new}}, 0) & \text{— Create} \\ (b_1, t_1), (b_2, t_2) \to (b_3, t_3) & \text{— Merge} \\ \varnothing \ \to \ (b_{\text{new}}, 0) & \text{— Stage} \end{array} \right\} \tag{5}$$

Модель управления, описанная в данном параграфе, вдохновлена как системами учета происхождением данных, так и системами контроля версий исходного кода. Хоть и сама идея систем контроля версий исходного кода имеет довольно долгую историю, именно распределенная система управления версиями Git в настоящий момент является практически технологическим стандартом [12-14].

Предложенная автором адаптация отодвигает вопрос идентификации изменений на второй план и основной своей задачей рассмотреть процесс управления учета данными. Цель - управлением абстрактными отношениями между наборами данных. Главным качеством данного отношения является его прямая связь с процессом изменений в данных, проходящим естественным образом в уже происходящих производственных процессах. Данное отношение

может не иметь семантической связи (или не иметь четко сформулированной связи), но может служить указанием на то, что данная связь присутствует.

Так же, предложенная модель управления происхождением данных, имеет схожесть с организацией процесса интеграцией в терминах ETL. Графовая модель изменений, лежащая в основе данной модели, позволяет как выполнять декомпозицию задач, так и организовывать управление самим процессом изменения. Тогда процесс можно организовать в три этапа:

1. Предоставление необходимого сегмента информации для редактирования из некоторой среды (пользовательской или программной) ($Stage$ -> $Extract$);
2. Ожидание выполнения средой всех необходимых изменений над данными ($Commit$ -> $Transform$);
3. Выполнение операции слияния ($Merge$ -> $Load$) с системной ветвью.

3. Разработанная интегрированная система

На основе разработанной модели была создана система интегрированной обработки информации о лесах и лесных ресурсах – Лес-Sense. Целью системы является управление процессами сбора, обработки и интеграции данных о лесном фонде, обновляемых и редактируемых параллельно.

Интегрированную система включает следующие подсистемы:

Подсистему обработки лесоустроительной информации;

4. Подсистему управления базами данных;
5. Подсистему управления версиями данных;
6. Подсистему визуализации информации.

Наиболее важным компонентом системы является подсистема управления версиями. Она была разработана в рамках конкретных производственных задач как независимая, но интегрируемая система. Логически информация в ней разделена на модель сущности, модель версий и модель артефактов (Рисунок 3.3). Каждая версия идентифицируется хэш-суммой своего содержимого (артефактов), имеет указание на автора изменений (имя пользователя, рабочей группы и компьютера), время внесение изменений

Каждая ветвь дерева изменений представляет собой набор “слепков” рабочего пространства оператора на определенный момент времени. Это может быть результат проектирования или работы специалистов ГИС. Рабочее пространство может быть разделено и храниться на разных компьютерах, при этом сохраняя идентичный набор веток (Рисунок 2). Такой подход позволяет вести историю изменений по каждому из участков на разных компьютерах.

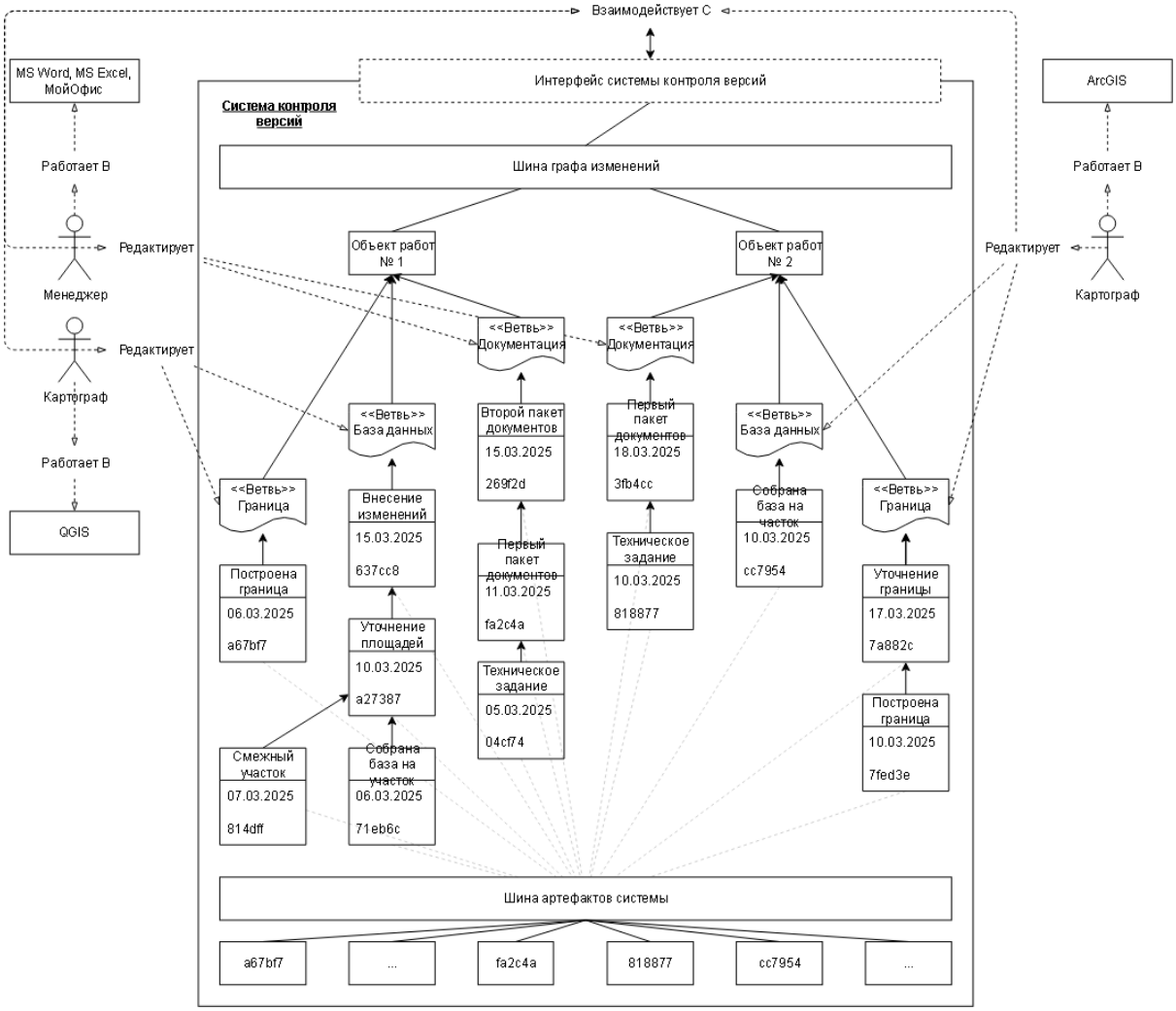


Рисунок 2: Сценарий использования системы Лес-Sense

Алгоритм сценарий использования системы следующий:

Оператор выполняет staging операцию для своего рабочего пространства.

7. Система выполняет операцию pull tree и сравнивает локальный граф изменений пользователя с системным.
8. Если все корректно, оператору оператору предлагается предоставить информацию (текстовое сообщение) о сущности вносимых изменений, можно оставить пустым.
9. После этого система выполняет операции commit и push.
10. Если графы не совпадают, ветви пользователя принимаются как поврежденные или скомпрометированные и заменяются системными.
11. Оператору предоставляется возможность принять решение, продолжить внесение изменений или провести анализ данных на предмет

Таким образом выполняется синхронизация лишь графа изменений с системным репозиторием, а сами данные можно загрузить только при необходимости. Для этого используется механизм синхронизации репозиториев. Он обновляет системную базы с версиями, хранящихся у исполнителей. Системное хранилище в этом случае является базой данных всех изменений по исполнителям. Несмотря на то, что сами репозитории могут храниться изолированно друг от друга, но функционируют они в рамках логики единой системы контроля версий. Это позволяет данные объединять или наоборот сегментировать. Таким образом, оператор может увидеть результат своей работы в подсистеме визуализации. Данная подсистема реализована на базе Web-ГИС. (Рисунок 3)

Рисунок 3: Интерфейс системы Лес-Sense

Важной проблемой при работе с данными инвентаризации лесов, является проверка их достоверности. Это может включать в себя следующие виды работ: таксация лесов, проектирование лесничеств, проектирование целевого назначения и особо защитных участков (ЦН и ОЗУ), научно-исследовательские работы.

Большой интерес при анализе представляет возможность рассмотрения всех видов артефактов или типов ветвей, которые есть в системе. Для анализа в этом ключе требуется рассмотреть возможность создания наиболее общего хранилища, структура которого бы определялась из существующих в системе данных.

Система учета происхождения позволяет преобразовать разнородную информацию о лесах и лесных ресурсах в унифицированный формат – квартальную сеть. Поскольку данная информация основана на реальных данных, целесообразно рассмотреть возможность ее сопоставления для оценки достоверности и выявления статистических закономерностей.

Рисунок 4: Граф изменений и виртуальный граф изменений поквартально

На основе анализа связанной литературы касающегося анализа как лесной растительности, анализа климатических измерений, можно выделить следующий типовой алгоритм оценки достоверности данных лесного учета применительно к возможностям разработанной системы [15-19]:

Выбор эталонных участков: Определяются участки с достоверной информацией, подтвержденной экспертным путем. В качестве эталонных могут использоваться участки с заранее заложенными пробными площадями.

12. Моделирование распределения: На основе данных эталонных участков строится статистическая модель распределения. Возможны как параметрические, так и непараметрические подходы. В рамках параметрического подхода можно применить законы распределения (например, нормальное, Пуассона), подбирая параметры на основе данных эталонного участка. Непараметрический подход может включать использование методов оценки плотности вероятности (KDE).

13. Выбор участка для анализа: Выбирается участок, данные которого нуждаются в оценке достоверности. Данный участок может содержать аномалии или недостоверные данные.
14. Вычисление плотности распределения: Используя построенную на шаге 2 модель, вычисляется плотность распределения для данных выбранного участка.
15. Построение таблицы правдоподобия: Для каждого набора данных на выбранном участке строится таблица правдоподобия, отражающая вероятность принадлежности этих данных к модели, построенной на основе эталонных участков.
16. Оценка хозяйственной важности и определение рисков: Определяется хозяйственная важность квартала (защитные леса, леса с редкими породами) и оцениваются риски, связанные с недостоверными данными лесного учета.
17. Построение матрицы принятия решений: На основе таблицы правдоподобия и оценки рисков строится матрица принятия решений, которая позволит выбрать оптимальный вариант действий в зависимости от уровня достоверности данных.

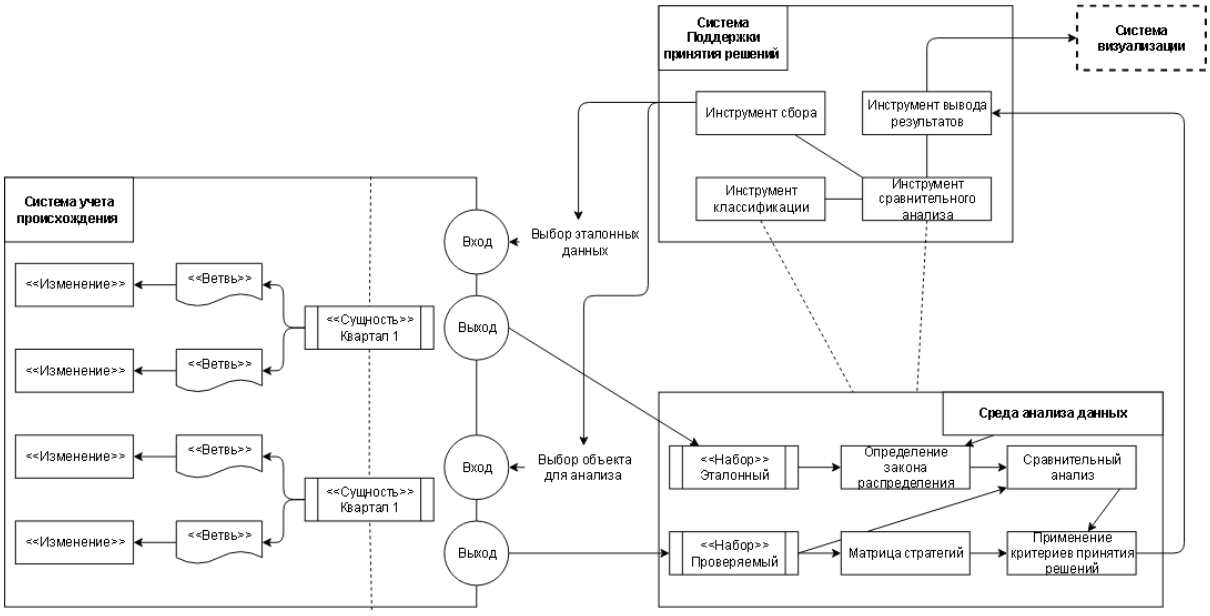


Рисунок 5: Схема работы системы контроля качества данных

4. Заключение.

В данной работе была рассмотрена задача создания интегрированной системы сопровождения инвентаризации лесов. В качестве основы системы учета происхождения данных была предложена модель направленного ацикличного графа. Конкретным практическим решением стал разработанный прототип геоинформационной системы Лес-Sense. Система использует граф изменений как для анализа хода выполнения работ, так и для реализации механизма синхронизации изменений между исполнителями. Было также показано, как данную систему можно расширить для реализации механизма контроля качества данных. Дальнейшие исследования могут быть связаны с более детальным изучением видов данных, их организации в ветвях графа и выявлением проблем, возникающих при их слиянии.

5. Список литературы

[1] В. В. Михелёв, “Системно-объектный подход к системному анализу: Особенности и преимущества,” ЭИ, vol. 49, no. 1, pp. 145–152, Mar. 2022, doi: 10.52575/2687-0932-2022-49-1-145-152.

[2] Н. П. Путивцева, Т. В. Зайцева, and Л. А. Губкина, “К вопросу о выборе метода системного анализа,” Теория и практика современной науки, vol. 8, no. 38, pp. 139–143, 2018.

[3] V. Vysotska et al., “Formal Data Integration Models Development for Intelligent Electronic Commerce Systems,” presented at the 8th International Conference on Computational Linguistics and Intelligent Systems, Lviv, Ukraine, Apr. 2024.
[4] A. Simitsis and P. Vassiliadis, “A Methodology for the Conceptual Modeling of ETL Processes,” in A Methodology for the Conceptual Modeling of ETL Processes, Jan. 2003, pp. 305–316.
[5] А. Е. Вовченко, Л. А. Калиниченко, and Д. Ю. Ковалев, “Методы разрешения сущностей и слияния данных в ETL-процессе и их реализация в среде HADOOP,” Информатика и ее применения, vol. 8, no. 4, pp. 94–109, Nov. 2014, doi: 10.14357/19922264140412.
[6] P. Christen, Data matching: Concepts and techniques for record linkage, entity resolution, and duplicate detection. in Data-centric systems and applications. Berlin ; New York: Springer, 2012.
[7] J. Bleiholder and F. Naumann, “Data fusion,” ACM Comput. Surv., vol. 41, no. 1, pp. 1–41, Jan. 2009, doi: 10.1145/1456650.1456651.
[8] M. Pivovar, “Automated Data Transformation in the Execution of ETL Operations in the Aviation Industry,” in Automated Data Transformation in the Execution of ETL Operations in the Aviation Industry, 2024.
[9] M. Pivovar, B. Misnevs, and I. Pticina, “Methodology for Calculating ETL Indicators in the Process of Implementation of Aircraft Maintenance Information Systems,” in Reliability and Statistics in Transportation and Communication, vol. 117, I. Kabashkin, I. Yatskiv, and O. Prentkovskis, Eds., Cham: Springer International Publishing, 2020, pp. 44–53. doi: 10.1007/978-3-030-44610-9_5.
[10] M. Pivovar, L. Rozhkova, B. Misnevs, and I. Pticina, “Data Quality Indicators of ETL in the Process of Implementation of Information Systems for Aircraft Maintenance and Operation,” in Reliability and Statistics in Transportation and Communication, vol. 195, I. Kabashkin, I. Yatskiv, and O. Prentkovskis, Eds., Cham: Springer International Publishing, 2021, pp. 130–139. doi: 10.1007/978-3-030-68476-1_12.
[11] М. Р. Вагизов, Е. П. Истомин, Н. В. Яготинцева, О. Н. Колбина, А. Е. Морщихина, and К. В. Конжоголадзе, “Разработка интеллектуальной геоинформационной системы для отрасли лесного хозяйства,” Geoinformatika, no. 3, pp. 4–13, Oct. 2021, doi: 10.47148/1609-364X-2021-3-4-13.
[12] В. С. Новиков and М. В. Паринов, “Применение систем управления версиями при разработке информационной системы радиационного контроля и безопасности,” in Материалы XVIII Международной научно-методической конференции, Воронеж, 8-9 февраля 2018.
[13] С. В. Картамышев, “Системы управления репозиториями проектов на основе распределенной схемы контроля версий,” presented at the Международная научно-техническая конференция молодых ученых БГТУ им. В.Г. Шухова, 01-30 мая 2015, pp. 2856–2860.
[14] N. M. Zolkifli, A. Ngah, and A. Deraman, “Version Control System: A Review,” Procedia Computer Science, vol. 135, pp. 408–415, 2018, doi: 10.1016/j.procs.2018.08.191.
[15] C. Mohamedou, A. Kangas, A. Hamedianfar, and J. Vauhkonen, “Potential of Bayesian formalism for the fusion and assimilation of sequential forestry data in time and space,” Can. J. For. Res., vol. 52, no. 4, pp. 439–449, Apr. 2022, doi: 10.1139/cjfr-2021-0145.
[16] A. El Hachem, J. Seidel, F. Imbery, T. Junghänel, and A. Bárdossy, “Technical Note: Space–time statistical quality control of extreme precipitation observations,” Hydrol. Earth Syst. Sci., vol. 26, no. 23, pp. 6137–6146, Dec. 2022, doi: 10.5194/hess-26-6137-2022.
[17] H. Y. Teh, A. W. Kempa-Liehr, and K. I.-K. Wang, “Sensor data quality: A systematic review,” J Big Data, vol. 7, no. 1, p. 11, Dec. 2020, doi: 10.1186/s40537-020-0285-1.
[18] K. A. Sudduth and S. T. Drummond, “Yield Editor: Software for Removing Errors from Crop Yield Maps,” Agronomy Journal, vol. 99, no. 6, pp. 1471–1482, Nov. 2007, doi: 10.2134/agronj2006.0326.
[19] А. Н. Греков, Е. В. Вышкваркова, and В. В. Трусевич, “Применение метода ядерной оценки плотности для обнаружения аномалий в данных активности двустворчатых моллюсков,” vol. 58, no. 4, pp. 135–144, 2024, doi: 10.33075/2220-5861-2024-4-135-144.

Проблемно-ориентированная среда для создания систем поддержки принятия решений на основе таблиц решений

Никита Дородных[1] и Александр Юрин[1]

[1] *Институт динамики систем и теории управления имени В.М. Матросова СО РАН, ул. Лермонтова, 134, Иркутск, 664033, Россия*

Аннотация

В работе описывается проблемно-ориентированная среда типа программа-оболочка – iDSS.Desktop, обеспечивающая возможность создания интеллектуальных систем поддержки принятия решений диагностического типа. Для формирования решений и хранения знаний используются формализмы логических правил и прецедентов, представленные в форме таблиц решений. Приведено описание программной системы, включая: основные функции; архитектуру; структуру используемых логических правил; способы формирования баз знаний. Представлен иллюстративный пример использования iDSS.Desktop при создании прототипа приложения АвиаТехПом.Терминал для поиска и устранения неисправностей гражданских воздушных судов на примере Cessna 182T.

Ключевые слова

Программа-оболочка, интеллектуальная система, диагностирование авиационных систем и комплексов, таблица решений, логические правила, iDSS.Desktop, АвиаТехПом.Терминал, Cessna

1. Введение

Одним из факторов, влияющих на эффективность решения задач технической диагностики при помощи программных систем, является степень вовлеченности специалистов-предметников в процесс их создания или настройки (адаптации) под условия конкретной проблемы. В качестве способа повышения вовлеченности может быть рассмотрено создание программных решений в соответствии с идеями направления, известного как End-User Development (EUD) [1-2], и его разновидностей: программирования для конечных пользователей (End-User Programming) и разработки (включая моделирование и проектирование) для конечных пользователей (End-User Software Engineering). EUD предлагает следующие основные способы реализации идей в рамках автоматизации разработки интеллектуальных систем:

- создание инструментальных средств компонентной сборки, в том числе на основе облачных сервисов и их композиций [3-4];
- создание специализированных и проблемно/предметно-ориентированных языков для описания логики интеллектуальных систем [5-6];
- создание средств программирования, в том числе основанных на принципах интеллектуального интерфейса, шаблонах, мастерах и модельных трансформациях [6-8];
- создание шаблонов проектирования и онтологий [5, 9];

7th International Workshop on Information, Computation, and Control Systems for Distributed Environments (ICCS-DE 2025), July 7–11, 2025, Irkutsk, Russia
EMAIL: nikidorny@icc.ru (A. 1); iskander@icc.ru (A. 2)
ORCID: 0000-0001-7794-4462 (A. 1); 0000-0001-9089-5730 (A. 2)
© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).
ICCS-DE 2025 Workshop Proceedings
DOI: 10.47350/ICCS-DE.2025.11

- создание методов, обеспечивающих использование различных источников информации (электронных таблиц, текстов, баз данных, концептуальных моделей) в контексте парадигмы больших данных [7, 9, 10].

Реализуя ряд направлений EUD, авторами была предложена технология [6, 11] построения прототипов баз знаний интеллектуальных систем на основе модельных трансформаций концептуальных (информационных) моделей. В данной работе представлено программное инструментальное средство технологии – проблемно-ориентированная среда iDSS.Desktop типа программа-оболочка для построения настольных интеллектуальных систем поддержки принятия решений диагностического типа [12], основанная на способе моделирования и использования предметных знаний в форме таблиц решений. Выбор формы таблиц решений [13] обусловлен их наглядностью, а при наличии соответствующего инструментария (например, «СИМПР-2015» [14], «ПРОБАЗ» [15] и др.) и простотой использования, что выгодно их отличает от визуальных предметно/проблемно-ориентированных языков. Также необходимо отметить перспективность использования таблиц решений для построения баз знаний интеллектуальных ассистентов и чат-ботов [16]. Таким образом, основное назначение iDSS.Desktop – создание интеллектуальных систем, реализующих поиск решения на основе логических правил (продукций), которые, в свою очередь, описывают взаимосвязи между внешними проявлениями неисправностей и причинами их возникновения, причинами и мероприятиями (операциями, работами) по их локализации, а также прецеденты, фиксирующие подобные неисправности в прошлом.

2. Проблемно-ориентированная среда

2.1. Используемая специализация таблиц решений

Таблицы решений относятся к известным способам представления предметных знаний, который, в том числе, пользуется популярностью у специалистов-предметников благодаря своей простоте и наглядности. При этом применяется как их классическая форма [13, 17], так и ее модификации, например, eXtended Tabular Trees (XTT2) [18]. В данной работе используется модификация из [19], обеспечивающая:

- выделение зависимых столбцов символом «#»;
- возможность задания одноуровневой иерархии в условиях «плоских» CSV-таблиц путем указания составных имен столбцов, включающих имя сущности (или имя класса сущностей) и наименование ее свойства, с помощью разделителя «::»;
- использование произвольных значений в качестве значений ячеек, а не только указывающих на наличие или отсутствие определенного свойства (компонента) в структуре правила (да/нет или 0/1).

2.2. Основные функции, архитектура и особенности реализации

Проблемно-ориентированная среда реализует следующие основные функции:

- Ввод информации в таблицы решений определенной структуры, в частности о:
    a. проявлениях неисправностей и возможных отказавших системах,
    b. проявлениях неисправностей и мероприятиях (операциях);
    c. произошедших ранее отказах (опыт эксплуатации).
- Поиск решения на основе логических правил.
- Поиск решения по прецедентам.
- Формирование и контроль плана устранения неисправности (отказа).
- Формирование специализированных отчетов.

Архитектура, обеспечивающая реализацию указанных функций, включает следующие модули:

- подсистему управления данными, реализующую взаимодействие с таблицами решений (доступ к ним в процессе их интерпретации и поиска решения) и конфигурационными файлами;

- подсистему генерации графического интерфейса пользователя, формирующую отдельные элементы управления для ввода информации о внешних проявлениях неисправности (отказа), а также отображения результатов;
- подсистему поддержки поиска и устранения неисправностей, включая модули для поиска решения на основе логических правил и прецедентов;
- редактор таблиц решений;
- модуль генерации отчетов по работам.

Программная реализация осуществлена в среде Lazarus, для хранения данных и знаний в форме таблиц решений используется формат CSV.

2.3. Структура таблиц решений и способы их формирования

iDSS.Desktop предполагает использование таблиц решений, описывающих:

- зависимости между внешними проявлениями неисправности (отказа) и определенными системами – правила типа «проявление-система»;
- последовательности операций (работ) по устранению неисправностей (отказов) – правила типа «работа-работа»;
- зафиксированные ранее неисправности (прецеденты).

При этом среда обеспечивает интерпретацию таблиц с произвольным набором полей, удовлетворяющих следующим ограничениям:

- обязательное наличие полей с признаком-символом «#», обозначающим поля (свойства), формирующие консеквент (действие) логических правил и часть «решение» прецедентов;
- совпадение хотя бы части полей антецедента (условия) логических правил и части «описание проблемы» прецедента, что необходимо для поиска решения по одинаковому набору внешних проявлений или диагностических признаков, формирующих контекст;
- обязательное наличие полей: «Работа::Наименование»/«Наименование», «Работа::Результат»/«Результат», «#Работа::Наименование»/«#Наименование» в структуре правил «работа-работа», состав остальных полей, образующих контекст применения правила, вариативен.

Для наполнения таблиц решений данными и знаниями использовались следующие способы:

1. Использование табличного редактора Microsoft Excel, способ характеризуется большим количеством ошибок, обусловленных ручным вводом данных.
2. Использование редактора продукционных баз знаний Personal Knowledge Base Designer (PKBD) [20], поддерживающего принципы модельно-ориентированной разработки. Способ обладает большей трудоемкостью в части формализации разветвленных сценариев операций.
3. Использование встроенного редактора таблиц решений. Благодаря специализированным функциям (коррекция данных, проверка целостности, копирование, ввод данных для продолжения цепочки правил, отображение цепочки правил, отображение следующего правила) способ характеризуется меньшим количеством ошибок.
4. Использование веб-ориентированной программной системы Knowledge Modeling System (KMS) [21], а именно, редактора диаграмм переходов состояний.

На основе опыта применения разных способов был сделан вывод, что использование KMS перспективно при необходимости моделирования сценариев (цепочек правил), описывающих разветвленные последовательности операций со сложными условиями переходов.

3. Иллюстративный пример применения

Одним из примеров применения и апробации инструментальной среды является выполнение работ по проекту Фонда поддержки инноваций (заявка С1ИИ-301071) и компании АЛЬТАИР-ИИ цель которого – разработка прототипа программы «АвиаТехПом» [22] для обучения и интеллектуальной поддержки принятия оперативных решений техником при

диагностировании и обслуживании воздушного судна. С помощью iDSS.Desktop был создан настольный вариант системы АвиаТехПом – АвиаТехПом.Терминал. При этом рассматривались несколько целевых типов воздушных судов: Сухой Суперджет (RRJ-95) и Cessna 182T.

В качестве исходных данных использовались технические руководства по эксплуатации и поиску и устранению неисправностей [23, 24]. Для системы кондиционирования воздуха Cessna 182T разработанный сегмент базы знаний содержал 23 записи в таблице решений «проявление-система» и 52 – «работа-работа». При решении подобной задачи для Сухой Суперджет для систем кондиционирования воздуха и электроснабжения сегменты составили 178, 687 и 173, 828 записей соответственно.

На рисунке 1 приведены примеры экранных форм среды/прототипа приложения, а именно: (1) основной экран с введенными внешними проявлениями неисправности (исходные данные) и результатами поиска причин неисправности на основе логических правил (согласно документации) в форме возможных причин и перечня мероприятий; (2) встроенный редактор таблиц решений; (3) цепочка выполнения мероприятий (операций).

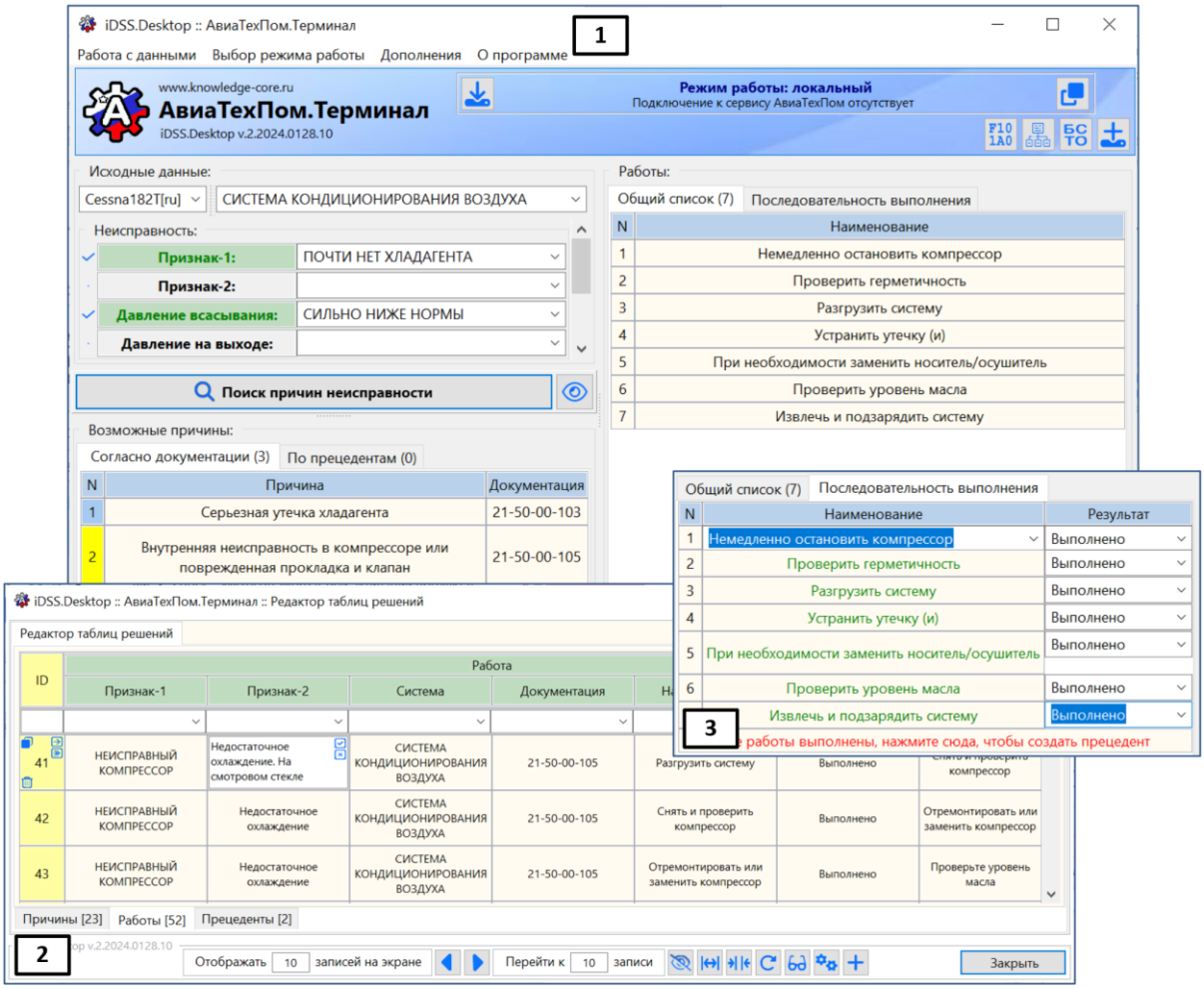


Рисунок 1: Экранные формы среды/прототипа приложения

4. Заключение

В работе рассматривается проблемно-ориентированная инструментальная среда iDSS.Desktop типа «оболочка», предназначенная для построения настольных интеллектуальных диагностических систем поддержки принятия решений. Среда поддерживает принципы EUD, реализуемые через интерпретацию таблиц решений и генерацию элементов интерфейса пользователя. Для редактирования таблиц решений могут быть использованы как табличные редакторы, в частности, Microsoft Excel и LibreOffice Calc, а

также авторский инструментарий: PKBD, KMS и встроенный редактор. Для поиска решений используются механизмы логического вывода на основе продукций и прецедентов. Приведен один из примеров апробации в рамках проекта разработка программы «АвиаТехПом» [22], в рамках которого программа показала свою перспективность.

5. Благодарности

Работа выполнена в рамках госзадания Минобрнауки России по проекту «Методы и технологии облачной сервис-ориентированной цифровой платформы сбора, хранения и обработки больших объёмов разноформатных междисциплинарных данных и знаний, основанные на применении искусственного интеллекта, модельно-управляемого подхода и машинного обучения» (№ гос. регистрации: 121030500071-2).

6. Список литературы

[1] B.R. Barricelli, F. Cassano, D. Fogli, A. Piccinno, End-user development, end-user programming and end-user software engineering: A systematic mapping study, Journal of Systems and Software 149 (2019) 101–137.

[2] E. Coronado, F. Mastrogiovanni, B. Indurkhya, G. Venture, Visual Programming Environments for End-User Development of Intelligent and Social Robots, a Systematic Review, Journal of Computer Languages 58 (2020) 100970.

[3] И.В. Бычков, А.Г. Феоктистов, С.А. Горский, Р.О. Костромин, Р.К. Федоров, Автоматизация интеграции сервисов веб-обработки данных экологического мониторинга с распределенными научными приложениями, Автометрия 58(4) (2022) 67–75.

[4] О.А. Николайчук, А.И. Павлов, Сервис автоматизации создания интеллектуальных систем на платформе Яндекс.Облако, Материалы VI Международного семинара по информационным, вычислительным и управляющим системам для распределенных сред (ICCS-DE 2024), Иркутск, 2024, 153–159.

[5] В.В. Голенков, Н.А. Гулякина, Проект открытой семантической технологии компонентного проектирования интеллектуальных систем. Часть 1: принципы создания, Онтология проектирования 1 (2014) 42–64.

[6] А.Ф. Берман, Н.О. Дородных, О.А. Николайчук, А.Ю. Юрин, Технология создания основанных на знаниях систем для исследования динамики состояния технических объектов, Вычислительные технологии 28(4) (2023) 94–108.

[7] А.А. Коробко, А.В. Коробко, Оригинальный подход к построению модельно-ориентированной системы консолидации данных, Информатизация и связь 4 (2017) 232–238.

[8] B. Ruiz-Mezcua, A. Garcia-Crespo, J. Lopez-Cuadrado, I. Gonzalez-Carrasco, An expert system development tool for non AI experts, Expert Systems with Applications 38 (2011) 597–609.

[9] Yu.A. Zagorulko, Development of a method for automatic extraction of ontology entity names from natural language texts, Mechanics and Technology 3(85) (2024) 486–494.

[10] A.O. Shigarov, A.Yu. Yurin, N.O. Dorodnykh, Semi-Automated Formalization and Representation of the Engineering Knowledge Extracted From Spreadsheet Data, IEEE Access 9 (2021) 157468-157481.

[11] М.А. Грищенко, Н.О. Дородных, О.А. Николайчук, А.Ю. Юрин, Применение модельно-управляемого подхода для создания продукционных экспертных систем и баз знаний, Искусственный интеллект и принятие решений 2 (2016) 16–29.

[12] А.Ю. Юрин, Н.О. Дородных, iDSS.Desktop, Свидетельство о государственной регистрации программы для ЭВМ, М, Рег. № 2022617130 от 19.04.22.

[13] А.А.Башлыков, А.П. Еремеев, Основы конструирования интеллектуальных систем поддержки принятия решений в атомной энергетике, М.: Инфра-М, 2018.

[14] А.П. Еремеев, Н.В. Чибизова Инструментальный программный комплекс СИМПР-2015 и его применение при обучении студентов по направлению «Прикладная математика и

информатика», Труды Международной научно-практической конференции ИНФОРИНО-2016, 125–130.
[15] В.А. Игошин, М.В. Игошин Инструментальные средства создания баз знаний, экспертных систем и интеллектуальных систем поддержки принятия решений, В сборнике: Искусственный интеллект в решении актуальных социальных и экономических проблем XXI века, 2018, 291–293.
[16] A.Yu. Yurin, O.A. Nikolaychuk, N.O. Dorodnykh, A.B. Stolbov, D.A. Denisova, Using an Intelligent Assistant for Aircraft Diagnostics and Maintenance, Lecture Notes in Networks and Systems 776 (2023) 325–333.
[17] А.П. Еремеев, Развитие аппарата таблиц решений в контексте создания гибридных интеллектуальных систем, Гибридные и синергетические интеллектуальные системы. Материалы III Всероссийской Поспеловской конференции с международным участием, 2016, 121–132.
[18] G.J. Nalepa, K. Kluza, UML representation for rule-based application models with XTT2-based business rules, International Journal of Software Engineering and Knowledge Engineering 22(4) (2012) 485–524.
[19] А.Ю. Юрин, Применение трансформаций таблиц решений при создании интеллектуального программного модуля «Детектор» для веб-приложений, Программные продукты и системы 4 (2020) 573–581.
[20] A.Yu. Yurin, N.O. Dorodnykh, Personal knowledge base designer: Software for expert systems prototyping, SoftwareX 11 (2020) 100411.
[21] Н.О. Дородных, О.А. Николайчук, А.Ю. Юрин, Система визуального моделирования знаний Knowledge Modeling System, Материалы VI Международного семинара по информационным, вычислительным и управляющим системам для распределенных сред (ICCS-DE 2024), Иркутск, 2024, 97–102.
[22] Ю.В. Котлов, А.Ю. Юрин, АвиаТехПом: Состояние и перспективы, Crede Experto: транспорт, общество, образование, язык 1 (2024) 146–156.
[23] AirComm Corporation Cessna 182T Maintenance Manual. URL: https://www.usermanuals.tech/d/aircomm-corporation-cessna-182t-t182t-maintenance-manual.
[24] Cessna 182 ServiceManual. URL: https://www.live2fly.ru/_files/ugd/74aa7b_a4e02584e5704be39c4bc60ca0effda6.pdf.

Методика формирования исходных данных для алгоритмов машинного обучения при идентификации элементов ландшафта

Виктор Мочалов,[1] Ольга Григорьева[1]

[1]*Военно-космическая академия имени А.Ф.Можайского, г. Санкт-Петербург, 197198, Россия*

Аннотация

Представлен типовой состав исходных данных для алгоритмов машинного обучения в части обучающих и контрольных выборок. Предложена методика обоснованного формирования исходных данных и пример её практической реализации. В качестве источника исходных данных приняты материалы отечественной мультиспектральной космической съёмки. Приведен пример автоматизированной идентификации элементов ландшафта и числовые значения показателей качества результатов обработки.

Ключевые слова

Мультиспектральная космическая съёмка, обучающая и контрольная выборки, семантическая сегментация, качество результатов обработки, цифровая карта

1. Введение

Развитие методов автоматизированной обработки данных дистанционного зондирования Земли является одним из основных направлений совершенствования новых информационных технологий, обеспечивающих оперативное решение актуальных практических задач. Часто возникает необходимость идентификации или семантической сегментации элементов ландшафта с последующим представлением результатов работ в виде цифровых топографических карт местности. Автоматизированная обработка материалов космической съёмки осуществляется, как правило, на основе алгоритмов машинного обучения, в том числе с использованием нейросетевых технологий. Успешная эксплуатация алгоритмов во многом зависит от исходных данных (ИД), в состав которых входят, в частности, обучающие и контрольные выборки. Вместе с тем ИД часто формируются из непроверенных источников. Кроме того, иногда считается, что расширение объёма ИД может привести к повышению качества результатов обработки материалов съёмки. Также не всегда осуществляется контроль достоверности данных, входящих как в обучающие, так и в контрольные выборки. Показатели качества результатов эксплуатации нейронных сетей часто определяются автоматически на основе заложенных в программные модули алгоритмов расчета обобщенных показателей. В данной работе предложена методика, опирающаяся как на проведённые теоретические исследования, так и на опыт решения практических задач обработки материалов мультиспектральной аэрокосмической съёмки. Представлена структура и основные этапы применения методики, апробированной при идентификации состояния лесных массивов. Обрабатывались материалы отечественной космической съёмки, выполненной с помощью космического аппарата «Канопус-В». Обеспечено формирование ИД, которые использовались для семантической сегментации элементов ландшафта. Результаты семантической сегментации представлены в виде фрагмента цифровой топографической карты.

7th International Workshop on Information, Computation, and Control Systems for Distributed Environments (ICCS-DE 2025), July 7–11, 2025, Irkutsk, Russia
EMAIL: vicavia@yandex.ru (A. 1); email2@mail.com (A. 2)
ORCID: 0009-0009-8553-1153 (A. 1); 0000-0001-6638-5204 (A. 2)
©2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).
ICCS-DE 2025 Workshop Proceedings
DOI: 10.47350/ICCS-DE.2025.12

Проведено предварительное оценивание показателей качества результатов обработки и намечены пути их улучшения. Разработанная методика направлена на совершенствование технологий искусственного интеллекта, имеет важное значение при обработке пространственных данных и решении задач информационного обеспечения в различных областях практической деятельности.

2. Основная часть

2.1. Методический аппарат

Структурная схема методики формирования ИД для свёрточной нейронной сети (CNN) приведена на рисунке 1.

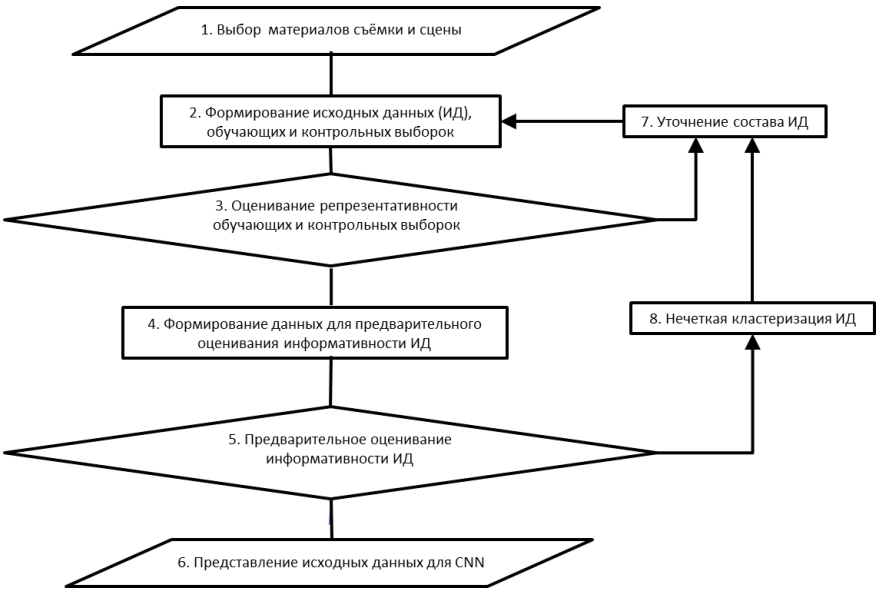


Рисунок 1: Структурная схема методики

На этапе 1 оператор осуществляет выбор сцены и её фрагмента в пределах условного кадра материалов съёмки с учётом содержания прикладной задачи. Фрагмент сцены содержит преимущественно идентифицируемые элементы ландшафта.

На этапе 2 оператор выбирает ИД. Состав ИД для двух элементов ландшафта приведен на рисунке 2. Исходные данные представляются в виде результатов измерения спектральных отражательных характеристик элементов ландшафта, имеющих строгую географическую привязку.

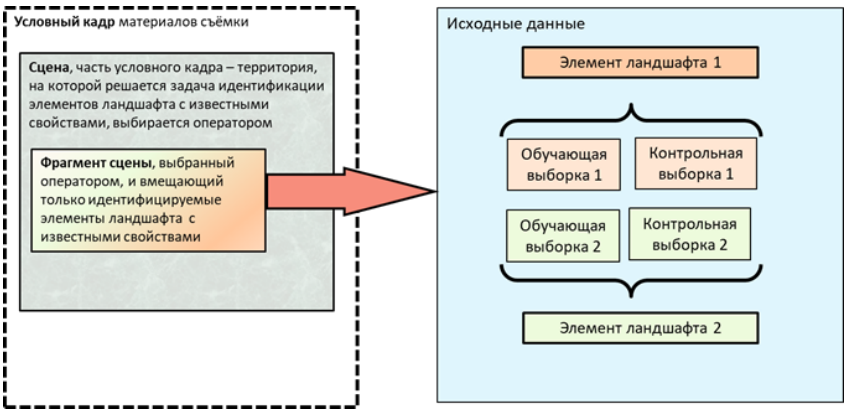


Рисунок 2: Состав исходных данных

На этапе 3 осуществляется оценивание репрезентативности ИД. Выборки считаются репрезентативными, если их статистические характеристики в части средних значений, дисперсии и среднеквадратичного отклонения отличаются от характеристик генеральной совокупности в пределах сцены не более чем на 5%. При невыполнении этого условия производится уточнение состава исходных данных.

На этапах 4 и 5 осуществляется формирование данных и предварительное оценивание показателей качества решения задачи идентификации элементов ландшафта. На основе результатов расчета информативных индексов или оцениваемых параметров для традиционных алгоритмов идентификации строятся гистограммы и с учётом планируемого размещения гиперпараметра осуществляется расчет базовых и обобщенных показателей качества решения задачи идентификации элементов ландшафта [1].

При невыполнении требований по информативности на этапе 8 выполняется операция корректировки ИД с помощью математического аппарата нечёткой кластеризации (FCM) [2].

2.2. Пример реализации методики

Методика апробирована на примере формирования ИД и идентификации трёх элементов ландшафта в пределах сцены, приведенной на рисунке 3. Рассматриваются материалы мультиспектральной космической съёмки, выполненной с помощью космического аппарата «Канопус-В» KV5_20109_19337_03_KANOPUS_20220812_081812_081849.SCN12.MS.L2.

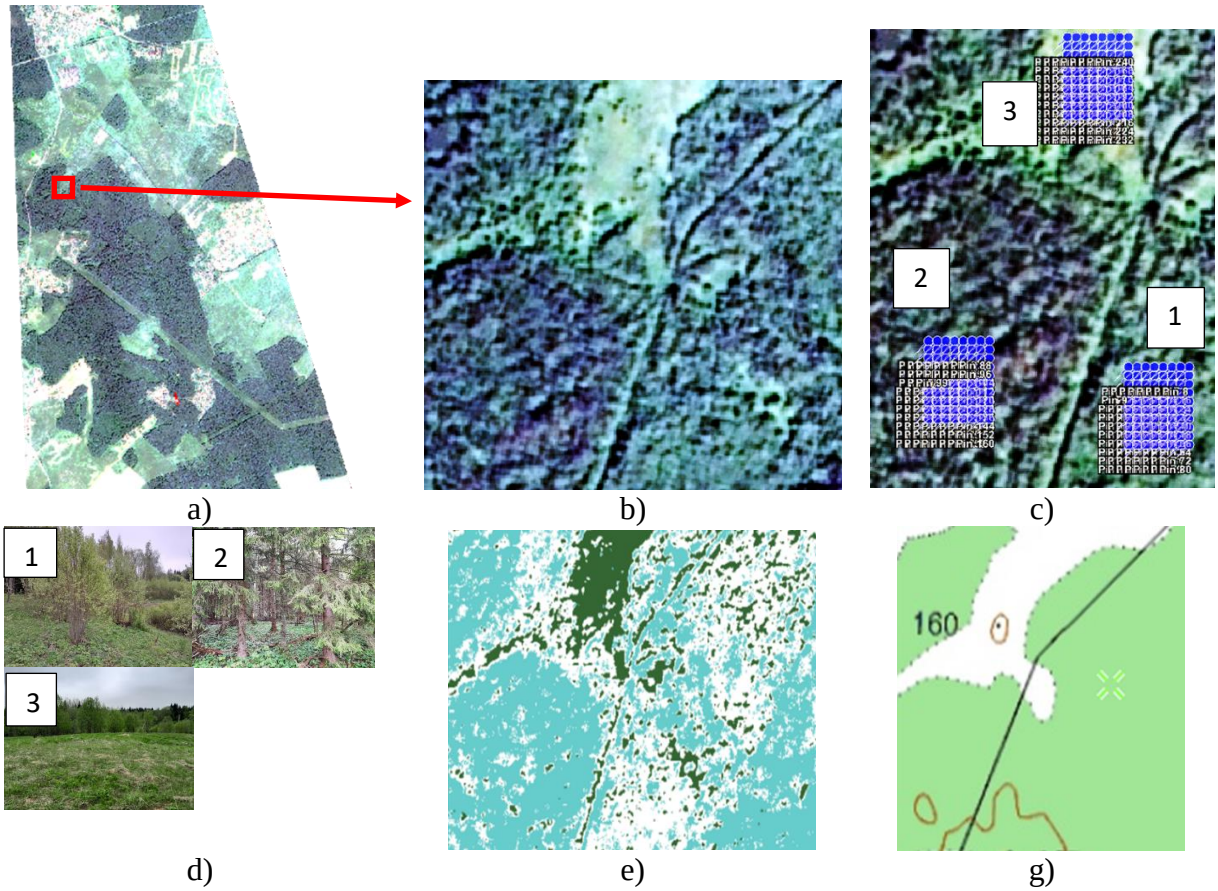


Рисунок 3 Пример формирования исходных данных и результаты идентификации элементов ландшафта: a) часть условного кадра материалов съёмки; b) сцена; c) исходные данные, сформированные оператором; d) результаты наземных обследований для семантического описания элементов ландшафта; e) результаты идентификации элементов ландшафта;g) фрагмент топографической карты сцены. Цифрами на рисунке обозначены

элементы ландшафта с условными наименованиями: 1 - редколесье; 2 - мощный лес; 3 - поляна.

На этапе 3 проведена проверка репрезентативности ИД, которая показала удовлетворительные результаты. При этом размер каждой выборки исходных данных составляет 80 единиц (пикселей).

На этапах 4 и 5 проведено предварительное оценивание качества результатов идентификации элементов ландшафта. Для этого построены гистограммы, представленные на рисунке 4, отражающие размещение пикселей из состава исходных в зависимости от спектрального индекса NDVI [3], характеризующего состояние растительности.

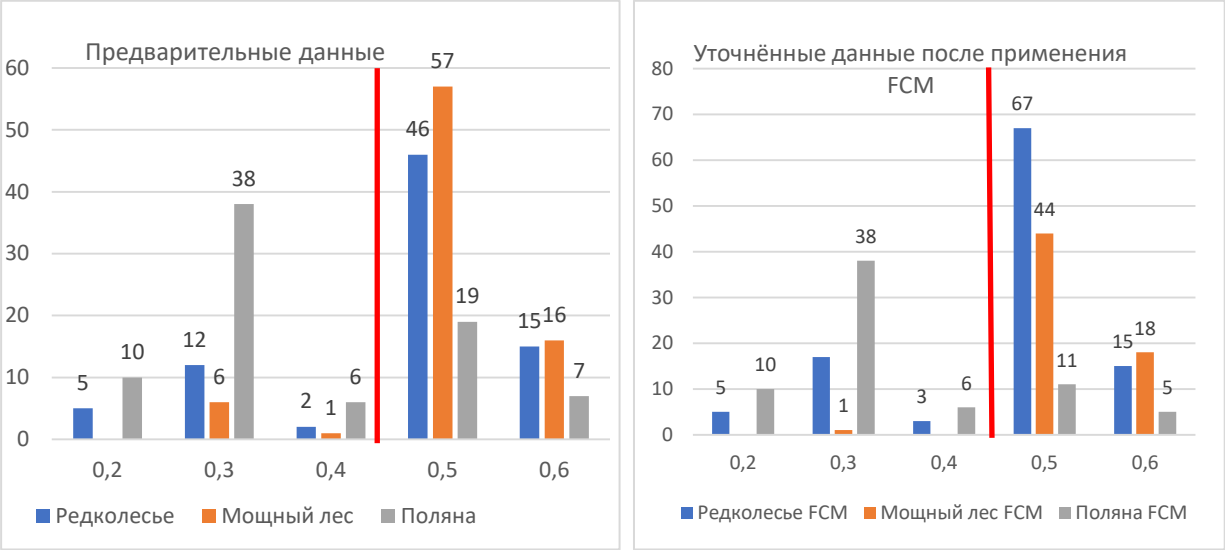


Рисунок 4 Гистограммы для расчета базовых показателей качества результатов идентификации элементов ландшафта: слева – на основе предварительного формирования исходных данных; справа – после применения математического аппарата нечёткой кластеризации (FCM). По оси абсцисс отложены значения индекса NDVI, по оси ординат – количество пикселей, характеризующихся соответствующим значением индекса. Красной линией обозначено значение гиперпараметра, назначаемое оператором.

В ходе анализа гистограмм получены предварительные оценки базовых и обобщенных показателей качества результатов идентификации. Значения показателей приведены в таблице 1. Анализ этих показателей позволил сделать вывод о необходимости повышения качества результатов идентификации элементов ландшафта путём уточнения состава исходных данных на основе применения математического аппарата нечёткой кластеризации. Результаты уточнения ИД отображены на рисунке 5.

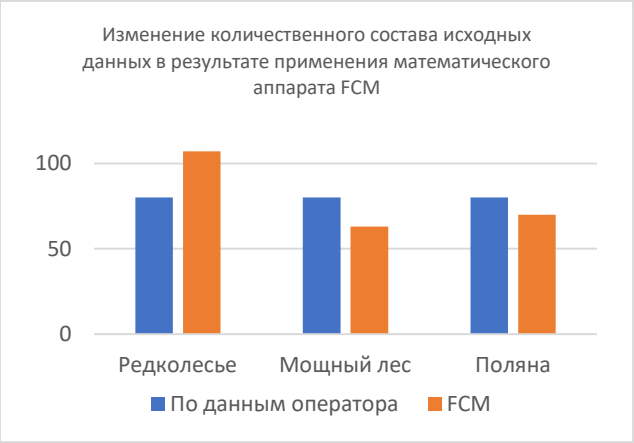


Рисунок 5 Результаты уточнения состава исходных данных после применения алгоритма нечёткой кластеризации (FCM).

В результате уточнения состава исходных данных улучшились предварительные значения обобщенных показателей качества результатов идентификации рассматриваемых элементов ландшафта.

Таблица 1

Показатели предварительного оценивания качества результатов идентификации элементов ландшафта с условными наименованиями «поляна» - «мощный лес» на основе исходных данных, сформированных оператором и после применения математического аппарата FCM

	TP	FP	TN	FN	OA
Оператор	54	7	73	26	0,79
FCM	54	1	62	16	0,87

В таблице приняты следующие обозначения:

TP (True Positive) - количество пикселей, представляющих действительно «поляну»;

FP (False Positive) - количество пикселей, представляющих ошибочно «поляну»;

TN (True Negative) - количество пикселей, представляющих действительно «мощный лес»;

FN (False Negative) - количество пикселей, представляющих ошибочно «мощный лес»;

OA (Overall Accuracy) – обобщенный показатель, качества результатов идентификации, рассчитываемый по формуле:

$$OA = \frac{(TP + TN)}{(TP + FP + TN + FN)}. \quad (1)$$

3. Заключение

Разработанная методика позволяет сформировать исходные данные для алгоритмов машинного обучения с обеспечением: репрезентативности объёма исходных данных; возможности предварительного оценивания качества результатов идентификации элементов ландшафта на основе сформированных выборок; возможности уточнения состава исходных данных.

4. Список литературы

[1] Maxwell A.E., Warner T.A., Guillén L.A. Accuracy Assessment in Convolutional Neural Network-Based Deep Learning Remote Sensing Studies—Part 2: Recommendations and Best Practices. Remote Sens. 2021, 13, 2591. https://doi.org/10.3390/rs13132591

[2] Mochalov, V., Grigorieva, O., Zhukov, D., Markov, A., Saidov, A. Remote Sensing Image Processing Based on Modified Fuzzy Algorithm Advances in Intelligent Systems and Computing, 2020, 1225 AISC, стр. 563–572 Conference Paper

[3] https://www.indexdatabase.de/

Исследование влияния грозовых явлений на лесные пожары на основе методов машинного обучения

Юлия Пестова [1,2], Ольга Николайчук [1,2], Яна Ветвицкая[1]

[1] *Иркутский государственный университет, Россия, 66400, г. Иркутск, ул. Карла Маркса 1*
[2] *Институт динамики систем и теории управления им. В.М. Матросова Сибирского отделения Российской академии наук, Россия, 664033, г. Иркутск, ул. Лермонтова, 134*

Аннотация

Исследование посвящено анализу влияния грозовых явлений на риск возникновения лесных пожаров с использованием методов машинного обучения. Проведено сопоставление данных о грозах и лесных пожарах в Иркутской области за период с 2011 по 2023 год с применением алгоритма ближайшего соседа KD Tree. Показано, что включение грозовых признаков незначительно улучшает точность прогнозирования риска, однако ведущую роль играют погодные условия, предшествующие грозе. Полученные результаты могут быть использованы для совершенствования моделей прогнозирования и предупреждения лесных пожаров.

Ключевые слова

Риск лесных пожаров, сухие грозы, грозовые пожары, машинное обучение, прогнозирование пожаров, KD Tree, AutoGluon

1. Введение

Лесные пожары остаются серьёзной экологической проблемой, представляющей значительный риск для природных экосистем и хозяйственной деятельности человека. Среди причин возникновения пожаров, особую роль играют молниевые разряды, вызывающие до 10% лесных пожаров в России [1]. С ростом температуры и изменением циркуляции атмосферы, согласно прогнозам, количество грозовых явлений и связанных с ними пожаров в ближайшие годы будет расти [2, 3]. Целью исследования является оценка влияния грозовых признаков на точность модели прогнозирования риска возникновения лесных пожаров с помощью машинного обучения.

2. Методология исследования и предварительная обработка данных

В исследовании собраны и обработаны данные о лесных пожарах и грозах в Иркутской области за период с 2011 по 2023 год. Исходный набор данных для исследования содержит записи о грозах, полученных из архива NASA GHRC (WWLLN) [4], о лесных пожарах, предоставленных ИКИ РАН [5] и физико-географических факторов территории, включая: топографические (высота, крутизна и ориентация наклона), характеристики растительного покрова [6] и метеорологические факторы [7, 8].

На этапе обработки данных проведена пространственно-временная фильтрация данных о пожарах, генерация «не пожарных» событий в равном соотношении с образцами «пожарных» событий, и их агрегирование с метеорологическими показателями за периоды 3, 7 и 14 дней. В результате был получен набор данных о событиях с учетом пространственных характеристик территории и метеорологических показателей, таких как сумма осадков, максимальные и

7th International Workshop on Information, Computation, and Control Systems for Distributed Environments (ICCS-DE 2025), July 7–11, 2025, Irkutsk, Russia
EMAIL: yupest@icc.ru (A. 1); nikoly@icc.ru (A. 2); yanavetvitskaya@gmail.com (A. 3)
ORCID: 0000-0001-5874-7965 (A. 1); 0000-0002-5186-0073 (A. 2); 0009-0008-9546-5586 (A. 3)
©2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).
ICCS-DE 2025 Workshop Proceedings
DOI: 10.47350/ICCS-DE.2025.13

минимальные температуры воздуха, средние показатели скорости ветра, влажности воздуха, почвы и количество засушливых дней.

Для пространственно-временного сопоставления событий пожаров и гроз использовался алгоритм ближайшего соседа KD Tree. KD Tree представляет собой структуру данных, основанную на многомерном бинарном дереве, эффективно работающую при поиске ближайших соседей в пространствах с умеренным числом измерений и при использовании стандартных евклидовых метрик. В результате проведённых экспериментов, алгоритм KD Tree продемонстрировал высокую скорость работы и точность для решения задачи сопоставления событий пожаров и гроз в трёхмерном пространстве, на основе географических координат и даты событий.

В данном исследовании гроза считается причиной пожара, если выполняются условия: наличие сухой грозы (при осадках менее, чем 0,5 мм), дистанция до пожара менее 15 км или принадлежность к одной ячейке регулярной сетки 0,25°, временной период между грозой и пожаром менее 7 дней и наличие как минимум 3-х засушливых дней до пожара.

Различные исследования подтверждают, что большинство пожаров, вызванных молниями, возникает в течение первых 2–4 дней после грозового разряда и на расстоянии до 20 км [2]. В данной работе также было установлено, что большинство пожаров возникают в первые 3 дня после грозы, особенно часто — непосредственно в день грозы. Наибольшее количество пожаров от грозы возникает в пределах 5–10 км. Согласно статье [2], максимальная вероятность возгорания наблюдается при условии, если грозе предшествовал период без значительных осадков, что подтверждает важность учёта засушливых дней перед грозой при прогнозировании риска пожаров.

Результаты анализа показали, что в среднем число грозовых пожаров составляет до 6% от общего числа возгораний, в определенные годы этот процент увеличивается до 8-9%. Рисунок отражает статистику грозовых пожаров на территории Иркутской области за 2013 г. (рис. 1).

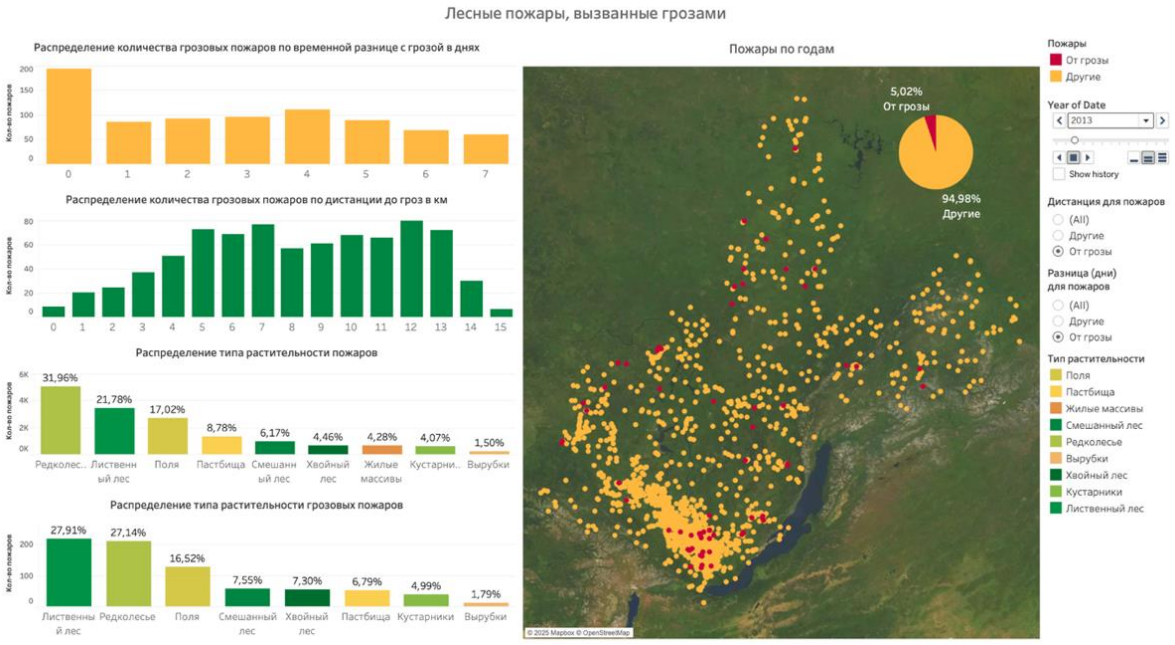


Рисунок 1: Визуализации распределений и картирование пожаров, в том числе от гроз

Исследование пространственного распределения сухих гроз по территории Иркутской области показал, что молниевые разряды распределяются неравномерно и сконцентрированы преимущественно в центральной и южной частях региона. Максимальная плотность сухих гроз наблюдается в районе, прилегающем к озеру Байкал, а также вдоль рек Ангара и Лена, что обусловлено особенностями местного рельефа и микроклимата этих территорий. Рельеф, в частности, наличие возвышенностей и горных систем, таких как Восточный Саян и Прибайкалье, способствует формированию конвективных облаков и, следовательно,

увеличивает вероятность сухих гроз и связанных с ними пожаров. В северной части области частота сухих гроз заметно снижается, что связано с климатическими особенностями этой территории – более прохладными и влажными условиями, менее способствующими развитию засух и образованию сухих гроз. Рисунок позволяет ознакомиться с картированием сухих гроз (рис. 2).

Рисунок 2: Картирование сухих гроз за 2014 год

3. Результаты моделирования и анализ важности признаков

Для построения моделей классификации риска возникновения лесных пожаров была использована логистическая регрессия как базовая модель. Точность данной модели составила 0,92. Затем была применена библиотека автоматического машинного обучения AutoGluon, позволяющая автоматически подбирать гиперпараметры моделей машинного обучения, обучать их, в том числе ансамбли и деревья решений, и выявлять наиболее производительную. В процессе автоматического подбора моделей AutoGluon использовались модели Random Forest, LightGBM, ExtraTrees, CatBoost, где наиболее эффективной оказалась модель случайного леса (Random Forest [9]), обученная на метеорологических показателях, агрегированных за последние три дня перед пожаром. Эффективность моделей измерялась на основе метрики точности f1-score на тестовой выборке, которая составляет 20% от исходного набора данных.

На первом этапе модель обучалась без грозовых признаков. Наибольшее значение метрики f1 на тестовой выборке было получено при использовании погодных признаков, агрегированных за последние 3 дня, что составило 0,93. При увеличении периода агрегации до 7 и 14 дней наблюдалось последовательное снижение точности, на тестовой выборке значение уменьшилось до 0,92.

На втором этапе модель была обучена с учетом грозовых явлений и проведено сравнение моделей с целью понимания, как наличие факторов о грозах влияет на точность модели. Максимальное значение f1 вновь было получено при 3-дневной агрегации и составило 0,94 на тестовой выборке для модели Random Forest. Однако с увеличением временного интервала до 7 и 14 дней метрика точности снижалась аналогичным образом, как и при обучении без грозовых признаков: до 0,938 при учете интервала 7 дней и 0,935 – 14 дней. Остальные модели показали менее точный результат: LightGBM – 0,938, ExtraTrees – 0,912, CatBoost – 0,931.

Необходимо отметить, что при добавлении грозовых признаков качество модели улучшилось на 0,01 по метрике f1 на тестовой выборке. Это говорит о том, что грозовые признаки имеют положительное, но не решающее влияние на точность прогнозирования.

По важности наибольшее влияние оказали признаки: засушливые дни, температура воздуха за последние 3-7 дней до пожара, влажность воздуха и почвы (3 и 7 дней) и наличие сухой грозы имеет значимую, но не ведущую роль. Низкая значимость бинарного признака грозы объясняется тем, что модель получает аналогичную информацию из более детальных погодных признаков, таких как дни засухи и температура воздуха. Таким образом, сам факт сухой грозы менее информативен, чем условия, при которых она возникла. Анализ SHAP выявил, что зоны за ветром (низкая ветровая экспозиция) на крутых склонах создают локально более сухие условия, повышающие вероятность возникновения грозовых пожаров во всех рассмотренных периодах. Дашборд с результатами исследования влияния грозовых признаков на прогнозирование пожара представлен на рисунке (рис. 3).

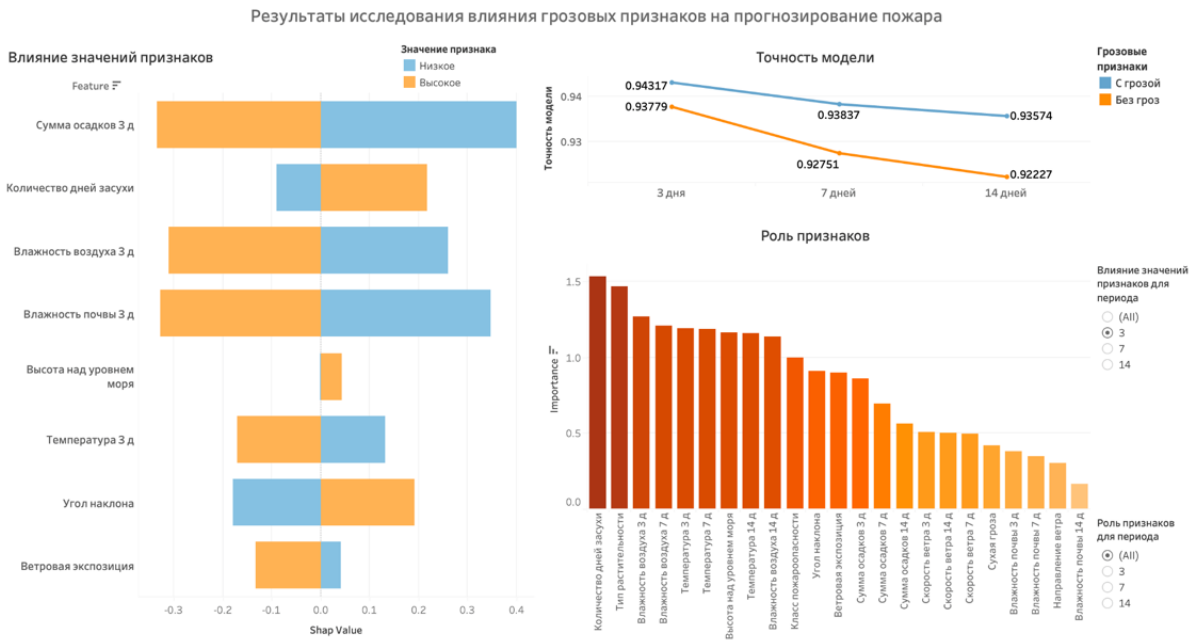


Рисунок 3: Визуализации важности признаков и точности прогнозирования риска пожаров

4. Заключение

Проведённое исследование подчёркивает сложность связи грозовой активности с возникновением лесных пожаров. Несмотря на интуитивную значимость молний как причины возгораний, наибольшую роль играют именно погодные условия, предшествующие грозе. Важным направлением дальнейших исследований может стать интеграция данных о длительности и мощности разряда молнии, а также их детального пространственного распределения. Отдельного внимания заслуживает исследование возможности возникновения вторичной грозовой активности над очагами крупных пожаров. Согласно исследованию [2], такие параметры могут значительно повысить качество прогнозирования.

Результаты исследования могут обеспечить разработку комплексных моделей, учитывающих динамическое взаимодействие между пожаром и атмосферными явлениями, и

существенно повысить эффективность прогнозирования и предупреждения лесных пожаров в условиях изменяющегося климата.

5. Благодарности

Работа выполнена в рамках гранта No 075-15-2024-533 Министерства науки и высшего образования РФ на выполнение крупного научного проекта по приоритетным направлениям научно-технологического развития (проект «Фундаментальные исследования Байкальской природной территории на основе системы взаимосвязанных базовых методов, моделей, нейронных сетей и цифровой платформы экологического мониторинга окружающей среды»).

6. Список литературы

[1] Е.Г. Швецов, А.И. Сухинин, Е.И. Пономарев, Исследование влияния погодных условий на возникновение пожаров от гроз, Хвойные бореальной зоны том 25 1-2 (2008) 47–51.
[2] Р.В. Васильев, М.А. Тащилин, А.В. Татарников, Сопоставление динамики термальных точек и зарегистрированных лесных пожаров с динамикой молниевых разрядов на Байкальской природной территории, Вычислительные технологии том 28 6 (2023) 37–45. doi:10.25743/ICT.2023.28.6.004.
[3] А.Ю. Бибаева, Пирогенное преобразование геосистем Западного Прибайкалья: факторы, условия, современное состояние, Известия Иркутского государственного университета. Серия Науки о Земле 35 (2021) 3–18. doi: 10.26516/2073-3402.2021.35.3.
[4] K. Virts, R. Holzworth, M. McCarthy, World Wide Lightning Location Network (WWLLN) Monthly Thunder Hour Data, Dataset available online from the NASA Global Hydrometeorology Resource Center DAAC, Huntsville, Alabama, U.S.A. 2024. doi: 10.5067/WWLLN/DATA101
[5] Е.А. Лупян, Д.В. Лозин, С.А. Барталев, И.В. Балашов, Ф.В. Стыценко, Оценка повреждений российских лесов пожарами в XXI веке на основе анализа интенсивности горения по данным прибора MODIS, Современные проблемы дистанционного зондирования Земли из космоса том 21 6 (2024) 233–249. doi: 10.21046/2070-7401-2024-21-6-233-249.
[6] I.V. Bychkov, G.M. Ruzhnikov, R.K. Fedorov, A.K. Popova, Y.V. Avramenko, On classification of Sentinel-2 satellite images by a neural network ResNet-50, Computer Optics volume 47 3 (2023) 474–481.
[7] S. Saha et. al. The NCEP Climate Forecast System Reanalysis, Bull. Amer. Meteor. Soc, volume 91 8 (2010) 1015–1057.
[8] S. Saha et. al. The NCEP Climate Forecast System Version 2., J. Climate 27 (2014) 2185-2208.
[9] L. Breiman, Random forests, Mach. Learn 45 (2001) 5–23. doi: 10.1023/A:1010933404324.

Разработка препроцессора языка программирования python для распределенной обработки данных

В.С. Симонов[1], М.С. Хайретдинов[1,2]

[1]*Новосибирский государственный технический университет (Новосибирск, Россия)*
[2]*Институт вычислительной математики и математической геофизики СО РАН (Новосибирск, Россия)*

Аннотация

Постановка проблемы. Появление парадигм распределенных вычислений потребовало эволюции методов разработки программного обеспечения, более того, возникла необходимость эффективного преобразования традиционно последовательных приложений в распараллеливаемые и масштабируемые архитектуры. Это преобразование является ключевым в использовании вычислительной мощности современных многоядерных и распределенных систем. Python, с его обширной экосистемой и простотой, становится мощным языком, который облегчает этот переход. Результаты. В этой статье рассматриваются методологии и разработка инструмента препроцессора Python, предназначенного для преобразования последовательных приложения в формы, подходящие для крупномасштабных вычислительных сред. Практическая значимость. Благодаря подробному анализу и внедрению мы демонстрируем использование функций Spark и сторонних библиотек для автоматизации этого перехода, что делает распределенные вычисления более доступными для более широкого круга приложений и разработчиков. Препроцессор включает в себя набор правил, которые определяют различные шаблоны ввода кода (например, циклический переход по списку) и преобразуют код, пригодный для платформ параллельной обработки данных. Результаты. В данной статье представлен препроцессор Python, предназначенный для автоматизации преобразования из последовательных приложений на Python в те, которые используют возможности распределенных вычислительных сред.

Ключевые слова

Формальные языки, распределенные вычисления, параллельное программирование, python,

Введение

В области разработки программного обеспечения переход от последовательных или однопоточных приложений к приложениям, использующим распределенные вычислительные ресурсы, таким как многопоточные, параллельные или облачные среды, сопряжен с рядом трудностей. Тем не менее, это необходимая эволюция, позволяющая справляться с растущими объемами данных и вычислительными требованиями современных приложений. Последовательные приложения работают

7th International Workshop on Information, Computation, and Control Systems for Distributed Environments (ICCS-DE 2025), July 7–11, 2025, Irkutsk, Russia
v.s.simonov@corp.nstu.ru (Simonov V.); marat@opg.sscc.ru (Khairetdinov M.)
0009-0004-0452-7819 (Simonov V.); 0009-0004-0452-7819 (Khairetdinov M.)
© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).
ICCS-DE 2025 Workshop Proceedings (iccs-de.icc.ru)
DOI: 10.47350/ICCS-DE.2025.14

поэтапно, выполняя одну операцию за раз. Этот подход, несмотря на свою простоту, становитя все более неадекватным для решения задач, требующих обработки больших наборов данных или выполнение сложных вычислений в разумные сроки [1].

MapReduce - популярная модель для разработки распределенных приложений. Она отличается разнообразием и высокой эффективностью работы с такими реализациями [2,3,4,5]. Переход к парадигмам распределенных вычислений, таким как параллельная обработка и облачные вычисления, может значительно сократить время выполнения за счет разделения задач на более мелкие блоки, которые могут выполняться одновременно.

Распределенные вычисления охватывают целый ряд архитектур, включая параллельные, грид-, облачные и распределенные вычисления. Каждая парадигма обладает уникальными преимуществами в обработке обширных вычислений и больших наборов данных на нескольких вычислительных модулях.

Роль Python в параллельных и распределенных вычислениях определяется его богатым набором библиотек и фреймворков, таких как PySpark, интерфейс передачи сообщений (MPI) для Python (mpi4py), Dask и Joblib. Эти инструменты абстрактны сложность взаимодействия и управления процессами делает Python идеальным языком для разработки масштабируемых приложений.

Последовательные приложения, обычно предназначенные для однопоточных сред, не используют многоядерные современные процессоры или возможности распределенных систем, что приводит к неэффективным вычислениям. Преобразование этих приложений в параллельную или распределенную архитектуру может значительно повысить производительность и масштабируемость.

Распределенные вычисления охватывают целый ряд архитектур, включая параллельные, грид-, облачные и распределенные вычисления. Каждая парадигма обладает уникальными преимуществами в обработке обширных вычислений и больших наборов данных на нескольких вычислительных модулях.

Роль Python в параллельных и распределенных вычислениях определяется его богатым набором библиотек и фреймворков, таких как PySpark, интерфейс передачи сообщений (MPI) для Python (mpi4py), Dask и Joblib. Эти инструменты абстрактны сложность взаимодействия и управления процессами делает Python идеальным языком для разработки масштабируемых приложений.

1. Постановка задачи

Преобразование последовательного приложения в распределенное вычислительное приложение вручную сопряжено с преодолением ряда трудностей:

1. Сложность кода: параллельный или распределенный код создает сложности, связанные с такими проблемами, как синхронизация, совместное использование данных и шаблоны взаимодействия.
2. Сложность отладки: отладка параллельных и распределенных систем по своей сути является более сложной задачей из-за параллельных операций и проблем с синхронизацией.

3. Знание инфраструктуры: разработчики должны иметь глубокое представление о целевой крупномасштабной вычислительной среде.
4. Оптимизация производительности: простое преобразование распределенный или параллельный запуск приложения не гарантирует повышения производительности; требуется тщательная оптимизация.

Для решения этих задач мы предлагаем препроцессор Python, который позволяет абстрагироваться от сложностей преобразования последовательного кода в формат, подходящий для распределенных вычислений.

2. Структура препроцессора и реализация

Препроцессор анализирует последовательный код на языке Python, находит фрагменты, которые можно распределить и выполняет преобразование кода, включая соответствующие библиотеки и внося необходимые изменения.

2.1. Обзор

Препроцессор Python, разработанный в данной работе, представляет собой инструмент, который анализирует и преобразует код Python, написанный для последовательного выполнения, в форму, которая может быть выполнена в параллельной или распределенной среде. Его базовая архитектура показана на рисунке 1.

Он включает в себя функции анализа кода, распознавания образов и автоматического преобразования кода. Препроцессор, предназначен для преобразования входного кода в эффективный код в рамках парадигмы MapReduce, поддерживаемый платформой PySpark. Конвейер выполнения состоит из трех основных модулей, архитектура которых будет описана далее.

Анализатор программ. Данный начальный этап включает в себя разбор входного кода с помощью абстрактного синтаксического дерева (AST) и выполнение статического анализа программы. Цель этого этапа - определить сегменты кода, которые подходят для перевода. Для каждого идентифицированного сегмента анализатор программ выполняет две ключевые задачи:

1. Создание описания пространства поиска, используя высокоуровневое промежуточное представление (IR). Это описание направляет процесс синтеза в следующем модуле , определяя границы и правила для создания корректной сводки программы.
2. Генерация условий проверки (VC), чтобы гарантировать, что сводная информация о синтезированной программе семантически эквивалентна исходному входному коду, тем самым гарантируя правильность процесса преобразования.

Поиск эквивалента. На этом этапе препроцессор синтезирует сводки программ на основе пространства поиска и VC, предоставляемых анализатором программ [10]. Данный модуль воплощает основную логику для преобразования высокоуровневой программной логики в форму, которая легко преобразуется в исполняемый код для

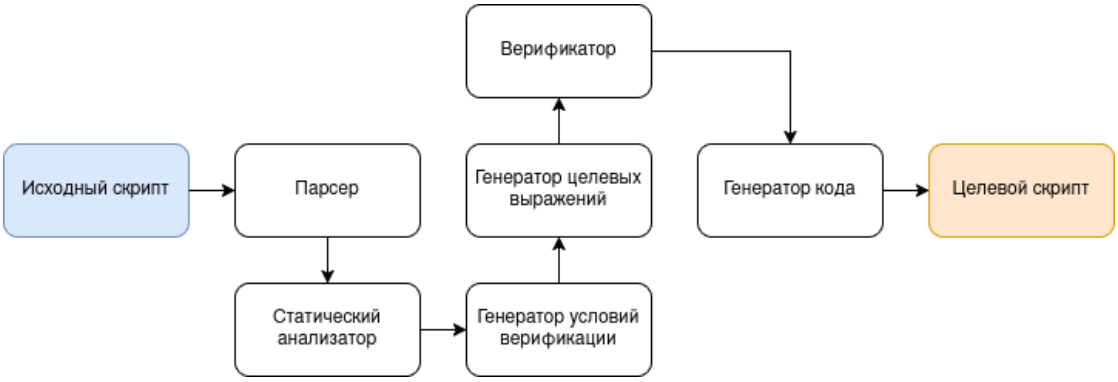


Figure 1: Архитектура системы.

платформ параллельных вычислений. В нем используется алгоритм поэтапного синтеза для эффективной навигации в пространстве поиска, гарантирующий, что процесс поиска корректных сводок программ будет быстрым и точным. Цель состоит в том, чтобы создать сводку, отражающую суть оригинала вычисления таким образом, чтобы обеспечить максимальную эффективность при развертывании в рамках платформы MapReduce.

Генерация кода. После того, как найдено и проверено краткое описание подходящей программы, последним шагом является преобразование этого краткого описания в исполняемый код. Этот модуль легко адаптируется и поддерживает генерацию кода для трех основных платформ MapReduce: Spark, Hadoop. Такое разнообразие позволяет препроцессору быть универсальным и применимым в различных вычислительных средах. Кроме того, этот модуль отвечает за генерацию кода, который отслеживает статистику данных во время выполнения. Эта функция позволяет осуществлять динамический выбор между различными стратегиями внедрения на основе показателей производительности в реальном времени, что еще больше оптимизирует эффективность выполнения переведенных программ [6].

Вместе эти модули образуют комплексный конвейер для системы препроцессора, предназначенный для оптимизации процесса перехода от высокоуровневого кода к эффективному, распараллеливаемому исполняемому коду, подходящему для распределенных задач обработки данных. Этот подход делает упор на автоматизацию в обеспечении семантической эквивалентности за счет использования условий проверки и направлен на оптимизацию производительности во время выполнения путем адаптивного выбора между различными стратегиями выполнения на основе статистики текущих данных.

2.2. Восходящая верификация

Восходящая верификация [11, 12] представляет собой методологию, направленную на анализ семантики программного кода путём его преобразования в абстрактные спецификации высокого уровня. Данный подход заключается в генерации

постусловий для отдельных участков кода, формально описывающих их влияние на выходные переменные. Разрабатываемый язык спецификаций ориентирован на решение двух ключевых задач:

1. Формирование простых абстракций, совместимых с целевой предметно-ориентированной платформой (DSL). Это означает, что отбрасываются корректные абстракции, не поддающиеся преобразованию в синтаксис целевого языка. Соответственно, язык спецификаций должен избегать конструкций, не имеющих прямых аналогов в DSL.
2. Генерация содержательных абстракций, поддерживающих параллельную обработку данных. Из рассмотрения исключаются спецификации, ориентированные исключительно на последовательные вычисления.

2.3. Генерация кода

Препроцессор также включает механизмы для выборки входных данных и динамического переключения между исходным кодом и оптимизированной версией на основе показателей производительности.

Это объяснение подчеркивает логику препроцессора и его подход к автоматизации оптимизации кода Python для распределенных вычислительных сред, предоставляя как общий обзор его возможностей, так и некоторое представление о технических деталях, связанных с его работой. Препроцессор не только упрощает переход от автономного кода на Python к оптимизированной распределенной версии но также включает в себя ряд функциональных возможностей, предназначенных для повышения производительности и обеспечения надежности в различных вычислительных средах.

Препроцессор разумно распределяет данные по узлам, принимая во внимание размер, сложность и вычислительные требования каждой задачи, тем самым сводя к минимуму накладные расходы и максимально повышая эффективность обработки данных. Используя базовую архитектуру распределенных вычислительных платформ, таких как Spark, этот инструмент реализует стратегии обеспечения отказоустойчивости, такие как информация о происхождении для RDD, что позволяет вычислениям корректно восстанавливаться после сбоев. отказы узлов без существенных потерь производительности.

3. Результаты

В качестве примера того, как работает препроцессор, приведем код из листингов 1 и 2. В листинге 1 функция select предназначена для поиска по списку записей (предполагается, что они хранятся в переменной с именем table) в поисках тех записей, первый столбец которых соответствует указанному ключу. Когда найдено совпадение, эта запись добавляется в список результатов, который возвращается в конце работы функции.

Listing 1: Выбор записи из таблицы.

```
def select(table, key):
    result = []
    for record in table:
        if record.columns[0] == key:
            result.append(record)
    return result
```

В листинге 2 приведен код после запуска препроцессора. И теперь код стал пригоден для выполнения на PySpark.

Listing 2: Предварительно обработанный код для PySpark.

```
def select(rdd, key):
    result = rdd.flatMap(lambda data_index:
        [data_index] if data_index.columns[0]
        == key else [])
    return result.collect()
```

3.1. Производительность

Здесь описан один из тестов. Среднее время обработки каждого фрагмента кода, составляющее 7,3 минуты , в сравнении со значительно меньшим средним временем, составляющим 1,8 минуты, свидетельствует о неравномерном распределении времени компиляции. Это указывает на то, что, хотя многие компиляции выполняются довольно быстро, некоторые из них занимают непропорционально больше времени, возможно , из-за сложных задач оптимизации или проверки, которые препроцессор выполняет для определенных тестов.

Заключение

Разработанный в этом исследовании препроцессор Python представляет собой значительный шаг на пути к тому, чтобы сделать распределенные вычисления более доступными для широкого спектра приложений. Автоматизируя процесс преобразования, этот инструмент не только упрощает переход от последовательной архитектуры к параллельной и распределенной, но и открывает новые возможности для повышения производительности и масштабируемости. По мере дальнейшего развития крупномасштабных вычислений такие инструменты, как этот препроцессор, будут играть решающую роль, позволяя приложениям использовать весь потенциал современных вычислительных инфраструктур.

References

[1] J. Dean, S. Ghemawat. MapReduce: Simplified Data Processing on Large Clusters. Commun. ACM 51, 01.01.2008, с. 107–113.

[2] Apache Spark 2025. (2025). [Электронный ресурс]. URL: https://spark.apache.org (дата обращения 28.03.2025).
[3] A.V. Aho, M.S. Lam, R. Sethi, J. D. Ullman. Compilers: Principles, Techniques, and Tools (2Nd Edition). Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, июнь 2006, с. 38–39.
[4] W. Blume, R. Eigenmann, J. Hoeflinger, D.A. Padua, P. Petersen, L. Rauchwerger, and P. Tu. Automatic Detection of Parallelism: A grand challenge for high performance computing. IEEE PDT 2, 3 1994, с. 37–46. 10.18127/j19997493-201904-08
[5] Simonov V. S. Automatic decomposition of a sequential algorithm for MapReduce Frameworks / V. S. Simonov, M. S. Khairetdinov. – ISBN 978-1-6654-6480-2. DOI 10.1109/SIBIRCON56155.2022.10017034. – Text : electronic // International multi-conference on engineering, computer and information sciences (SIBIRCON–2022) : proc., Novosibirsk–Yekaterinburg, 11–13 Nov. 2022. – IEEE, 2022. – с. 1780–1783.
[6] A. Alexandrov, A. Katsifodimos, G. Krastev, V. Markl. 2016. Implicit Parallelism Through Deep Language Embedding. SIGMOD Rec. 45, 1 (Июнь 2016), с. 53–54.
[7] ROSE. An open source compiler infrastructure. [Электронный ресурс]. URL: http://rosecompiler.org/ROSE$_{H}TML_{R}eference/index.html$(18.03.2025).
[8] M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.
[9] R. Bodik, B. Jobstmann. 2013. Algorithmic program synthesis: introduction. International Journal on Software Tools for Technology Transfer 15 (2013), с. 399–403.
[10] Alvin Cheung, Armando Solar-Lezama and Samuel Madden (2013): Optimizing Database-backed Applications with Query Synthesis. In: Proceedings of the 34th ACM SIGPLAN Conference on Pro- gramming Language Design and Implementation, PLDI '13, ACM, New York, NY, USA, с. 3–14, doi:10.1145/2491956.2462180.
[11] Shoaib Kamil, Alvin Cheung, Shachar Itzhaky and Armando Solar-Lezama (2016): Verified Lifting of Stencil Computations. SIGPLAN Not. 51(6), с. 711–726, doi:10.1145/2980983.2908117.

Конструирование программных интерфейсов

Игорь Николаевич Скопин [1]

[1] *Новосибирский государственный университет, НГУ, Новосибирск, 630090, Новосибирская область, г. Новосибирск, ул. Пирогова, 2, Российская Федерация*
[2] *Институт вычислительной математики и математической геофизики Сибирского отделения Российской академии наук ИВМиМГ СЩ РАН 630090, г. Новосибирск, проспект Академика Лаврентьева, 6 Country, Российская Федерация*

Аннотация

Изучаются проблемы повышения полезности программных систем, связанные с разработкой интерфейсов. Предлагается методика разработки интерфейсов, преодолевающая эти проблемы за счет предварительного анализа операционных маршрутов пользователей разрабатываемой системы.

Ключевые слова

Юзабилити, абстрактный интерфейс, конкретный интерфейс, деятельность, полнота поддержки деятельности, интерфейсные стандарты.

1. Введение

Интерфейс программной системы — важная составляющая качества, как основы оценки полезности приложения для пользователя. В то же время, при разработке интерфейсам должное внимание уделяется не всегда. В результате в принципе полезные программы часто оказываются неудобными или даже не востребованными пользователями. Интерфейсные ошибки сплошь и рядом «украшают» популярные программы. Положение усугубляется тем, что реальное использование таких программ создает предпосылки формирования стихийных стандартов, закрепляющих и тиражирующих интерфейсные ошибки.

Сведения о том, какая доля трудозатрат при создании системы приходится на интерфейс, довольно разноречивы. Есть мнение, что в среднем она составляет более половины времени реализации проекта. Проверить это трудно, в частности потому, что продуманный интерфейс обычно требует большего времени, чем непродуманный. Разработка интерфейса, основанного принципах, которые не удовлетворяют пользователя, может усложнять реализацию. Для разных приложений роль интерфейсных характеристик различна, а потому различна трудоемкость интерфейсной части проекта.

В одной из немногих книг [1], посвященных интерфейсам, приводятся рекомендации и предостережения, обоснованные хорошим анализом. Однако автор обсуждает действия пользователя, не связывая интерфейс с функциональностью и архитектурой, что ограничительно. Мы предлагаем восполнить этот пробел, рассматривая интерфейсы в двух аспектах. Во-первых, это абстрактное управление поведением приложения, а во-вторых — отображение абстрактного управления в конкретные интерфейсные формы. Соответственно, вводятся понятия *абстрактного* и *конкретного интерфейсов* и их *полноты*. Полнота абстрактного интерфейса — это соответствие функциональности всем элементам деятельности, автоматизируемой приложением, а полнота конкретного интерфейса — отражение в его элементах всех аспектов абстрактного поведения в формах, отвечающих пользовательской потребности автоматизируемой деятельности.

Ключевым положением предлагаемого подхода является анализ всех видов деятельности пользователя, на которые влияет включение приложения в качестве средства или инструмента.

7th International Workshop on Information, Computation, and Control Systems for Distributed Environments (ICCS-DE 2025), July 7–11, 2025, Irkutsk, Russia
EMAIL: iskopin@gmail.com
© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).
ICCS-DE 2025 Workshop Proceedings
DOI: 10.47350/ICCS-DE.2025.15

Цель анализа — определить место разрабатываемого приложения в контексте работы пользователя и, в частности, обосновать необходимые элементы интерфейса и построить типовые сценарии взаимодействия с приложением. Эти сценарии рассматриваются как основа отображения абстрактного управления в конкретный интерфейс. В результате разработчик имеет возможность обосновать использование активных, пассивных и декоративных элементов интерфейса для тех или иных целей.

2. Интерфейс как составляющая качества приложения

Обычные критерии качества программного изделия связываются, в первую очередь, с его функциональностью и производительностью. Что же касается интерфейса, то, признавая существенный его вклад в полезность программной разработки равным функциональности и производительности, часто считают, что качество, привносимое интерфейсом, гарантируется использованием при его реализации сложившихся стандартов и хорошо себя зарекомендовавших библиотек поддержки. Причины тому следующие. Если реализацию функций системы можно спланировать и специфицировать на предпроектной стадии и фиксировать результаты этой работы в виде вполне проверяемых требований (во времена советского программирования такие требования предписывалось включать в технические задания), то, за редким исключением, запланировать проверяемые интерфейсные характеристики не удается. Предпроектные требования к интерфейсам сводятся к утверждениям об их соответствии хорошо *зарекомендовавшим* себя образцам, играющим роль стандарта. Чаще всего программные проекты организуются лишь для предоставления новых функций и/или более производительных версий существующих систем. Интерфейсные проекты сводятся к предложению стандартных решений и их библиотечных реализаций, которые отражают довольно произвольное мнение о стандартах элементов интерфейса, но не специфику приложений, для которых применение библиотек оправдано.

Характерными примерами поддержки конструирования интерфейсов могут служить библиотеки, использованные для разработки многочисленных интегрированных сред разработки, так называемых IDE-систем (Integrated development environment см. [3]), предназначенных для разработчиков программного обеспечения. Свое начало все они берут от Turbo Vision [4] и послушно наследуют его средства стандартизованного диалога. Однако в такой среде все приложения становятся похожими друг на друга, их специфика теряется. Безусловно, стандартизация элементов интерфейса нужна, но ее роль должна ограничиваться рамками объективно обусловленных эргономических и когнитивных критериев, учитывать складывающиеся у пользователей привычки и предпочтения. Так же нет сомнения в том, что должны быть выработаны унифицированные правила компоновки элементов и технологически выверенные приемы работы пользователя в интерфейсной среде, подобно тому, как стандартизованы расположение клавиш на пишущей машинке и правила печатания "вслепую". Но унификация не должна вступать в противоречие с требованием обеспечивать максимально полную поддержку конкретной пользовательской деятельности, для которой предназначена программная система.

Такое положение дел нельзя считать удовлетворительным. Большинство интерфейсных решений весьма неоднозначны, во многих случаях оценка качества интерфейса субъективна, о приемлемости или непригодности интерфейса часто удается говорить лишь в целом, без обоснованного указания на четкие критерии.

Потребительская ценность программного продукта, его полезность с точки зрения практического использования, складывается из трех составляющих:

- *функциональность* — способность программы выполнять требуемые от нее функции (набор задач из области приложения, решению которых способствует данное программное изделие, адаптивность для решения новых задач, уровень автоматизации деятельности пользователя при применении программы и другие подобные характеристики);

- *эффективность* — скоростные характеристики и требования памяти, связанные с выполнением программы (они в конечном итоге определяют границы возможного практического использования программы на данном оборудовании);
- *интерфейс* — средства управления работой программы, благодаря которым осуществляется включение ее в человеко-машинную систему.

Эти составляющие качества программного обеспечения не являются независимыми. К примеру, когда программа мало пригодна для практических целей из-за низкой эффективности (о непригодности вообще здесь речи нет, если выполнены функциональные требования), правомерна постановка задачи разработки новой программы, для которой уровень эффективности фиксируется как функциональное требование. Аналогичная ситуация возможна с мало приемлемым интерфейсом.

Функциональная составляющая доминирует в оценке программного изделия: если не выполнены функциональные требования, то говорить об эффективности и интерфейсе не приходится. Функциональные требования естественно рассматривать как регламент, фиксирующий *модель вычислений программы*, а саму функциональность отождествлять с такой моделью. В свою очередь, требования эффективности — это ограничения на допустимость реализации функциональной модели на данном оборудовании, а реализация ее есть *отображение модели вычислений на* реальное *оборудование*, удовлетворяющее требованиям эффективности. Наконец, требования к интерфейсу есть не что иное, как *спецификации языка управления поведением модели*.

Важно отметить, что под языком здесь понимается произвольная знаковая система для передачи информации, включающая не только символьные последовательности, но и визуальные образы (пиктограммы и иные изображаемые знаки), указание картин и их фрагментов с помощью программно-аппаратных средств, задание расцветки изображений, использование меню, транспарантов, табло, обрамлений и др., а также звуковые сигналы и, возможно, речевые средства ввода/вывода. Данный перечень относится к уровню человеко-машинного интерфейса программной системы. Когда речь идет о программах, обеспечивающих доступ к внешнему оборудованию, например, о драйверах, роль языка играют сигналы-сообщения от и для этого оборудования, а также система прерываний. Рассматривая межмодульные интерфейсы, мжно заметить, что для окружения, использующего модуль, предоставляемые им средства являются расширением языка реализации системы. Таким образом, всякий раз, когда речь идет об интерфейсе, нужно иметь в виду тот или иной язык, предоставляемый на уровень использования, и как следствие, если есть возможность выбирать язык, критерием выбора его средств должны быть требования со стороны пользователя.

Выделение модели вычислений, ее отображения как реализации в программе и языка управления поведением модели имеет ряд принципиальных следствий.

1. При проектировании программной системы эти три составляющие разграничивают требования к программному изделию, что создает предпосылки для независимого построения модели, повышения эффективности реализации и разработки интерфейса.

В реальной практике независимость указанных составляющих относительна, поскольку связи между ними существуют на уровне взаимовлияния. Так, можно запланировать вычисления, которые не реализуемы при заданных требованиях к эффективности. Управляемость поведением модели также имеет свои пределы: нельзя говорить об интерфейсе, если не фиксировать управляемые объекты, их возможное поведение и т. п., нельзя строить интерфейс без учета приемлемых для функционирования системы характеристик эффективности. Тем не менее при заданной функциональности отображение модели на реальную операционную среду можно рассматривать как выбор реализации, удовлетворяющей требованиям эффективности, а проектирование интерфейса — как выбор его языковых средств, адекватных использованию программы. Эти две задачи не только можно, но и целесообразно ставить как независимые, в частности по той причине, что для них различны критерии отбора подходящих решений.

2. Явное разграничение трех составляющих качества позволяет ставить задачу разработки серий программ с общей моделью вычислений, предназначенных для различного применения.

Примерами серий такого рода являются системы, учитывающие особенности перерабатываемой информации для повышения эффективности вычислений. Пользователю предоставляется право выбора метода решения задачи, а каждый из методов, если их особенности не проявляются в управлении, — это вариант реализации одной и той же вычислительной модели (если указанное условие не выполняется, т.е. выбор метода является частью управления, то имеет место модель вычислений, включающая с себя вариантные методы, а не серия программ-вариантов). В ряде случаев возможен автоматический выбор метода на основе анализа входных данных, и тогда сама серия для пользователя выступает как единая система.

Другой пример серийности на основе общей модели вычислений — различные программы, использование которых связывается с применением оборудования с несопоставимыми интерфейсными возможностями. Выбор варианта программы таких серий целесообразно возлагать на установку системы.

Серийность, учитывающая особенности перерабатываемых данных, отражает неоднозначность критериев эффективности программ. Они не сводятся лишь к требованию оптимизации — минимизации для конкретных вычислений времени выполнения и/или необходимой памяти. Выбор того или иного алгоритма должен учитывать наряду с особенностями обрабатываемых данных архитектуру вычислительной системы, на которой выполняется программа. Если ставится задача соблюдения оптимальности для разных архитектур — это характерно для большинства вычислительных библиотек — то серийность часто удается обеспечить за счет компиляции, и тогда можно говорить, что пользователю предлагается единая модель вычислений. В противном случае решение о выборе варианта программы должен принимать пользователь, для которого все такие варианты естественно рассматривать как серию. Примером серийности в обоих аспектах может служить библиотека Krylov [5], предназначенная для решения сверхбольших систем линейных уравнений. Алгоритмы решения формально одной и той же задачи специализированы для различных особенностей матриц с целью повышения производительности расчетов. Для заданных параметров задачи в автоматизированном или ручном режиме может быть выбран тот вариант, который дает минимальное время счета.

Еще более неоднозначны интерфейсные критерии, и это обусловливает серии, связанные с вариантами управления. Разные пользователи обладают индивидуальными и групповыми психологическими особенностями восприятия, и, как следствие, для их продуктивной работы требуются различные способы взаимодействия с программной системой. На пользовательские критерии предпочтения влияют и уровень предварительной подготовки, и привычки, и то, что по мере освоения системы требуются различные уровни пояснений, помощи и т.д. В этой связи можно ставить задачу конструирования динамичного интерфейса, т.е. явно и/или неявно настраиваемого на особенности пользовательского предпочтения и восприятия информации.

В качестве наглядной иллюстрации этого положения уместно упомянуть о том, как, к примеру, браузер Chrome [6] или YouTube [7] отбирает и упорядочивает информацию для предъявления пользователю с учетом истории взаимодействия с ним. По понятным причинам первая из этих систем делает это на основе более глубокого анализа.

3. При сравнении программных изделий следует разграничивать сопоставление их функциональности, эффективности и интерфейсов.

На практике это разграничение зачастую игнорируется, что снижает полезность информации для пользовательского выбора подходящего изделия. Если при проектировании функциональность является заданным критерием пригодности будущего изделия, то при сравнении программ ее роль двояка.

Во-первых, функциональные характеристики определяют правомерность использования программы для автоматизации конкретной деятельности. Здесь, скорее, следует сравнивать назначение сопоставляемых программных продуктов, а не такие свойства, как применимость их для массового оперирования (эффективность) и удобство для пользователя (интерфейс).

Во-вторых, сравнение эффективности невозможно без выявления точек пересечения в назначении анализируемых программных изделий, т.е. без выделения их функциональной общности. Этот аспект сравнения достаточно ясен, и обычно эффективность проверяется на общих задачах. Тем не менее, иногда не учитывается, что функциональная общность —

понятие относительное. Так, сравнение эффективности компилятора, который работает параллельно с пользовательским вводом программы, и автономного компилятора, скорее всего, будет в пользу последнего, а простой редактор программ можно сделать более быстрым, чем текстовый процессор. Разная быстродействия здесь не является критерием качества: по внешней схожести применения программ нельзя судить об их функциональности в целом. Ситуация меняется, когда сравнительная эффективность не является критерием качества, — достигнута приемлемая скорость работы программы, т.е. когда с точки зрения полезности применения нет смысла повышать ее эффективность.

Не менее важно разграничение трех составляющих качества для сравнения интерфейсов программ. Проектирование интерфейса в большей степени искусство, чем технология, поэтому для него и критерии качества более субъективны. Но когда сравнение интерфейсов отделено от сопоставления программ по другим составляющим качества, возможна постановка общих вопросов, ответы на которые позволяют сделать оценку интерфейса объективнее.

4. При оценке качества программной системы безотносительно к другим аналогичным изделиям нет смысла говорить о хорошем или плохом продукте, если явно не обозначить деятельность, в которой изделие должно использоваться. Только с этой точки зрения правомерны утверждения о функциональной пригодности системы, об эффективной или неэффективной ее реализации, об удобствах пользовательского управления.

Уместно разграничивать два рода деятельности, для поддержки которых предназначается программа: *автоматизируемая деятельность*, существующая и без применения программных средств, и *новая деятельность*, которая была бы невозможна без оцениваемой программы.

В первом случае программа — один из инструментов деятельности, и ее можно сравнивать с позиции заменяемых или дополняемых неавтоматизированных аналогов. Иными словами, здесь аналогами фиксирована функциональная модель, а о качестве изделия можно судить по тому, насколько эффективнее оказалась выполняемая с помощью программы деятельность по сравнению с неавтоматизированным вариантом. Понятно, что имеется в виду эффективность деятельности, а не программы. как воплощения модели. В частности, эффективность такой деятельности существенно зависит от адекватности управляющих воздействий, т.е. от предлагаемого интерфейса. Можно классифицировать возможные интерфейсные средства по степени их влияния на эффективность деятельности, *подбирать интерфейс* под конкретные условия применения программы, анализируя получаемые результаты.

Во втором случае аналогов программного инструмента нет, а потому критерии качества меняются: оценкой программного изделия становятся результаты новой деятельности, то, как она встраивается в систему других деятельностей. Таким образом, вклад эффективности и интерфейса в оценку качества программы менее весом по сравнению с функциональностью. Тем не менее, если изделие претендует на достаточно долгое существование, вопросы эффективности и интерфейса также должны рассматриваться как существенные. Поэтому вместо подбора интерфейсных средств может быть рекомендована стратегия настройки интерфейса, осуществляемая пользователем: в ходе эксплуатации программной системы, будут выявлены желательные "точки роста" ее эффективности и дополнительные управляющие воздействия. Возможность и целесообразность такого развития программы надо планировать.

3. Задача конструирования интерфейса

Развитые технологии программирования предоставляют разнообразные средства реализации моделей вычислений сложных программных систем и достижения требуемой эффективности. Как правило, они исходят из внешних спецификаций конструируемого программного продукта, в которых можно достаточно точно фиксировать, что должно быть сделано, а следовательно, проверка функциональности и определение достигнутого уровня эффективности в принципе не затруднительны.

Иначе обстоит дело с интерфейсом. Когда интерфейсные критерии удается описать явно (например, в случае межмодульного интерфейса), его разработка не требует особых усилий и сводится к поддающимся рационализации согласованиям. Интерфейсные трудности возникают

при конструировании пользовательского представления системы. Главная причина их в том, что разработчики часто слабо представляют себе, кто и как будет применять их изделие. Иногда они даже заявляют примерно следующее: "Вам предоставлена программа с определенным набором средств, а как распорядиться ими — ваша проблема". Это в корне неверное утверждение — забота о технологии применения предлагаемого (программ, оборудования, документации и др.) — важная задача ответственных разработчиков, именно они могут и должны точнее других ответить на вопросы о назначении изделия.

Без учета при разработке "портрета" пользователя, без продумывания регламентов и методов применения программы потребительская ценность приложения снижается. Функционально хорошие программы не всегда в состоянии удовлетворить пользователя с его запросами, обусловленными особенностями его конкретной деятельности. Иногда приемлемый для одних целей интерфейс оказывается неудобным для других из-за смещений между назначением и применением программы (соответствующие примеры удачных и неудачных интерфейсных решений можно найти в [8]).

Проблема разработки интерфейса, соответствующего потребностям пользователей, состоит в том, что заранее определить истинные потребности получается далеко не всегда. Для решения этой проблемы мы предлагаем воспользоваться специальным приемом, отвечающем рассмотрению системы как совокупности активных и/или пассивных *элементов*. В каждый момент элемент находится в одном из заранее определенных *состояний*, характеризующих возможное *поведение элемента*, т.е. его способность выполнять определенные *действия*, в том числе *установку* и *разрыв связей* элементов, а также их создание и удаление элементов и доступность их действий для выполнения. Все это соответствует хорошо известному шаблону проектирования программных систем MVC (Model, View, Controller) [9]. Его основное требование сводится к необходимости при проектировании разграничивать три компонента системы:

- *Модель вычислений* (Model) – определяет и фиксирует требуемую функциональность разрабатываемой системы как множество допустимых последовательностей состояний элементов модели в целом;
- *Представление* (View) – определяет и фиксирует формы задания и предъявления перерабатываемых данных и другой информации элементов системы, необходимую для реализации модели;
- *Управление* (Controller), — определяет и фиксирует действия и воздействия элементов, которые при функционировании модели активизируют фрагменты вычислений в зависимости от поступающих для управления данных и приводят к изменению состояния модели вычислений в целом.

Информация о состоянии элементов доступна для принятия решений о том, какие воздействия нужно осуществить для перевода системы из текущего состояния в желаемое. Она передается как со стороны модели, так от управления в представление для внешнего использования. Взаимодействие пользователя с приложением осуществляется через представление. Из этого следует необходимость учета того, что для разных типов пользователей могут требоваться различные представления.

При разработке представлений необходимо понимать, что каждый пользователь понимает модель и управление приложения по-своему. Это связано с тем, что он встраивает приложение в свою деятельность в качестве инструмента. А поскольку эту деятельность пользователя и связанные с ней другие деятельности предсказать невозможно, не стоит надеяться на то, что пользовательское понимание модели и управления будут совпадать с тем, что предлагает программа приложения. Задача каждого из представлений, предлагаемых разным типам пользователей, состоит в том, чтобы построить соответствие между моделью и управлением приложения и пользователей. Имея ввиду необходимость поддержки этого соответствия при работе приложения как основы его полезности, или, что то же, юзабилити программной системы в целом, *задача конструирования интерфейса разбивается на две части:*

- Разработка *средств взаимодействия системы* с пользователем в виде программных интерфейсов, которые обеспечивают осуществимость выполнения всей

функциональности приложения. Комплект таких средств называется *абстрактным интерфейсом*.

- Разработка визуальных и/или иных *средств предъявления информации* для пользователя и задания воздействий на систему, которые реализуются как обращение к элементам абстрактного интерфейса. Комплект таких средств называется *конкретным интерфейсом*.

Понятия абстрактного и конкретного интерфейсов в точности соответствуют понятиям абстрактного и конкретного синтаксисов из области конструирования компиляторов (см. [10]): и в том, и в другом случае абстрактное является инвариантом множества конкретного, приводящего к одним и тем действиям модели и управления. Аналогия распространяется и на разделение задачи конструирования интерфейса на две части: в архитектуре компиляторов ему соответствуют оперирование семантической информацией и синтаксический анализ.

На рисунке 1 представлена схема, иллюстрирующая задачу конструирования интерфейса, согласованного с шаблоном проектирования MVC. Шаблон естественным образом задает «водораздел» между абстрактным (относящимся к вычислениям) и конкретным (связанным с деятельностью пользователя) уровнями приложения, к которым относятся две части интерфейсной задачи. На схеме они разделены жирной линией. Модель и управление с точки зрения пользователя обозначены заштрихованными блоками U-Model и U-Control, соответственно.

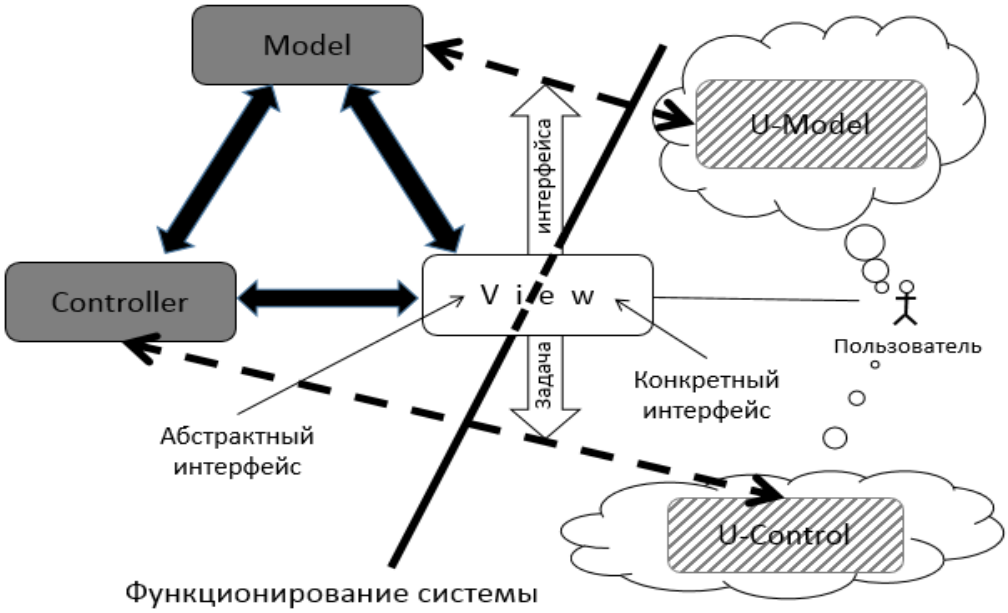


Рисунок 1: Модель MVC, абстрактный и конкретный интерфейсы

4. Деятельность пользователя и разработка интерфейса приложения

Два аспекта задачи конструирования интерфейса, которые мы только что определили, требуют различных методов решения. Разработка конкретного интерфейса исходит из таких понятий, как дизайн экрана, эргономичность, психологические ограничения, использование стандартов и сложившейся практики оформления взаимодействий с программной системой. При построении абстрактного интерфейса ситуация иная. Здесь главным является обеспечение автоматизируемой приложением деятельности максимально возможной поддержкой. Понятно, что это только потенциальная возможность, которая может быть начисто уничтожена плохим конкретным интерфейсом. Подтверждение тому можно найти на каждом шагу. И это довольно поучительно. Так, примеры, представленные Дж. Раскиным в [1], позволили сформулировать ряд универсальных принципов разработки интерфейсов с учетом когнитивных особенностей

восприятия информации. К сожалению, этот автор обсуждает действия пользователя, не связывая интерфейс с архитектурой, и не учитывая различия типов пользователей. Подобные ошибки, рассуждая об интерфейсах, допускает и Дж. Сполски в своей замечательной серии статей про программное обеспечение [2]. В результате разумные рекомендации обоих авторов для интерфейсных решений зачастую оказываются несовместимыми, а порою прямо противоположными. В свое время нам также не удалось избежать подобного недостатка: анализ эргономических и психологических ограничений, а также влияния привычек и сложившихся стихийно стандартов на качество программной разработки, представленный в работе [8], мог быть более точным, если бы он выполнялся раздельно для конкретного и абстрактного интерфейсов.

Понятно, что в связи с понятиями абстрактного и конкретного интерфейсов есть необходимость развить информацию, представленную в указанных публикациях, а также других более поздних работах. В последующем изложении мы обсуждаем в основном вопросы конструирования абстрактных интерфейсов и лишь частично касаемся разработки конкретных интерфейсов. Для этого есть две причины. Во-первых, разработка абстрактного интерфейса *всегда должна предшествовать* решению вопросов о конкретизации пользовательских средств и методов конструируемой программной системы. Можно утверждать, что в качественно проработанном программном проекте абстрактный интерфейс становится базой для проектируемых средств конкретного интерфейса. Вторая причина связана с различием стиля работ, выполняемых в рамках конструирования двух интерфейсных аспектов: абстрактный интерфейс является более технологичной деятельностью, чем конкретный, который можно с полным на то основанием считать искусством. Исходя из этих причин мы рассматриваем методы развития проектом с позиции методологической теории деятельности.

4.1 Общие положения деятельностного подхода

Понятие деятельности — ключевое для проектирования программных систем, и это тема отдельного рассмотрения. Здесь мы рассмотрим лишь те ее аспекты, которые имеют отношение к разработке интерфейсов. И первое, на что стоит обратить внимание, — это *структура деятельности*: из каких элементов она состоит? Естественное ограничение при ответе на этот вопрос — рассматривать только целенаправленные деятельности. Как следствие, появляются следующие *элементы деятельности* (см. рис.2):

- *Цель* — для чего организуется деятельность, т.е. какие продукты и с каким уровнем качества предписано получить при ее выполнении;
- *Результат* — что фактически продуцируется в ходе выполнения деятельности. Результат не всегда соответствует цели. Если он включает все, что предписано целью, деятельность считается успешной, иначе — неуспешной. В обоих случаях при выполнении деятельности не только те продукты, которые соответствует цели (например, опыт разработчиков, отходы и др.);
- *Субъект* — тот, кто выполняет деятельность. Это необязательно индивидуум, поскольку деятельность может выполняться автоматически, и необязательно реальный индивидуум, т.к. деятельность может быть коллективной, и тогда удобно говорить о виртуальном субъекте. Не любой субъект пригоден для выполнения деятельности, а значит, он должен обладать вполне определенными атрибутами (например, соответствующей квалификацией), которые необходимы для выполнения деятельности;
- *Материалы и ресурсы* — то, из чего продуцируются результаты деятельности. Материалы могут быть как материальные, так и информационные, а ресурсы — расходуемыми или не ограниченные (считаются такими);
- *Средства и инструменты* — с помощью чего продуцируются результаты деятельности;
- *Методы* — указывают на то, как выполнять деятельность для получения целевых результатов, потребляя ее материалы и ресурсы.

Деятельность может находиться в одном из ее *состояний*, которые можно рассматривать в качестве характеристики выполнения деятельности. На множестве состояний определено

бинарное отношение возможности перехода из одного состояния в другое. Тем самым задается граф состояний. Переход между состояниями (если он возможен) происходит в результате *действий* субъекта, использующего элементы деятельности. Допускаются изменения состояний и без участия субъекта (пример — конкурентное использование ресурса, общего для разных деятельностей).

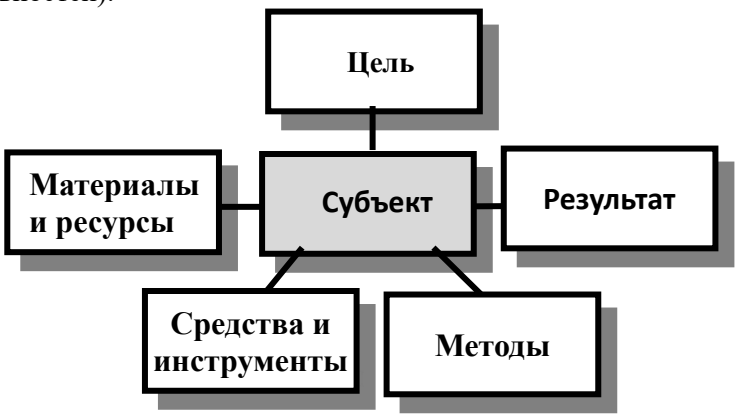


Рисунок 2: Элементы деятельности

В графе состояний выделены *начальносе состояние*, в котором деятельность активизируется, и множество *заключительных состояний*, в каждом из которых деятельность может завершиться. Некоторые из заключительных состояний определяются как *целевые* — считается, что в них цели деятельности достигнуты.

Последовательность состояний, которая выстраивается в результате действий субъекта, называется *операционным маршрутом* деятельности. Операционный маршрут называется *полным*, если он ведет от начального состояния в заключительное. Полный операционный маршрут, приводящий к целевому состоянию, считается *успешным*. В отличие от так называемых *сценариев*, которые для попадания в целевое состояние предписывают субъекту определенные последовательности действий, операционные маршруты выстраиваются субъектом, т.е. допускают свободный выбор очередного состояния, что более точно отражает реальность целенаправленных деятельностей.

Деятельность всегда выполняется в некотором *окружении*, из которого поставляются ее элементы, и в которое передаются ее результаты в качестве элементов других деятельностей из окружения. Связанные этими отношениями деятельности могут группироваться, и такие группы можно трактовать как самостоятельные деятельности со своими окружениями и элементами. Таким образом образуются *система деятельностей*, в которой каждая из деятельностей составляется из других деятельностей, актуализирующих элементы групп.

Системы деятельностей образуются на основе отношения использования продуктов и результатов, передаваемых от одной деятельности другой. Эти и связанные с ними понятия уточняются следующим образом:

- *Результат деятельности* — все, что производится в ней, независимо от того, используется это или нет в какой-либо другой деятельности (в последнем случае использование рассматривается вне изучаемой системы);
- *Продукт деятельности* $Д_п$ — та часть результата $Д_п$, которая *используется* в другой деятельности $Д_и$ в качестве какого-то ее элемента. Множество таких деятельностей $A = \{Д_{и1}, \ldots, Д_{ип}\}$ очень велико. В частности, оно содержит все варианты использования всех пользователей продукта;
- *Целевой (основной) продукт деятельности* $Д_п$ — то, что заявлено в $Д_п$ как ее цель, т.е. обозначены некие деятельности $= \{Д_{иi}, i \in \{1, \ldots, n\} \mid Д_{иi} \in A\}$, о которых предполагается, что они будут использовать продукты $Д_п$);
- *Побочный продукт* $Д_п$—та часть результата, не являющаяся целевым продуктом, для которой есть использующая его деятельность $Д_и \in A \setminus \{Д_{и1} \cup \ldots \cup Д_{ип}\}$ $i \in \{1, \ldots, n\}$.
- *Сопутствующий продукт* — тот продукт (возможно, другой деятельности), без которого использование целевого продукта затруднительно (в разной степени).

Рис. 3 иллюстрирует введенные понятия. Светлые стрелки указывают на блоки, раскрывающие результаты деятельностей, которые предоставляют целевые и побочные продукты другим деятельностям (они выделенных овалами). Темные стрелки отражают использование продуктов.

Автоматизация деятельности — это подмена каких-то ее элементов программными аналогами. Такая подмена неизбежно влечет изменение и других элементов за исключением ее цели — новый инструмент требует новых методов работы с ним, другой квалификации субъекта и пр. Для эффективного (во всех смыслах) выполнения субъектом деятельности необходимо, чтобы его новые операционные маршруты были бы сходными с ранее реализуемыми маршрутами, в частности, они не должны ломать привычные стандарты взаимодействия с системой. Это справедливо и при конструировании новой деятельности, которая предполагает достижение новых целей. В обоих случаях аналогия помогает освоению автоматизированной деятельности.

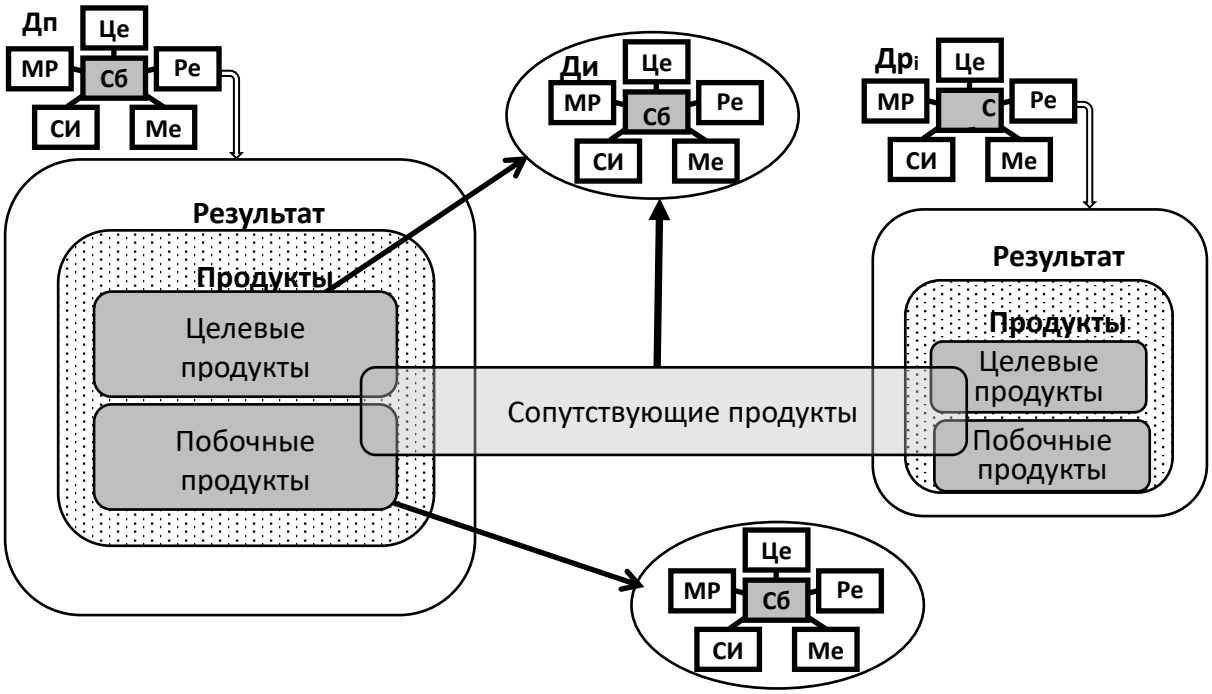


Рисунок 3: Результат и продукты деятельности

Сходство нового операционного маршрута с ранее освоенными пользователям, указывает на то, что деятельность, которую реализует этот маршрут, разбивается на фрагментные деятельности, и некоторые из них соответствуют сформировавшимся у пользователя поведенческим или иным стереотипам. Когда стереотипы отвечают эргономическим и когнитивным ограничениям, производительность труда пользователя повышается. Это наблюдение распространяется и на решение задач конкретного интерфейса. Для разработки абстрактного интерфейса оно может служить рекомендацией для выбора подходящих шаблонов в качестве атомарных единиц действий, так называемых виджетов — примитивов графического интерфейса пользователя, имеющий стандартный внешний вид и выполняющий стандартные действия [11].

4.2 Конструирование абстрактного интерфейса

Предыдущее обсуждение показывает, что для разработки абстрактного интерфейса необходим *анализ операционных маршрутов*, при котором решается, *что* должно предъявляться пользователю в качестве информации и для выбора вариантов действий, но не *как* это предъявляется: показывается, рассказывается и др. Последнее — задача анализа и

формирования представлений конкретного интерфейса, которую нужно решать после того, как приняты решения на абстрактном уровне.

Вариантов операционных маршрутов слишком много, чтобы пытаться охватить их все. Вместо этого целесообразно строить типовые сценарии пользовательской деятельности, которые ограничивают рассмотрение последовательностями состояний деятельности при движении к цели. В отличие от сценариев, предлагаемых пользователю в качестве руководства, аналитические сценарии для каждого состояния сопровождаются информацией о тех состояниях, в которые можно перейти в принципе. С точки зрения отдельного сценария большая часть переходов в такие состояния — это ошибки. Однако не исключено, что другие сценарии организуют последовательности состояний, для которых данные переходы правомерны. Полезное состояние не будет упущено не только, когда другой сценарий уже входит в число типовых, но и случаях, когда его полезность выяснится в дальнейшем, и придется модернизировать систему.

В результате анализа операционного маршрута разработчики должны для каждого состояния определить:

- Какие элементы деятельности используются в пользовательском действии, как изменяются элементы деятельности;
- Какой результат продуцируется при переходе в новое состояния и в каких деятельностях используются продукты;
- В какой программной поддержке нуждаются эти действия.

На основе информации об операционных маршрутах выполняется следующий шаг анализа — *проверка функциональной полноты*, т.е. соответствие предлагаемых средств полному набору автоматизируемых функций, необходимых для пользовательских действий. Полнота проверяется исходя из текущего понимания потребностей пользователя, или, что то же, на основе всех действий субъекта на всех проанализированных операционных маршрутах. Если в процессе разработки программного продукта или даже в ходе его эксплуатации выясняется нарушение функциональной полноты, то комплект предлагаемых средств расширяется, и проверка повторяется.

Из функционально полного набор средств выделяются *базовые элементы* модели вычислений и требуемого управления абстрактного уровня. Набор этих элементов должен быть достаточен для реализации всех автоматизируемых функций, что означает требование *реализационной полноты*. Это достигается путем унификации и комбинирования элементов, а при необходимости, их декомпозиции. Реализация базовых элементов осуществляется в рамках принятых для проекта архитектурных решений и требований эффективности с использованием как общесистемных, так и специальных программных средств. В результате работ, связанных с обеспечением реализационной полноты, в архитектуре фиксируется достигнутый уровень функциональных возможностей. В терминах шаблона MVC это означает спецификацию модели, управления и реализационной части представления. Последнее есть основа конструирования конкретного интерфейса — вторая часть представления.

Наряду с функциональной и реализационной полнотой имеет смысл рассматривать *интерфейсную полноту*, понимая ее как задание языка управления поведением модели, адекватного восприятию программной системы пользователем. Если задача определения абстрактного интерфейса решена, то определена и семантика такого языка. И тогда можно подбирать варианты возможного интерфейса, учитывая эргономические и когнитивные ограничения, о которых убедительно говорит Дж. Раскин [1]. Самый главный аспект интерфейсной полноты — поддержка всех сторон автоматизируемой деятельности, в частности, в интерфейсе должны быть распознаваемы все элементы этой деятельности.

В связи с предложенной методикой конструирования абстрактного интерфейса необходимо подчеркнуть следующее. Разработчик программной системы не должен полагать, что его решения об операционных маршрутах окажутся единственно правильными, а выполненная им проверка функциональной и реализационная полноты станет гарантией этих качеств для пользователя. Практика показывает, что действительно нужные для пользователей программные системы – это продукты, которые постоянно развиваются для удовлетворения пользовательских потребностей. При переходе к разработке конкретного интерфейса, для каждого варианта предлагаемого пользователям и удовлетворяющего ограничениям нужна

проверка того, достигнута ли поддержка всех сторон автоматизируемой деятельности. И первым шагом в этой проверке является выяснение его соответствия абстрактному интерфейсу. В то же время, среди критериев качества не последнее место занимает точность: среди предоставляемых элементов изображений, звуковых и иных сигналов не должно быть ничего лишнего.

Представленная только что схема применения принципов полноты нуждается в дополнении, когда результаты проверки полноты и точности неудовлетворительны и необходима ревизия проекта. В этом случае следует вернуться к этапу, который стал причиной дефекта. Уточненная схема поддержки конструирования интерфейса представлена на рис. 4.

5. Заключение

В данной публикации представлены лишь те аспекты разработки программных систем, которые затрагивают конструирование интерфейсов. Как показывает анализ многих программных проектов, очень часто поддержка деятельности в целом подменяется реализацией лишь требуемых функций без заботы о том, как и для чего пользователь активизирует эти функции. Эта позиция приемлема, когда в деятельности пользователя используется, к примеру, лишь вычислительный результат, получение которого не требует управления. Если же программа предлагается как инструмент деятельности, необходимо понимание всех аспектов этой деятельности, т.е. комплексный системный подход к организации предъявляемых средств.

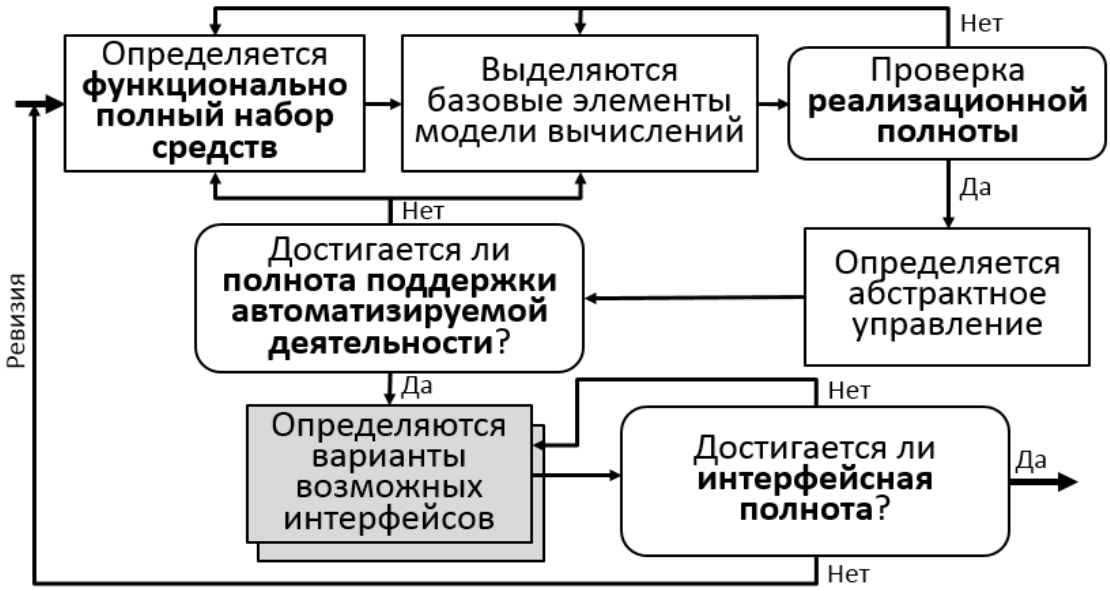


Рисунок 4: Схема применения принципа полноты

Комплексность автоматизации пользовательской деятельности предполагает, что разработчики при конструировании программ не ограничиваются учетом эргономических, когнитивных и психологических ограничений, а стремятся к достижению соответствия предлагаемых средств структуре деятельности, понимают, что автоматизируемая деятельность есть часть большой системы деятельностей, игнорирование которой приводит необходимости приспосабливаться к особенностям программы. На это уже довольно давно указывал Г.П.Щедровицкий в исследованиях по общей теории деятельности и ее приложению к проектировочной деятельности. Его статью [12] можно рассматривать как методологическую основу проведенного в настоящей работе анализа и выдвигаемых принципов конструирования интерфейсов.

Рассмотрение задачи конструирования интерфейсов с позиций теории деятельности приводит к тому, что хорошо себя зарекомендовавший шаблон проектирования MVC должен

быть дополнен понятиями абстрактного и конкретного интерфейсов, на основе которых предлагается особый подход к проектированию и реализации интерфейсов. Одним из наиболее важных соглашений этого подхода является принцип полноты поддержки автоматизируемой деятельности, который естественным образом проявляется в функциональной, реализационной и интерфейсной полноте.

Подход, который мы предлагаем для конструирования не только интерфейсов, но и других составляющих разработки программных проектов в ряде аспектов перекликается с тем, что предлагается в документе PMBOK [13] (Project Management Body of Knowledge). Однако нельзя не отметить принципиальное отличие: PMBOK ориентирован на процессное представление проектной деятельности, в котором нет места субъектам, разграничению продукта и результата. В результате проектные деятельности и их части оказываются черными ящиками со входами и выходами, разбавленными инструментами, которые влияют на выполнение процесса. По этой причине концепция деятельностого представления проектов представляется более перспективной тем более, когда речь идет о конструировании интерфейсов.

Более гибким по сравнению с PMBOK, хотя одновременно и более специализированным является стандарт SWEBOK [14] (Software Engineering Body of Knowledge), претендующий на роль Руководства к своду знаний по программной инженерии. Тем не менее, и к этому стандарту можно предъявить аналогичные претензии.

Подводя итог нашему исследованию проблемы адекватной разработки программных интерфейсов, можно сделать вывод о том, что эта тематика развивается недостаточно интенсивно и пока почти не выходит за рамки экспериментов и случайных решений.

6. Благодарности

Исследования выполнены в рамках государственного задания ИВМиМГ СО РАН FWNM-2025-0005 и представлены на 7th International Workshop on Information, Computation, and Control Systems for Distributed Environments (ICCS-DE 2025), Июль 7–11, 2025, Иркутск, Россия.

7. Список использованных источников

[1] Raskin J. The Humane Interface: New Directions for Designing Interactive System // Addison-Wesley Professional,2000.–256 p. ISBN 0201379376 (ISBN13: 9780201379372)
[2] Перевод на русский: Раскин Дж. Интерфейс: новые направления в проектировании компьютерных систем. // Символ-Плюс. 2005. — 272 с. ISBN 5-93286-030-8
[3] Spolsky J. Joel on Software. // URL: https://m-i-uznetsov.livejournal.com/147976
[4] Перевод на русский: Джоэл о программировании, Символ-Плюс, 2006,
[5] ISBN 5-93286-063-4
[6] Интегрированная_среда_разработки. https://ru.wikipedia.org/wiki/Интегрированная_среда_разработки
[7] Фаронов В. В. Turbo Pascal. Наиболее полное руководство. // BHV-Санкт-Петербург. 2007. — 1054 с. ISBN 5-94157-295-6, CD.
[8] Бутюгин Д.С., Гурьева Я.Л., Ильин В.П., Перевозкин Д.В., Скопин И.Н. Функциональность и технологии алгебраических решателей в библиотеке Krylov. // Вестник ЮУрГУ, 2013, т. 2, №3. — с. 92 – 105.
[9] Обзор новой версии Google Chrome 26. // URL: http://www.comss.ru/page.php?id=1394
[10] YouTube.// URL: http://www.youtube.com/
[11] Скопин И.Н. Разработка интерфейсов программных систем. // В сб. Системная информатика. Вып. 6/Новосибирск: Наука. Сибирская издательская фирма. 1997. с.34 – 96.
[12] Burbeck S. Applications Programming in Smalltalk-80(TM): How to use Model-View-Controller (MVC). // URL: http://st-www.cs.illinois.edu/users/smarch/st-docs/mvc.html
[13] Ахо А.В., Лам М.С., Сети Р., Ульман Д.Д. Компиляторы. Принципы, технологии и инструментарий. // Вильямс. 2014. —1184 с. ISBN 978-5-8459-1932-8.

[14] Что такое виджеты // URL: http://widgetok.ru/2009/01/what-is-widgets/
[15] Щедровицкий Г. П. Автоматизация проектирования и задачи развития проектировочной деятельности. // Разработка и внедрение автоматизированных систем в проектировании (теория и методология). — М.: Стройиздат, 1975. — С. 3 – 177.
[16] Руководство к своду знаний по управлению проектами (Руководство PMBOK®). — Project Management Institute, Inc, 2008. — 465 с.
[17] Guide to the software engineering body of knowledge: 2004 version," Library of Congress Online Catalog, 2005, http://lccn.loc.gov/2005921729. Retrieved 16 July 2013.

Разработка веб-сервисов для комплексного исследовании окружающей среды на основе открытых стандартов моделирования

Александр Б. Столбов [1]

[1]*Институт динамики систем и теории управления имени В.М. Матросова Сибирского отделения Российской академии наук (ИДСТУ СО РАН), ул. Лермонтова 134, Иркутск, Россия*

Аннотация

Статья посвящена разработке прототипов веб-сервисов для комплексного моделирования окружающей среды, ориентированных на исследования Байкальской природной территории (БПТ). Предлагаемое решение основано на открытых стандартах моделирования (BMI - Basic Model Interface) и форматах данных (NetCDF), интегрированных в геопортал ИДСТУ СО РАН. Описаны ключевые типы сервисов: предобработки данных, конвертации и агрегации данных, а также реализации стандарта BMI. Практическая применимость демонстрируется на примере гидрологической модели MOSART-WM, реализующей сквозной цикл сценарного моделирования от исходных данных до визуализации результатов с использованием цифровой платформы геопортала.

Ключевые слова

Комплексное исследование окружающей среды, открытые стандарты моделирования, веб-сервисы, геопортал

1. Введение

Современные исследования в области комплексного моделирования окружающей среды требуют не только развития математических моделей, но и создания масштабируемой и доступной инфраструктуры, способной объединить многочисленные данные и обеспечить их оперативную обработку. В последние годы наметился устойчивый переход от локальных вычислительных сред к облачным веб-сервисам, предоставляющим доступ к моделям и их компонентам через стандартизированные интерфейсы. Ключевой проблемой практического применения такого подхода остается сложность интеграции разнородных источников данных и моделей. Решение этой проблемы требует создания единой среды, чему посвящены коллективные усилия научного сообщества, в частности развитие такой междисциплинарной дисциплины как комплексного моделирования окружающей среды (Integrated Environmental Modeling – IEM) [1-3].

Суть IEM заключается в создании методологии, требующей интеграции разнородных компонентов, включая математические модели различных природных и антропогенных процессов, пространственные данные, результаты мониторинга и сценарии воздействий для изучения взаимозависимости между экологическими, гидрологическими, атмосферными и социально-экономическими факторами. Конечной целью является получение научно обоснованных прогнозов состояния окружающей среды и оценки последствий природных или антропогенных изменений в зависимости от различных условий.

7th International Workshop on Information, Computation, and Control Systems for Distributed Environments (ICCS-DE 2025), July 7–11, 2025, Irkutsk, Russia
EMAIL: stolboff@icc.ru (A. 1)
ORCID: 0000−0001−6513−7030 (A. 1)
© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).
ICCS-DE 2025 Workshop Proceedings
DOI: 10.47350/ICCS-DE.2025.16

Задача разработки веб-сервисов является естественным развитием результатов, полученных при создании информационной системы о математических моделях окружающей среды, поддерживающих открытые стандарты моделирования – ИС КМОС [4]. Данная статья вносит вклад в это направление в контексте разрабатываемой в рамках гранта Министерства науки и высшего образования РФ цифровой платформы исследования Байкальской природной территории (БПТ).

2. Веб-сервисы для комплексного исследовании окружающей среды

Основная задача разработки веб-сервисов предоставить унифицированный и доступный через веб-интерфейс инструментарий для выполнения полного цикла сценарного анализа, опираясь на открытые стандарты моделирования [5-6] и концепции, заложенные в концептуальной модели комплексного моделирования окружающей среды (КМ КМОС) [4, 7]. Так, элементы КМ КМОС описывают информацию о требованиях к вычислительной инфраструктуре и совместимости типов данных и форматов файлов; специфике применяемых каркасов моделирования; о характеристиках математических уравнений, численных методов, ограничениях на область определения переменных состояния и допустимые диапазоны параметров моделей

В настоящем исследовании основными системами, к изучению которых будут применятся разрабатываемые сервисы, являются объекты Байкальской природной территории, а в качестве целевой информационно-коммуникационной платформы используется геопортал ИДСТУ СО РАН [8]. Таким образом, предлагаемые сервисы для комплексного исследовании окружающей среды фактически будут сервисами геопортала с расширенной функциональностью, обеспечивающейся за счёт возможностей ИС КМОС. В качестве целевого стандарта выбран BMI (Basic Model Interface) – это открытый и унифицированный программный интерфейс, разработанный сообществом CSDMS (Community Surface Dynamics Modeling System) [5], так как по результатам предыдущего анализа [4] является на данный момент наиболее популярным и активно использующимся или планируемым к внедрению в среде специалистов.

Рассмотрим принципиальную схему взаимодействия пользователей с сервисами на примере некоторой пространственно-распределённой модели объекта окружающей среды, представленную на рисунке 1.

Основными источниками данных являются:

- Внешние хранилища включают цифровые модели рельефа (DEM), данные дистанционного зондирования, климатические ряды, карты землепользования и другие геопространственные данные, релевантные для БПТ.
- Эксперт идентифицирует и предоставляет специфические параметры моделирования, например, специалист-гидролог может указать коэффициент Маннинга для различных типов поверхности и каналов, точки стока, параметры водохранилищ и др. Эксперт также оценивает качество исходных и подготовленных данных.

Основными типами сервисов являются:

1. Сервисы предобработки данных получают сырые данные (например, DEM) и выполняют критически важные преобразования. Например, на основе библиотеки pysheds [9] может быть реализована следующая функциональность:

- устранение артефактов-впадин в DEM для обеспечения непрерывного стока;
- обработка плоских участков DEM, добавление минимальных уклонов для определения направления стока;
- расчет направлений стока (формат D8);
- автоматическое или полуавтоматическое (с заданием точки стока экспертом) выделение границ водосборных бассейнов;
- расчет накопленного стока (upstream area);
- извлечение структуры речной сети по заданным пороговым значениям накопленного стока.

2. Сервисы конвертации и агрегации преобразуют обработанные геопространственные данные и другие входные данные в форматы, требуемые конкретной моделью. Например, для

большинства моделей, поддерживающих открытые стандарты моделирования, применяются следующие форматы:

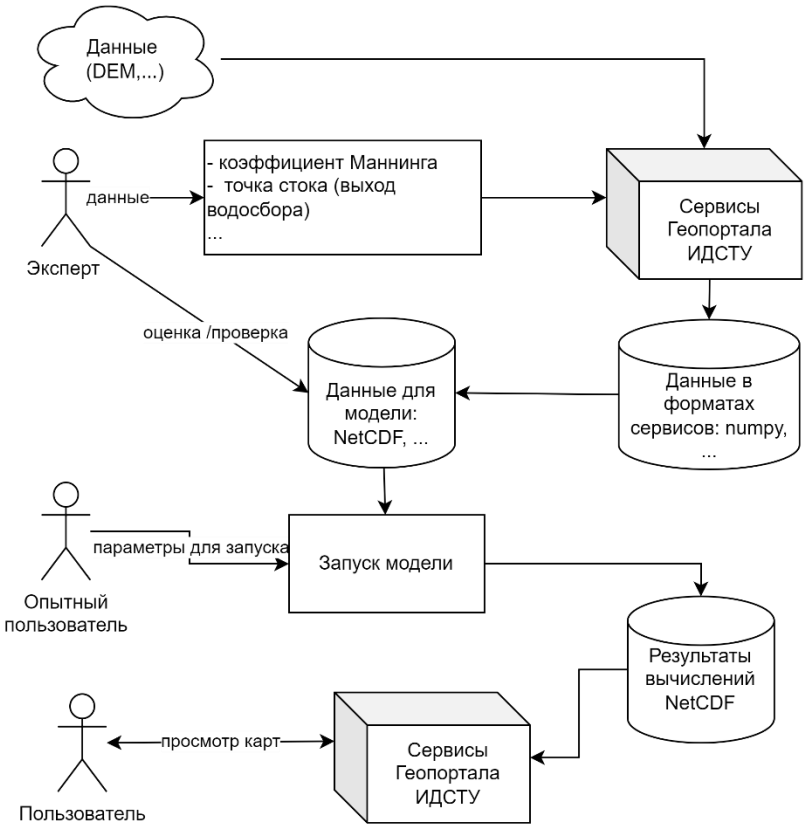


Рисунок 1: Прототип схемы взаимодействия пользователей с сервисами геопортала ИДСТУ СО РАН на примере гидрологической модели.

- NetCDF (Network Common Data Form) – основной целевой формат для описания входных/выходных данных и параметров для вычислительных моделей ІМЕ. При этом сервисы должны учитывать CF-conventions (Climate and Forecast Metadata Conventions): набор рекомендаций и стандартов для оформления метаданных в файлах NetCDF, разработанный для обеспечения совместимости, переносимости и однозначного понимания климатических, гидрологических, океанографических и других геофизических наборов данных.
- Популярные сервисы визуализации используют numpy, GeoTIFF, GeoJSON.

2. Сервисы для реализации открытого стандарта моделирования как минимум должны включать следующие методы:

- Методы для управление процессом вычислений: update(), initialize() и т.п.
- Методы для управление данными (переменными и параметрами) модели: get_start_time(), set_value(…), get_input_var_names() и т.п.
- Методы для управления стекой (каждая переменная модели определяется с привязкой к 1D, 2D или 3D): get_var_location(…), get_var_grid(…) и т.п.

Основными типами пользователей являются:

- Эксперт – участвует на этапе настройки параметров запуска, может выполнять предварительные запуски для оценки адекватности данных и параметров перед основными расчетами.
- Опытный пользователь – задаёт параметры запуска, выбирает подготовленный набор входных данных и запускает модель через сервис.
- Обычный пользователь – просматривает исходные и подготовленные данные на интерактивной карте; запускает предопределенные сценарии визуализации (карты,

временные графики) для результатов моделирования; проводит сравнительный анализ разных сценариев; экспортирует данные и графики.

3. Иллюстративный пример

В качестве иллюстративного примера была выбрана MOSART-WM (Model for Scale Adaptive River Transport – Water Management) [10] вычислительная гидрологическая модель, предназначенная для расчёта речного стока и управления водохранилищами в составе комплексных ESM (Earth System Models). В настоящее время тестирование работы сервисов осуществляется на условных данных. Результаты приведены на рисунках 2 и 3.

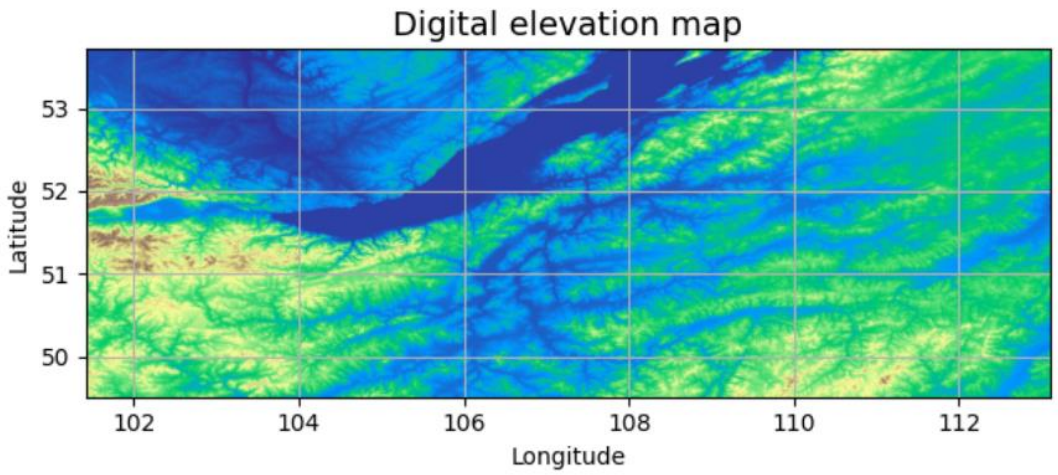


Рисунок 2: Пример применения сервиса для расчёта направления стока (на условных данных)

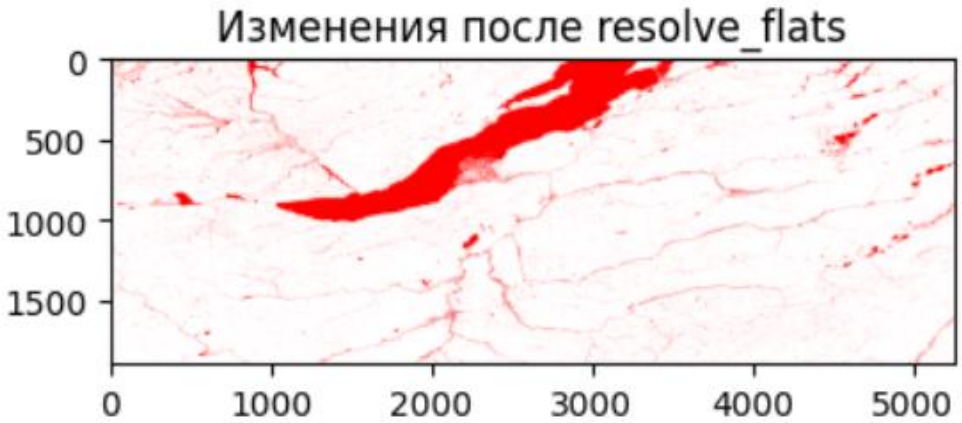


Рисунок 3: Пример применения сервиса для устранения плоских участков (на условных данных)

Модель обладает следующими отличительными особенностями: поддержка BMI API; адаптивный временной шаг (один цикл базовой модели разбивается на 3 шага MOSART; внутри каждого шага расчёта стока со склона выполняется 4 шага вычислений относительно русла; локальное изменение шага по уклону, ширине и длине канала); масштабируемость – работает на разрешениях от 1 км (локальные бассейны) до 100 км (глобальные расчёты). Библиотеки для расчётов по модели были успешно установлены и запущены на цифровой платформе геопортала ИДТСУ СО РАН. При этом в качестве модели рельефа также были использованы данные геопортала.

4. Заключение

В статье представлены первые результаты автора по разработки прототипов веб-сервисов для комплексного моделирования окружающей среды на основе программных библиотек, реализующих открытые стандарты моделирования, в частности на данный момент применялись NetCDF (для хранения и обмена данными) и BMI (для организации вычислений).

Для демонстрации принципиальной возможности применения созданных прототипов была выбрана простая гидрологическая модель MOSART-WM. На её основе был реализован сквозной цикл сценарного моделирования: от исходных данных (DEM) через сервисы предобработки и формирования входных NetCDF файлов до запуска вычислений на цифровой платформе геопортала ИДСТУ СО РАН.

5. Благодарности

Работа выполнена в рамках гранта № 075-15-2024-533 Министерства науки и высшего образования РФ на выполнение крупного научного проекта по приоритетным направлениям научно-технологического развития (проект «Фундаментальные исследования Байкальской природной территории на основе системы взаимосвязанных базовых методов, моделей, нейронных сетей и цифровой платформы экологического мониторинга окружающей среды»).

6. References

[1] R. Argent, An overview of model integration for environmental applications - Components, frameworks and semantics, Environmental Modelling & Software 19 (2004) 219–234. doi: 10.1016/S1364-8152(03)00150-6.

[2] R. Moore, A. Hughes, Integrated environmental modelling: Achieving the vision, Geological Society London Special Publications 408 (2016) 17–34. doi:10.1144/SP408.12.

[3] T. Iwanaga, H.H. Wang, S.H. Hamilton, V. Grimm, T.E. Koralewski, A. Salado, S. Elsawah, S. Razavi, J. Yang, P. Glynn, J. Badham, A. Voinov, M. Chen, W.E. Grant, T.R. Peterson, K. Frank, G. Shenk, C.M. Barton, A.J. Jakeman, J.C. Little, Socio-technical scales in socio-environmental modeling: Managing a system-of-systems modeling approach, Environmental Modelling & Software 135 (2021).

[4] Столбов А.Б. Разработка информационной системы о математических моделях окружающей среды, поддерживающих открытые стандарты моделирования, Материалы VI Международного семинара по информационным, вычислительным и управляющим системам для распределенные сред (ICCS-DE 2024), 1-5 июля, 2024, Иркутск, Россия. Иркутск: Изд-во ИДСТУ СО РАН, 2024. С.34-38.

[5] G.E. Tucker, E.W.H. Hutton, M.D. Piper, B.Campforts, T. Gan, K.R. Barnhart, A.J. Kettner, I. Overeem, S.D. Peckham, L. McCready, J. Syvitski, CSDMS: a community platform for numerical modeling of Earth surface processes, 2022. doi:10.5194/gmd-15-1413-2022.

[6] Q. Harpham, A. Hughes, R.V.Moore, Introductory Overview: The OpenMI 2.0 Standard for Integrating Numerical Models, Environmental Modelling & Software 122 (2019) 1–13. doi:10.1016/j.envsoft.2019.104549.

[7] А.А. Лемперт, А.Б. Столбов, В.В. Козлов, Использование онтологического подхода для интеллектуальной поддержки сценарного моделирования водных объектов Приангарья, Информационные и математические технологии в науке и управлении 19 (2020) 14–24. doi: 10.38028/ESI.2020.19.3.002.

[8] Feoktistov A., Gorsky S., Kostromin R., Fedorov R., Bychkov I. Integration of web processing services with workflow-based scientific applications for solving environmental monitoring problems, ISPRS International Journal of Geo-Information (2022). Vol. 11, № 1 DOI: 10.3390/ijgi11010008

[9] Pysheds – Simple and fast watershed delineation in python, 2025. URL: https://mattbartos.com/pysheds/.

[10] Thurber et al., mosartwmpy: A Python implementation of the MOSART-WM coupled hydrologic routing and water management model, Journal of Open Source Software 6 (2021) 3221. https://doi.org/10.21105/joss.03221

Методика создания композиций сервисов для ботанических исследований

Роман Константинович Фёдоров,[1] Алла Васильевна Верхозина[2]

[1] *Институт динамики систем и теории управления имени В.М. Матросова Сибирского отделения Российской академии наук (Иркутск, Россия), Иркутск, ул. Лермонтова 134, 664033, Россия*
[2] *Сибирский институт физиологии и биохимии растений Сибирского отделения Российской академии наук (Иркутск, Россия), Иркутск, ул. Лермонтова 132, 664033, Россия*

Аннотация

Большое количество сервисов обработки данных уже приводит к сложности поиска релевантных сервисов. При полном соответствии описания сервиса решаемой задаче необходимо проверить совместимость параметров, что значительно замедляет поиск сервиса. Реализация методики для ботанических исследований позволила для предметных специалистов показывать в первую очередь связанные с их данными сервисы. В рамках методики использование меток для создания композиции сервисов объединяет преимущества применения экспертных знаний, онтологических концептов, анализа статистики применения сервисов. Предложенная методика является эвристической, окончательное решение по применению сервиса ложится на пользователя. Тем не менее, реализация модели предметной области и алгоритмы позволяют выявить наиболее релевантные сервисы, распространить успешный опыт применения сервисов среди предметных специалистов и увеличить количество применяемых сервисов.

Ключевые слова

Сервис-ориентированная архитектура, OGC WPS, обработка пространственных данных, автоматизация ботанических исследований

1. Введение

В области ботанических исследований ученые активно ведут работу по сбору данных о распространении различных видов. Накоплены большие массивы данных, в том числе в электронном виде. Для анализа данных ботанических исследований активно растет количество методов анализа в рамках Сервис-ориентированной архитектуры [1]. Например, это сервисы предоставления данных дистанционного зондирования земли, обработки пространственных данных и т. д. Для обработки пространственных данных разработан и активно используется стандарт OGC WPS [2], унифицирующих работу с сервисами. Созданные сервисы значительно упрощают решение многих задач. Однако созданные сервисы достаточно редко применяются по следующим причинам:

- большое количество сервисов, среди которых необходимо выявить сервисы для проведения ботанических исследований часто при отсутствии или частичном наличии метаданных;
- необходимость проверки совместимости параметров сервисов;
- отсутствие необходимой квалификации в информационных технологиях.

Поиск релевантных предметной области сервисов и упрощение их применения является актуальной задачей, позволяющей повысить эффективность процессов подготовки и

7th International Workshop on Information, Computation, and Control Systems for Distributed Environments (ICCS-DE 2025), July 7–11, 2025, Irkutsk, Russia
EMAIL: fedorov@icc.ru (A. 1); allaverh@list.ru (A. 2)
ORCID: 0000-0002-2944-7522 (A. 1); 0000-0002-0872-4455 (A. 2)
© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).
ICCS-DE 2025 Workshop Proceedings
DOI: 10.47350/ICCS-DE.2025.17

проведения научных экспериментов по решению задач в области ботанических исследований за счет автоматизации построения и применения композиций сервисов [3].

В рамках Сервис-ориентированной информационно-аналитической среды обработки пространственных данных (СОИАС) разработана и апробирована методика создания композиций для ботанических исследований. Данная методика основывается на методе создания композиций сервисов [4], который проводит комплексный анализ метаданных, онтологий, экспертных знаний и статистики применения сервисов. Далее опишем основные этапы методики.

2. Построение модели предметной области

На первом этапе производится построение модели предметной области, которое включает:

- определение основных моделей данных и создание сервисов данных;
- разработка сервисов обработки данных;
- описание онтологической модели и формирование экспертных знаний.

Построение модели предметной области позволяет создать основу для формирования композиций сервисов, предложит пользователю первоначальное множество возможных композиций пользователю.

2.1. Создание сервисов данных

При проведении ботанических исследований основными данными являются гербарные листы, которые определены стандартами TDWG Darwin Core и Access to Biological Collections Data (ABCD) [5] и состоят из следующих полей: название вида, местонахождение, местообитание, даты сбора, определения, коллектор, автор определения, гербарная коллекция, координаты, число хромосом, полевой номер, примечания и т. д. Для интеграции данных разных коллективов создана модель данных «Core collection», с набором полей TDWG и ABCD. С помощью Фабрики сервисов ввода и редактирования реляционных данных на основе предложенной модели созданы сервисы данных (Рисунок 1), которые позволили коллективам организовать распределенный сбор данных. Для каждого сервиса данных определены пользователи и их права.

Большинство исследователей дополнительно к этому набору полей собирают специфичные данные, необходимые для проведения их исследований. При добавлении новых полей сформированы пользовательские дочерние модели.

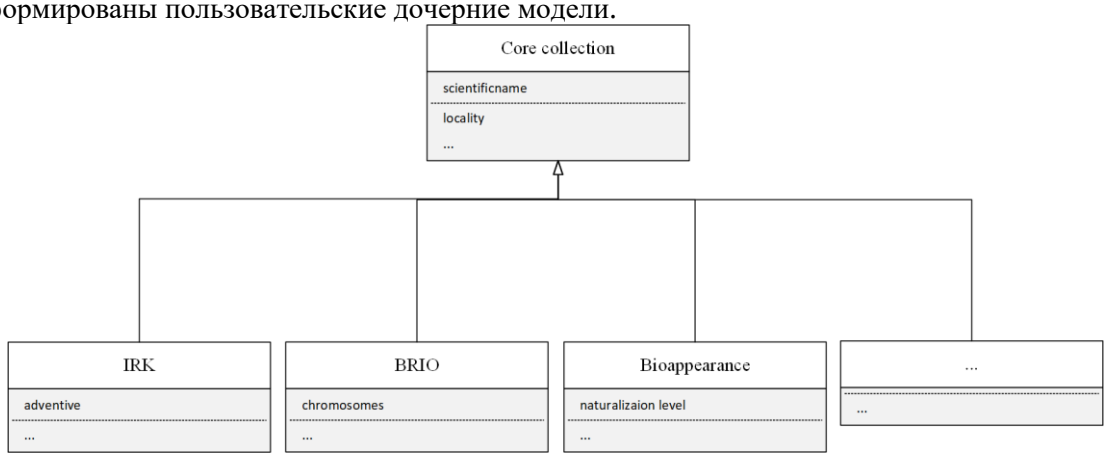


Рисунок 1: Иерархия таблиц, созданных на основе «Core collection»

Использование общей модели позволяет быстро создать новые таблицы, учитывающие пожелания исследователей, унифицировать структуру собираемых данных, а затем объединить данные и проводить совместный анализ. С помощью Фабрики сервисов ввода и редактирования реляционных данных созданы следующие сервисы:

1. БД «Гербарий сосудистых растений Сибирского института физиологии и биохимии растений СО РАН» (IRK);
2. БД «Гербарий мхов Сибирского института физиологии и биохимии растений СО РАН» (IRK);
3. БД «Адвентивные виды сосудистых растений Южной Сибири»;
4. БД «Первые находки чужеродных видов растений на территории Байкальской Сибири (Хронология, география)»;
5. БД «Флора сосудистых растений Заповедного Прибайкалья»;
6. БД «Рыбы»;
7. БД «Лишайники»;
8. БД «Печеночные»;
9. БД «Мохообразные»;
10. БД «Грибы» и т. д.

2.2. Описание онтологической модели и формирование экспертных знаний

Для всех сервисов данных, созданных на основе модели Core Collection, указана метка http://rs.tdwg.org/ontology/voc/Collection. Все метки базового сервиса автоматически указываются в метаданных дочерних сервисов. В качестве экспертных знаний для таблиц определены следующие метки:

1) geoportal:mark:relationaltable – данная метка присваивается всем сервисам, созданных с помощью Фабрики сервисов ввода и редактирования реляционных данных, а также входному параметру сервисов table2json, table2csv и table2xls. Применение этой метки позволяет предлагать пользователям воспользоваться одним из способов сохранения данных таблицы.

2) geoportal:mark:geometryrelationaltable – данная метка присваивается всем сервисам, созданным с помощью Фабрики сервисов ввода и редактирования реляционных данных и содержащим атрибут с типом GEOMETRY. Данная метка указана у входного параметра сервиса table2shp. Данные таблицы могут быть переданы сервису table2shp, который сохраняет пространственные данные в формате SHAPE.

2.3. Разработка сервисов обработки данных

Реализованы сервисы конвертации данных: table2shp – сохраняет данные в формате SHAPE; table2json – сохраняет данные в формате GeoJSON, который используется для композиций сервисов, заданных с помощью JavaScript; table2csv – конвертирует данные в текстовый формат Comma-Separated Values (CSV); table2xls – используется для экспорта данных в формат Excel с указанием необходимых типов данных.

Пространственный анализ распространения видов является наиболее часто используемым для проведения ботанических исследований. Для проведения пространственного анализа на основе библиотеки GDAL [6] разработаны сервисы операций алгебры GRID над файлами в формате GeoTIFF: raster_sum, raster_subtraction, raster_division и raster_multiplication. Для растеризации точечных векторных данных разработан сервис vector2grid, который принимает на вход данные в формате SHAPE и производит конвертацию данных в растровый формат GeoTIFF. Данный сервис используется, например, для подсчета числа видов в ячейке, анализа конфигурации и плотности ареала видов.

Разработан сервис relief_get_points, который производит обработку данных рельефа. Применяется сервис для вычисления уклона (анг – slope) и экспозиции (анг – aspect), а также для определения высоты точки.

Разработан сервис diagram для отображения диаграмм и графиков, параметрами которого являются данные и модель отображения. Разделение данных и модели отображение позволяет автоматически создавать диаграммы и графики на основе заданной модели для разных данных.

3. Композиции сервисов

На основе построенной модели предметной области создаются композиции сервисов в автоматическом и ручном режиме.

3.1. Композиция сервисов для построения ареала распространения

При изучении того или иного вида растений является важным исследование ареала его распространения. Одним из способов анализа пространственного распределения объектов является построение карты плотности (тепловая карта). С помощью сервисов данных специалистами собирается информация о находках различных видов растений в реляционные таблицы. Для создания карты плотности распространения вида в рамках методики пользователь использовал следующие сервисы:

1) table – сервис данных, в котором можно настроить фильтрацию для получения данных одного вида и на определенную территорию;
2) table2shp – сохраняет отфильтрованные данные в векторный формат SHAPE;
3) vector2grid – выполняет подсчет количества растений в ячейках регулярной сетки, который сохранен в формате GeoTIFF;
4) Map – создает карту ареала распространения вида.

Информация об использовании сервисов сохранена в таблицу Log. Далее произведен анализ таблицы Log. На основе анализа автоматически создана композиция сервисов (Рисунок 2).

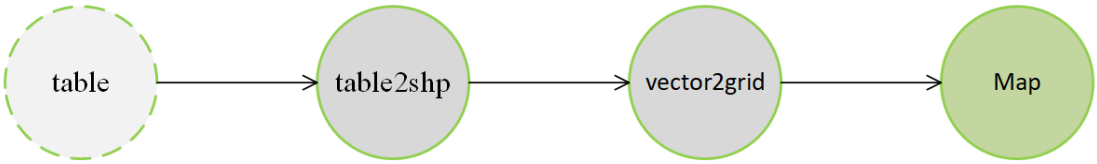


Рисунок 2: Композиция сервисов для построения ареала распространения

3.2. Композиция сервисов прогнозирования изменения флористического состава Байкальского региона

Разработана композиция сервисов анализа распространения вида. В композиции (Рисунок 3) реализованы следующие шаги:

1) формирование карты флористической изученности;
2) формирование регулярной сетки, где в каждой ячейке ставится флаг наличия исследуемого вида;
3) для каждой ячейки регулярной сетки производится операция логического «и» с соответствующей ячейкой карты флористической изученности. В результате получаем карту с ячейками, где в ячейки установлен флаг при наличии вида и флористической изученности;
4) производится подсчет ячеек с установленным флагом и вычисление отношения с количеством ячеек, флористически изученных.

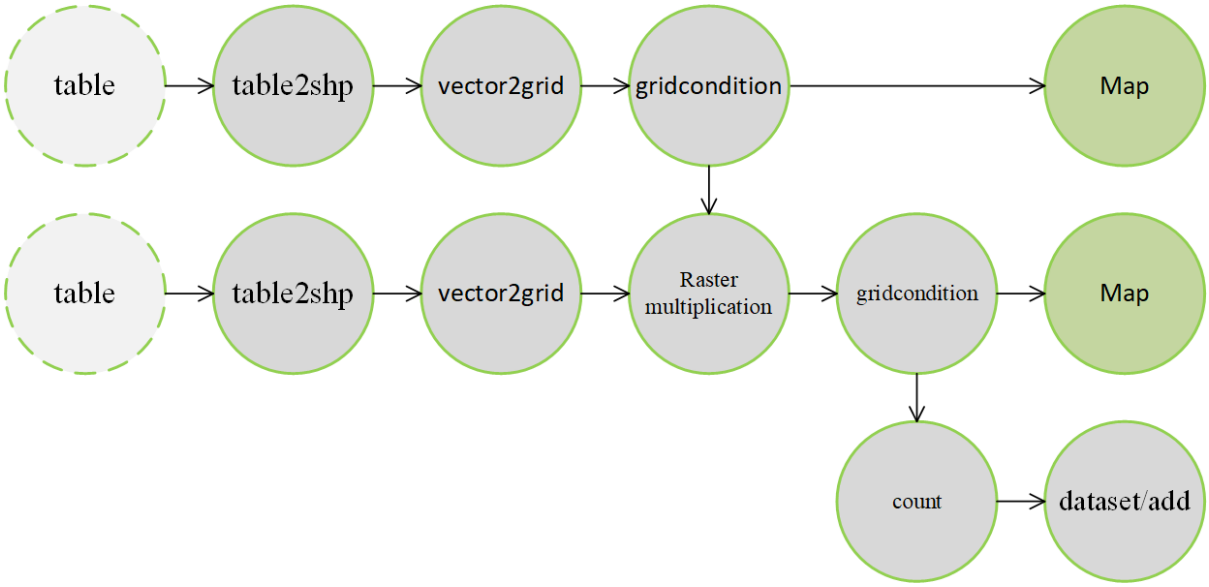


Рисунок 3: Композиция сервисов анализа распространения вида

Данный сервис вычисляет, на какую часть территории распространился вид, при этом рассматривая только изученные фрагменты территории. При выполнении данной композиции пользователю необходимо выбрать композицию сервисов и указать таблицу.

3.3. Композиция сервисов для расчета гистограммы высот

Для любых табличных сервисов данных, содержащих атрибут типа Point (координаты точечного объекта) создана композиция сервисов для отображения графика распределения высот (Рисунок 4). Для решения этой задачи используется WPS сервис relief_get_points, который для точек, заданных в формате WKT, выдает высоту над уровнем моря в метрах. Создание графиков производится сервисом diagram, в котором входными данными являются модель отображения и данные в формате GeoJSON. На выходе сервиса генерируется ссылка, по которой можно посмотреть график (Рисунок 5).

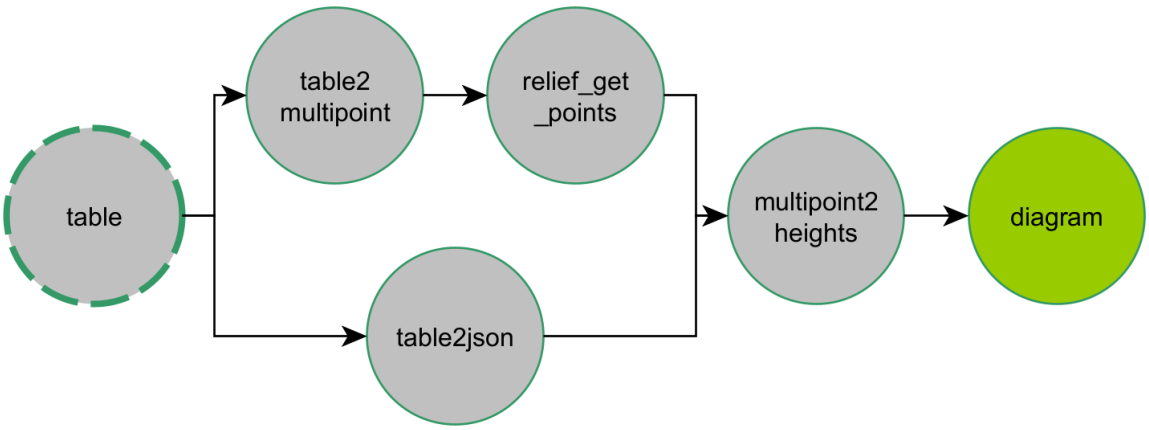


Рисунок 4: Композиция сервисов для расчета гистограммы высот

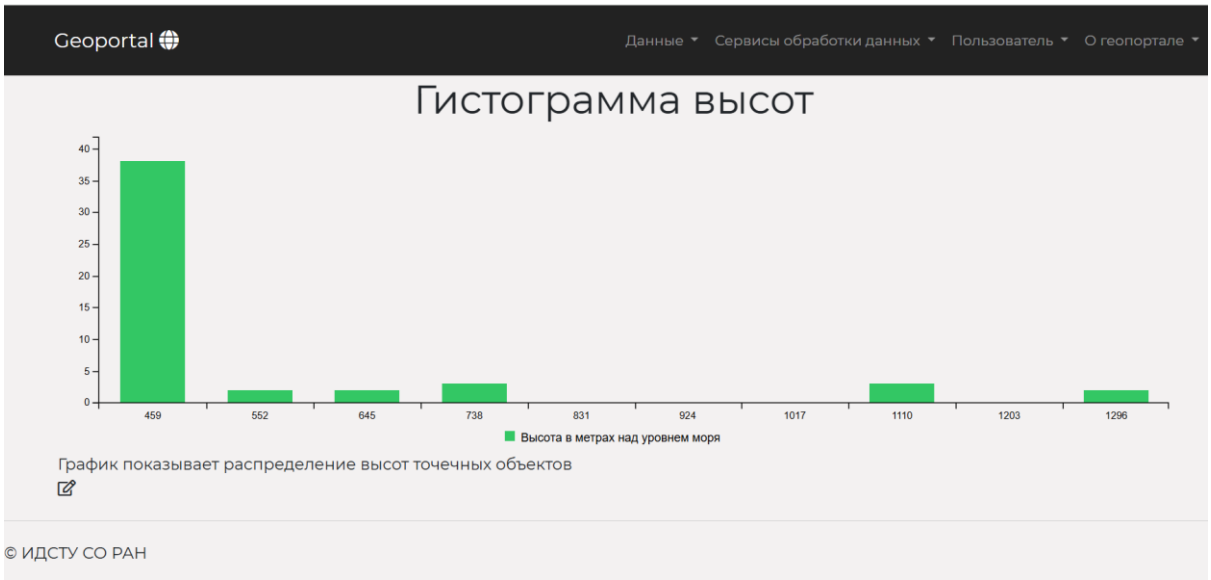


Рисунок 5: Гистограмма для вида Waldsteinia ternata (Stephan)

4. Автоматическое связывание сервисов

Для автоматического связывания сервисов в композиции используется механизм меток. Метка – это строковое уникальное значение. Она используется для идентификации возможности передачи данных между сервисами, т. е. если выходной параметр одного сервиса и входной параметр другого сервиса имеет одну и ту же метку, то между ними возможна передача данных. Каждому параметру сервиса может соответствовать несколько меток. Метками могут являться заданные пользователем значения, автоматически сгенерированные значения и онтологические концепты, заданные с помощью URI. Пользовательский интерфейс геопортала для сервисов данных производит поиск среди входных параметров сервисов и композиций, у которых имеются те же метки сервиса данных. Найденные сервисы предлагаются пользователю для применения. Приведем несколько примеров.

Композиция за счет экспертных знаний. Пользователь может явно указать собственные метки для связывания сервисов. Композиция сервисов для расчета гистограммы высот может применяться для всех табличных данных имеющих атрибут, задающий точку на карте в формате WKT. Чтобы данная композиция предлагалась всем пользователям таких сервисов данных, в параметре сервиса table указана метка «geoportal:mark:geometryrelationaltable». Данная метка имеется у всех сервисов данных, имеющих атрибут точка. В результате композиция сервисов для расчета гистограммы высот будет предлагаться пользователю для всех таких сервисов данных.

Запоминание применения сервисов. Успешное выполнение сервиса с использованием результатов другого сервиса запоминается системой, генерируется уникальная метка, которая присваивается обоим сервисам. Это позволяет успешную композицию пары сервисов одного пользователя автоматически предлагать другим пользователям.

Композиция сервисов с общей моделью данных. Создание сервисов данных производится на основе модели данных. Например, для ботанических исследований часто используется модель «Core Collection». Для всех сервисов данных, созданных с помощью этой модели, указана метка «http://rs.tdwg.org/ontology/voc/Collection». Применение композиции сервисов построения ареала распространения для сервиса данных с данной меткой привело к тому, что данная метка записана в метаданных композиции. После этого данная композиция предлагается всем пользователям, работающим с данными на основе модели «Core Collection». Результат работы композиции представлен на Рисунке 6.

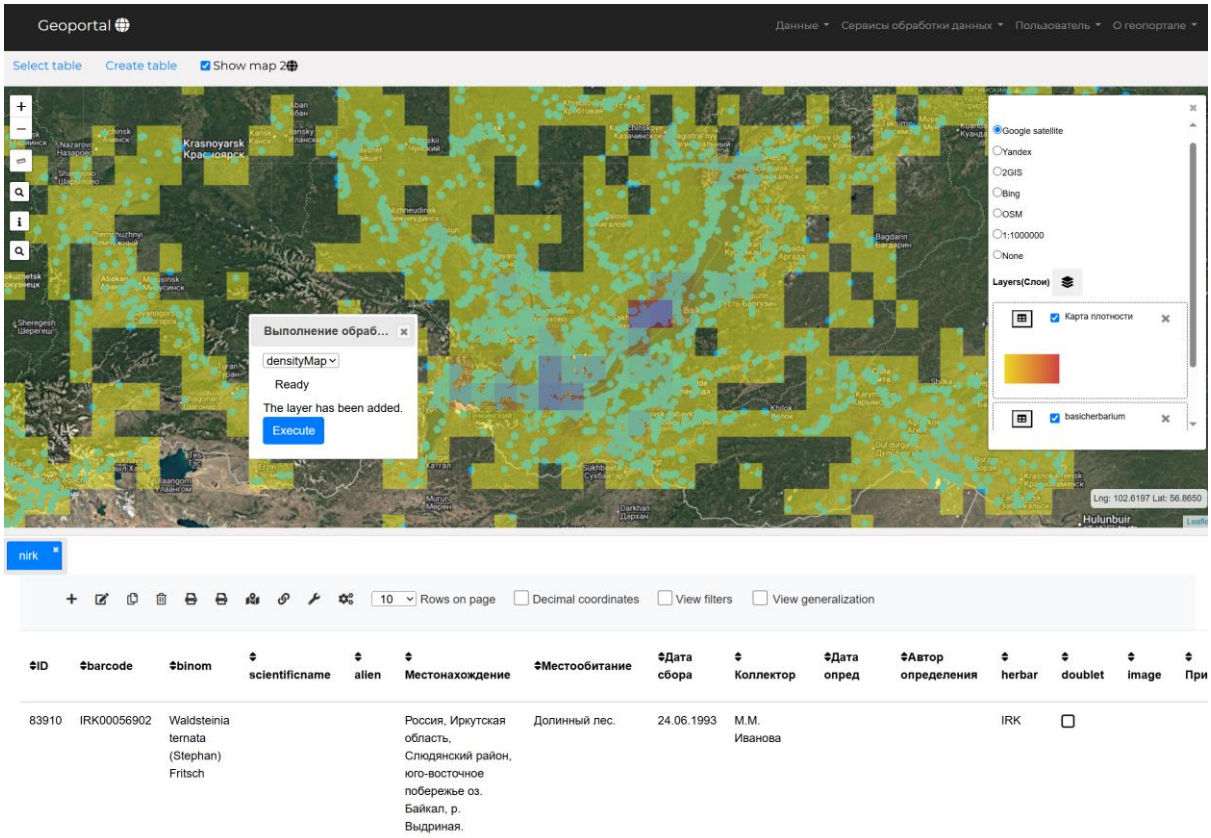


Рисунок 6: Создание карты ареала распространения видов

5. Заключение

Большое количество сервисов уже приводит к сложности поиска релевантных сервисов. При полном соответствии описания сервиса решаемой задаче необходимо проверить совместимость параметров, что значительно замедляет поиск сервиса. Реализация методики для ботанических исследований позволила для предметных специалистов показывать в первую очередь связанные с их данными сервисы. В рамках методики использование меток для создания композиции сервисов объединяет преимущества применения экспертных знаний, онтологических концептов, анализа статистики применения сервисов. Предложенная методика является эвристической, окончательное решение по применению сервиса ложится на пользователя. Тем не менее, реализация модели предметной области и алгоритмы позволяют выявить наиболее релевантные сервисы, распространить успешный опыт применения сервисов среди предметных специалистов и увеличить количество применяемых сервисов.

6. Благодарности

Работа выполнена в рамках гранта № 075-15-2024-533 Министерства науки и высшего образования РФ на выполнение крупного научного проекта по приоритетным направлениям научно-технологического развития (проект «Фундаментальные исследования Байкальской природной территории на основе системы взаимосвязанных базовых методов, моделей, нейронных сетей и цифровой платформы экологического мониторинга окружающей среды»).

7. Литература

[1] Bih J. Service oriented architecture (SOA) a new paradigm to implement dynamic e-business solutions // Ubiquity. 2006. Vol. 4. P. 1–17. DOI: 10.1145/1162511.1159403
[2] Schut P. OpenGIS ® Web Processing Service // Open Geospatial Consortium. 2007. № 6. P. 1-3.
[3] Edmond D., ter Hofstede A. Service composition for electronic commerce // Proceedings of the Pacific Asia Conference on Information Systems (PACIS-2000). Hong Kong, 2000.
[4] Фёдоров Р.К. Метод композиции сервисов обработки пространственных данных // Матер. VI междунар. семинара по информационным, вычислительным и управляющим системам для распределенных сред (ICCS-DE 2024). 2024. С. 160-168. DOI: 10.47350/ICCS-DE.2024.25.
[5] The Access to Biological Collections Data (ABCD) Schema, 2025. URL: https://www.dcc.ac.uk/resources/metadata-standards/abcd-access-biological-collection-data.
[6] GDAL, 2025 URL: https://gdal.org/.

Разработка сервис-ориентированной среды моделирования природно-технических систем

Александр Феоктистов[1], Михаил Воскобойников[1], Игорь Бычков[1], Алексей Еделев[1,2]

[1] *Институт динамики систем и теории управления им. В.М. Матросова СО РАН, ул. Лермонтова, д. 134, 664033, Иркутск, Россия*
[2] *Институт систем энергетики им. Л.А. Мелентьева СО РАН, ул. Лермонтова, д. 130, 664033, Иркутск, Россия*

Аннотация

Предложен новый подход к созданию сервис-ориентированной вычислительной среды для моделирования и оценки функционирования и развития природно-технических систем с использованием высокопроизводительных вычислительных систем. Представлена схема прогнозирования развития и оценки состояния локального энергетического комплекса. Апробация предложенного подхода выполнена на примере решения задачи комплексной оценки критичности элементов транспорта локальной системы газоснабжения.

Ключевые слова

Природно-технические системы, моделирование, сервисы, приложения

1. Введение

В настоящее время обеспечение поддержки решения больших ресурсоемких задач исследования функционирования и развития природно-технических систем (ПТС) в высокопроизводительной вычислительной среде специалистами разных предметных областей является актуальной проблемой. В частности, важным классом ПТС являются энергетические комплексы разных уровней их структурно-отраслевой иерархии, т. к. они относятся к критически важным инфраструктурам. Крупные внешние возмущения могут вызывать отказы элементов энергетических комплексов, которые в свою очередь могут приводить с существенным недопоставкам энергии потребителям и негативным воздействиям на окружающую среду.

В этой связи в докладе предлагается подход к разработке и применению сервис-ориентированной среды для исследования функционирования и развития локальных энергетических комплексов, располагающихся на Байкальской природной территории (БПТ). В рамках предложенного подхода особое внимание уделяется созданию средств автоматизации разработки и применения сервис-ориентированных приложений для моделирования, структурно-параметрической оптимизации, экологической и технико-экономической оценки локальных энергетических комплексов.

2. Методы и средства

В рамках исследования энергетический комплекс представлен его физической моделью – стендом имитационного моделирования [1]. Стенд моделирует наиболее важные процессы работы комплекса в условиях возможных внешних воздействий из окружающей среды.

7th International Workshop on Information, Computation, and Control Systems for Distributed Environments (ICCS-DE 2025), July 7–11, 2025, Irkutsk, Russia
EMAIL: agf@icc.ru (A. 1); mikev1988@mail.ru (A. 2), bychkov@icc.ru (A. 3), flower@isem.irk.ru (A. 4)
ORCID: 0000-0002-9127-6162 (A. 1); 0000-0003-3034-4907 (A. 2); 0000-0002-1765-0769 (A. 3); 0000-0003-2219-9754 (A. 4)
© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).
ICCS-DE 2025 Workshop Proceedings
DOI: 10.47350/ICCS-DE.2025.17

Модели и методы анализа, прогнозирования развития и оценки состояния локальных энергетических комплексов реализуются в виде сервис-ориентированных приложений. Приложения выполняются в ГРВС, которая может интегрировать суперкомпьютерные ресурсы центров коллективного пользования, ресурсы Grid-систем, ресурсы облачных и туманных платформ в процессе решения больших научных и прикладных задач.

Результаты моделирования предоставляются экспертам, которые проводят их анализ с помощью системы GeoARM и средств геопортала ИДСТУ СО РАН [2]. На основе проведенного анализа вырабатываются рекомендации по формированию мероприятий по развитию исследуемых комплексов.

Данные о состоянии и процессах функционирования стенда и ГРВС предоставляются специализированными системами мониторинга. Эти системы позволяют также вносить необходимые воздействия по управлению процессами функционирования стенда и ГРВС.

Разработка специализированных API систем мониторинга, сервисов и приложений, а также организация ГРВС осуществляется с помощью инструментального комплекса FDE-SWFs [3], который относится к классу систем управления научными рабочими процессами (НРП). Валидация данных и верификация прикладных модулей и сервисов осуществляется на испытательных стендах, которые также разрабатываются средствами FDE-SWFs в виде системных НРП. В отличие от известных инструментов подобного назначения, FDE-SWFs обеспечивает поддержку ряда стандартов разработки и применения НРП. Он позволяет представлять НРП на языке Business Process Execution Language (BPEL) и поддерживает стандарт Web Processing Service (WPS) от консорциума Open Geospatial Consortium. Поддержка WPS-сервисов особо важна для решения задач экологического мониторинга природных территорий, чрезвычайно актуальных в настоящее время.

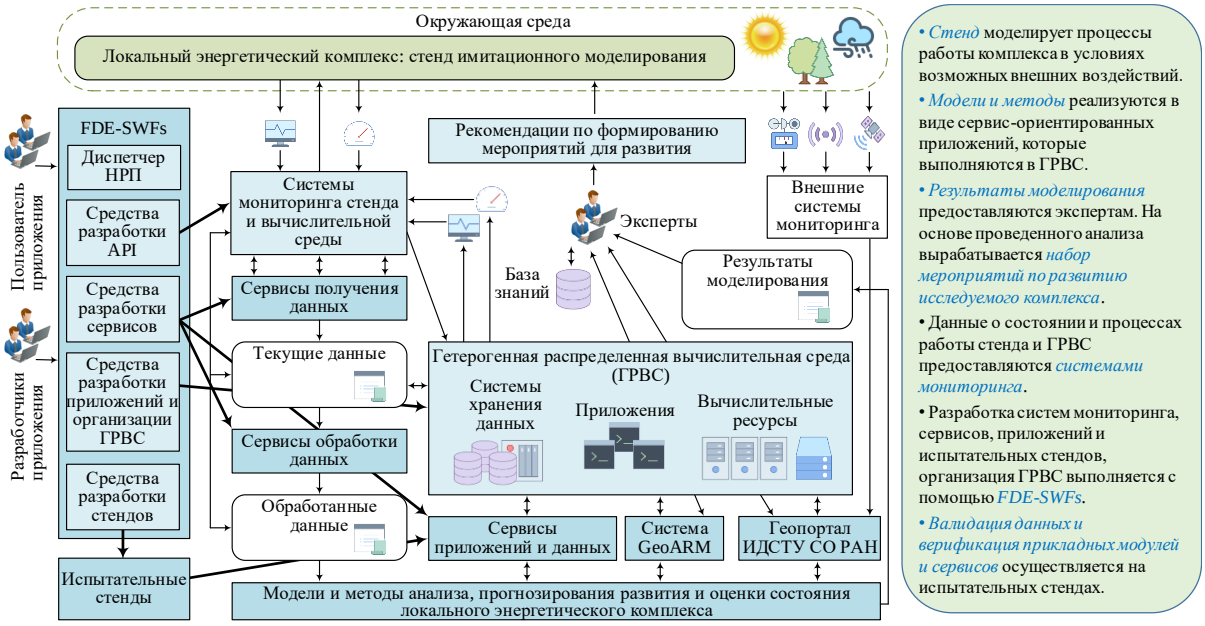


Рисунок 1: Общая схема прогнозирования развития и оценки состояния локального энергетического комплекса

3. Экспериментальные исследования

В качестве примера рассмотрим сервис-ориентированную среду (рис. 2) для комплексной оценки критичности элементов транспорта системы газоснабжения микросети (рис. 3) с точек зрения его топологии и функциональности. Конфигурация системы описывается ориентированным графом $G = (V, E)$, где V – множество узлов (источников и потребителей газа), E – множество дуг (объектов транспорта газа). На рис. 2 источники и потребители газа показаны соответственно овалами и прямоугольниками. Новизна постановки задачи

заключается в том, что критичность оценивается как по отношению к системе в целом, так и относительно отдельных различных групп потребителей. Различные методы оценки критичности элементов системы реализованы в виде сервис-ориентированных приложений. Рабочие процессы выполнены под управлением диспетчера НРП FDE-SWFs, взаимодействующего с системой управления прохождением заданиями HTCondor. Задача решена применительно к классификации дуг исследуемой системы относительно их критичности.

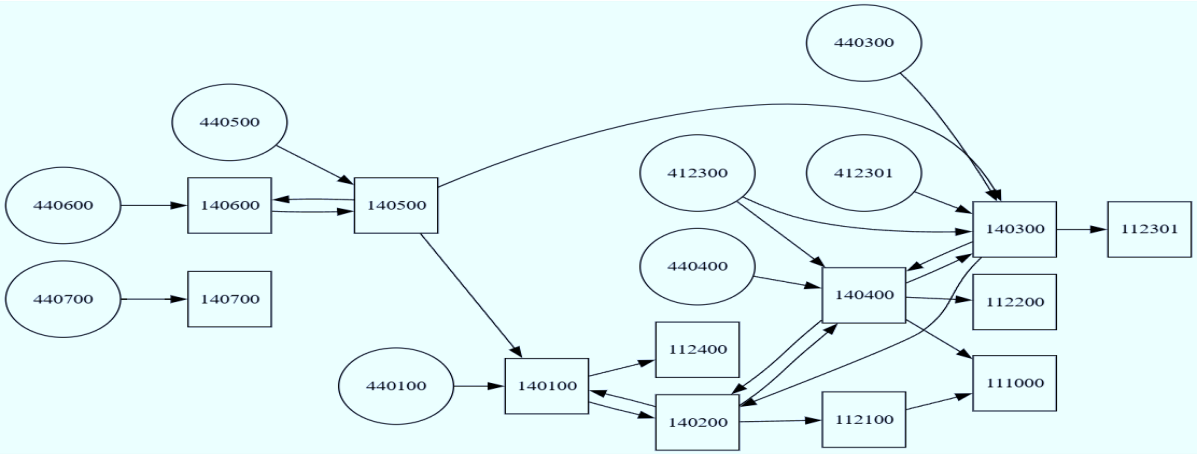


Рисунок 2: Газораспределительная сеть

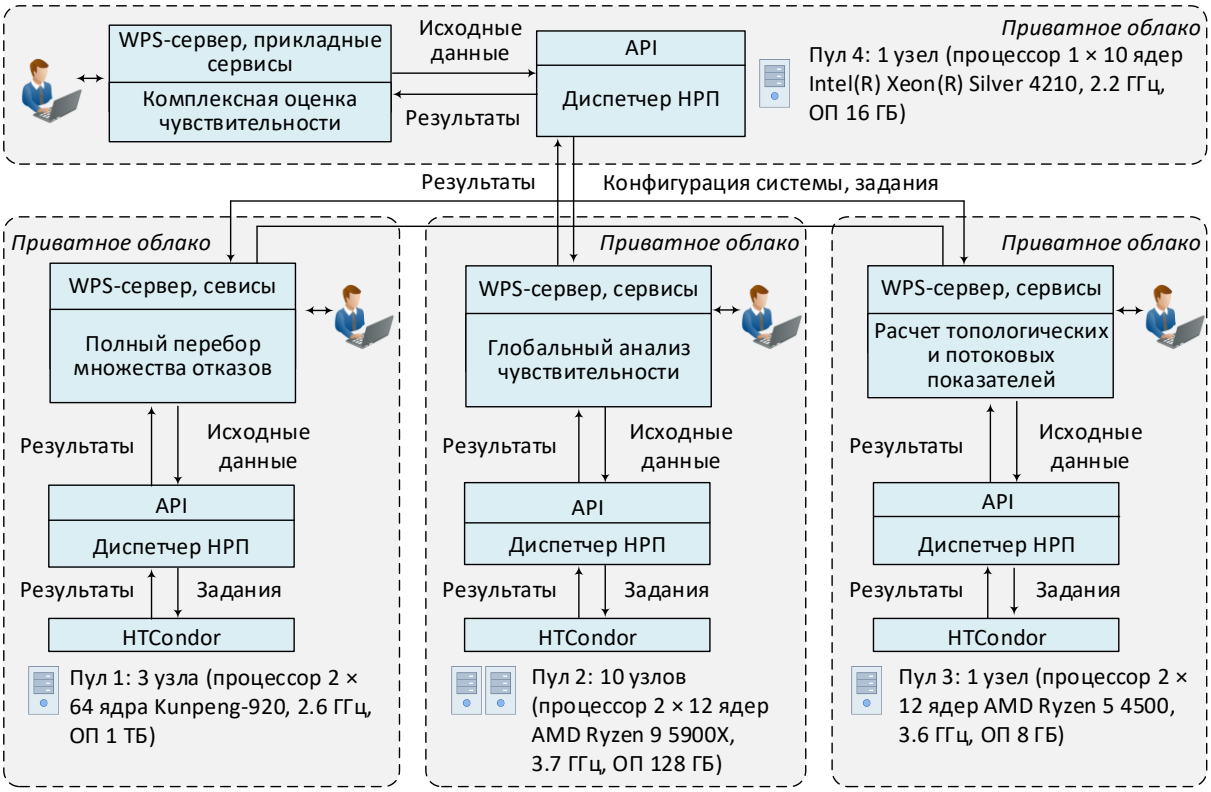


Рисунок 3: Сервис-ориентированная среда

Приложения, реализующие разные подходы к оценке критичности элементов выполнены на вычислительных ресурсах, наиболее подходящих с точки зрения вычислительной сложности решаемых задач. Результаты экспериментов приведены в таблице 1. Для удобства восприятия все значения показателей критичности масштабированы с коэффициентом 1000, наибольшие значения в каждом столбце выделены жирным шрифтом, а максимальные значения выделены цветом. Анализ значений позволяет выделить две группы уязвимых элементов и одну группу

узких мест. Первая группа уязвимых элементов (первые две строки таблицы) имеет наивысшие значения перечисленных показателей. За ней следует вторая по важности группа уязвимых элементов (строки таблицы с третьей по пятую). Узких места системы представлены в таблице последними тремя строками. Они являются резервными путями транспорта газа в случае отказа уязвимых элементов из первой группы.

Таблица 1. Результаты сравнительного анализа

Объект	p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8	p_9	p_{10}	p_{11}	p_{12}	p_{13}
440500-140500	**331**	**682**	**1054**	**747**	**872**	**524**	29	5	**323**	5	**4**	32	0
140500-140300	**207**	**438**	**692**	**63**	**149**	**188**	47	3	**409**	3	**6**	**70**	**0.010**
112100-111000	**106**	**235**	**390**	2	10	79	37	13	**169**	13	3	7	0.004
140200-112100	**135**	**290**	**468**	0	16	98	**68**	11	**312**	11	**8**	**42**	0.004
140300-140200	**135**	**298**	**489**	5	**36**	**108**	**92**	8	**401**	8	**9**	**53**	0.006
140400-112200	0	43	127	0	1	74	34	**34**	1	**34**	0	38	0
140400-140300	0	43	127	0	3	79	50	**37**	15	**39**	1	29	0.005
140600-140500	0	43	127	1	0	84	53	**47**	2	**47**	0	**58**	0

Показатели критичности элементов интерпретируются следующим образом: p_1 – оценка вклада элемента в суммарные последствия от реализации набора множеств отказов F размера k; p_2 и p_3 – оценки вклада этого элемента во множества отказов размерности 2 и 3; p_4 и p_5 – оценки по методу Соболя влияния изменения производственной мощности элемента на системную производительность; p_6 – индекс чувствительности по методу Delta; p_7 – число проходов максимального потока по дуге сети; p_8 – оценка числа вхождений дуги в минимальные разрезы сети; p_9 – вклад дуги в максимальный поток по сети; p_{10} – потенциальное узкое место (отношение потока по дуге к ее пропускной способности; p_{11} – показатель, интегрирующий p_9 и p_{10}; p_{12} – центральность дуг по посредничеству; p_{13} – степень падения топологической эффективности сети вследствие удаления дуги.

Результаты проведенной комплексной оценки критичности объектов транспорта исследуемой системы газоснабжения послужили базисом для выделения значимых для живучести исследуемой системы элементы. Классификация элементов показала, что ключевые объекты транспорта, входящие в группу уязвимых элементов, одновременно могут быть зависимыми элементами. Кроме того, группа уязвимых элементов не пересекается с группой «узких» мест, которые обеспечивают резервные пути поставки природного газа потребителям в случае отказа ключевых уязвимых элементов. Результаты экспериментов показали, что разработанная среда позволяет за приемлемое время производить комплексную оценку критичности элементов не только по отношению к производительности системы в целом, но и также для отдельных групп потребителей.

4. Заключение

Проведенное исследование показало, что разработка сервис-ориентированных приложений, основанных на научных рабочих процессах, является перспективной парадигмой для решения широкого спектра фундаментальных и прикладных задач в различных областях знаний. Использование высокопроизводительных вычислительных систем обеспечивает высокую эффективность процесса решения задач. Применение инструментального комплекса FDE-SWFs обеспечило гибкость в разработке и применении сервис-ориентированных приложений на основе научных рабочих процессов.

Эффективность разработки и применения сервис-ориентированных приложений продемонстрирована на примере решения задач оценки критичности объектов транспорта локальной системы газоснабжения. Разработанная вычислительная среда позволила за приемлемое время производить комплексную оценку критичности элементов энергетической

инфраструктуры не только по отношению к производительности системы в целом, но и также для отдельных групп потребителей.

Предложенный подход к организации сервис-ориентированной вычислительной среды для моделирования и оценки функционирования и развития локальных энергетических комплексов допускает свое естественное развитие и обобщение применительно к другим классам природно-технических систем.

5. Благодарности

Работа выполнена при поддержке Министерства науки и высшего образования РФ, грант № 075-15-2024-533 на выполнение крупного научного проекта по приоритетным направлениям научно-технологического развития (проект «Фундаментальные исследования Байкальской природной территории на основе системы взаимосвязанных базовых методов, моделей, нейронных сетей и цифровой платформы экологического мониторинга окружающей среды», рег. № 124052100088-3), проект № FWEW-2021-0005 «Технологии разработки и анализа предметно-ориентированных интеллектуальных систем группового управления в недетерминированных распределенных средах» (рег. № 121032400051-9).

6. Список литературы

[1] И. В. Бычков, Г. А. Феоктистов, М. Е. Боднюк, Д. Н. Карамов Система мониторинга физической модели микросети, Автометрия, 2025, т. 61, № 2, с. 5–12. doi:10.15372/AUT20250201.

[2] И. В. Бычков, С. А. Горский, А. В. Еделев, Р. О. Костромин, И. А. Сидоров, А. Г. Феоктистов, Е. С. Фереферов, Р. К. Федоров. Поддержка управления живучестью систем энергетики на основе комбинаторного подхода, Известия РАН. Теория и системы управления, 2021, № 6, с. 122–135. doi:10.31857/S000233882106007X.

[3] A. Feoktistov, M. Voskoboinikov, A. Tchernykh, Framework for Development and Execution of Scientific WorkFlows: Designing Service-Oriented Applications, Programming and Computer Software, 2024, vol. 50, no. 8, pp. 900–913. doi: doi.org/10.1134/S0361768824700828.

Application of information and computing systems in a hybrid power grid

Vladislav Badenko[1], Alexander Kozlov [1]

[1] *ESI SB RAS, 130 Lermontov str., Irkutsk, 664033, Russia*

Abstract

The work focuses on a hybrid energy system containing a biomass-based gas generator and solar panels. To control and optimize such a multi-component system, analysis, monitoring, and modeling tools are needed. A monitoring system is proposed for the hybrid microgrid, which collects information on various characteristics. This system allows data analysis, prediction of system behavior, and decision making for control or modification of the energy system. A mathematical model was proposed for the biomass utilization component of the microgrid, and an original numerical method was applied to obtain the input data of the model. This approach allows to increase the efficiency of power generation in this unit, which directly improves the efficiency of the microgrid as a whole. Thereby, the paper proposes methods and tools aimed at analyzing, modeling and management of a multicomponent hybrid energy system.

Keywords

Hybrid energy system, kinetic analysis, CFD-modeling

1. Introduction

This paper focuses on a power system based on renewable energy sources such as biomass and solar energy. During the implementation of renewable energy sources in existing energy systems or during the creation of new energy systems, the key role is played by the collection and interpretation of information on various system parameters, both during normal operation and during testing [1]. The importance of analytical studies of the operation of power systems with renewable energy sources is due to the structure of these systems: the presence of units with different energy sources, interaction with energy storage systems, and inconsistent generation of some components (e.g., solar energy) [2].

In addition to the collection and analytical processing of data on the system as a whole, the consideration and optimization of individual units is also an important task. Special attention is required for units performing thermochemical conversion of biomass for fuel production and further electricity generation, since their efficiency is determined by the operating conditions, composition and properties of the feedstock used, reactor design and purification systems. Accordingly, there are tasks to improve the efficiency of energy system. When solving these problems, it is relevant to use instrumental complexes to study the properties of feedstock and mathematical modeling to assess the processes occurring in the reactor zone [3].

In this paper, a monitoring system that collects information on various characteristics has been proposed for a hybrid microgrid. This system allows data analysis, prediction of system behavior, and decision making for control or modification of the power system. A mathematical model has been proposed for the microgrid component involved in biomass usage, and an original numerical method has been applied to obtain the input data of the model. This approach enables to increase efficiency in this component, which directly improves the performance of the microgrid as a whole.

7th International Workshop on Information, Computation, and Control Systems for Distributed Environments (ICCS-DE 2025), July 7–11, 2025, Irkutsk, Russia
EMAIL: badenko@isem.irk.ru (A. 1); kozlov@isem.irk.ru (A. 2)
ORCID: 0009-0007-0992-564X (A. 1)
© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).
ICCS-DE 2025 Workshop Proceedings
DOI: 10.47350/ICCS-DE.2025.19

Therefore, the paper proposes methods and tools aimed at analyzing, modeling and controlling a multi-component energy system on renewable energy sources. The schematic diagram of the hybrid microgrid is shown in Fig. 1.

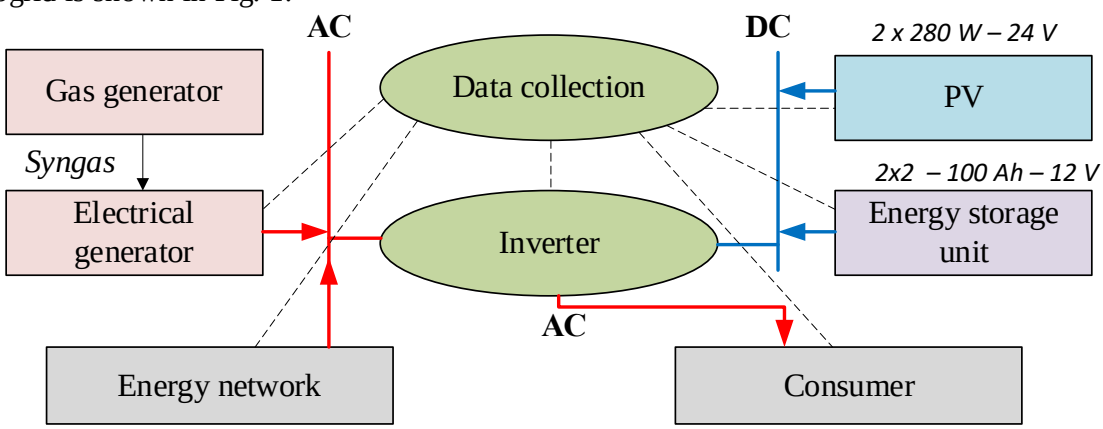


Figure 1: Schematic diagram of a hybrid microgrid

2. Monitoring system

Because the ability to graphically visualize and access key parameters of the hybrid microgrid is of great importance, a monitoring system was created. An example of the hybrid microgrid monitoring system interface is shown in Fig. 2.

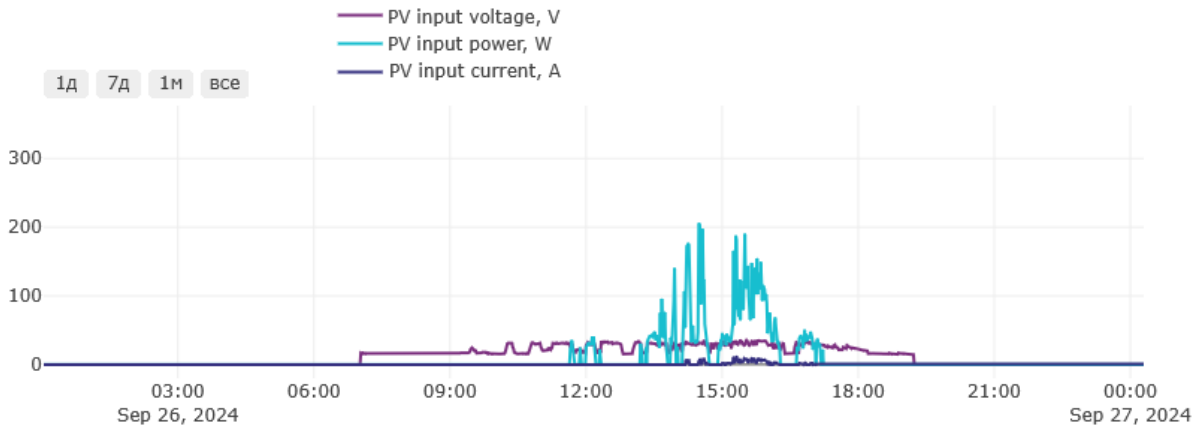


Figure 2: An example of using a monitoring system to study a microgrid. The characteristics of solar cells for a daily period are shown

At the stage of designing the hybrid microgrid, the task of the highest unification of the recorded data to assist in analytical work was established. This task was solved by means of correct selection of a control node in the form of an inverter and its corresponding software, as well as auxiliary subroutines for data processing.

Data on microgrid parameters require processing by the server structure for correct display. Conversions of data formats, writing to arrays and filtering are performed using the JavaScript language. Reflection of the hybrid microgrid parameters is performed by means of a web page implemented in the HTML hypertext markup language with CSS.

When working with a hybrid microgrid, there is an important task of identifying patterns and forming predictive recommendations. Combining monitoring data, environmental data (for systems with photovoltaic cells), and experimental observations into a database allows to solve this task. The collected analytical data, together with subsequent studies, provide a basis for experiments on scaling up, expansion and implementation of the hybrid microgrid.

3. The modeling of the gas generator

In order to perform computational experiments in modeling any object, reliable input data are required. These data can be obtained by using various instrumental complexes. In the context of thermochemical conversion reactors, the input data are properties of feedstock, and as instrumental complexes synchronous thermal analysis complexes and application of standardized methods of technical analysis are used. On the basis of the data characterizing the behavior of the investigated sample under thermochemical conversion conditions, a kinetic analysis can be performed. In this way, the inverse kinetic problem is solved under conditions of a regular heating regime.

A model of computational hydrodynamics of the pyrolysis reactor was created using data from instrumental studies and kinetic analysis. Modeling of pyrolysis specifically is conditioned by the fact that during active release of gaseous conversion products, access of oxidant to fuel particles is difficult [4]. Therefore, pyrolysis data are of great importance and determine the energy efficiency of the microgrid node. The result of the model is shown in Fig. 3.

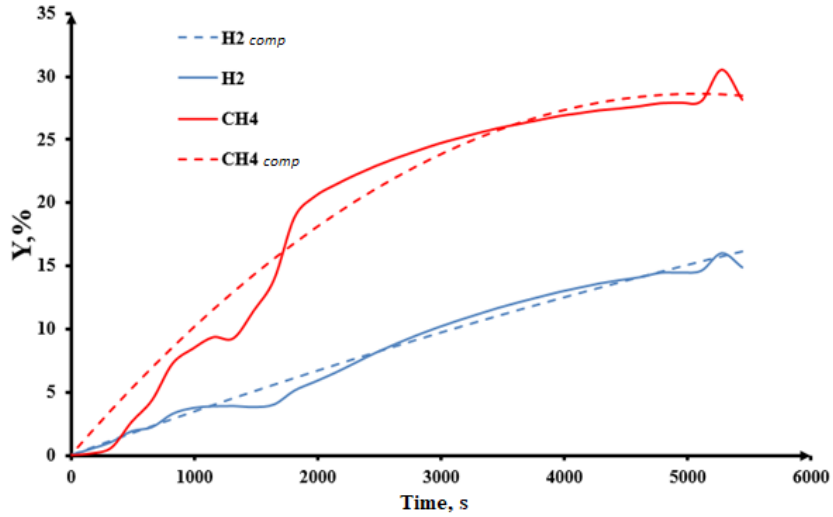


Figure 3: Comparison of modeling results and experimental values on formation of H_2 and CH_4 at biomass pyrolysis (the computed data are labeled as "comp")

4. Acknowledgements

The research was carried out under State Assignment Project (no. FWEU-2021-0005) of the Fundamental Research Program of Russian Federation 2021-2030 using the resources of the High-Temperature Circuit Multi-Access Research Center (Ministry of Science and Higher Education of the Russian Federation, project no 13.CKP.21.0038).

5. References

[1] Ivanov R. Parameters monitoring organization of the solar panels experimental stand, Information and mathematical technologies in science and management 4 (2021) 77-87. doi: 10.38028/ESI.2021.24.4.008

[2] Suslov K.V. Development of power supply systems of Russian isolated territories using renewable energy sources, Proceedings of Irkutsk State Technical University 21 (2017) 131–142. doi: 10.21285/1814-3520-2017-5-131-142

[3] Yu Z. T. Essential aspects of the CFD software modelling of biomass gasification processes in downdraft reactors, RSC advances 14 (2024) 28724-28739. doi: 10.1039/D4RA04886E

[4] Antunes E. et al. A circular economy approach for phosphorus removal using algae biochar, Cleaner and Circular Bioeconomy 1 (2022). doi: 10.1016/j.clcb.2022.100005

On an Improvement to the Next-step Procedure

Nikita Kosyanov [1]

[1] *Institute for System Dynamics and Control Theory of SB RAS, Lermontov str., 134, Irkutsk, 664033, Russia*

Abstract

In this paper, we address a next-step procedure for solving nonsmooth integer programming problems, where the objective function is given as a maximum and the constraints include both equalities and inequalities. We propose two modifications of the original procedure aimed at significantly reducing the number of operations required to obtain a solution.

Keywords

Integer programming, nonsmooth optimization, minimax problem, next-step procedure

1. Introduction

Modern computing is inseparable from parallel processing. It is now virtually impossible to find a computer or smartphone equipped with a single-core processor. Moreover, many devices are equipped with heterogeneous cores—high-performance and energy-efficient—requiring specially designed software to balance the computational load across them.

Load balancing problems of this kind are relevant not only for large-scale computing centers [1], where workloads must be distributed across multiple compute clusters, but also for service systems such as healthcare, where, for example, patients in infectious disease hospitals must be allocated among available physicians [2].

In many cases, such multi-resource systems can be described mathematically, and the task of assigning jobs or requests can be formulated as an optimization problem—typically, one that minimizes the maximum load across all resources. However, these problems often include the constraint that individual tasks (or patients) cannot be divided among multiple resources, which necessitates the development and application of specialized solution methods that explicitly account for such indivisibilities.

2. Next-step procedure

We consider the problem $(\mathcal{TP}_N)$ of performance optimization in the case of parallel processing of N computational scenarios using n processing resources:

$$(\mathcal{TP}_N): f_N(x) = \max\left\{\frac{x_1}{a_1}; \frac{x_2}{a_2}; \dots; \frac{x_n}{a_n}\right\} \downarrow \min_{x \in X_N}, \quad X_N \subset \mathbb{N}_0^n, \tag{1}$$

$$X_N: \sum_{i=1}^{n} x_i = N, \quad 0 \le x_i \le c_i, \quad i = \overline{1,n}. \tag{2}$$

where x_i is the number of scenarios processed by resource i; $a_i \in \mathbb{N}$ denotes the performance (capacity) of resource i; X_N is the available set of values for x, which is a subset of the n-dimensional vector space with non-negative integer coordinates ($\mathbb{N}_0^n$) and satisfies the constraints (3.2); $N \ge 0$ represents the total number of scenarios to be processed; $c_i \ge 0$ is the maximum number of scenarios (in terms of execution time or memory usage) that resource i can handle.

7th International Workshop on Information, Computation, and Control Systems for Distributed Environments (ICCS-DE 2025), July 7–11, 2025, Irkutsk, Russia
EMAIL: kosyanov.nik@gmail.com (A. 1);
ORCID: 0009-0005-3448-5573 (A. 1);
© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).
ICCS-DE 2025 Workshop Proceedings
DOI: 10.47350/ICCS-DE.2025.20

In addition, we define the function $p(x)$:

$$p(x) \triangleq (p_1, p_2, \dots, p_n) = \left(\frac{x_1+1}{a_1}, \frac{x_2+1}{a_2}, \dots, \frac{x_n+1}{a_n}\right), \quad x \in \mathbb{N}_0^n. \tag{3}$$

We are now in a position to describe the solution procedure for problem $(\mathcal{TP}_N)$.

Next-step Procedure (NSP)

Step 0. Set $k := 0$, $x^k := 0$, $p(x^k) := \left(\frac{1}{a_1}, \frac{1}{a_2}, \dots, \frac{1}{a_n}\right)$.

Step 1. IF $k < N$, THEN find $j = \underset{i \in \{1,2,\dots,n\}}{\operatorname{argmin}} \{p_i\}$, where $x_j + 1 \le c_j$, ELSE STOP.

Step 2. Set

$$x^{k+1} := \begin{cases} x_i^{k+1} = x_i^k, & \forall i \neq j, \\ x_j^k = x_j^k + 1; \end{cases} \quad p(x^{k+1}) := \begin{cases} p_i^{k+1} = p_i^k, & \forall i \neq j, \\ p_j^{k+1} = p_j^k + \dfrac{1}{a_j}; \end{cases}. \tag{4}$$

$k := k + 1$ and loop to Step 1.

2.1. First improvement

It is evident that applying the Next-step procedure to solve the problem requires $o(N)$ operations. In practical cases, the value of N is often significantly large (e.g., exceeding 100,000), which inevitably leads to a considerable slowdown in code execution. Nevertheless, it is not required to begin the search with $k = 0$; rather, it is enough to identify an initial vector x^{k_1} with a uniform distribution of the k scenarios. For example, let $N_0 = N$ and

$$x^{k_1} : \begin{cases} x_i^{k_1} = q_1 a_i, & if\ q_1 a_i, < c_i, \\ x_i^{k_1} = c_i, & if\ q_1 a_i, \ge c_i; \end{cases} \quad q_1 = \left\lfloor \frac{N_0}{\sum_{i=1}^n a_i} \right\rfloor, \quad k_1 = \sum_{i=1}^n x_i^{k_1}. \tag{5}$$

where $\lfloor \dots \rfloor$ denotes the floor function, which rounds a number down to the nearest integer.

If, for some $x_i^{k_1}$, we have $x_i^{k_1} = c_i$, this means that the inequality corresponding to this component holds as an equality, and the associated resource processes the maximum number of scenarios it can handle. Hence, the number of unprocessed scenarios is calculated as:

$$N_1 = N_0 - k_1. \tag{6}$$

We can repeat this allocation process (excluding fully loaded resources), while $N_s < N_{s+1}$, where s is a counter. Thus, the computational cost of the NSP is equal to $o(n) + o(N_s)$.

2.2. Second improvement

The second improvement to the Next-Step procedure concerns the vector function $p(x)$. It can be observed that at each iteration, only a single component of $p(x)$ changes, while the others remain unchanged. Thus, if we sort the components of $p(x)$ in ascending order $\tilde{p}(x) = (p(x), \le)$, the desired component j (Step 1 in NSP) will appear in the first position.

Then, after each step, there is no need to search for the minimum in the entire array (which has a computational complexity of $o(n)$); it is sufficient to compare the updated first component with the second one (with computational complexity $o(1)$) and, if necessary, adjust its position relative to the others (with complexity $o(n)$). This implementation can lead to a significant reduction in the number of operations in certain cases.

3. Numerical testing

We conducted numerical testing on a set of test examples for the original procedure, as well as for its three modifications: with Improvement 1, with Improvement 2, and with both improvements combined. The testing results are presented in Table 1.

Table 1
NSP modification testing

Type	Average execution time (s)	Memory usage (GB)
Original NSP	4.2174 s	1.1824
NSP with Improvement 1	0.0114 s	1.1825
NSP with Improvement 2	3.9546 s	1.1824
NSP with both Improvements	0.0107	1.1825

4. Conclusions

The numerical results demonstrate that the first improvement to the NSP significantly accelerates code execution—by a factor of 371. In contrast, the second improvement alone yields only a modest speedup of 1.06 times. However, when both improvements are applied simultaneously, the overall performance increases dramatically, resulting in a 394-fold reduction in execution time. At the same time, the memory usage of the computing system increases only slightly.

5. Acknowledgements

The research was funded by the Ministry of Education and Science of the Russian Federation within the framework of the project "Theoretical foundations, methods and high-performance algorithms for continuous and discrete optimization to support interdisciplinary research" (No. of state registration: 121041300065-9, code FWEW-2021-0003).

6. References

[1] G. A. Oparin, A. G. Feoktistov, S. A. Gorsky, Management of asynchronous computational process in ORLANDO TOOLS, Vestnik kompyuternyh i informacionnyh tekhnologiy (2011) 48–55. doi:10.1145/1188913.1188915.

[2] N. Kosyanov, Epidemic optimal control problem with vaccination and isolation processes, in: Proceedings of the 8th International School-Seminar on Nonlinear Analysis and Extremal Problems (NLA-2024), ISDCT SB RAS, Irkutsk, Russia, 2024, pp. 137–139.

Обучение представлениям электронных документов

Андрей Евгеньевич Алексеев[1], Андрей Анатольевич Михайлов[1,2]

[1] *Институт динамики систем и теории управления им. В.М. Матросова Сибирского отделения Российской академии наук, Россия, 664033, г. Иркутск, ул. Лермонтова, 134*
[2] *Институт системного программирования им. В. П. Иванникова Российской академии наук, Россия, 109004, г. Москва, ул. Александра Солженицына, д. 25*

Аннотация

В работе рассматривается метод обучения представлений электронных документов с использованием самообучения и инвариантной кластеризации. Основное внимание уделяется применению метода IIC (Invariant Information Clustering) для анализа электронных документов. Эксперименты показывают, что метод IIC способен выявлять информативные признаки из изображений документов и снижать зависимость от ручной разметки. Результаты классификации демонстрируют высокую точность при минимальном количестве размеченных данных.

Ключевые слова

самообучение, инвариантная кластеризация, электронные документы, обучение без учителя, анализ документов, классификация документов, контрастное обучение.

1. Введение

В современном мире работа с документами является неотъемлемой частью многих сфер деятельности. Однако автоматизация обработки документов всё ещё требует значительных усилий. Одной из ключевых задач является обучение компьютера анализу и интерпретации документов. В данной работе рассматривается метод обучения представлений электронных документов с использованием подхода инвариантной кластеризации (IIC).

2. Обзор работ

Самообучение включает различные подходы, среди которых выделяются методы на основе предварительных задач [2,3,4], контрастное обучение [5,6,7], генеративные методы и комбинированные подходы [8,9,10,11,12]. В области обучения представлений электронных документов модель LayoutLM стала первой трансформерной моделью, обученной с совместным использованием информации на уровне текста документа и двумерного позиционного представления текста. Метод IIC реализует подход к обучению без учителя, основанный на максимизации взаимной информации между кластерными назначениями парных образцов данных. Проведённые эксперименты демонстрируют значительное превосходство IIC по сравнению с существующими методами.

3. Метод обучения представлениям электронных документов

Наш подход основан на работе [1]. Авторы представили метод обучения представлениям, основанный на контрастном обучении. Отличительная особенность данного метода заключается в использовании взаимной информации в качестве функции потерь.

7th International Workshop on Information, Computation, and Control Systems for Distributed Environments (ICCS-DE 2025), July 7–11, 2025, Irkutsk, Russia
EMAIL: solmir505@gmail.com (A. 1); mikhailov@icc.ru (A. 2)
© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).
ICCS-DE 2025 Workshop Proceedings
DOI: 10.47350/ICCS-DE.2025.21

Основная идея заключается в обучении отображения $\Phi : X \rightarrow Y$, которое сохраняет общие для x и x′ черты и отбрасывает уникальные для каждого экземпляра особенности.

4. Эксперименты

Архитектура фреймворка, предназначенного для сопровождения научных мероприятий,

Для нашего подхода важно, чтобы классы хорошо кластеризовались. Следовательно, нужно отобрать классы, которые визуально отличаются друг от друга и содержат чётко выраженные уникальные признаки, характерные для данного класса. Исходя из этого отобраны следующие три класса: чек, паспорт и водительское удостоверение. Эксперименты проводились на компьютере с GPU NVIDIA GeForce RTX 4070 (12GB) и 8-ядерным процессором AMD Ryzen 7 5700X. Конфигурация включала 32GB оперативной памяти. Для оценки качества кластеризации использовались следующие метрики: Adjusted Rand Index (ARI) – 0.909, Homogeneity – 0.909, Completeness – 0.893. На основе кластеризации была решена задача классификации документов. Для этого понадобились метки классов 5% от общего размера датасета. Соотношение меток классов с результатами кластеризации показывает точность модели (accuracy) 0.969, что является хорошим результатом для классификации с учётом сбалансированной выборки.

5. Заключение

В ходе проведённого исследования был изучен и адаптирован метод IIC для задачи обучения представлений электронных документов. Эксперименты подтвердили высокую эффективность метода в условиях отсутствия размеченных данных, что является значительным преимуществом в контексте анализа сложных неструктурированных данных. Метод IIC продемонстрировал способность выявлять информативные признаки из изображений документов, что позволяет снизить зависимость от ручной разметки и повысить качество векторных представлений. Результаты экспериментов показали высокую точность классификации даже при минимальном количестве размеченных данных, что подтверждает практическую применимость метода в области анализа документов. Таким образом, метод IIC представляет собой перспективное решение для построения масштабируемых и устойчивых систем анализа документов на основе методов обучения без учителя. Полученные результаты открывают новые возможности для автоматизации обработки и анализа электронных документов, что может найти широкое применение в различных сферах деятельности.

6. Благодарности

Работа выполнена в рамках государственного задания Министерства науки и высшего образования Российской Федерации (тема № 1023110300006-9).

7. Список использованных источников

[1] Xu Ji, Joao F Henriques, and Andrea Vedaldi. 2019. Invariant information clustering for unsupervised image classification and segmentation. In Proceedings of the IEEE/CVF International Conference on Computer Vision. 9865–9874.

[2] S. Gidaris, P. Singh, and N. Komodakis, “Unsupervised representation learning by predicting image rotations,” in Int. Conf. Learn. Represent., pp. 1–14, 2018.

[3] R. Zhang, P. Isola, and A. A. Efros, “Colorful image colorization,” in Eur. Conf. Comput. Vis., pp. 649–666, 2016.

[4] T. Nathan Mundhenk, D. Ho, and B. Y. Chen, “Improvements to context based self-supervised learning,” in IEEE Conf. Comput. Vis. Pattern Recognit., pp. 9339–9348, 2018.

[5] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, “Momentum contrast for unsupervised visual representation learning,” in IEEE Conf. Comput. Vis. Pattern Recognit., pp. 9729–9738, 2020.
[6] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, “A simple framework for contrastive learning of visual representations,” in Int. Conf. Mach. Learn., pp. 1597–1607, 2020.
[7] J. Zbontar, L. Jing, I. Misra, Y. LeCun, and S. Deny, “Barlow twins: Self-supervised learning via redundancy reduction,” in Int. Conf. Mach. Learn., 2021
[8] K. He, X. Chen, S. Xie, Y. Li, P. Doll´ar, and R. Girshick, “Masked autoencoders are scalable vision learners,” in IEEE Conf. Comput. Vis. Pattern Recognit., pp. 16000–16009, 2022.
[9] H. Bao, L. Dong, S. Piao, and F. Wei, “Beit: Bert pre-training of image transformers,” in Int. Conf. Learn. Represent., pp. 1–13, 2022.
[10] J. Zhou, C. Wei, H. Wang, W. Shen, C. Xie, A. Yuille, and T. Kong, “ibot: Image bert pre-training with online tokenizer,” in Int. Conf. Learn. Represent., pp. 1–12, 2022.
[11] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” in NAACL, 2019.
[12] A. Baevski, H. Zhou, A. Mohamed, and M. Auli, “wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations,” in NeurIPS, 2020.

РАЗВИТИЕ СИСТЕМЫ МОНИТОРИНГА СТЕНДА ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ МИКРОСЕТИ

Максим Боднюк

Институт динамики систем и теории управления им. В.М. Матросова СО РАН, ул. Лермонтова, д. 134, 664033, Иркутск, Россия

Аннотация

Представлено развитие архитектуры и функциональных возможностей СМ стенда имитационного моделирования микросети. Показаны преимущества модифицированной архитектуры в сравнении с базовой. Достигнутые преимущества подтверждены экспериментальными результатами.

Ключевые слова

Микросеть, имитационный стенд, информационно-измерительная система, мониторинг

1. Введение

Применение микросетей является актуальным направлением развития энергетики как в России, так и за рубежом. При этом возникают проблемы с выбором оптимальной структуры и параметров работы сетей. Проведение экспериментов с реальным энергетическим комплексом сопряжено с высокими временными и финансовыми затратами, а также с необходимостью обеспечения надежности и безопасности работы с его оборудованием. Поэтому исследования зачастую проводят на имитационном стенде микросети, который позволяет проанализировать структуру сети, определить основные характеристики и оптимальные режимы ее работы. Важным связующим звеном между стендом и специалистами, исследующими микросеть, является система мониторинга (СМ) работы стенда. В данной работе предложено развитие базовой архитектуры (БА) СМ стенда имитационного моделирования микросети технопарка Иркутского национального исследовательского технического университета [1].

2. Результаты исследования

Существует широкий спектр систем мониторинга инженерной инфраструктуры энергетических комплексов [2]. Однако полнофункциональные системы являются, как правило, платными. Другие системы зачастую не подходят для мониторинга конкретного оборудования из-за отсутствия необходимых программно-аппаратных компонентов, несовместимости форматов и протоколов передачи данных. Поэтому возникает необходимость разработки новой или существенной модификации существующей СМ [3]. На рис. 1 показана БА. Оборудование стенда включает блоки однофазных источников питания, однофазных трансформатороров, измерителей параметров однофазной сети, моделей трансформаторных подстанций и нагрузок, линий электропередачи, преобразователя интерфейсов. В рамках БА приложение MasterOPC выполняет сбор данных со стенда в режиме реального времени по протоколу Modbus и их передачу в приложение оператора. Оператор может получать значения наблюдаемых параметров блоков и регулировать в ручном режиме изменяемые параметры блоков в приложении оператора. Включение/отключение блоков оператором возможно только в ручном режиме. К сожалению, БА реализует только часть необходимых возможностей.

7th International Workshop on Information, Computation, and Control Systems for Distributed Environments (ICCS-DE 2025), July 7–11, 2025, Irkutsk, Russia
EMAIL: maks.bodnyuk@mail.ru (A. 1)
ORCID: 0009-0008-4337-9352 (A. 1)
© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).
ICCS-DE 2025 Workshop Proceedings
DOI: 10.47350/ICCS-DE.2025.22

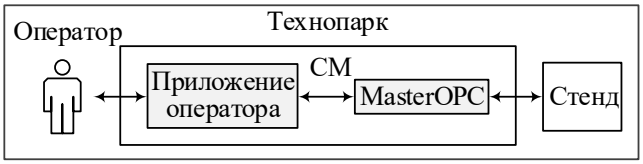


Рисунок 1. БА.

В работе предложена модифицированная архитектура (МА) СМ (рис. 2). В рамках МА разработаны следующие новые компоненты: приложение управления парметрами блоков и их включением/выключением в автоматическом режиме на основе прямого обмена данными между оператором и стендом по протоколу Modbus, а также между оператором и внешними системами; приложение для визуализации как данных в реальном времени, так и исторических данных за указанный период; реляционная база данных (БД) SQLite для хранения результатов мониторинга; API для обеспечения доступа специалистам к СМ через внешние системы, обеспечивающий получение данных мониторинга по запросу, их агрегирование, структурирование и фильтрацию по заданным временным интервалам. В отличие от БА, возможности МА не ограничивают число измеряемых параметров блоков стенда, обеспечивают возможность хранения данных мониторинга, поддерживают обмен данными в телекоммуникационной сети, повышают точность измеряемых параметров с 0.1 до 0.001 и позволяют визуализировать временные рядов как текущих, так и ретроспективных данных.

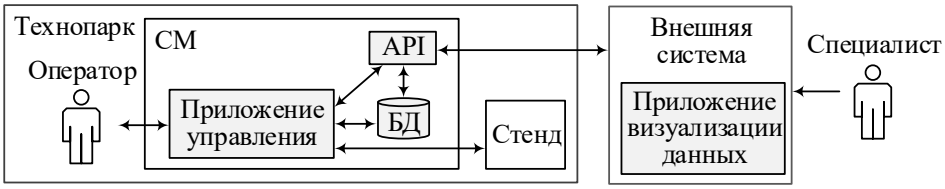


Рисунок 2. МА.

МА апробирована при моделировании работы автономной микросети туристического лагеря, расположенного на Байкальской природной территории (БПТ), в условиях крупных возмущений. Блоки стенда использовались для имитации работы трансформаторов, линий электропередачи и потребителей микросети с учетом разного рода нагрузок. В результате имитационного моделирования определены агрегированные показатели основных периодов ее работы в условиях крупных возмущений и выявлены наиболее уязвимые элементы сети.

3. Заключение

Осуществлено развитие БА СМ стенда технопарка. В результате ускорен обмен данными и повышена их точность до 0.001. Работоспособность и эффективность работы МА СМ апробированы в процессе моделирования работы автономной микросети туристического лагеря, расположенного на БПТ. Работа выполнена в рамках гранта № 075-15-2024-533 Минобрнауки РФ на выполнение крупного научного проекта по приоритетным направлениям научно-технологического развития (№ гос. рег. 124052100088-3).

4. Список литературы

[1] О технопарке. URL https://www.istu.edu/deyatelnost/nauka/baza/tekhnopark.

[2] В. М. Левин, П. А. Петушков, М. А. Швец. Дистанционный мониторинг и управление состоянием трансформаторов в распределительных электрических сетях. Известия высших учебных заведений. Проблемы энергетики, 2023, т. 25, № 2, с. 97–109. doi:10.30724/1998-9903-2023-25-2-97-109.

[3] Ю. А. Клименко, Я. Е. Львович, А. П. Преображенский. Проектирование контрольно-измерительных компонент распределительных энергетических систем. Advanced Engineering Research (Rostov-on-Don), 2024, т. 24, № 1, с. 88–97. doi:10.23947/2687-1653-2024-24-1-88-97.

Сегментация КЛКТ-снимков на основе U-Net архитектуры

Антон Булах [1]

[1] *Институт математики и информационных технологий Иркутского государственного университета, Бульв. Гагарина, д. 20, Иркутск, 6640003, Россия*

Аннотация

В данной работе рассматривается задача автоматической сегментации медицинских изображений, полученных с помощью конусно-лучевой компьютерной томографии (КЛКТ). Для выделения анатомических структур черепа используется свёрточная нейронная сеть с архитектурой U-Net. Представлены основные этапы подготовки данных, обучения модели и её применения для решения задачи семантической сегментации. Приведены примеры результатов работы модели и обсуждаются перспективы дальнейшего развития подхода.

Ключевые слова

Сегментация медицинских изображений, U-Net, конусно-лучевая компьютерная томография, глубокое обучение, обработка медицинских данных

1. Введение

Конусно-лучевая компьютерная томография (КЛКТ) широко применяется в стоматологии, ортодонтии и челюстно-лицевой хирургии для получения объёмных изображений костных структур черепа. Традиционно процесс анализа таких снимков требует ручной разметки интересующих областей — верхней и нижней челюстей, зубов, корневых каналов и других костных структур черепа.

Ручная сегментация является трудоёмкой и зависит от квалификации специалиста. Автоматизация этого процесса позволяет повысить эффективность работы, снизить риск ошибок и ускорить получение результатов.

2. Методы и материалы

Для достижения поставленной цели использовались методы на основе глубокого обучения. Использовалась библиотека MONAI [1], производилось обучение модели нейронной сети U-Net. [2] Для того, чтобы собрать новый набор данных, использовались размеченые КЛКТ-снимки ToothFairy2 Dataset [3]. Обучение модели проводилось в GoogleColab.

[1] 7th International Workshop on Information, Computation, and Control Systems for Distributed Environments (ICCS-DE 2025), July 7–11, 2025, Irkutsk, Russia
EMAIL: antonionykid@gmail.com
ORCID: 0009-0000-3121-6582
© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).
ICCS-DE 2025 Workshop Proceedings
DOI: 10.47350/ICCS-DE.2025.23

3. Результаты

В результате проведённого исследования были получены следующие результаты:

- Создан и подготовлен набор данных, содержащий КЛКТ-снимки и соответствующие им размеченные маски;
- Обучена модель на основе U-Net, демонстрирующая сегментацию анатомических структур;
- Выполнена постобработка результатов, которая позволит улучшить чёткость границ и целостность выделенных объектов.

4. Выводы

В ходе работы был создан новый набор данных и обучена модель, однако результаты тестирования показали средние значения. Также, на данном этапе работы, модель не включает в себя сегментацию таких отделов как зубы и их корневая система. Это показывает важность пордолжения исследований и дообучения модели. Необходимо подтвердить гипотезу о положительном влиянии автоматического преобразования КЛКТ-снимков, оптимизации архитектуры модели и исследовании методов обработки изображений, которые ещё не были использованы в работе.

5. Выводы

Полученные результаты подчёркивают сложность задачи сегментации КЛКТ-снимков и необходимость использования комплексного подхода, который включает в себя не только глубокое обучение, но и тщательное предобучение и подготовку данных.

6. Список литературы

[1] MONAI Consortium. MONAI: Medical Open Network for AI // https://github.com/Project-MONAI/MONAI .

[2] O. Ronneberger, P. Fischer, T. Brox, *U-Net: Convolutional Networks for Biomedical Image Segmentation* , MICCAI, 2015.

[3] M. S. Petrov, D. A. Kopylov. ToothFairyDataset2: Dataset for Dental Image Segmentation and 3D Reconstruction // Data in Brief, 2023. — Vol. 47. — P. 108952. DOI:10.1016/j.dib.2023.108952.

Разработка методов программной симуляции работы LIDAR
Development of software simulation methods LIDAR operation

Елизавета Викулова [1], Антон Толстихин [1]

[1] *Институт динамики систем и теории управления им. В.М. Матросова Сибирского отделения Российской академии наук, Россия, 664033, г. Иркутск, ул. Лермонтова, 134*

Аннотация
В рамках данной работы рассматривается задача разработки методов программной симуляции работы LIDAR для применения в робототехническом комплексе TEMAR. В данном комплексе установка реальных LIDAR-датчиков невозможна по экономическим и конструктивным причинам. Для решения задачи предлагаются два подхода: итеративный и аналитический. В результате проведенной серии экспериментов был выявлен наиболее успешный по точности и скорости работы подход.

Ключевые слова
LIDAR, компьютерное зрение, моделирование, препятствие

1. Введение

Современные технологии автоматизации требуют эффективных систем восприятия окружающей среды, среди которых LIDAR [1, 2] является ключевым инструментом для точного измерения расстояний и создания трехмерных моделей. В условиях высокой стоимости и сложности интеграции реальных датчиков в робототехнические системы разрабатываются методы программной симуляции LIDAR, позволяющие моделировать его работу на основе данных с камер и алгоритмов компьютерного зрения. В данной работе предлагаются итеративный и аналитический подходы к симуляции LIDAR, что позволяет повысить точность и эффективность систем восприятия в условиях невозможности использования реальных датчиков.

2. Исследуемые подходы и условия проведения экспериментов

Данная работа заключается в разработке методов программной симуляции работы LIDAR для дальнейшего их применения в рамках комплекса TEMAR [3]. В данном случае установка реальных датчиков на мобильных роботов комплекса невозможна по конструктивным и экономическим соображениям. Однако, присутствует система камер и программное обеспечение с использованием алгоритмов компьютерного зрения, способные распознавать препятствия на стенде. Таким образом, задачей модуля симуляции LIDAR является отбор данных, доступных роботу, а также их преобразование в соответствующий формат.

Итеративный подход [4, 5], основанный на последовательном построении лучей с фиксированным шагом, является базовым методом, который позволяет достаточно точно воспроизвести работу реального LIDAR. Этот метод прост в реализации, но из-за фиксированного шага может обеспечивать недостаточно высокую точность и требует значительных вычислительных ресурсов, особенно при обработке сложных сцен с большим

7th International Workshop on Information, Computation, and Control Systems for Distributed Environments (ICCS-DE 2025), July 7–11, 2025, Irkutsk, Russia
EMAIL: vikulizavet85@gmail.com (A. 1); madstayler93@gmail.com (A. 2)
©2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).
ICCS-DE 2025 Workshop Proceedings
DOI: 10.47350/ICCS-DE.2025.24

количеством объектов. В данной работе он используется как эталон для сравнения с другими двумя методами.

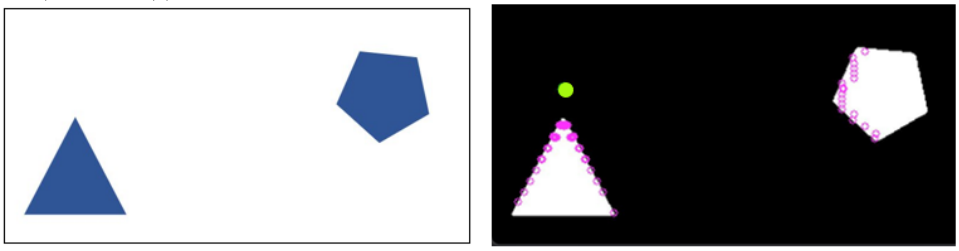
Предложенный аналитический подход, в свою очередь, включает два метода: описание препятствий с использованием фазового пространства и аппроксимацию контуров препятствий. Метод с использованием фазового пространства, основанный на преобразовании Хафа [6, 7], позволяет эффективно находить прямые, описывающие границы объектов, за счет анализа пересечений синусоидальных кривых. Аппроксимация контуров с помощью алгоритма Рамера-Дугласа-Пекера [8] позволяет сократить количество точек, описывающих препятствия, сохраняя их форму, что снижает вычислительную нагрузку. После определения границ препятствий предлагается метод поиска точек, который позволяет получить массив координат этих препятствий. Стоит отметить, что сложные вычисления проводятся единожды, в случае стационарных препятствий, что существенно снижает вычислительную нагрузку и повышает эффективность работы алгоритма. После чего требуется только анализировать пересечения прямых.

Для проверки качества работы предложенных подходов была проведена серия тестов на подготовленных изображениях в количестве 10 штук. Тесты проводились с различными сложностями сцен, а именно: одно или несколько препятствий выпуклой и невыпуклой формы. Формат изображений JPG, каждое из которых представляет собой стандартный кадр в цветовой модели RGB и представлено в трех разрешениях: 450x300, 800x500 и 1800x900 пикселей. Разработана программа в интегрированной среде разработки IntelliJ IDEA на языке программирования Java с применением алгоритмов компьютерного зрения, в частности, используется библиотека OpenCV [9]. Требований к операционной системе не выдвигалось.

Ключевым критерием для оценки эффективности применяемых методов является время обработки одного изображения, выраженное в миллисекундах. Для повышения объективности результатов рассчитывается среднее время на основе ста тестовых запусков для каждого изображения. Это позволяет снизить влияние случайных факторов и получить более точную оценку производительности.

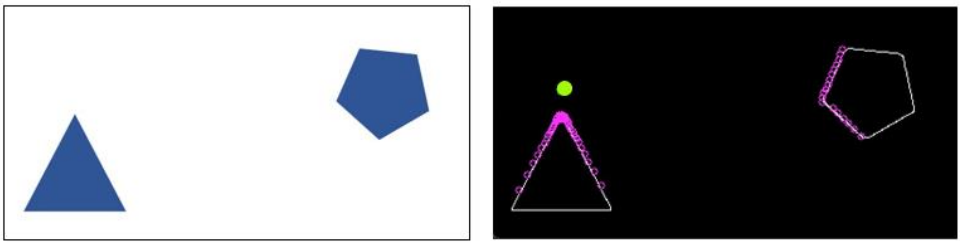
3. Результаты

На рисунках 1-3 приведены примеры работы модуля, для каждого из исследуемых подходов. На них зеленая точка обозначает положение LIDAR, сиреневые круги представляют собой найденные точки границы препятствия, которые были выявлены с помощью соответствующего метода.



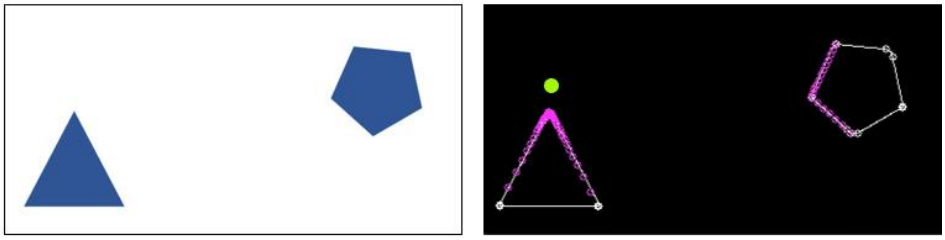
a) **b)**

Рисунок 1: Пример входного изображения для двух выпуклых препятствий с различным количеством углов (a) и результат работы итеративного подхода (b)



a) **b)**

Рисунок 2: Пример входного изображения для двух выпуклых препятствий с различным количеством углов (a) и результат работы аналитического подхода с использованием фазового пространства (b)



a) **b)**

Рисунок 3: Пример входного изображения для двух выпуклых препятствий с различным количеством углов (a) и результат работы аналитического подхода с использованием аппроксимации (b)

На рисунке 4 представлены сравнительные результаты предложенных алгоритмов. В случае аналитических методов время обработки измерялось дважды: с нахождением границ препятствий и без него. Так как в комплексе TEMAR препятствия остаются статичны, сложные вычисления нужно проводить единожды. Поэтому справедливо оценивать время как с построением карты (поиском препятствий), так и без него.

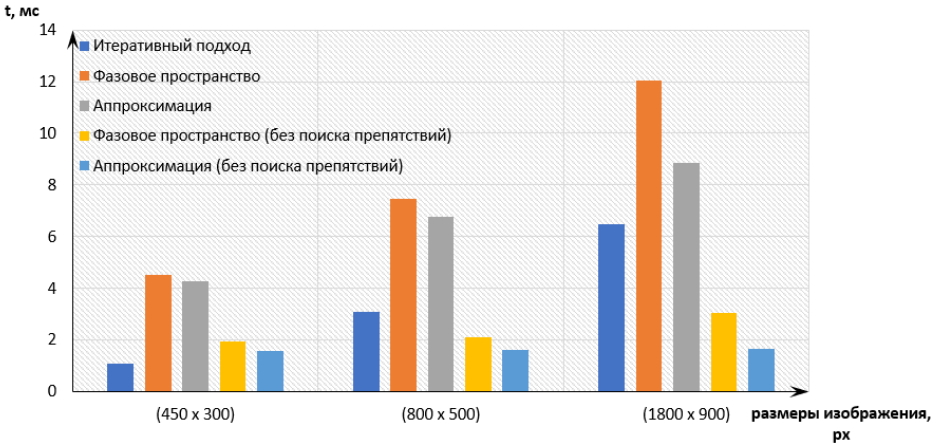


Рисунок 4: Результаты работы алгоритмов

На основе полученных результатов, можно сделать вывод, что итеративный подход отличается линейном ростом сложности O(n), зависящим от размера изображения. Данный метод требует значительных вычислительных ресурсов из-за необходимости проверки большого количества точек. Фиксированный шаг усложняет точное определение границы объекта. Уменьшение шага для повышения точности ведет к увеличению количества проверяемых точек и, как следствие, времени обработки. В результате, достижение четкой границы объекта становится практически невозможным из-за отсутствия компромисса между размером шага и вычислительными затратами. Однако, он будет более эффективен в случае

нестационарных препятствий, когда объекты изменяются или меняют свое положение во времени или пространстве.

Предложенные аналитические методы в случае, когда осуществляется поиск препятствий (построение карты) характеризуются линейной сложностью O(n), которая также как и у итеративного метода зависит от размера входного изображения. Однако, при условии работы со стационарными препятствиями, вычислительная сложность значительно снижается, поскольку сложные вычислительные процедуры требуется проводить только один раз. В дальнейшем осуществляется только анализ пересечения прямых, что позволяет достичь практически константной сложности O(1), не зависящей от размера изображения на входе. Такой подход обеспечивает более четкое определение границ объектов и является его явным преимуществом. Метод с использованием фазового пространства обладает более низкой скоростью работы за счет большего количества найденных границ препятствий. В целом, оба метода показывают хорошую стабильность и эффективность, что подтверждается результатами экспериментов на различных примерах.

Учитывая, что в рамках комплекса TEMAR препятствия являются статичными объектами, наиболее подходящим для выбора является аналитический подход с использованием метода аппроксимации множества точек контура. Который показал лучшее время обработки изображения среди всех рассмотренных подходов.

4. Источники

[1] Современные методы геодезической измерительной техники [Электронный ресурс] // StudWork : [сайт]. – URL: https://studwork.ru/spravochnik/geodeziya/sovremennye-metody-geodezicheskoy-izmeritelnoy-tehniki (дата обращения: 02.05.2025).

[2] Антонов А. Сканирующие лазерные дальномеры (LIDAR) [Электронный ресурс] / А. Антонов // Современная электроника : спец. проект. - Волгоград, [б.г.]. – URL: https://www.cta.ru/articles/soel/2016/2016-1/4475/ (дата обращения: 01.05.2023).

[3] Kostylev D. A. Development of the complex modelling system for intelligent control algorithms testing / D. A. Kostylev, A. A. Tolstikhin, S. A. Ul'yanov // 42nd International Convention MIPRO 2019 : proceedings. – Opatija, 2019. – Vol. 1. – No. 1. – P. 1091–1096.

[4] Schmitt A. Ray Tracing Algorithms - Theory and Practice / A. Schmitt, H. Muller, W. Leister // Theoretical Foundations of Computer Graphics and CAD. – Berlin: Springer Verlag, 1988. – P. 997–1030.

[5] OpenCV: Sobel Derivatives [Электронный ресурс] // OpenCV Documentation. – URL: https://docs.opencv.org/4.x/d2/d2c/tutorial_sobel_derivatives. html (дата обращения: 06.05.2025).

[6] Дуда Р. О. Использование преобразования Хафа для обнаружения линий и кривых на изображениях / Р.О. Дуда, П.Э. Харт // Comm. ACM. – 1972. – Т. 15. – С. 11–15.

[7] OpenCV: Hough Line Transform [Электронный ресурс] // OpenCV Documentation. – URL: https://docs.opencv.org/4.x/d6/d10/tutorial_py_houghlines.html (дата обращения: 14.05.2025).

[8] Douglas D.H. Algorithms for the Reduction of the Number of Points Required to Represent a Digitized Line or its Caricature / D.H. Douglas, T.K. Peucker // Cartographica: The International Journal for Geographic Information and Geovisualization. – 1973. – Vol. 10, № 2. – P. 112–122.

[9] Прохоренок Н. А. OpenCV и Java. Обработка изображений и компьютерное зрение / Н. А. Прохоренок. СПб.: БХВ-Петербург, 2018. – 320 с.

Обзор методов предобработки мультимодальных данных, содержащих климатическую информацию, при генерации векторных представлений данных

Ольга Юрьевна Гавенко[1,2], Наталья Александровна Шашок[2]

[1] *Институт динамики систем и теории управления им. В.М. Матросова СО РАН, ул. Лермонтова, 134, а/я 292, Иркутск, 664033, Россия*
[2] *Федеральный исследовательский центр информационных и вычислительных технологий, пр-т Академика Лаврентьева, 6, Новосибирск, 630090, Россия*

Аннотация

Генерация векторных представлений является важной технологией обработки данных, позволяющей преодолевать семантические разрывы между формально разными данными и обеспечивающей эффективный семантический поиск и анализ взаимосвязей. Это создает основу для интеграции мультимодальных данных и разработки сложных интеллектуальных систем. Качественная предобработка данных выступает важным этапом, напрямую влияющим на качество векторного поиска среди итоговых эмбеддингов. Особенно это актуально для работы с мультимодальными климатическими данными, где необходимо учитывать специфику различных форматов.

В статье представлен обзор на методы предобработки данных, содержащих климатическую информацию, в частности, текстовых, графических и картографических. Эти методы необходимы при разработке интеллектуальных информационных систем хранения, обработки и анализа климатической информации.

Ключевые слова

Векторные представления, предобработка данных, мультимодальные данные

1. Введение

Задача генерации векторных представлений (эмбеддингов) из разнородных данных — из текста, изображения, аудиозаписи, видео или иных — занимает центральное место в современном машинном обучении и искусственном интеллекте. Эта технология является фундаментальной, поскольку позволяет решать задачи семантического поиска и анализа взаимосвязей между объектами. Ключевая ценность такой технологии заключается в том, что эмбеддинги дают возможность находить документы или элементы данных, обладающие близким смысловым содержанием (семантической близостью), даже если на поверхностном уровне (например, в лексике текста или визуальных деталях изображения) они кажутся далекими по своим формальным или синтаксическим характеристикам.

Процесс генерации эмбеддингов заключается в трансформации неструктурированных или слабоструктурированных данных (которые сложно обрабатывать алгоритмами напрямую) в формализованное векторное пространство высокой размерности. В этом пространстве математическое расстояние между векторами (например, вычисляемое через косинусную близость или евклидову метрику) напрямую коррелируется с содержательной (семантической) близостью представляемых ими объектов. Таким образом, семантические отношения между реальными данными материализуются в виде геометрических отношений их векторных представлений. Именно эта технология обеспечивает эффективную интеграцию разнотипных

7th International Workshop on Information, Computation, and Control Systems for Distributed Environments (ICCS-DE 2025), July 7–11, 2025, Irkutsk, Russia
EMAIL: olga.yu.gavenko@mail.ru (A. 1); n.shashok@alumni.nsu.ru (A. 2)
ORCID: 0000-0003-3619-1120 (A. 1); 0009-0007-3658-6110 (A. 2)
©2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).
ICCS-DE 2025 Workshop Proceedings
DOI: 10.47350/ICCS-DE.2025.25

данных (мультимодальность), создавая основу для разработки сложных мультимодальных систем анализа данных и интеллектуальных систем принятия решений, способных обрабатывать и синтезировать информацию из различных источников.

Предобработка входных данных является важным этапом перед извлечением признаков и непосредственно генерацией эмбеддингов. Качество предобработки напрямую определяет качество итоговых векторных представлений. Цель этого этапа — подготовить "сырые" данные таким образом, чтобы модель могла выделить из них наиболее значимые и релевантные признаки для последующего обучения. Важно, чтобы, в результате работы модели, интуитивное понимание смысловой близости объектов человеком находило свое точное отражение в близости соответствующих векторов в созданном пространстве. Иными словами, если человек считает два документа или изображения семантически похожими, их эмбеддинги при качественной генерации должны быть расположены в векторном пространстве близко друг к другу.

В данной работе представлен обзор наиболее часто применимых алгоритмов генерации векторных представлений из исходных данных с целью дальнейшего их использования в интеллектуальной системе, обрабатывающих мультимодальные климатические данные различного формата.

2. Предобработка данных

Поскольку речь идет именно о данных, которые могут содержать в себе климатическую информацию, центральными типами данных определим текстовые, графические и геоданные, в частности, картографические. Представляется, что типов данных, полезных при работе с экологическими данными, существует больше, чем рассмотренных, поскольку возможно использование и таких данных, как числовые, в виде таблиц.

2.1. Текстовые данные

Перед извлечением признаков из текста необходимо его предварительно подготовить таким образом, чтобы текстовые данные стали пригодными для использования специальных алгоритмов, предназначенных для использования в рамках искусственного интеллекта и машинного обучения. В случае с текстовыми данными нет каких-либо особенностей, связанных с нахождением и выделением климатической информации в сырых данных, поэтому предобрабатка представляется стандартной.

Одним из основных шагов в предобработке данных является токенизация, то есть преобразование текста в последовательность отдельных блоков данных — токенов. Предполагается, что при токенизации база знаний разделяется на некоторые семантически самостоятельные фрагменты текста, и токен должен соответствовать логически завершенному элементу знаний. В качестве токенов могут быть определены слова, символы, числа (зачастую с использованием дополнительных математических символов; так, токен «+50» и токен «-50», вероятно, будут считаться разными, так как несут разную смысловую нагрузку), формулы или подстроки, в зависимости от задачи. Существуют различные модели токенизации, например, используемые OpenAI в инструменте tiktoken [1]; такие модели, как GigaChat также используют токенизацию при создании эмбеддингов [2].

Следующим шагом в предобработке может быть нормализация. Каждый токен может подвергнуться нормализации, то есть приведению слов и символов к единообразному виду, который упрощает дальнейший анализ и генерацию эмбеддингов. В данном случае речь идет не о масштабировании числовых данных, а о стандартизации текстовых элементов, таких как регистр, пунктуация или написание. Нормализация обеспечивает консистентность генерируемых эмбеддингов, что влияет на качество работы моделей. Однако при её реализации следует учитывать специфику текста, поскольку в некоторых случаях «лишние» элементы (например, эмоциональная пунктуация) могут быть важны для задач анализа тональности или стиля.

Следующие шаги – стемминг, (удаление флексий); лемматизация (приведение слов к канонической форме) и чистка (удаление не несущих смысловой нагрузки слов) – применяются не так часто, поскольку могут приводить к потере лингвистически релевантной информации.

2.2. Графические данные

Одним из наиболее ранних подходов генерации векторов из изображений с целью дальнейшего поиска похожих изображений было использование цветовых гистограмм [3], позволяющее осуществлять поиск по цветовым профилям, что упрощает работу с большими объемами данных. В данном подходе предобрабатывать изображения не требуется, поскольку при генерации эмбеддингов в данном случае изображение конвертируется в RGB-формат, после чего разбивается на три цветовых канала: красный, зелёный и синий, после чего для каждого канала строится гистограмма, где по оси X — значение цвета (0–255), а по оси Y — количество пикселей с этим значением. Конечные вектора получаются конкатенацией гистограмм.

Иначе происходит при использовании более поздних моделей генерации векторов.

Локальные дескрипторы, такие как SIFT (Scale-Invariant Feature Transform) [4] и SURF (Speeded-Up Robust Features) [5], выделяют ключевые точки и текстуры, устойчивые к масштабу, повороту и освещению, что позволяет сравнивать изображения по локальным признакам. Однако зависимость от ручной настройки параметров и высокая вычислительная сложность делают их непрактичными для работы с большими массивами данных.

Такие архитектуры, как AlexNet [6] и ResNet [7], автоматически извлекают иерархические, вложенные друг в друга признаки — от краёв и текстур до сложных семантических объектов. Векторы, полученные из последних слоёв таких сетей, кодируют высокоуровневую информацию, что повышает точность поиска, поскольку вектор в данном случае кодирует в себе некоторое «смысловое» значение изображения.

Как итог, предобработка изображений сильно зависит от тренировочного пайплайна модели. Например, такие подходы, как нормализация пикселей — приведение значений к диапазону [0, 1] или стандартизация (для ImageNet: вычитание средних [0.485, 0.456, 0.406] и деление на стандартные отклонения [0.229, 0.224, 0.225]) — могут улучшать сходимость моделей. Конвертация цветового пространства (RGB, HSV, Lab) позволяет выделять специфические признаки, такие как текстуры или освещение. Изменение размера изображения до фиксированного формата (например, 224×224 для ResNet) с использованием интерполяции или обрезки обеспечивает совместимость с входными слоями нейросетей, при этом сохранение аспектного соотношения критично для задач, где важны пропорции объектов.

2.3. Графические данные

Предобработка географических и картографических данных для генерации эмбеддингов представляет собой комплекс шагов по подготовке пространственной информации. Начинается она с унификации и конвертации разнородных исходных данных (векторных форматов и растровых изображений карт или снимков) в структурированное представление, пригодное для моделей машинного обучения.

В качестве релевантных семантических признаков для векторных данных могут выступать координаты, геометрические характеристики (площадь, периметр), топологические отношения (соседство, пересечение) и категориальные атрибуты; для растровых данных применяются методы компьютерного зрения и глубокого обучения, следовательно, релевантные признаки могут быть схожими с графическими данными, а шаги по предобработке могут совпадать с шагами предобработки изображений. Однако важным также представляется своего рода «соединение» разных видов карт в один набор данных, и, соответственно, обогащение данных: в качестве дополнительных потоков в изображении карты возможно добавление графической информации о выбросах газов, высоте над уровнем моря, типах зданий на определенной территории, что повышает семантическую насыщенность. Такой подход используется, в

частности, в моделях типа Loc2Vec [8], в которых эмбеддинги несут в себе закодированную информацию как о географических, так и семантических признаков, описывающих окружение места.

3. Заключение

Представленный обзор специфических особенностей процесса предобработки климатических данных некоторых форматов (текст, изображения, геоданные) предоставляет необходимую основу для построения эффективных интеллектуальных систем. Такие системы, опирающиеся на качественные векторные представления, способны решать сложные задачи анализа, поиска и прогнозирования в сфере экологии и климатологии, синтезируя знания из разнородных источников информации.

4. Список литературы

[1] OpenAI, Tiktoken: Fast bpe tokenizer for openai's models, 2022. URL: https://github.com/openai/tiktoken

[2] Руководства GigaChat API, Подсчет токенов | Документация для разработчиков, 2025. URL: https://developers.sber.ru/docs/ru/gigachat/guides/counting-tokens

[3] Н. С. Костюкова, Использование 2D-цветовых гистограмм для представления цветового содержимого изображений, Известия ЮФУ. Технические науки, №1, 2005.

[4] T. Lindeberg, Scale Invariant Feature Transform, Scholarpedia, 7(5), 2012.

[5] H. Bay, T. Tuytelaars, L. Van Gool, SURF: Speeded Up Robust Features, in: Leonardis, A., Bischof, H., Pinz, A. (Eds), Computer Vision – ECCV 2006. Lecture Notes in Computer Science, vol 3951, Springer, Berlin, Heidelberg, 2006.

[6] A. Krizhevsky, I. Sutskever, G. E. Hinton, ImageNet classification with deep convolutional neural networks, in: Commun. ACM 60, 6, 2017, pp. 84–90.

[7] H., Kaiming, X. Zhang, S. Ren, J. Sun, Deep Residual Learning for Image Recognition, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, pp. 770-778, doi: 10.1109/CVPR.2016.90.

[8] Sentiance, Loc2Vec: Learning location embeddings with triplet-loss networks, 2020. URL: https://sentiance.com/loc2vec-learning-location-embeddings-w-triplet-loss-networks

Методы построения матрицы перспективного преобразования для сшивки изображений

Надежда Душкина [1], Антон Толстихин [1]

[1] *Институт динамики систем и теории управления Им. В.М. Матросова, СО РАН, ул. Лермонтова, 134, Академгородок м-н, Свердловский район, Иркутск, 664033, Россия*

Аннотация

В данной статье рассматриваются методы построения матрицы перспективного преобразования для автоматической сшивки изображений. Проведен анализ традиционных подходов, включая RANSAC и метод наименьших квадратов (МНК), а также предложен гибридный алгоритм, сочетающий МНК с роевым методом оптимизации GWO (Grey Wolf Optimizer). Экспериментально подтверждено, что комбинированный метод МНК+GWO обеспечивает лучшее соотношение скорости и качества сшивки по сравнению с классическими решениями. Разработанный подход может быть применен в системах компьютерного зрения и робототехнике, в частности, в программно-аппаратной системе TEMAR. Перспективными направлениями дальнейших исследований являются адаптация алгоритма для работы с видеопотоками в реальном времени и интеграция с методами глубокого обучения.

Ключевые слова

Сшивка изображений, перспективное преобразование, матрица преобразований, RANSAC, метод наименьших квадратов, роевые алгоритмы, GWO, оптимизация, компьютерное зрение, TEMAR

1. Введение

В современном мире, где визуальная информация играет ключевую роль, потребность в широкоугольных и детализированных изображениях постоянно растет. Одним из эффективных методов получения таких изображений является технология сшивания (склейка) нескольких изображений в одно целое [1]. Такая технология является одной из распространённых задач обработки, подразумевает под собой получение единой панорамы или изображения с высоким разрешением из нескольких перекрывающихся изображений, которые могут быть получены в разное время, с разных точек зрения или с разных датчиков.

Сшивание изображений имеет большое значение в области медицинской визуализации, компьютерного зрения, спутниковых данных, а также возникает при аэрофотосъёмке или микросъемке длинной детали. Практическая значимость работы заключается в возможности применения разработанного алгоритма для автоматической сшивки изображений внутри программно-аппаратной системы моделирования TEMAR (Testbed Environment for Multiple Autonomous Robots) [2], существующей на базе Института динамики систем и теории управления имени В. М. Матросова Сибирского отделения Российской академии наук в городе Иркутске.

2. Этапы создания сшитого изображения

В целях решения задачи сшивки изображений используется метод склеивания изображений на основе ключевых точек и дескрипторов, поскольку этот метод является универсальным и

7th International Workshop on Information, Computation, and Control Systems for Distributed Environments (ICCS-DE 2025), July 7–11, 2025, Irkutsk, Russia
EMAIL: dushckina.nad@yandex.ru (A. 1); madstayler93@gmail.com (A. 2)
© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).
ICCS-DE 2025 Workshop Proceedings
DOI: 10.47350/ICCS-DE.2025.26

подходит для решения большинства задач. Для выполнения склеивания изображений необходимо решить следующие задачи:

1. найти координаты ключевых точек, выделив некоторое число их пар;
2. построить матрицу перспективного преобразования для искажения второго изображения на основе найденных ключевых точек;
3. найти координаты новых ключевых точек для второго изображения, при помощи построенной матрицы из пункта 2;
4. применить матрицу ко всему второму изображению, получить его искаженную копию;
5. сшить изображения, сопоставляя ключевые точки;
6. оценить величину ошибки сопоставления ключевых точек.

Ключевые точки представляют собой значимые характеристики, такие как угловые точки или текстурные элементы, которые можно использовать для сопоставления. Другими словами, ключевые точки – характерные точки изображения, т.е. точки, которые сильнее всего отличаются от других областей [3]. Обычно это углы, пересечения линий или текстуры с резкими переходами, например, на шахматной доске характерной точкой будет переход из черной клетки в белую. Эти точки служат опорными маркерами, которые помогают машинам анализировать и интерпретировать визуальный контент. Определение ключевых точек осуществляется, например, при помощи алгоритмов ORB, SURF, SIFT и др.

Для установления соответствий между ключевыми точками двух изображений выполняется попарное сравнение их дескрипторов с последующим выделением наиболее схожих пар на основе заданной метрики сходства.

По результатам российского исследования [4] алгоритм FAST из группы традиционных алгоритмов проявил себя как лучший алгоритм для поиска ключевых точек. В данной работе применяется алгоритм BRISK (Binary Robust Invariant Scalable Keypoints) [5], который эффективен для масштабирования и поворотов, в его основе использован улучшенный алгоритм FAST. Он сочетает в себе эффективность бинарных методов (как ORB, FAST) с устойчивостью к изменениям масштаба и поворота (как SIFT, SURF).

По наличию артефактов на изображении, используя визуальную оценку, можно без дополнительных усилий или оборудования определить качественно сшито изображение или нет. Артефакты на изображении – это дефекты, которые проявляются в виде искажённых пикселей или фрагментов с изменёнными цветами.

Помимо визуальной оценки качества сшитого изображения, существуют различные численные метрики, позволяющие оценивать качество сшитого изображения по различным критериям. Измерение несоответствия между предсказанными позициями на изображении и фактически наблюдаемыми позициями будем рассматривать как величины следующих ошибок, которые выражены формулами (1 – 3):

- среднеарифметическое отклонение;
- среднеквадратическое отклонение;
- максимальное отклонение.

$$E_1 = \frac{\sum_{i=1}^{n} \left\| x_i - x_i^{'} \right\|}{n} \to \min, \tag{1}$$

$$E_2 = \frac{\sum_{i=1}^{n} \left\| x_i - x_i^{'} \right\|^2}{n} \to \min, \tag{2}$$

$$E_3 = \max_{1 \le i \le n} \left\| x_i - x_i^{'} \right\| \to \min. \tag{3}$$

Здесь n – это количество ключевых точек, x_i – это координата ключевой точки на первом изображении, x_i' – это координата ключевой точки на втором изображении, выраженная в координатной плоскости первого изображения.

2.1. Методы построения матрицы преобразования

Матрица преобразований применяется для вычисления новых координат объекта при его трансформации. Изменяя элементы матрицы преобразования, можно применять к объектам любые трансформации, такие как масштабирование, зеркальное отражение, поворот, перемещение и другие. Преобразование между двумя системами координат представлено в виде матрицы 3×3 и записывается следующим образом:

$$H = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix}. \quad (4)$$

Координатные преобразования выражаются в виде матричных умножений. Применяя к изображению единичную матрицу, получим неискажённое изображение. При этом качество полученного изображения во многом зависит от нахождения и построения матрицы перспективного преобразования. В данной работе для её построения сравниваются следующие методы:

— RANSAC (RANdom SAmple Consensus) [6] – широко используемый в настоящее время итерационный метод, предложенный в 1981 году Фишлером и Боллесом.

— Метод наименьших квадратов (МНК) [7] – базовый метод регрессионного анализа для оценки неизвестных параметров регрессионных моделей по выборочным данным.

Проведенные тесты подтвердили эффективность предложенных методов для решения задачи сшивки изображений. МНК продемонстрировал высокую скорость работы и неплохое качество сшивки, в то же время RANSAC, являясь алгоритмом устойчивым к выбросам, показал хорошие результаты в критерии качества сшивки, но значительно уступил по скорости предыдущему методу. Имея в качестве преимущества скорость работы, метод МНК предоставляет возможность для улучшения.

2.1.1. Гибридный метод

Роевые алгоритмы стохастической оптимизации [8] – алгоритмы, стремительно набирающие популярность в задачах оптимизации, главным преимуществом которых является возможность параллелизации вычислений. В данном исследовании будет использован алгоритм поиска серых волков (Grey Wolf Optimizer) [9].

Предложим применение роевого алгоритма для нахождения матрицы перспективного преобразования в задаче сшивки изображений для улучшения матрицы, найденной методом МНК. В этом случае на вход роевому алгоритму подается найденная ранее матрица и алгоритм начинает поиск решения, исходя из этих данных. В результате такого использования роевого алгоритма была получена сшивка, представленная ниже (рис. 1). В сравнении с предыдущими сшивками этой пары изображений, наблюдается значительное улучшение: выровнена этажная лестничная площадка, сопоставлены перспективные линии в верхней части изображения. На изображении присутствует артефакт, который не сразу бросается в глаза – несовпадение тени от окна, находящегося за спиной фотографирующего.

Таблица 1

Расчет величины отклонений для каждого метода

Ошибка	RANSAC	МНК	МНК+GWO
Среднеарифметическая (E1)	4,62	5,97	2,84
Среднеквадратическая (E2)	87,41	47,34	11,56
Максимальная (E3)	28,75	16,69	8,55

Рисунок 1: Сшитое изображение по гибридному методу МНК+GWO

Время, затраченное на поиск матрицы по методу МНК, составило 377 600 нс, улучшение при помощи алгоритма GWO – 1 633 000 нс, в сумме получаем 2 010 600 нс. Отметим, что характерное для роевых алгоритмов стохастическое поведение приводит к статистически значимым вариациям времени выполнения, которое при повторных вычислениях матрицы преобразования проявляется в колебаниях порядка 500 000 наносекунд. Таким образом, обеспечивается выигрыш по времени в сравнении с алгоритмом RANSAC, который составил 7 270 900 нс, при этом качество сшитого изображения значительно повысилось.

3. Заключение

Поскольку матрица преобразования является девятимерной, важнейшим критерием для её подбора является точность при работе с многомерными задачами. В то же время задача сшивная изображения вносит ограничение по времени.

В результате данного исследования были определены достоинства каждого метода в контексте решаемой задачи, а также проведено их сравнение с используемым базовым методом сшивания изображений. Реализации данных алгоритмов на языке Java были оформлены в виде библиотеки для удобства использования в дальнейшем исследовании.

Список источников

[1] C. Y. Chen, R. Klette, Image stitching – comparisons and new techniques, Computer Analysis of Images and Patterns (CAIP), (1999) 615–622.

[2] D. Kostylev, A. Tolstikhin, S. Ul'yanov, Development of the complex modelling system for intelligent control algorithms testing, Journal of Physics: Conference Series, (2020) 943–948. doi:10.1088/1742-6596/1515/5/052008.

[3] Р. К. Гонсалес, Р. Э. Вудс, Цифровая обработка изображений, 3-е изд., Москва: Техносфера, 2012.

[4] А. Н. Жуковская, М. Н. Фаворская, Алгоритм сшивания двух изображений при создании панорамных изображений, Материалы XXII международной научно-практической конференции. – Конференция Решетневские чтения, Красноярск: СибГАУ, 2018, стр. 123-124.

[5] S. Leutenegger, M. Chli, R. Y. Siegwart, BRISK: Binary Robust Invariant Scalable Keypoints, Proc. IEEE Int. Conf. Comput. Vis. (ICCV), 2011, pp. 2548–2555. doi:10.1109/ICCV.2011.6126542.

[6] M. A. Fischler, R. C. Bolles, Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography, Commun. ACM, (1981) 381–395. doi:10.1145/358669.358692.
[7] A. M. Legendre, Nouvelles methodes pour la determination des orbites des cometes, Paris: F. Didot, 1805.
[8] А. П. Карпенко, Современные алгоритмы оптимизации: подходы, основанные на природе, 2-е изд., Москва: Издательство МГТУ им. Н.Э. Баумана, 2017.
[9] S. Mirjalili, S. Mirjalili, A. Lewis, Grey wolf optimizer, Advances in Engineering Software, vol. 69, (2014), 46–61. doi:10.1016/j.advengsoft.2013.12.007.

Интеграция метода восстановления текста из PDF-документов с нарушенной кодировкой в систему Dedoc

Михаил Викторович Загородников,[1] Андрей Анатольевич Михайлов[1,2]

[1] *Институт динамики систем и теории управления им. В.М. Матросова Сибирского отделения Российской академии наук, Россия, 664033, г. Иркутск, ул. Лермонтова, 134*
[2] *Институт системного программирования им. В. П. Иванникова Российской академии наук, Россия, 109004, г. Москва, ул. Александра Солженицына, д. 25*

Аннотация

В работе рассматривается проблема восстановления текста из PDF-документов со сложной структурой, нестандартными шрифтами [1] и нарушенной кодировкой [2], где традиционные подходы основанные на OCR [3,4] оказываются неэффективными. Разработан и интегрирован в систему Dedoc модуль, использующий анализ внутренних данных PDF-файлов (CMap-таблиц, метаинформации шрифтов, потоков содержимого). Для распознавания глифов применяется сверточная нейронная сеть, реализованная на PyTorch. Проведён рефакторинг архитектуры, тестирование и интеграция с API системы. Метод показал прирост точности на 45% по сравнению с OCR-подходами Dedoc.

Ключевые слова

PDF, извлечение текста, глифы, CMap, нейронная сеть, PyTorch, Dedoc, OCR, распознавание

1. Введение

Извлечение текста из PDF-документов представляет сложную задачу при наличии нарушенной кодировки шрифтов и сложного оформления. Такие документы трудно обрабатывать традиционными средствами на основе OCR, поскольку те не справляются с искажениями и потерями в информации. Целью работы является разработка модуля, который позволяет восстанавливать текст без применения распознавания изображений, напрямую анализируя внутренние данные PDF-документа.

2. Интеграция метода восстановления текста в Dedoc

В рамках работы был разработан модуль восстановления текста, использующий внутреннюю информацию PDF-документа вместо изображений. Этот модуль позволяет преодолеть ограничения OCR и обеспечить более точное извлечение текстовой информации из файлов с нарушенной кодировкой или нестандартными шрифтами. Модуль интегрирован в систему Dedoc — платформу для интеллектуального анализа документов, что обеспечило его тестирование в реальной архитектуре.

2.1. Описание предложенного подхода

Метод основан на анализе шрифтов и кодировки в PDF-документе (рис 1.), включая:

7th International Workshop on Information, Computation, and Control Systems for Distributed Environments (ICCS-DE 2025), July 7–11, 2025, Irkutsk, Russia
EMAIL: mishazagorodnikov@mail.ru (A. 1); mikhailov@icc.ru (A. 2)
ORCID: 0009-0003-9076-4863 (A. 1); 0000-0003-4057-4511 (A. 2)
© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).
ICCS-DE 2025 Workshop Proceedings
DOI: 10.47350/ICCS-DE.2025.27

- Извлечение встроенных шрифтов и глифов;
- Построение изображений глифов;
- Распознавание глифов сверточной нейросетью (CNN);
- Сопоставление глифов с символами на основе метаданных;
- Восстановление текстового слоя.

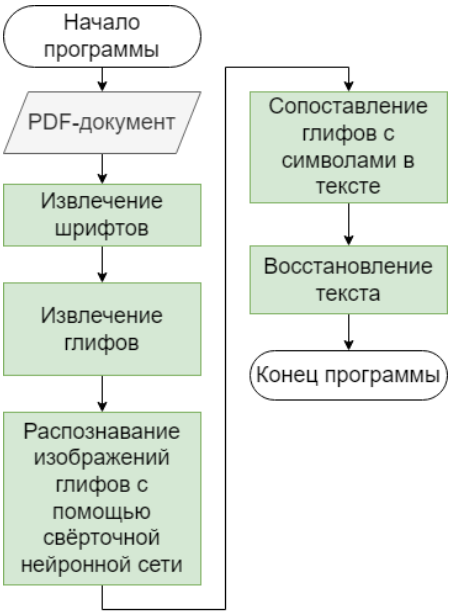


Рисунок 1: Схема метода

Таким образом, удается обойти ограничения OCR-систем и восстановить текст с высокой точностью.

2.2. Архитектура реализации

2.2.1. Основные компоненты

Модуль построен на основе следующих компонентов:

- *PDFReader* — класс, извлекающий все необходимые данные из PDF-документа, к которой относятся, шрифты и их глифы, и восстанавливающий текст по полученной из PDF-документа информации;
- *FontForgeWrapper* — класс для взаимодействия с приложением fontforge, позволяющий работать с шрифтами любых форматов;
- *CNNModel, Model* — свёрточная нейросеть и интерфейс к ней;
- *PDFTextCorrector* — модуль для постобработки текста;
- *PDFBrokenEncodingReader* — класс, наследуемый от класса PDFBaseReader из проекта Dedoc, для взаимодействия с архитектурой системы, использующий Reader, который восстанавливает текст PDF-документ.

2.2.2. Оптимизация и перенос

Для повышения скорости и совместимости с Dedoc были внесены следующие изменения:

- Перевод нейросети с TensorFlow на PyTorch;
- Поддержка пакетной обработки изображений (батчи);
- Переход на временные директории хранения данных;
- Рефакторинг с соблюдением PEP-8 и улучшением читаемости кода.

2.3. Интеграция и тестирование

2.3.1. Интеграция с Dedoc

Для совместимости с архитектурой Dedoc были реализованы интерфейсы can_read и read, создан конфигурационный файл, модифицирован Dockerfile, а также настроена загрузка модели с HuggingFace.

2.3.2. Результаты тестирования

Тестирование проведено на 14 PDF-документах, содержащих нарушенную кодировку, сложный фон [5] и нестандартную структуру [6]. Средняя точность восстановления текста, рассчитанная по метрике Левенштейна, составила 95% против 50% у OCR Dedoc. Это доказывает применимость предложенного метода к "проблемным" документам.

3. Благодарности

Работа выполнена при поддержке государственного задания Министерства науки и высшего образования Российской Федерации (тема № 1023110300006-9).

4. Список литературы

[1] Haralambous Y. Fonts & encodings. — "O'Reilly Media, Inc.", 2007.
[2] Tauber J. K. Character encoding of classical languages // 2019). Digital classical philology: Ancient Greek and Latin in the digital revolution. — 2019. — С. 137—158.
[3] Awel M. A., Abidi A. I. Review on optical character recognition // International Research Journal of Engineering and Technology (IRJET). — 2019. — Т. 6, No 6. — С. 3666—3669.
[4] A detailed review on text extraction using optical character recognition / C. Thorat [и др.] // ICT Analysis and Applications. – 2022. – С. 719-728.
[5] An adaptive thresholding algorithm-based optical character recognition system for information extraction in complex images / D. Akinbade [и др.] // Journal of Computer Science. — 2020. — Т. 16, No 6. — С. 784—801.
[6] DocBed: A multi-stage OCR solution for documents with complex layouts / W. Zhu [и др.] // Proceedings of the AAAI Conference on Artificial Intelligence. Т. 36. – 2022. – С. 12643–12649.

Цифровая трансформация научно-организационной деятельности: теоретические подходы к разработке специализированного фреймворка

Иван Николаевич Игнатьев [1]

[1] *Институт динамики систем и теории управления им. В.М. Матросова Сибирского отделения Российской академии наук, Россия, 664033, г. Иркутск, ул. Лермонтова, 134*

Аннотация

В условиях стремительного развития цифровых технологий возрастает потребность в адаптации научно-организационной деятельности к новым условиям информационного взаимодействия. В статье рассматриваются теоретические основы создания программного фреймворка, направленного на поддержку ключевых процессов научных мероприятий: регистрацию участников, подачу и рецензирование материалов, генерацию отчетной документации. Обоснована актуальность цифровизации данной сферы и выделены основные принципы проектирования цифровых решений, ориентированных на повышение эффективности и прозрачности научных коммуникаций. Проведён анализ существующих решений, выявлены их ограничения и сформулированы требования к новой архитектуре фреймворка.

Ключевые слова

Научные мероприятия, фреймворк, информационные системы, автоматизация, рецензирование, научная коммуникация.

1. Введение

Современная научно-организационная деятельность требует пересмотра традиционных подходов к проведению конференций, семинаров и других научных мероприятий. Рост объёма информации, необходимость работы с удалёнными участниками, высокая нагрузка на организаторов — всё это обостряет потребность в цифровизации процессов взаимодействия между участниками, авторами, редакторами и рецензентами. Тем не менее, во многих научных учреждениях по-прежнему используются ручные методы регистрации, приёма материалов и обмена документами, что приводит к неэффективности, потерям данных и ограничению масштаба проводимых мероприятий [1].

Цифровая трансформация этой сферы требует не просто автоматизации отдельных операций, а создания комплексных решений, учитывающих специфику научной коммуникации. Одним из таких решений является разработка специализированного программного фреймворка, предназначенного для поддержки полного жизненного цикла научного мероприятия — от регистрации участников до формирования итоговой документации. Такой фреймворк должен быть гибким, расширяемым и адаптированным под задачи конкретной организации. Его теоретическая основа должна опираться на современные концепции цифровизации, процессного моделирования и принципов архитектурной модульности, что делает тему актуальной как с прикладной, так и с методологической точки зрения. Программный комплекс

7th International Workshop on Information, Computation, and Control Systems for Distributed Environments (ICCS-DE 2025), July 7–11, 2025, Irkutsk, Russia
EMAIL: ignatievanyaya@mail.ru (Ignatev I.)
ORCID: 0009-0002-5323-9122 (Ignatev I.)
© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).
ICCS-DE 2025 Workshop Proceedings
DOI: 10.47350/ICCS-DE.2025.28

2. Актуальность и проблематика

Научные мероприятия — важный элемент академической жизни, обеспечивающий обмен знаниями, обсуждение результатов исследований и формирование профессиональных связей. Однако организационные процессы, связанные с их проведением, часто остаются на уровне ручного труда. Регистрация участников, приём тезисов, рецензирование и оформление итоговых материалов традиционно выполняются через электронную почту или с использованием формальных офисных инструментов, не обеспечивающих целостности и прослеживаемости данных.

Такие методы сопряжены с рядом проблем: высокий риск ошибок при ручном вводе информации, невозможность эффективно отслеживать статусы заявок и материалов, сложности при взаимодействии между авторами, редакторами и рецензентами. Кроме того, отсутствие централизованной системы хранения и обработки данных приводит к фрагментации информации и затрудняет повторное использование накопленных данных при проведении новых мероприятий.

В контексте глобального перехода к цифровым форматам взаимодействия особенно остро встаёт задача модернизации инструментов, поддерживающих научно-организационные процессы. Цифровизация этой сферы должна не только повысить эффективность работы организаторов, но и обеспечить участникам удобный доступ к функциям подачи материалов, получения обратной связи и сопровождения своего участия в мероприятии.

Однако существующие платформы, такие как EasyChair, ConfTool или Google Forms, часто либо чрезмерно усложнены и ориентированы на международные конференции, либо не обладают необходимой гибкостью и интеграцией с внутренними системами научных учреждений. В результате возникает потребность в разработке специализированных цифровых решений, опирающихся на потребности конкретного академического контекста.

Таким образом, создание фреймворка, ориентированного на автоматизацию и упрощение научно-организационной деятельности, становится важной задачей, сочетающей в себе как прикладные, так и методологические аспекты. Такой фреймворк должен учитывать не только технические требования, но и особенности научного взаимодействия, что требует привлечения междисциплинарных подходов к его проектированию.

3. Теоретические основы проектирования фреймворков для научной сферы

Проектирование фреймворков для поддержки научно-организационной деятельности должно опираться на системный подход к моделированию процессов и архитектурных решений. В основе лежит концепция фреймворка как набора взаимосвязанных компонентов, обеспечивающих решение типовых задач в определённой предметной области. В научной сфере это задачи регистрации, подачи и рецензирования работ, информационного взаимодействия между участниками, а также автоматизированного формирования выходных документов — программ, сборников и сертификатов.

Одним из ключевых понятий, применяемых в теоретическом обосновании необходимости цифровизации, является модель AS-IS / TO-BE. Она позволяет описать текущее состояние процессов (AS-IS), выявить их недостатки и сформировать целевое состояние (TO-BE) — то есть оптимизированную, автоматизированную модель, которую должен реализовать программный продукт. Применение этой модели помогает системно подойти к выбору функциональности, архитектурных компонентов и последовательности внедрения.

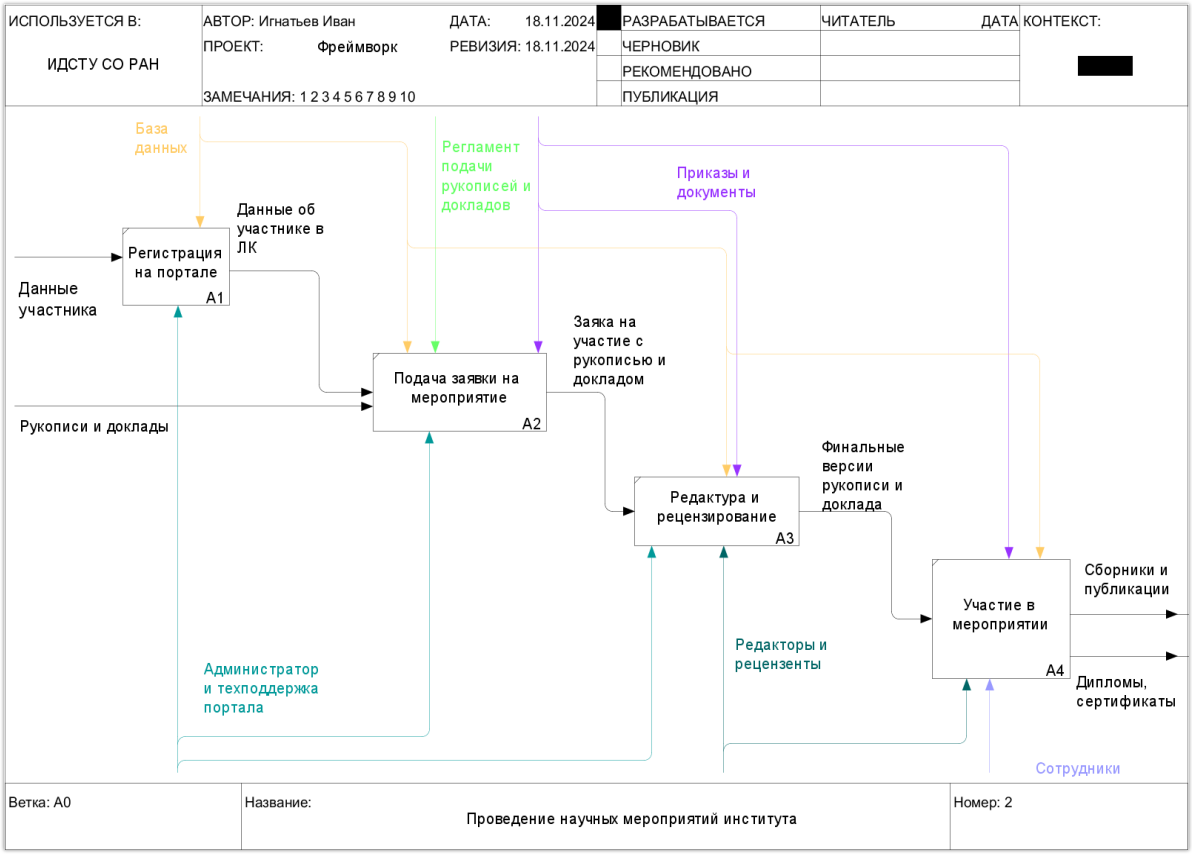


Рисунок 1: Модель TO-BE разрабатываемого фреймворка

Фреймворк, ориентированный на научные мероприятия, должен реализовывать принципы модульности, повторного использования и расширяемости. Это предполагает выделение функциональных блоков (регистрация, подача работ, рецензирование, документы), которые можно адаптировать под разные типы мероприятий без переписывания всей системы. Теоретически такая архитектура позволяет обеспечить масштабируемость решения и его долгосрочную применимость в различных контекстах.

Особое внимание в теоретических моделях уделяется роли пользователя. Система должна быть ориентирована на взаимодействие с разными категориями пользователей: участниками, организаторами, редакторами и рецензентами. Это требует создания ролевой модели и обеспечения контроля доступа к функциям в зависимости от роли, что является стандартной практикой в проектировании систем с распределённой ответственностью.

Таким образом, основой для проектирования фреймворка выступает совокупность методологических принципов цифровой трансформации, архитектурных концепций разработки и прикладных требований научной среды. Такая теоретическая база необходима для создания устойчивого, воспроизводимого и адаптируемого цифрового решения.

4. Методологические требования к архитектуре фреймворка

Архитектура фреймворка, предназначенного для сопровождения научных мероприятий, должна опираться на чётко структурированную модель жизненного цикла мероприятия и соответствовать требованиям гибкости, надёжности и масштабируемости. На методологическом уровне это означает необходимость реализации подхода, ориентированного на процессы (process-oriented design), при котором каждый этап мероприятия представлен как модуль со своими входами, выходами и интерфейсами взаимодействия с другими модулями.

Ключевым требованием является наличие единого пользовательского пространства, в котором участник может создать и редактировать профиль, подать заявку на участие, прикрепить доклад, отслеживать этапы его рассмотрения и получать итоговые документы. Для организаторов, в свою очередь, должно быть предусмотрено административное представление событий: управление регистрацией, распределение заявок, контроль статуса рецензирования и генерация выходной документации. Весь этот функционал должен быть организован вокруг централизованной базы данных и системы ролевого доступа.

Существенным элементом архитектуры становится механизм обработки научных работ. Он должен предусматривать возможность загрузки и замены файлов, назначения редакторов и рецензентов, отображения истории изменений и фиксации рецензий в структурированном виде. Это требует создания подсистемы согласованного документооборота, исключающей рассылку файлов по электронной почте и обеспечивающей прозрачность взаимодействия.

Автоматизация формальной части мероприятий также требует методологического подхода. Формирование программы мероприятия, списков участников, бейджей и сертификатов должно осуществляться на основе шаблонов, связанных с данными из базы. Такая организация позволяет снизить временные затраты и минимизировать ошибки при оформлении документации, особенно при массовых мероприятиях с сотнями участников.

Таким образом, методологическая основа архитектуры фреймворка включает в себя процессный подход, модульную структуру, ориентацию на роли пользователей и централизованную работу с данными. Такой подход обеспечивает устойчивость и воспроизводимость решения, облегчая его адаптацию под нужды конкретных учреждений и расширение функциональности в будущем.

5. Обзор существующих решений и постановка задачи разработки

На сегодняшний день на рынке представлено несколько цифровых платформ, предназначенных для поддержки научных мероприятий, среди которых наиболее известны EasyChair, ConfTool и Microsoft Forms [2]. Эти решения используются для проведения международных конференций, но имеют определённые ограничения в контексте применения в академических учреждениях, особенно в условиях ограниченных ресурсов и отсутствия поддержки интеграции с внутренними системами. Как правило, такие платформы ориентированы на англоязычную аудиторию, требуют оплаты или обладают закрытым исходным кодом, что ограничивает возможности их адаптации.

Простые инструменты вроде Google Forms или Microsoft Excel часто применяются в малых научных коллективах. Однако они не обеспечивают целостности процесса: регистрация, подача материалов, рецензирование и документирование осуществляются разрозненно, что приводит к увеличению нагрузки на организаторов и снижению прозрачности. Отсутствие механизма авторизации, централизованного хранения данных, истории изменений и автоматического формирования отчётных материалов делает такие решения временными и неэффективными в долгосрочной перспективе.

Исходя из вышеуказанных недостатков, формируется чёткая задача — разработать специализированный фреймворк, ориентированный на потребности российских научных организаций. Такой фреймворк должен быть открытым, модульным, легко расширяемым и соответствовать требованиям цифровой трансформации научной сферы. Его архитектура должна предусматривать многоуровневую ролевую модель, интеграцию с локальными системами (например, базами участников, архивами публикаций), поддержку многоязычного интерфейса, а также возможности автоматического формирования документов по шаблонам.

Особое внимание в постановке задачи уделяется удобству использования и минимизации ручного труда. Пользовательский интерфейс должен быть интуитивно понятным как для участников, так и для организаторов мероприятия. Система должна позволять повторное использование данных, исключая необходимость ввода одной и той же информации при каждом новом мероприятии, а также обеспечивать связь между автором и его работами даже при поздней регистрации [3].

Таким образом, обоснование необходимости собственного фреймворка базируется на анализе текущих решений и их несоответствия требованиям локальных научных институтов. Задача проектирования заключается не только в технической реализации, но и в формализации процессов научной коммуникации, адаптированных под цифровую среду.

6. Заключение

Цифровая трансформация научно-организационной деятельности представляет собой важное направление развития современной академической инфраструктуры. Как показано в теоретическом анализе, существующие инструменты либо не охватывают весь жизненный цикл научного мероприятия, либо оказываются недостаточно гибкими и адаптированными к специфике конкретных учреждений. Это создаёт необходимость в разработке специализированного фреймворка, способного обеспечить комплексную поддержку процессов — от регистрации участников до рецензирования и выпуска итоговой документации.

Теоретические основания проектирования такого фреймворка включают в себя принципы модульности, ролевого доступа, централизованной работы с данными и ориентации на повторное использование информации. Использование моделей бизнес-процессов, таких как AS-IS/TO-BE, позволяет системно подойти к цифровизации научных мероприятий, устраняя узкие места традиционных методов организации и обеспечивая прослеживаемость всех этапов.

Таким образом, проектирование и внедрение фреймворка для научных мероприятий — это не просто задача автоматизации, а шаг к формированию устойчивой цифровой среды, способной обеспечить более высокое качество научной коммуникации, снизить нагрузку на организаторов и повысить прозрачность академических процедур. Разработка такого решения требует не только технических, но и методологических усилий, что подчёркивает актуальность темы как в прикладном, так и в научном контексте.

7. Список использованных источников

[1] Игнатьев И.Н., Башарина О. Ю. Методы и средства веб-поддержки научно-организационной деятельности // Современные наукоемкие технологии. — 2024. — № 11. — С. 33—39. DOI: 10.17513/snt.40207.

[2] Multidisciplinary Digital Publishing Institute [Электронный ресурс]. URL: https://www.mdpi.com/ (дата обращения: 10.06.2025).

[3] EasyChair [Электронный ресурс]. URL: https://easychair.org/ (дата обращения: 10.06.2025).

Защита от незаметных атак на модели NLP с использованием омоглифов

Андрей Александрович Кузнецов, [1] Андрей Анатольевич Михайлов [1,2]

[1] *Институт динамики систем и теории управления им. В.М. Матросова Сибирского отделения Российской академии наук, Россия, 664033, г. Иркутск, ул. Лермонтова, 134*
[2] *Институт системного программирования им. В. П. Иванникова Российской академии наук, Россия, 109004, г. Москва, ул. Александра Солженицына, д. 25*

Аннотация
В работе рассматривается проблема атак на системы обработки естественного языка (NLP), реализуемых с применением омоглифов — символов, которые визуально идентичны или схожи, но обладают различными значениями. Предлагается новый подход, основанный на применении модели трансформера T5, обученной для корректировки орфографических ошибок на синтетически созданном корпусе данных, в котором символы были заменены на их омоглифы. Результаты эксперимента демонстрируют высокую точность модели в исправлении атакованных примеров

Ключевые слова
обработка естественного языка, защита от атак на языковые модели, омоглифы

1. Введение

С развитием глубоких нейронных сетей в области обработки естественного языка стало очевидным, что подобные системы уязвимы для атак, искажающих входные данные. В отличие от атак в области компьютерного зрения, атаки на языковые модели в NLP могут быть более заметными, например, нарушение орфографии или перефразирование. Возможность изменить текст, не изменяя его визуально, может быть использована злоумышленниками для обхода механизмов фильтрации контента, некорректного машинного перевода, ухудшения качества запросов и препятствия индексации поисковых систем [1].

2. Обзор работ

Современные работы в области атак на модели обработки естественного языка [2,3,4] предлагают различные способы защиты, которые можно разделить на следующие подходы:

- Система, основанная на правилах замены (rule-based recovery). Суть заключается в символьном проходе по входной строке и замене символа, отличного от языка оригинала. Проблема данного решения заключается в том, что подобная система строится только для одного языка.
- Состязательное обучение (Adversarial Training). Базовый вариант защиты NLP моделей от состязательных атак – добавление «враждебных» примеров в процесс обучения модели. Недостатками данного решения являются увеличение затрат на обучение исходной модели и неполноценная защита от состязательных примеров.

7th International Workshop on Information, Computation, and Control Systems for Distributed Environments (ICCS-DE 2025), July 7–11, 2025, Irkutsk, Russia
EMAIL: viirtualkuz@yandex.ru (A. 1); mikhailov@icc.ru (A. 2)
ORCID: 0000-0003-4057-4511 (A. 2)
©2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).
ICCS-DE 2025 Workshop Proceedings
DOI: 10.47350/ICCS-DE.2025.29

- Создание векторных представлений для каждого символа. Следующая группа работ предлагает создавать векторные представления для каждого символа и, сравнивая с соседями, находить ближайшие аналоги среди других представлений.
- Коррекция визуально схожих слов. В работе по коррекции визуально схожих слов в арабском языке упоминается способ на основе двунаправленных рекуррентных нейронных сетей (BiLSTM), анализирующий контекст слова как в прямом, так и в обратном направлении.
- Коррекция и детекция омоглифов в тексте на основе свёрточных нейронных сетей (CNN). В процессе работы генерируется изображение на основе входного текста, в котором могут присутствовать омоглифы, затем оно поступает на вход модели для последующей детекции и коррекции.

3. Метод коррекции омоглифов

Для решения задачи коррекции омоглифов предлагается использовать языковую модель на базе трансформера, а именно модель семейства T5. Главной её особенностью является то, что она предобучена на большом массиве текстовых данных, покрывающем объёмный спектр NLP задач и используется для дообучения под конкретные цели. Наш метод, T5-GlyF (Glyph Fixing), основан на проекте SAGE команды SberDevices, в котором модель на базе T5 выполняет коррекцию орфографических и пунктуационных ошибок на русском и английском языках. Для создания большого обучающего корпуса авторами был разработан алгоритм аугментации данных, который имитировал ошибки человека в правописании.

4. Эксперименты и результаты

- Разработан алгоритм генерации возмущений с использованием омоглифов.
- Сгенерирован датасет с помощью разработанного алгоритма.
- Проведено обучение модели на сгенерированном датасете.
- Результаты обучения демонстрируют высокую точность модели в исправлении атакованных примеров: 99% для английского языка и 96% при добавлении русского языка.

5. Заключение

Искажение входных данных на основе омоглифов в области NLP является серьёзной угрозой для языковых моделей. Решение на основе трансформера, T5-GlyF, оказалось отличным вариантом системы защиты против атак с использованием омоглифов согласно проведённым тестированиям на реальных NLP задачах. Наша система защиты успешно справляется с атакованными с помощью омоглифов входными данными, а в случае, когда никакой атаки нет, эффективность целевой языковой модели остаётся неизменной.

6. Благодарности

Работа выполнена при поддержке государственного задания Министерства науки и высшего образования Российской Федерации (тема № 1023110300006-9).

7. Список литературы

[1] Boucher N. et al. Bad characters: Imperceptible nlp attacks //2022 IEEE Symposium on Security and Privacy (SP). – IEEE, 2022. – С. 1987-2004.

[2] Eger S. et al. Text processing like humans do: Visually attacking and shielding NLP systems //arXiv preprint arXiv:1903.11508. – 2019.
[3] Almanea M. M. Automatic methods and neural networks in Arabic texts diacritization: a comprehensive survey //IEEE Access. – 2021. – Т. 9. – С. 145012-145032.
[4] Han X. et al. Text adversarial attacks and defenses: Issues, taxonomy, and perspectives //Security and Communication Networks. – 2022. – Т. 2022. – №. 1. – С. 6458488.

Планирование на основе GOAP в задаче поиска объекта группой мобильных роботов

Алексей Линявский [1,2] и Надежда Нагул [2]

[1] *Иркутский государственный университет путей сообщения, ул. Чернышевского, 15, Иркутск, 664074, Россия*
[2] *ИДСТУ СО РАН им. В.М. Матросова, ул. Лермонтова, 134, 664033, Иркутск, Россия*

Аннотация
В докладе рассматривается подход к автоматическому планированию поведения автономного мобильного робота на основе архитектуры GOAP (Goal-Oriented Action Planning). Показаны примеры формализации возможных действий роботов для применения GOAP, приведены примеры получаемых планов для решения задачи поиска и исследования объектов, расположенных в некоторой области с препятствиями.

Ключевые слова
Автономные мобильные роботы, автоматическое планирование, принятие решений

1. Введение

Goal-Oriented Action Planning – широко применяемая в компьютерных играх архитектура планирования действий так называемых NPC (Non-Player Character), позволяющая достичь видимости естественного интеллектуального поведения персонажей [1, 2]. GOAP был представлен в 2005 году как модификация планировщика STRIPS (Stanford Research Institute Problem Solver [3]), направленная на ускорение построения планов в динамических средах и их упрощение для более эффективной реализации [4]. Как и в других планировщиках, суть GOAP состоит в составлении плана достижения выбранной цели из доступного в текущей ситуации ограниченного набора действий, имеющих условия для их выполнения и эффекты, наступающие после выполнения. В отличие от STRIPS, каждому действию была назначена стоимость, которая использовалась в качестве эвристики для алгоритма A* для поиска в пространстве состояний мира. Кроме того, GOAP использует процедурные предусловия, с помощью которых система в процессе выполнения плана может использовать более сложную логику, например, вызывать функции планирования пути или распознавания объекта.

Преимуществом GOAP является возможность реализации сложных поведений объектов, для которых неприменимы традиционные алгоритмы планирования. Известным подходом к планированию и принятию решений является описание задач в языке PDDL и применение специализированных солверов. Однако разработка PDDL-планировщика представляет собой трудозатратную работу, при этом его функции могут оказаться невостребованными в решаемых задачах. В докладе рассматривается возможность использования подхода на основе GOAP для управления автономными мобильными роботами в процессе исследования заданной области и некоторых объектов в ней. В то время как для решения части задач, составляющих миссию роботов, например, для планирования пути и назначения заданий, используются известные алгоритмы [5, 6], принятие решений о той или иной последовательности высокоуровневых действий осуществляется с помощью GOAP.

7th International Workshop on Information, Computation, and Control Systems for Distributed Environments (ICCS-DE 2025), July 7–11, 2025, Irkutsk, Russia
EMAIL: inetlsec@mail.ru (A. 1); sapling@icc.ru (A. 2)
ORCID: 0009-0008-0412-9158 (A. 1); 0000-0003-1439-3274 (A. 2)
©2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).
ICCS-DE 2025 Workshop Proceedings
DOI: 10.47350/ICCS-DE.2025.30

2. Goal-Oriented Action Planning

В качестве входных данных планировщик GOAP принимает: множество агентов, множество возможных действий каждого агента, множество задач, начальные состояния и цели. Цели представляют собой состояния, которое агент должен достичь из начального или текущего состояния (рис. 1). В общем случае, состояния равны тогда и только тогда, когда наборы параметров состояний совпадают, а также совпадают значения параметров внутри наборов.

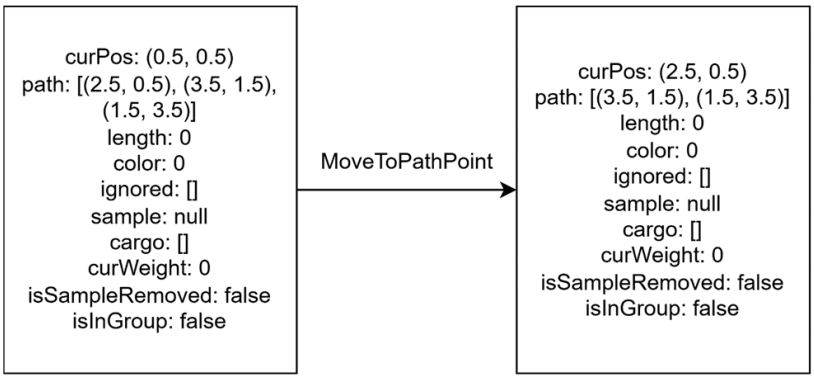


Рисунок 1: Пример состояний робота. Каждая строка представляет собой некоторый параметр.

Каждое действие в GOAP описывается следующими компонентами (рис. 2):

- вес действия - используется для оценки релевантности действия в момент планирования;
- предусловия - условия, необходимые для выполнения действия;
- эффекты - состояние, которое агент принимает после выполнения действия;
- алгоритм реализации действия.

Рисунок 2: Описание действия в GOAP

Представление в GAOP предварительных условий и эффектов действий в виде массивов переменных состояния системы фиксированного размера позволяет быстрее найти действие, которое могло бы уменьшить разницу между начальным и целевым состоянием посредством прямого сравнения массивов. План из действий составляется таким образом, чтобы последовательность действий привела начальное состояние к конечному - цели. Для построения плана в GOAP, как и STRIPS, используется обратное планирование, так называемое планирование от цели [1].

3. Задача исследования объектов группой АМР

Рассмотрим следующую миссию для группы из трех автономных мобильных роботов (АМР). Пусть задана некоторая область, и построены маршруты движения каждого из АМР для совместного сканирования области в поисках мест скопления некоторых предметов, возможно, представляющих интерес для дальнейшего исследования. При обнаружении скопления образцов робот может отделиться от группы для самостоятельного их исследования, либо вся группа проводит исследование совместно. Приведем несколько возможных действий каждого АМР группы:

1. **Движение по пути группой:** группа АМР движется к точке point в path.
GroupAction: MoveToPathPointGrouped(point)

*Weight: dist(curPos, point) / (10 * (isInGroup ? 1.5 : 1))*
Preconditions:
isInGroup = true
!path.isEmpty()
pathExists(curPos, point)
!ingored.contains(point)
Effects:
curPos = point

2. **Выход из группы:** АМР перестаёт двигаться в группе и начинает работать отдельно от группы из точки point.

Action: LeaveGroup(point)
Weight: (isInGroup ? 0.25 : 1)
Preconditions:
isInGroup = true
curPos = point
Effects:
isInGroup = false

3. **Исследование образца:** АМР выполняет обследование образца в точке point.

Action: ScanSample(point)
Weight: 1
Preconditions:
isInGroup = false
color = 0
length = 0
Effects:
color != 0
length != 0
sample = getSampleByColorLength(color, length)

4. **Игнорирование образца:** АМР записывает образец sample в игнорируемые ignored.

Action: Ignore()
Weight: getWeight(sample)
Preconditions:
sample = null
Effects:
isSampleRemoved = true
ignored += [curPos]

5. **Поднятие образца:** АМР собирает образец sample в свой грузовой отсек.

Action: TakeSample(sample)
*Weight: getWeight(sample) * curWeight*
Preconditions:
sample != null
curWeight + getWeightCost(sample) <= 1
Effects:
cargo += [sample]
curWeight += getWeightCost(sample)
sample = null

В зависимости от текущего состояния системы, присвоенные действиям веса влияют на выбор одной из возможных последовательностей действий, формируя таким образом план. Для построения плана используется алгоритм A*. На рис. 3 цветом выделен реализуемый план, выбранный по параметрам исследованного образца и веса каждого из действий.

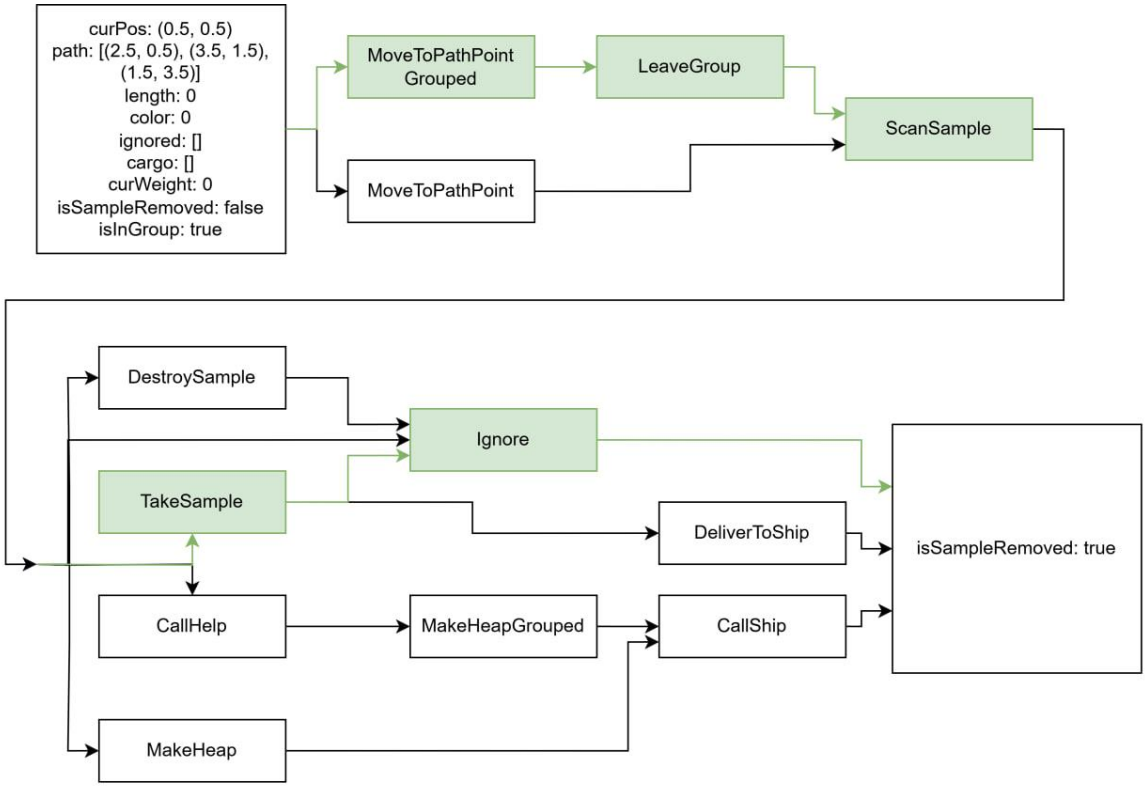


Рисунок 3: Возможные планы, полученные GOAP для группы AMP. Выбранный план помечен зеленым цветом.

4. Заключение

В докладе описаны принципы использования GOAP для планирования высокоуровневых действий мобильных роботов. Построение плана проводилось для индивидуального робота, однако в дальнейшем будут исследованы возможности применения GAOP с кооперирующимися агентами [7], а также рассмотрены более сложные подходы к планированию, такие как HTN (Hierarchical Task Network), в котором используются принципы иерархии задач и их декомпозиция [4]. Описанная в докладе миссия будет реализована в робототехническом комплексе TEMAR для тестирования предложенного подхода.

Список литературы

[1] J. Orkin. Applying goal-oriented action planning to games. AI Game Programming Wisdom, vol. 2, no. 2004, pp. 217–227, 2004.
[2] M.Ç. Uludağlı, K. Oğuz. Non-player character decision-making in computer games. Artif Intell Rev 56, 14159–14191 (2023). https://doi.org/10.1007/s10462-023-10491-7
[3] R. E. Fikes, N. J. Nilsson. STRIPS: A new approach to the application of theorem proving to problem solving, Artificial intelligence, vol. 2, no. 3-4, pp. 189–208, 1971.
[4] X. Neufeld, S. Mostaghim, D. L. Sancho-Pradel and S. Brand. Building a Planner: A Survey of Planning Systems Used in Commercial Video Games. IEEE Transactions on Games, 11, 2 (2019), 91-108. DOI: 10.1109/TG.2017.2782846.
[5] J. Li, Y. Zhao, Y. Shen. Multi-Factor Task Assignment and Adaptive Window Enhanced Conflict-Based Search: Multi-Agent Task Assignment and Path Planning for a Smart Factoryб Electronics, 14, 5 (2025), 842. DOI: https://doi.org/10.3390/electronics14050842.

[6] J. Tang, H. Ma. Mixed Integer Programming for Time-Optimal Multi-Robot Coverage Path Planning with Efficient Heuristics, Proceedings of the International Symposium on Combinatorial Search, 17 (2024), 289-290. DOI: https://doi.org/10.1609/socs.v17i1.31585.

[7] G. Boeda. Multi-agent cooperation in games with goal oriented action planner: use case in WONDER prototype project, Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment. 17, 1 (2021), 204-207. DOI: https://doi.org/10.1609/aiide.v17i1.18909.

Построение алгоритмов машинного обучения для детектирования сайтов-мошенников

Лукьянов Дмитрий А. , Лапина Мария А. , Лапин Виталий Г. , Кучеров Николай Н.

Северо-Кавказского федерального университета, Пушкина 1, 355000, Ставрополь

Аннотация

В настоящее время с развитием информационных технологий цифровые преступления стали серьёзной угрозой для конфиденциальной информации и данных пользователей. Искусственный интеллект способен анализировать параметры сайтов и определять наличие угроз для информации. В ходе выполнения исследования были разработаны методы машинного обучения по обнаружения фишинговых сайтов, построены схемы, которые позволяют моделям машинного обучения правильно преобразовывать данные для подачи их в модели. В результате работы разработаны методы поиска фишинговых атак с использованием моделей машинного обучения, которые были протестированы на имеющихся данных, на основе полученных результатов построены графики изменения точности обнаружения нелегитимных сайтов от изменения настроек моделей.

Ключевые слова:

машинное обучения, фишинг, сайт-мошенник, KNIME, киберпреступность, кибер-атаки, поиск уязвимостей, искусственный интеллект.

1. Введение

Фишинг – тип интернет-мошенничества, целью которого является получение персональных данных пользователя: пароли, данные кредитных карт, банковские счета, почты, паспортные данные и другая конфиденциальная информация [1]. Фишинговыми атаками могут являться: поддельные письма на электронную почту от банков, провайдеров с просьбой обновить персональные данные.Термин “Фишинг” впервые прозвучал 2 января 1996 года. Упоминание о данном явлении впервые появилось в группе новостей “Usenet” под названием “AOHello” [2]. В январе 1996 года была проведена первая в истории фишинговая атака.

Рисунок 1 – План фишинговой атаки.

7th International Workshop on Information, Computation, and Control Systems for Distributed Environments (ICCS-DE 2025), July 7–11, 2025, Irkutsk, Russia
EMAIL: ilia.alekseev.kenig@gmail.com (A. 1); mlapina@ncfu.ru (A. 2); vitlx@yandex.ru (A. 3) ; nik.bekesh@gmail.com (A. 4)
ORCID: 0009-0009-7203-0309 (A. 1); 0000-0001-8117-9142 (A. 2); 0000-0002-0611-7002 (A. 3); 0000-0003-0337-0093 (A. 4)
© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).
ICCS-DE 2025 Workshop Proceedings
DOI: 10.47350/ICCS-DE.2025.31

В настоящее время из-за развития компьютерных технологий и широкого распространения интернета защита от фишинговых атак стала надёжнее, но хакеры придумывают новые методы атак. Выделяют следующие типы фишинговых атак: целевой фишинг [3], vishing [4], клон-фишинг [5], фишинг в поисковых системах [6].

Существующие методы защиты (фильтрация URL [7], двухфакторная аутентификация [8], обучение пользователей [9], браузерные расширения [10]) не обеспечивают полной безопасности. В связи с этим актуальной задачей становится разработка автоматизированных систем на основе машинного обучения (МО), способных анализировать параметры сайтов и выявлять фишинговые угрозы.

2. Машинное обучение

Машинное обучение (МО) – это область искусственного интеллекта, которая позволяет ему находить закономерности в данных и делать прогнозы. Программа анализирует данные, находит в них последовательности и закономерности, обучается и применяет результаты на практике, прогнозируя данные.

Для работы с машинным обучением необходим набор данных (датасет) [11], в котором внесены основные характеристики сайтов и результат. Данный датасет создан для обнаружения фишинговых сайтов, которые маскируются под легитимными ресурсами, чтобы украсть данные пользователей.

Для работы с выбранным датасетом необходимо выбрать аналитическую платформу, которая предоставляет возможность работать с машинным обучением. В данном исследование для работы была выбрана аналитическая платформа с открытым кодом – KNIME [12].

3. Модели машинного обучения

Для работы с моделями необходимо построить схему, позволяющую анализировать данные. Обобщающая модель данных моделей показана на рисунке 2.

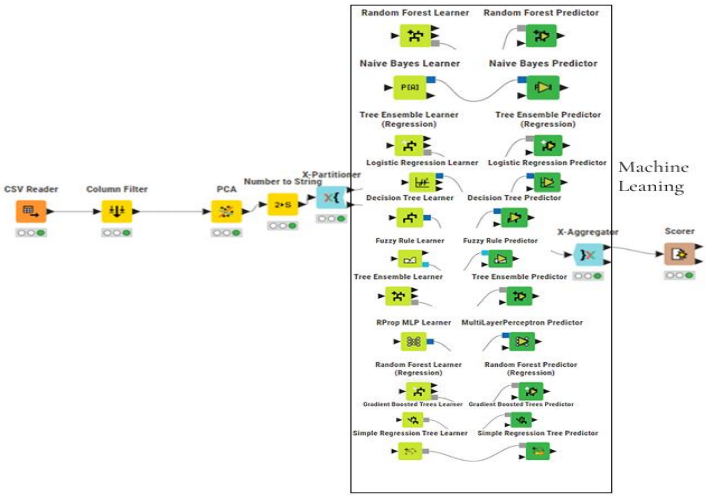


Рисунок 2 – Модели машинного обучения.

Наилучшие результаты показали ансамблевые методы:

1. Random Forest [13] — точность 96,6% (глубина дерева 20).
2. Gradient Boosted Trees [14] — точность 97,34% (глубина дерева 10).
3. Tree Ensemble [15] — точность 96,59% (глубина дерева 20).

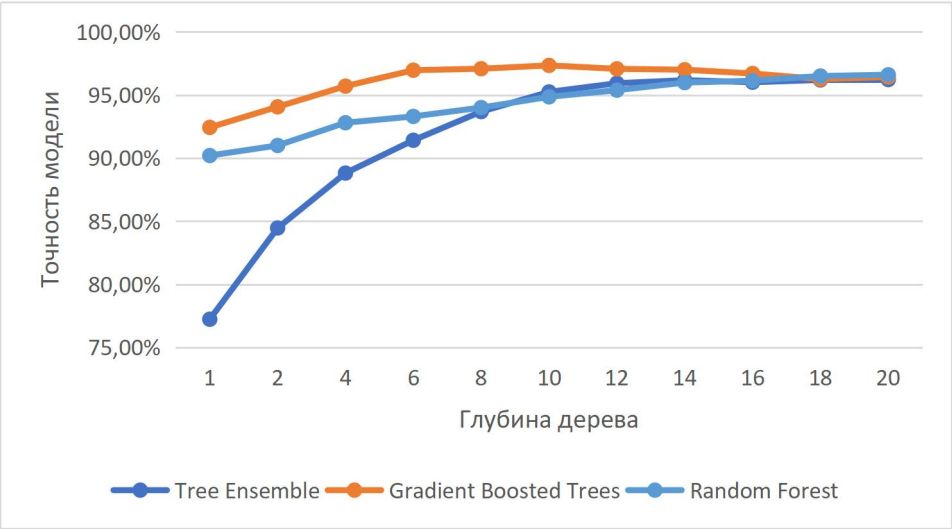


Рисунок 3 - Обобщающий график результатов.

Ансамблевые методы (Random Forest, GBM, Tree Ensemble) показали лучшие результаты благодаря:

1. Способности анализировать сложные нелинейные зависимости в данных.
2. Устойчивости к переобучению за счет бэггинга/бустинга.
3. Автоматическому отбору значимых признаков, что критично для задач с множеством параметров.

4. Вывод

Исследование показывает, что использование машинного обучения в обнаружении потенциально опасных сайтов и детектировании фишинговых атак перспективно, потому что обучение и дальнейшее использование искусственного интеллекта позволит развивать данную сферу, создавая и усовершенствуя модели машинного обучения, позволяя им быстрее и точнее находить закономерности в параметрах и прогнозировать легитимность сайтов, писем и уведомлений [16]

5. Благодарность

Исследование выполнено за счет гранта Российского научного фонда № 25-71-30007, https://rscf.ru/project/25-71-30007/

6. Список литературы

[1] Завьялов Александр Николаевич Интернет-Мошенничество (Фишинг): Проблемы Противодействия и Предупреждения // Baikal Research Journal. 2022. №2. URL: https://brj-bguep.ru/reader/article.aspx?id=25141

[2] History of Phishing // Phishing URL: https://www.phishing.org/history-of-phishing

[3] Cisco. Целевой фишинг. // ТЕХНО Н URL: https://technon.ru/upload/pdf/ironport_targeted_phishing.pdf

[4] Афанасьева Н. С., Елизаров Д. А., Мызникова Т. А. Классификация фишинговых атак и меры противодействия им //Инженерный вестник Дона. – 2022. – №. 5 (89). – С. 169-182.

[5] Фишинговые письма: как их распознать и не стать их жертвой // Kaspersky URL: https://www.kaspersky.ru/resource-center/preemptive-safety/phishing-prevention-tips

[6] Как работает фишинг через поисковые системы? // keeper URL: https://www.keepersecurity.com/blog/ru/2023/04/12/what-is-search-engine-phishing/

[7] Making the world's information safely accessible // Google Safe Browsing URL: https://safebrowsing.google.com
[8] Here's why people are saying two-factor authentication isn't perfect // Digitaltrends URL: https://www.digitaltrends.com/computing/why-two-factor-authentication-isnt-perfect/
[9] Корнюхина С. П., Лапонина О. Р. Исследование возможностей алгоритмов глубокого обучения для защиты от фишинговых атак //International Journal of Open Information Technologies. – 2023. – Т. 11. – №. 6. – С. 163-174.
[10] Boyle P., Shepherd L. A. Mailtrout: a machine learning browser extension for detecting phishing emails //34th British HCI Conference. – BCS Learning & Development, 2021. – С. 104-115.
[11] CyberSecurity : BookMyShow ads URL Analysis // kaggle URL: https://www.kaggle.com/datasets/shibumohapatra/book-my-show
[12] KNIME Analytics Platform // KNIME URL: https://www.knime.com/knime-analytics-platform
[13] Biau G., Scornet E. A random forest guided tour //Test. – 2016. – Т. 25. – №. 2. – С. 197-227.
[14] Natekin A., Knoll A. Gradient boosting machines, a tutorial //Frontiers in neurorobotics. – 2013. – Т. 7. – С. 21.
[15] Асито Ф. Предсказательная аналитика с KNIME / пер. с англ. А. Ю. Гинько. – М.: ДМК Пресс, 2025. – 360 с.: ил.
[16] Отахонов А. А. Обнаружение и оценка фишинговых url-адресов с использованием алгоритмов машинного обучения //Al-Farg'oniy avlodlari. – 2024. – №. 4. – С. 382-390.

Автоматизация DevOps в распределенной вычислительной среде применительно к научным приложениям

Евгений Николаев

Институт динамики систем и теории управления им. В.М. Матросова Сибирского отделения Российской академии наук, Россия, 664033, г. Иркутск, ул. Лермонтова, 134

Аннотация

Поддержка DevOps применительно к научным приложениям в гетерогенной распределенной вычислительной среде (ГРВС) является сложной и актуальной проблемой. Известен широкий спектр инструментов, реализующих отдельные этапы DevOps. К сожалению, готовое решение по их согласованному автоматизированному использованию применительно к научным приложениям, рабочие процессы которых выполняются в ГРВС с помощью систем управления прохождением заданиями, автору не известно. В этом контексте предложен подход к автоматизации процессов непрерывной интеграции и контейнеризации прикладного и системного программного обеспечения, используемого при выполнении рабочих процессов сервис-ориентированных научных приложений на вычислительном кластере. В рамках подхода применяется связка систем Ansible, Docker и Kubernetes. Эксперименты по апробации подхода показали его работоспособность и эффективность, а также существенное сокращение времени развертывания и настройки необходимого прикладного и системного программного обеспечения.

Ключевые слова

Сервис-ориентированные приложения, рабочие процессы, гетерогенная распределенная вычислительная среда, DevOps

1. Введение

Научная проблема поддержки DevOps применительно к научным приложениям в гетерогенной распределенной вычислительной среде (ГРВС) заключается в наличие существенных накладных расходов на доставку, развертывание, настройку, тестирование и запуск программного обеспечения (ПО) в ручном режиме [1]. Более того эти расходы стремительно возрастают при увеличении числа узлов среды. Ключевыми аспектами, влияющими на степень сложности решения данной проблемы, являются следующие:

- различия в характеристиках (архитектуре, производительности, размере оперативной и дисковой памяти, интерконнекте и др.) и программном окружении (ОС, версиях библиотек, зависимостях и т. п.) узлов среды;
- снижение надежности и отказоустойчивости программно-аппаратных средств среды, на которых ПО развернуто в ручном режиме, и в следствии этого возрастание риска возникновения нештатных аварийных ситуаций при выполнении этого ПО;
- специфика решения задач для каждой предметной области с помощью разворачиваемого ПО.

Разработка автоматизированной технологии управления жизненным циклом ПО в ГРВС позволит достичь следующих эффектов: существенно снизить накладные расходы, связанные с

7th International Workshop on Information, Computation, and Control Systems for Distributed Environments (ICCS-DE 2025), July 7–11, 2025, Irkutsk, Russia
EMAIL: zhenya_nikolaev_1995@list.ru

© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

ICCS-DE 2025 Workshop Proceedings
DOI: 10.47350/ICCS-DE.2025.32

ручной доставкой, развертыванием, настройкой, тестированием и запуском ПО; обеспечить воспроизводимость экспериментов в ГРВС; повысить надежность и отказоустойчивость программно-аппаратных средств узлов среды; адаптировать процессы DevOps под специфику решения задач а разных предметных областях.

В рамках данного исследования осуществлены выбор и интеграция DevOps-инструментов, определена технология непрерывной интеграции и контейнеризаци ПО, используемого при выполнении рабочих процессов сервис-ориентированных научных приложений на вычислительном кластере, и автоматизирована настройка ресурсов вычислительного кластера.

2. Методы и средства

Разработка научных приложений и их рабочих процессов осуществляется с помощью инструментального комплекса FDE-SWFs [2]. Для реализации отдельных этапов DevOps применительно к системному и прикладному ПО, необходимому для выполнения рабочих процессов, может использоваться ряд известных инструментов [3]. При этом их комплексное применение, как правило, определяется в каждом случае индивидуально, исходя из спектра поддерживаемых этапов DevOps, используемых вычислительных ресурсов, применяемого ПО и др. В рамках данного исследования выбрана связка систем «Ansible [4] + Docker [5] + Kubernetes [6]», использование которой рекомендовано в [1]. Настройка узлов, размещение на них ПО и модификация конфигурационных файлов осуществляется с помощью Ansible. Система Docker применяется для создания контейнеров, запуск которых и управление ими выполняется системой Kubernetes. В общем случае подготовка и проведение экспериментов в рамках предложенного подхода включает следующие основные этапы (рис. 1):

1. Подготовка узлов ГРВС и сбор информации о них с помощью системы Ansible, включая установку Kubernetes.
2. Интеграция системного ПО из Git-репозитория в базовый образ (БО) системы Docker. БО создается под ОС Linux. Он содержит компоненты диспетчера и СУПЗ.
3. Построение набора специализированных образов $СО_1$, …, $СО_n$ путем адаптации БО к особенностям используемых ресурсов среды путем настройки конфигурационных файлов с помощью системы Ansible. Отдельный образ содержит управляющий компонент центрального узла среды – Central Manager (CM). CM управляет рабочими узлами среды – Execute.
4. Интеграция прикладного ПО (модулей приложения, необходимых внешних библиотек из Git-репозитория и компонентов системы мониторинга вычислительных ресурсов и процессов), подключение временной базы данных (ВБД) для хранения промежуточных расчетов в $СО_1$, …, $СО_n$ и сборка на их основе интегрированных образов (ИО) $ИО_1$, …, $ИО_n$.
5. Размещение $ИО_1$, …, $ИО_n$ на ресурсах с помощью системы Ansible.
6. Проверка соединений и тестирование ресурсов.
7. Запуск контейнеров на основе $ИО_1$, …, $ИО_n$, их конфигурирование и оркестрация на ресурсах с помощью системы Kubernetes.
8. Запуск и выполнение рабочего процесса диспетчером рабочих процессов FDE-SWFs в ГРВС и его мониторинг.
9. Сохранение результатов расчетов и освобождение ресурсов.

3. Экспериментальный анализ

В качестве примера применения предложенного подхода к автоматизации DevOps рассматривается подготовка и проведение экспериментов по решению задачи глобального анализа чувствительности [7]. Приложение для решения задачи включает один научный рабочий процесс (рис. 2), состоящий из девяти параметров и реализуемый с помощью пяти модулей, представленных в виде сервисов. Вычислительная среда включает три узла со следующими характеристиками: операционная система – Ubuntu 22.04, размер оперативной памяти – 4012 Мб, число процессоров – 5, объем видеопамяти – 16 Мб. Управление заданиями осуществляется с помощью СУПЗ HTCondor [8]. Сопоставление оценок накладных расходов с накопительным

итогом на подготовку и проведение экспериментов, приведенных на рис. 2, показывает их сокращение за счет автоматизации DevOps в сравнении с ручным режимом более чем в два с половиной раза. Наибольшее сокращение достигается на этапах настройки конфигурационных файлов HTCondor и обеспечения сетевого взаимодействия между узлами кластера.

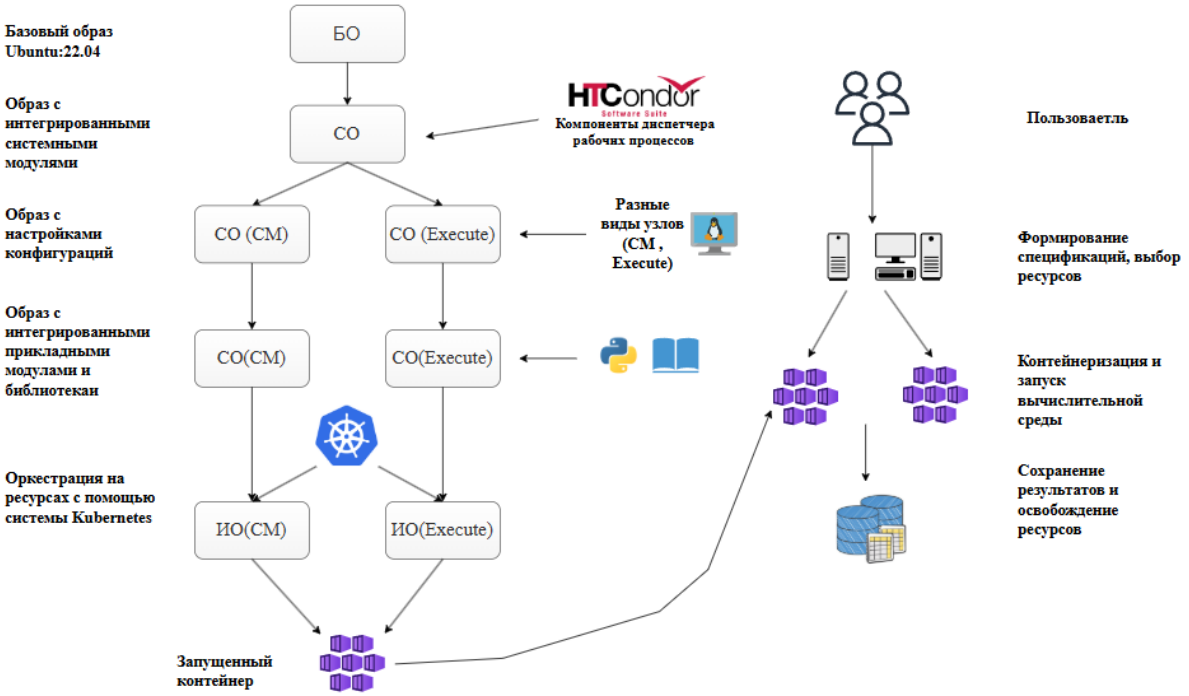


Рисунок 1: Основные этапы подготовки и проведения экспериментов

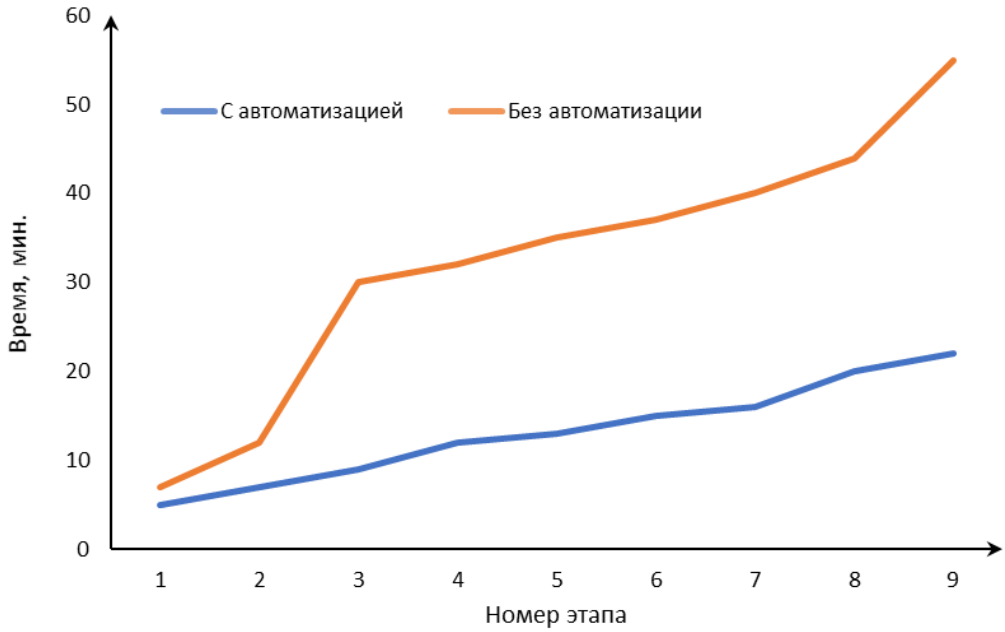


Рисунок 2: Оценка накладных расходов с накопительным итогом на подготовку и проведение экспериментов

4. Заключение

Предложен подход к автоматизации непрерывной интеграции и контейнеризации ПО, необходимого для выполнения рабочих процессов сервис-ориентированных приложений, с помощью связки «Ansible + Docker + Kubernetes». Работоспособность и эффективность применения данной связки продемонстрированы на реальном примере решения практической

задачи. Показано, что автоматизация непрерывной интеграции и контейнеризации ПО на основе предложенного подхода обеспечивает существенное сокращение времени подготовки и проведения экспериментов.

5. Благодарности

Исследование выполнено при поддержке Министерства науки и высшего образования Российской Федерации, проект № FWEW-2021-0005 «Технологии разработки и анализа предметно-ориентированных интеллектуальных систем группового управления в недетерминированных распределенных средах» (рег. № 121032400051-9) и грант № 075-15-2024-533 на выполнение крупного научного проекта по приоритетным направлениям научно-технологического развития (рег. № 124052100088-3).

6. Список использованных источников

[1] Р. Костромин. Подход к развертыванию программного обеспечения вычислительного кластера по облачной модели, Материалы VI Международного семинара по информационным, вычислительным и управляющим системам для распределенных сред (ICCS-DE 2024, 1-5 июля 2024 г., г. Иркутск, Россия), Иркутск: Изд-во ИДСТУ СО РАН, 2024, с. 135–140. doi:0.47350/ICCS-DE.2024.18.

[2] М. Л. Воскобойников, А. Г. Феоктистов, А. Н. Черных. Разработка и применение сервис-ориентированных научных приложений в инструментальном комплексе FDE-SWFs, Труды ИСП РАН, 2024 36(6) 195–214. doi:10.15514/ISPRAS-2024-36(6)-11.

[3] Инструменты DevOps. URL: https://learn.microsoft.com/en-us/devops.

[4] Ansible. URL: https://www.redhat.com/en/ansible-collaborative.

[5] Docker. URL: https://www.docker.com/.

[6] Kubernetes. URL: https://kubernetes.io/.

[7] A. Feoktistov, A. Edelev, A. Tchernykh, S. Gorsky, O. Basharina, E. Fereferov. An Approach to Implementing High-Performance Computing for Problem Solving in Workflow-based Energy Infrastructure Resilience Studies, Computation, 2023, vol. 11, no. 12, pp. 243. doi:10.3390/computation11120243.

[8] HTCondor. URL: https://htcondor.com/index.html.

Концепция подхода к решению задачи ответа на вопросы по табличной информации с использованием вычислительных графов.

Иван А. Поддубный[1], Никита О. Дородных[1]

[1]*Институт динамики систем и теории управления имени В.М. Матросова Сибирского отделения Российской академии наук (ИДСТУ СО РАН), 664033, Иркутская область, г. Иркутск, ул. Лермонтова, д. 134,*

Аннотация

В работе рассматривается метод предварительного обучения языковой модели для генерации ответов на табличные запросы с использованием линеаризованных вычислительных графов вместо традиционного SQL. Эксперименты с моделью BART-large на синтетическом датасете показали низкую точность на тестовых данных WiKiSQL, что указывает на ограниченность предложенного подхода и необходимость дальнейших исследований в области эффективного представления табличной информации для задач автоматического ответа на вопросы.

Keywords

Ответы на вопросы по табличной информации, Вычислительный граф, Предварительное обучение, Языковая модель,

1. Введение

Табличные данные широко используются для структурированного хранения и анализа информации в различных сферах, однако их неоднородность и отсутствие явной семантики затрудняют автоматическую обработку в интеллектуальных системах [1]. Одним из перспективных направлений решения этой проблемы является Table Question Answering (TQA)[2], позволяющее формировать ответы на вопросы к табличным данным на естественном или логическом языке. Современные методы TQA, включая семантический разбор и генеративные модели, сталкиваются с трудностями в числовых рассуждениях и учёте контекста, несмотря на достигнутый прогресс. Использование графовых представлений[3] показало потенциал для повышения эффективности за счёт переноса части вычислений на внешние модули.

Попытки комбинировать различные подходы для улучшения понимания запросов на естественном языке не привели к улучшению результатов, что актуализировало задачу более глубокого понимания табличной информации языковыми моделями. Наиболее заметным решением в этой области стала модель TAPEX [4], демонстрирующая, что выполнение SQL-запросов свидетельствует о понимании таблиц. В данной работе

7th International Workshop on Information, Computation, and Control Systems for Distributed Environments (ICCS-DE 2025), July 7–11, 2025, Irkutsk, Russia

poddubnyyiv@yandex.ru (Poddubny I.); nikidorny@icc.ru (Dorodnykh N.)

0000-0002-2431-8692 (Poddubny I.); 0000-0002-1825-0097 (Dorodnykh N.)

© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

ICCS-DE 2025 Workshop Proceedings (iccs-de.icc.ru)

DOI: 10.47350/ICCS-DE.2025.33

предлагается альтернативный подход: предварительное обучение языковой модели на графовых описаниях запросов вместо SQL, а также анализируется влияние такого представления на качество работы модели.

2. Метод генерации ответов по запросу в виде линеаризованного графа

Метод предварительного обучения генерации ответов на завпросы в формате линеаризованного вычислительного графа включает следующие этапы:

1. SQL запрос преобразуется в дерево с помощью SQLGlot.
2. Из дерева формируется граф.
3. Линеаризация графа. Полученная строка состоит из узлов вычислительного графа в формате "NODE тип узла | вложенность | аргументы. Узлы расположены в направлении от листьев к корню.
4. Генерация ответа моделью.
5. Сравнение результата генерации с ответом из набора данных.

3. Набор данных

В качестве данных был взят синтетический набор, который использовался в работе [4] для предварительного обучения. Он включает в себя 5 млн. примеров: SQL-запрос, таблица, ответ. Для данного набора выполнялась предварительная обработка по экранированию частей SQL запроса для его парсинга через SQLGlot, а так же фильтрация по превышению рамера входной последовательности токенов над максимально допустимым для модели. В результате набор был уменьшен до 3.5 млн.

4. Детали реализации

В качестве языковой модели был использован BART-large Предварительное обучение выполнялось на 3 GPU NVIDIA A100 80 GB PCIe с батчем 16 на каждую карту и коэффициентом аккумулирования градиента 5, что дает общий батч равный 240. При этом seed = 544, скорость обучения = 3e-5. Контрольные точки (checkpoints) сохранялись в конце каждой эпохи. Обучение на подготовленном наборе данных TAPEX длилось 3 эпохи и заняло 15 дней. В результате такого обучения была получена модель, которая генерирует ответ по запросу в виде линеаризованного графа.

5. Результаты

Оценка проводилась на тестовом и валидационном наборе WiKiSQL [5] после тонкой настройки в течении 20 эпох с аналогичными предварительному обучению параметрами

но с батчем 120 тренировочном наборе WiKiSQL. В качестве метрики оценки использовалась точность денотации (DA):

$$DA = P/N,$$

где P – количество точно совпадающих пар «вопрос-ответ»; N – общее количество пар «вопрос-ответ». Получены оценки для тестового и валижационного набора 12% и 14% соответственно для чекпоинта с минимальным лоссом на валидационном наборе. Полученные значения говорят о провальном обучении и деградации базовой модели.

6. Заключение

В ходе проведённого исследования был предложен и реализован метод предварительного обучения языковой модели для генерации ответов по табличным запросам в формате линеаризованного вычислительного графа. Несмотря на высокий потенциал использования вычислительных графов в аналогичных работах и использование современных инструментов (BART-large, SQLGlot, синтетический датасет TAPEX), результаты экспериментов показали низкую точность денотации (12% на тестовом и 14% на валидационном наборе WiKiSQL), что свидетельствует о неэффективности предложенного метода и деградации базовой модели. Полученные данные указывают на необходимость дальнейшего поиска и разработки более эффективных стратегий представления и обработки табличной информации для задач Table Question Answering.

Благодарности

Работа выполнена при поддержке государственного задания Министерства науки и высшего образования Российской Федерации (тема № 1023110300006-9).

Список литературы

[1] S. Zhang, K. Balog, Web table extraction, retrieval, and augmentation: A survey, ACM Transactions on Intelligent Systems and Technology (TIST) 11 (2020) 1–35.

[2] N. Jin, J. Siebert, D. Li, Q. Chen, A survey on table question answering: recent advances, in: China Conference on Knowledge Graph and Semantic Computing, Springer, 2022, pp. 174–186.

[3] R. Mouravieff, B. Piwowarski, S. Lamprier, Training table question answering via sql query decomposition, arXiv preprint arXiv:2402.13288 (2024).

[4] Q. Liu, B. Chen, J. Guo, M. Ziyadi, Z. Lin, W. Chen, J.-G. Lou, Tapex: Table pre-training via learning a neural sql executor, arXiv preprint arXiv:2107.07653 (2021).

[5] S. Yavuz, I. Gür, Y. Su, X. Yan, What it takes to achieve 100% condition accuracy on wikisql, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, 2018, pp. 1702–1711.

Адаптация подхода SimCLR к аннотированию столбцов русскоязычных таблиц

Кирилл Тобола[1], Никита Дородных[1]

[1] *Институт динамики систем и теории управления им. В.М. Матросова СО РАН, ул. Лермонтова, д. 134, 664033, Иркутск, Россия*

Аннотация

Таблицы часто лишены явной семантики, что затрудняет их машинную интерпретацию. Для интеграции с графами знаний особенно актуальна задача семантитческой интерпретации столбцов русскоязычных таблиц, однако существующие методы ограничены нехваткой размеченных данных и языковой спецификой. В работе предложен подход на основе адаптации контрастного обучения из работы SimCLR для русскоязычных таблиц, использующий DistilBERT в качестве базового кодировщика для обработки неразмеченных данных корпуса Russian Web Tables. Результаты для задачи семантического аннотирования русскоязычных столбцов превосходят базовые методы, подтверждая эффективность при работе с редкими типами и русскоязычными данными.

Ключевые слова

Русскоязычные таблицы, табличные данные, семантическое аннотирование столбцов, графы знаний, контрастное обучение, табличные представления.

1. Введение

Табличные данные представляют собой фундаментальный формат структурированного представления информации, широко используемый в научных исследованиях и бизнес-аналитике. Однако автоматизированная обработка таких данных сопряжена со значительными трудностями, обусловленными отсутствием явной семантической разметки, что особенно актуально для табличных данных на языках, отличных от английского [1]. В частности, русскоязычные таблицы сталкиваются с дополнительными ограничениями, связанными с недостатком специализированных инструментов и аннотированных корпусов. Существующие методы, основанные на архитектуре BERT [2–4], демонстрируют высокую эффективность, но требуют значительных объемов размеченных данных, которые для русского языка остаются ограниченными. Кроме того, англоязычные решения демонстрируют низкую адаптивность к русскоязычным таблицам из-за различий в токенизации и семантических особенностях языка.

2. Предлагаемый подход

Настоящее исследование предлагает инновационный метод семантической аннотации столбцов русскоязычных табличных данных, основанный на применении контрастного обучения в рамках архитектуры SimCLR [5]. Предлагаемый подход позволяет минимизировать проблему недостаточного объема размеченных данных за счет использования неразмеченных примеров и выявления скрытых семантических закономерностей. Ключевая задача семантической интерпретации столбцов заключается в автоматическом определении их семантического типа (например, «Спортсмен», «Литературный жанр» и аналогичных).

7th International Workshop on Information, Computation, and Control Systems for Distributed Environments (ICCS-DE 2025), July 7–11, 2025, Irkutsk, Russia
EMAIL: kirilltobola@icc.ru (A. 1); tualatin32@mail.ru (A. 2)
ORCID: 0009-0006-1014-451X (A. 1); 0000-0001-7794-4462 (A. 2)
© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).
ICCS-DE 2025 Workshop Proceedings
DOI: 10.47350/ICCS-DE.2025.34

Методологическая основа исследования заключается в разработке специализированного русскоязычного кодировщика для генерации устойчивых табличных представлений. Полученные векторные представления демонстрируют эффективность при решении задачи семантической интерпретации столбцов в русскоязычных таблицах.

2.1. Алгоритм обучения

Предлагаемый кодировщик табличных представлений был обучен на модифицированной версии набора данных RWT (Russian Web Tables) [6], содержащей 4,6 миллиона неразмеченных столбцов, с использованием метода контрастного обучения.

Контрастное обучение представляет собой метод самообучения, направленный на формирование информативных векторных представлений. Основной принцип данного подхода заключается в максимизации меры сходства между положительными парами данных и одновременной минимизации сходства между отрицательными. Такой механизм обеспечивает эффективное обучение на неразмеченных данных. Процедура обучения включает следующие этапы:

1. Создание двух аугментаций для каждого столбца (положительная пара).
2. Кодирование их в векторные представления.
3. Сближение векторов положительной пары и отдаление векторов отрицательных пар (отрицательными парами считаются аугментации, полученные из различных исходных столбцов).

В качестве двух аугментаций были выбраны: случайное удаление ячеек в столбце и перестановка строк внутри столбца.

2.2. Результаты

Экспериментальная часть исследования была выполнена на вычислительном кластере «Академик В. М. Матросов». Обучение модели кодировщика табличных представлений осуществлялось в течение 100 эпох с использованием четырёх графических ускорителей NVIDIA A100 (затрачено 9 дней 9 часов и 290 ГБ видеопамяти).

После завершения обучения кодировщик был интегрирован в архитектуру модели RuTaBERT [7] и дообучен на размеченном наборе данных RWT-RuTaBERT [8] для решения задачи семантической интерпретации столбцов русскоязычных таблиц (затрачено 2 дня 20 часов и менее 10 ГБ видеопамяти).

В качестве основного критерия эффективности модели использовалась F1-мера. Учитывая несбалансированность датасета RWT-RuTaBERT, применялись микро-F1 и макро-F1 метрики.

Для оценки эффективности предложенного подхода были проведены сравнительные тесты с моделями Doduo [4] и базовой версией RuTaBERT на наборе данных RWT-RuTaBERT. Результаты экспериментальной оценки представлены в Таблице 1.

Таблица 1

Результаты экспериментальной оценки для задачи семантической интерпретации столбцов русскоязычных таблиц на наборе RWT-RuTaBERT.

Модель	Микро F1	Макро F1
DODUO	96,2%	89,1%
RuTaBERT	96,4%	90,0%
Предлагаемый подход	**96,9%**	**91,0%**

Результаты исследования свидетельствуют о перспективности предложенного подхода. Наблюдаемое улучшение показателя макро F1 на 1-2% демонстрирует его повышенную эффективность в прогнозировании редко встречающихся семантических типов. Данное обстоятельство позволяет сделать вывод о более высокой обобщающей способности

разработанного метода, что обусловлено применением устойчивых табличных представлений кодировщика, сформированных в процессе контрастного обучения.

Также использование метода позволило достичь передовых показателей при меньших вычислительных затратах. В частности, предложенный метод требует примерно в три раза меньше видеопамяти в процессе обучения, что делает возможным его применение на стандартных рабочих станциях без необходимости использования специализированного оборудования.

3. Заключение

В данной работе представлен метод семантического аннотирования столбцов русскоязычных табличных данных, основанный на принципах контрастного обучения. Результаты проведенных экспериментов свидетельствуют о том, что предложенный подход позволяет преодолеть зависимость от значительных объемов размеченных данных благодаря механизму самообучения на неразмеченных таблицах.

Сравнительный анализ показал, что разработанный метод демонстрирует более высокую эффективность по сравнению с существующими базовыми решениями. Также достигается значительное повышение вычислительной эффективности: требования к объему памяти графических процессоров сокращаются на 60% относительно аналогов.

Предлагаемый подход создает перспективы для разработки универсальных систем семантической интерпретации табличных данных, что представляет особую актуальность для задач интеграции разнородной структурированной и слабоструктурированной информации.

4. Благодарности

Работа выполнена при поддержке государственного задания Министерства науки и высшего образования Российской Федерации (тема № 1023110300006-9).

5. Список источников

[1] Liu J., Chabot Y., Troncy R., Huynh V.-P., Labbe T., Monnin P. From tabular data to knowledge graphs: A survey of semantic table interpretation tasks and methods. Journal of Web Semantics, volume 76, 2023, 100761. doi: 10.1016/j.websem.2022.100761.

[2] Deng X., Sun H., Lees A., Wu Y., Yu C. TURL: Table Understanding through Representation Learning. Proc. the VLDB Endowment, volume 14, no. 3, 2020, pp. 307-319. doi: 10.14778/3430915.3430921.

[3] Yin P., Neubig G., Yih W. TaBERT: Pretraining for Joint Understanding of Textual and Tabular Data. Proc. the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 8413-8426. doi: 10.18653/v1/2020.acl-main.745.

[4] Suhara Y., Li J., Li Y. Annotating Columns with Pre-trained Language Models. Proc. the 2022 International Conference on Management of Data (SIGMOD’22), 2022, pp. 1493-1503. doi: 10.1145/3514221.3517906.

[5] Chen T., Kornblith S., Norouzi M., Hinton G. A simple framework for contrastive learning of visual representations. Proc. the 37th International Conference on Machine Learning (ICML’20), Online, 2020, pp. 1597-1607. doi: 10.5555/3524938.3525087.

[6] Fedorov, P., Mironov, A., Chernishev, G. Russian Web Tables: A Public Corpus of Web Tables for Russian Language Based on Wikipedia. Lobachevskii Journal of Mathematics, 2023, pp. 111-122.

[7] Tobola K., Dorodnykh N. Semantic Annotation of Russian-Language Tables Based on a Pre-Trained Language Model. Proc. the 2024 Ivannikov Memorial Workshop (IVMEM), Velikiy Novgorod, Russian Federation, 2024, pp. 62-68. doi: 10.1109/IVMEM63006.2024.10659709.

[8] RWT-RuTaBERT, 2025. URL: https://huggingface.co/datasets/sti-team/rwt-rutabert.

Методика применения инструментального комплекса для моделирования взаимодействующих микросетей

Михаил Андреевич Чекан [1]

[1] *Институт динамики систем и теории управления им. В.М. Матросова Сибирского отделения Российской академии наук, Россия, 664033, г. Иркутск, ул. Лермонтова, 134*

Аннотация

Доклад посвящён инструментальному комплексу автоматизации моделирования взаимодействия микросетей на основе мультиагентной среды. Рассмотрена архитектура комплекса, описана методика настройки и запуска сценариев моделирования. Комплекс предлагает визуальные средства конфигурации модельной среды, в том числе механизма обмена энергоресурсами и поведения агентов на основе иерархических машин состояний. Применение этих средств позволяет существенно снизить трудозатраты на разработку и проведение моделирования.

Ключевые слова

Микросеть, моделирование, средства автоматизации

1. Введение

Микросети (англ., microgrid) – это децентрализованные энергетические системы с небольшим числом генераторов (в том числе возобновляемых источников) и накопителей энергии, а также локальными нагрузками. Микросеть может взаимодействовать с глобальной системой энергетики, так и существовать изолированно от неё. Такие системы имеют ряд преимуществ, включающих в себя снижение потерь мощности и большую гибкость при управлении энергопотреблением. Микросеть обладает большей энергонезависимостью и живучестью при воздействии внешних возмущений. Исследование микросетей в уникальных природных зонах, таких как Байкальская природная территория (БПТ), направлено на улучшение экологичности и природосбережения. Будучи частью критической инфраструктуры (к которым относится энергетика), микросеть также является сложной взаимосвязанной системой, что обосновывает ведущую роль моделирования в проектировании и исследовании микросетей, а также их взаимодействия. Как следствие, существует потребность в развитых инструментальных средствах автоматизации исследования микросетей и их взаимодействия с применением высокопроизводительных вычислений. Одним из перспективных подходов в этой сфере является использование мультиагентного моделирования [1].

В данном докладе рассматривается инструментальный комплекс, автоматизирующий развёртывание мультиагентной среды моделирования взаимодействия микросетей, функционирующий на основе модели, представленной в [2].

2. Программный комплекс

Программный комплекс моделирования взаимодействия микросетей состоит из мультиагентной среды на основе платформы JADE [3] и надстройки в виде веб-приложения для

7th International Workshop on Information, Computation, and Control Systems for Distributed Environments (ICCS-DE 2025), July 7–11, 2025, Irkutsk, Russia
EMAIL: chekoopa@mail.ru (Chekan M.)
ORCID: 0000-0001-7622-421X (Chekan M.)
© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).
ICCS-DE 2025 Workshop Proceedings
DOI: 10.47350/ICCS-DE.2025.35

создания сценариев, конфигурирования среды и подготовки агентов. Архитектура комплекса представлена на рисунке 1.

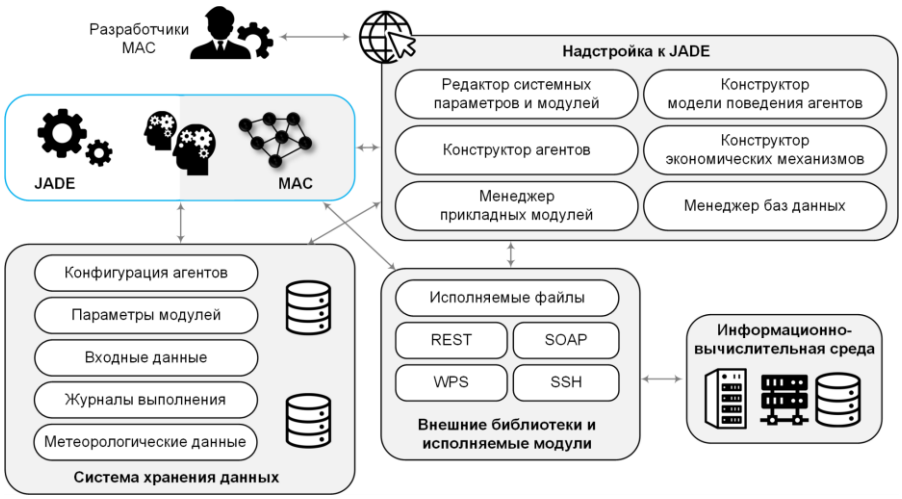


Рисунок 1: Архитектура программного комплекса

3. Методика

Процесс применения инструментального комплекса включает в себя этапы по развёртыванию самого комплекса, конфигурации его компонентов и непосредственного выполнения моделирования с последующим анализом его результатов.

Развёртывание платформы JADE и надстройки производится на вычислительном узле под управлением ОС на основе Linux с предустановленными Git и Java JDK версии 15 и выше. Развёртывание осуществляется автоматически посредством скрипта первичного развёртывания, выполняющего установку, преднастройку зависимостей, и организацию автозапуска сервисов.

Настройка комплекса включает в себя указание используемых контейнеров при распределённых сценариях моделирования, в этом случае для каждого узла требуется обеспечить доступ к нему посредством протокола SSH, выполнить развёртывание агентной платформы и сервиса мониторинга, а также убедиться в доступности запускаемого контейнера в пределах вычислительной сети.

Веб-интерфейс надстройки используется для добавления источников данных (для чтения параметров и записи результатов), подключения прикладных модулей (для выполнения вычислительных операций в рамках модели), а также конфигурации агентов и экономических механизмов. Агенты разрабатываются на языке Java вручную либо с использованием визуальных средств проектирования и кодогенерации на основе иерархических машин состояний [4]. Экономические механизмы настраиваются с помощью отдельного конфигуратора, который предлагает набор типовых механизмов с настройками и возможностью подключения внешнего прикладного модуля для реализации той или иной функциональности в рамках механизма.

Точкой сборки вышеописанных элементов является сценарий моделирования, который также конфигурируется через надстройку с указанием параметров, участвующих агентов, используемого механизма и входных данных. После запуска сценария надстройка позволяет отслеживать процесс выполнения моделирования (журнал исполнения платформы) и выгрузить результат по завершении моделирования.

4. Оценка трудозатрат

Для оценки качества автоматизации выполнено сравнение трудозатрат на разработку одного агента, представляющего взаимодействующую микросеть в платформе JADE. Результат сравнения представлен на рисунке 2. Разработка агента включает в себя:

- Создание каркаса агента (машина состояний с обработкой сообщений и регистрацией в Directory Facilitator). Ручная реализация требует около 4–6 человеко-часов, использование средств кодогенерации с визуальным редактором – 1–2 человеко-часа.
- Проектирование экономического механизма и логики принятия решений. Занимает 10–15 человеко-часов в ручном исполнении, 6–8 часов с использованием визуальных средств.
- Подключение агента к платформе и размещение его в вычислительной среде. Занимает около 0,5–1 час при ручной настройке мультиагентной среды, при использовании надстройки – около 0,25 часа.

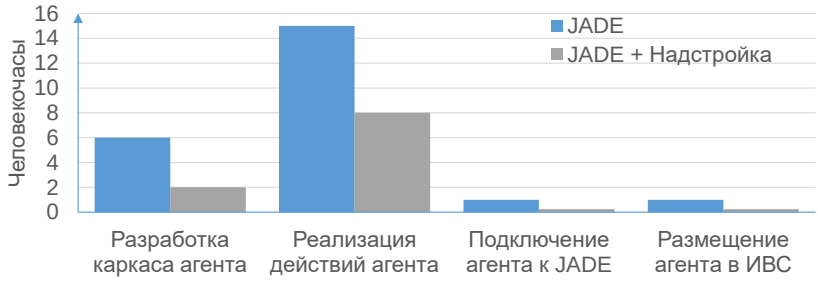


Рисунок 2: Сравнение трудозатрат по разработке агента микросети

Таким образом, использование комплекса сокращает затраты на разработку агента в среднем на 50-60%, что является важным фактором при масштабировании модели на большее число агентов. Отдельным преимуществом является инкапсуляция типовых операций с моделью, что сокращает вероятность ошибок и, как следствие, затраты на их анализ и исправление.

5. Заключение

В докладе рассмотрена архитектура программного комплекса для автоматизации агентного моделирования взаимодействия микросетей с использованием экономических механизмов регулирования спроса и предложения. Комплекс включает в себя средства организации МАС, автоматизации разработки агентов и конфигурирования экономических механизмов. Также описана методика применения этого комплекса для организации моделирования.

Применение средств в составе комплекса, в том числе веб-интерфейса и визуального редактора поведения агентов, позволяет снизить уровень необходимой технической квалификации при разработке МАС под конкретный сценарий моделирования, что с одной стороны позволяет предметному специалисту быстрее получить результат, с другой – избежать типовых ошибок, связанных с настройкой и программированием агентов.

Представленный комплекс можно применять при настройке мультиагентного моделирования в других предметных областях, предполагающих обмен ресурсами посредством экономического механизма. В дальнейшем планируется расширение предлагаемой функциональности для обеспечения большей гибкости проектирования агентов и механизма их взаимодействия.

6. Список использованных источников

[1] M.A. Jirdehi, V.S. Tabar, S. Ghassemzadeh, S. Tohidi, Different aspects of microgrid management: A comprehensive review, Journal of Energy Storage 30 (2020) 101457. doi: 10.1016/j.est.2020.101457.

[2] И.В. Бычков, А.Г. Феоктистов, М.А. Чекан, Модель поведения агента микросети, Вычислительные технологии, том 28, 6 (2023) 108–117. doi: 10.25743/ICT.2023.28.6.010.

[3] JAVA Agent DEvelopment Framework (JADE), 2025, URL: https://jade.tilab.com

[4] Чекан М.А., Комплекс автоматического проектирования агентов на основе иерархических конечных автоматов, Современные наукоемкие технологии 3 (2025) 59–65.

Модификация метода деления чисел в Системе Остаточных Классов

Егор М. Ширяев[1], Михаил Г. Бабенко[1]

[1]*Северо-Кавказский Федеральный Университет (СКФУ), ул. Пушкина, 1, Ставрополь, 355017, Российская Федерация*

Аннотация

Деление целых чисел в системе остаточных классов представляет собой математически и вычислительно сложную задачу, напрямую не реализуемую так же эффективно, как операции сложения и умножения. В данной работе рассматриваются три метода деления: классический метод деления с преобразованием, через обратный элемент, и предложенная модификация поканального деления. Оценка их производительности по времени и памяти проведена с использованием серии тестов на наборах взаимно простых модулей различной битности. Полученные результаты демонстрируют высокую эффективность поканального метода по производительности.

Ключевые слова

Система остаточных классов, Деление модулярных чисел, Китайская теорема об остатках, Криптография,

1. Введение

Система остаточных классов (СОК) — это представление чисел через набор их остатков по взаимно простым модулям. Такое представление позволяет выполнять сложение, вычитание и умножение параллельно, без необходимости обращения к полному значению числа. Однако операция деления в СОК остается одной из наиболее трудоемких и плохо формализуемых операций. Это связано с тем, что деление не является операцией, естественно определенной по модулю. В частности, результат деления двух чисел в СОК не может быть выражен через независимые деления остатков по модулю.

Тем не менее, эффективное деление в СОК необходимо в ряде задач. Среди них можно выделить криптографические протоколы [1], цифровую обработку сигналов [2] и ускоренные вычисления в остаточной арифметике [3]. В таких задачах требуется либо точное восстановление результата деления, либо приближенные методы, обеспечивающие стабильную скорость при контролируемой ошибке. Поэтому исследование алгоритмов деления и их сравнительный анализ являются актуальной задачей.

7-ой Международный семинар по информационным, вычислительным и управляющим системам для распределенных сред (ICCS-DE 2025), 7–11 Июля, 2025, Иркутск, Россия
shiryaev.jegor@yandex.ru (Shiriaev E.); mgbabenko@ncfu.ru (Babenko M.)
0000-0002-2359-1291 (Shiriaev E.); 0000-0001-7066-0061 (Babenko M.)
© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).
ICCS-DE 2025 Workshop Proceedings (iccs-de.icc.ru)
DOI: 10.47350/ICCS-DE.2025.36

2. Описание реализованных методов

2.1. Классические подходы

Традиционный подход подразумевает выполнение деления в позиционной форме с переводом чисел в позиционную систему и после деления обратно в СОК.

Классическим способом деления в СОК считается метод, в котором производится восстановление исходных чисел из остатков, далее осуществляется нахождение обратного элемента $Y^{-1} \bmod P$, где $P = \prod p_i$, и выполняется умножение $X \cdot Y^{-1} \bmod P$. Алгоритмически он базируется на использовании расширенного алгоритма Евклида и китайской теоремы об остатках [4]. Главный недостаток метода — необходимость перехода к полному представлению, что нарушает параллельность.

2.2. Поканальный метод (предлагаемая модификация)

В качестве альтернативы классическим подходам, основанным на восстановлении полного значения, нами была предложена модификация, заключающаяся в выполнении операции деления в каждом канале независимо. Иными словами, речь идет о вычислении результата в каждом модуле p_i без восстановления глобального представления чисел. При этом требуется, чтобы остаток делителя y_i был обратим по модулю p_i, то есть:

$$\gcd(y_i, p_i) = 1. \tag{1}$$

При выполнении этого условия можно вычислить обратный элемент $y_i^{-1} \in \mathbb{Z}_{p_i}$, такой что:

$$y_i \cdot y_i^{-1} \equiv 1 \pmod{p_i}. \tag{2}$$

Тогда результат деления в i-м канале формально выражается как:

$$z_i = x_i \cdot y_i^{-1} \bmod p_i, \tag{3}$$

а совокупность всех таких z_i образует представление результата:

$$\mathbf{z} = (z_1, z_2, \dots, z_k)\,. \tag{4}$$

Данный метод отличается отсутствием необходимости в операциях восстановления и последующего преобразования. Все вычисления выполняются поканально, что позволяет выполнять параллельную обработку.

3. Анализ и интерпретация результатов

Для каждого из описанных методов были проведены тесты на наборах взаимно простых модулей, состоящих из пяти чисел, длина которых варьировалась от 16 до 64 бит с шагом в 8 бит.

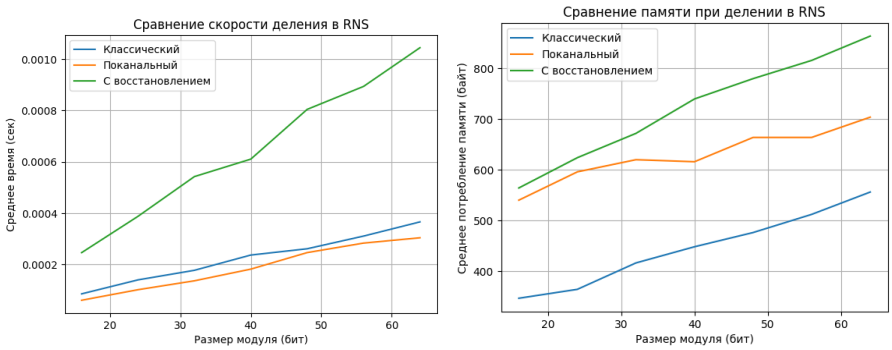


Рисунок 1: Результаты моделирования: (a) производительности; (b) потребления памяти

Результаты демонстрируют превосходство поканального метода по времени выполнения. Однако по параметру памяти он показывает результат хуже, чем классический.

4. Заключение

Проведенный анализ показал, что операция деления в системе остаточных классов может быть реализована различными способами с разными характеристиками. Предложенный поканальный метод оказывается наилучшим по производительности, при этом его недостатком является нестабильное потребление памяти. В задачах, критичных к скорости выполнения и допускающих компромисс по памяти, данный метод может использоваться как эффективная альтернатива классическим схемам.

Будущие исследования могут быть направлены на комбинированные с поканальным делением и приближенные методы.

References

[1] X. Zhang, K. K. Parhi, Implementation approaches for the residue number system, IEEE Circuits and Systems Magazine 17 (2017) 24–44. doi:10.1109/MCAS.2016.2631804.

[2] N. S. Szabo, R. I. Tanaka, Residue Arithmetic and Its Applications to Computer Technology, McGraw-Hill, 1967.

[3] M. D. Matteo, C. Fagiani, M. Lanuzzi, S. Pontarelli, Efficient rns arithmetic units for fpga-based high-speed computation, Journal of Signal Processing Systems 72 (2013) 111–120. doi:10.1007/s11265-013-0757-6.

[4] D. E. Knuth, The Art of Computer Programming. Volume 2: Seminumerical Algorithms, 3 ed., Addison-Wesley, 1997.

МЕТОДИКА РАЗРАБОТКИ ПРИЛОЖЕНИЙ ДЛЯ ПОЛУЧЕНИЯ, ОБРАБОТКИ И АНАЛИЗА МЕТЕОДАННЫХ

Евгений Юмашев

Институт динамики систем и теории управления им. В.М. Матросова СО РАН, ул. Лермонтова, д. 134, 664033, Иркутск, Россия

Аннотация

В работе предложена методика разработки сервис-ориентированных приложений для получения, обработки и анализа метеоданных. Сервисы приложения могут взаимодействовать с внешними программными системами и источниками данных. Они выполняются в гетерогенной распределенной вычислительной среде. Предложенная методика успешно апробирована для разработки и применения приложения структурно-параметрической оптимизации автономного энергетического комплекса.

Ключевые слова

Метеоданные, получение, обработка и анализ данных, сервисы, методика

1. Введение

Метеоданные широко используются при решении задач экологического мониторинга природных территорий. Известны различные открытые источники таких данных (rp5.ru [1], ВНИИГИ-МЦД [2] и др.). Однако автоматизированное получение качественных метеоданных из подобных источников и обеспечение высокоуровневой поддержки разработки гибких средств их дальнейшего использования являются в настоящее время актуальными проблемами. В данной работе предложена новая методика разработки сервис-ориентированных приложений для получения, обработки и анализа временных рядов метеоданных.

2. Методика

Схема применения методики получения, анализа и обработки временных рядов метеоданных включает следующие основные этапы (рис. 1): *разработка приложения* (описание вычислительной модели приложения (параметров и операций, выполняемых над этими параметрами) в инструментарии FDE-SWFs [3], выбор готовых модулей из программных систем и библиотек или разработка новых модулей на языке Python для реализации операций вычислительной модели, генерация WPS-сервисов, реализующих модули, их размещение на WPS-сервере, построение композиций сервисов, формирование гетерогенной распределенной вычислительной среды (ГРВС), доставка и размещение программного обеспечения (ПО) на узлы среды, тестирование и настройка программно-аппаратного обеспечения, контейнеризация ПО с помощью системы Docker); *подготовка эксперимента* (выбор сервиса или композиции сервисов, задание исходных данных); *проведение эксперимента* (запуск сервиса или композиции сервисов, контроль статуса выполнения, взаимодействие с внешними системами, сбор, хранение и передача расчетных данных, уведомление пользователей). Применение данной методики позволяет создавать сложные композиции сервисов и управлять их выполнением в ГРВС. Сервисы могут

7th International Workshop on Information, Computation, and Control Systems for Distributed Environments (ICCS-DE 2025), July 7–11, 2025, Irkutsk, Russia
EMAIL: yumashevgeny@mail.ru (A. 1)
ORCID: 0009-0003-8736-0155 (A. 1)
© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).
ICCS-DE 2025 Workshop Proceedings
DOI: 10.47350/ICCS-DE.2025.37

обмениваться данными с внешними системами посредством запросов. Контейнеризация ПО с помощью системы Docker осуществляет изоляцию вычислительных процессов, обеспечивает совместимость программных систем и библиотек, устраняет возможность возникновении конфликтов и повышает надежность вычислений. Стандарт WPS позволяет работать с геоданными и отслеживать статус выполнения сервисов с длительным временем расчетов.

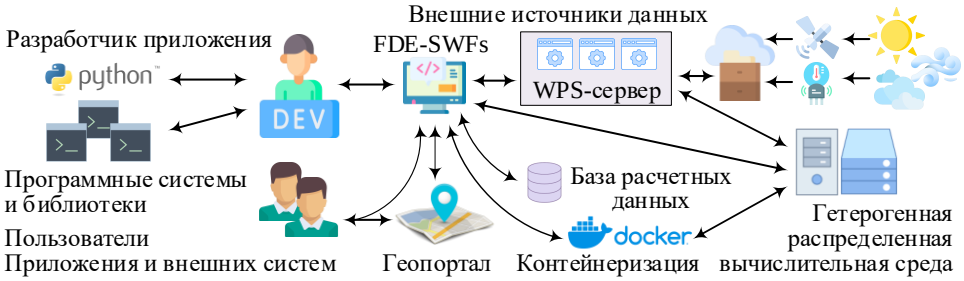


Рисунок 1: Схема реализации и применения методики

3. Приложение

Методика успешно применена при разработке приложения для получения, обработки и анализа метеоданных. Приложение включает сервисы получения массивов метеоданных (s_1), построения моделей метеорологического года (s_2), прогнозирования выработки электроэнергии солнечной батареи (s_3) и выработки электроэнергии ветрогенератором (s_4), а также формирования временных рядов метеоданных (s_5). Сервис s_1 получает метеоданные с открытого источника rp5.ru. Сервис производит нахождение пропусков данных и их заполнение (перевод категориальных данных в числовые, обучение модели прогнозирования пропусков данных, анализ временного ряда с целью выявления пропусков данных, заполнение пропусков данных на основе обученной модели), а также определение аномалий значений данных временного ряда (нормализацию данных, анализ временного ряда метеоданных с целью выявления индивидуальных аномалий, смягчение индивидуальных аномалий, анализ временного ряда метеоданных с целью выявления контекстных аномалий, смягчение контекстных аномалий). Проведен сравнительный анализ следующих характеристик rp5.ru и ВНИИГИ-МЦД: география сбора данных (c_1), представление данных для одной локации одним архивом (c_2), представление данных для нескольких локаций одним архивом (c_3), число доступных метеостанций (c_4), возможный период предоставления данных (c_5), периодичность данных (c_6) и процент успешного выполнения тестового набора запросов (c_7). Результаты сравнительного анализа характеристик c_i, весовых коэффициентов w_i и степеней d_i поддержки (1.00 – полная поддержка, 0.75 – большая степень поддержки, 0.50 – частичная поддержка, 0.25 – низкая степень поддержки, 0 – отсутствие поддержки) i-й характеристики для каждого источника данных представлены в табл. 1, $i = \overline{1,7}$. Суммарная оценка характеристик источника данных рассчитана по формуле $s = \sum_{i=1}^{7} d_i w_i$, где $\sum_{i=1}^{7} w_i = 1$. Весовые коэффициенты w_i отражают совокупные субъективные оценки пользователей источников данных. Суммарные оценки rp5.ru и ВНИИГИ-МЦД составили соответственно 0.83 и 0.45. Очевидно, что оценка rp5.ru существенно превосходит соответствующую оценку ВНИИГИ-МЦД, что во многом обусловлено степенью поддержки важных характеристик c_1, c_4, c_6 и c_7.

Приложение успешно применено в процессе решения задачи структурно-параметрической оптимизации автономного энергетического комплекса, условно расположенного на Байкальской природной территории. Этот комплекс включает солнечную электростанцию, солнечные инверторы, инверторы системы хранения энергии, аккумуляторные батареи и дизельную электростанцию. В рамках исследования использована композиция трех сервисов приложения: $s_1 \rightarrow s_2 \rightarrow s_3$. Оптимизация проведена на основе модели типичного метеорологического года, который строится сервисом s_2 с использованием массивов метеоданных, получаемых сервисом s_1 с rp5.ru. Сервисом s_3 выполнена оценка выработки электроэнергии солнечной электростанцией. В результате решения задачи определена оптимальная конфигурацию комплекса (рис. 2): солнечная электростанция мощностью 30 кВт,

аккумуляторные батареи емкостью 96 кВт * ч и дизельная электростанция мощностью 60 кВт. Выбранное оборудование обеспечивает минимальную себестоимость электроэнергии – 21.7 руб./кВт*ч при годовом потреблении дизельного топлива, не превышающем 19.5 тонн.

Таблица 1. Результаты сравнительного анализа

c_i	w_i	Аспекты характеристик		d_i	
		rp5.ru	ВНИИГИ-МЦД	rp5.ru	ВНИИГИ-МЦД
c_1	0.20	По России и за рубежом	По России	1.00	0.50
c_2	0.05	+	–	1.00	0
c_3	0.05	–	+	0	1.00
c_4	0.20	9800	13-521	0.75	0.25
c_5	0.10	С 2000 г. по н. в.	С 1950-70-х по н. в.	0.25	0.50
c_6	0.20	Почасовые	Суточные и месячные	1.00	0.50
c_7	0.20	99.9%	49.7%	1.00	0.50

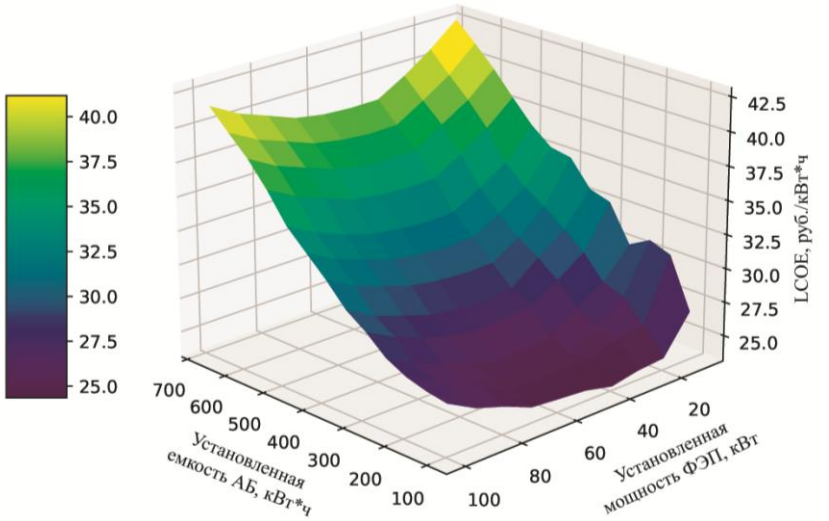


Рисунок 2: Результаты структурно-параметрической оптимизации

4. Заключение

Рассмотрены актуальные проблемы автоматизированного получения качественных метеоданных из открытых источников и обеспечения высокоуровневой поддержки разработки гибких средств их дальнейшего использования. С целью решения этих проблем предложена новая методика разработки сервис-ориентированных приложений для получения, обработки и анализа временных рядов метеоданных. Проведен анализ возможностей и выбор известных источников данных. Разработано приложение, которое успешно применено в процессе оптимизации конфигурации автономного энергетического комплекса, условно расположенного на Байкальской природной территории. Работа выполнена в рамках гранта № 075-15-2024-533 Минобрнауки РФ на выполнение крупного научного проекта по приоритетным направлениям научно-технологического развития (№ гос. рег. 124052100088-3).

5. Список литературы

[1] rp5 – Расписание погоды. URL: https://rp5.ru/Weather_in_the_world.
[2] Всероссийский научно-исследовательский институт гидрометеорологической информации. URL: http://meteo.ru/. doi: doi.org/10.25729/ESI.2023.31.3.007.
[3] A. Feoktistov, M. Voskoboinikov, A. Tchernykh, Framework for Development and Execution of Scientific WorkFlows: Designing Service-Oriented Applications, Programming and Computer Software, 2024, vol. 50, no. 8, pp. 900–913. doi: doi.org/10.1134/S0361768824700828.

СИСТЕМА МОНИТОРИНГА ПРОГРАММНО-АППАРАТНОЙ ИНФРАСТРУКТУРЫ ВЫЧИСЛИТЕЛЬНОЙ СРЕДЫ

Дмитрий Яковлев, Александр Феоктистов

Институт динамики систем и теории управления им. В.М. Матросова СО РАН, ул. Лермонтова, д. 134, 664033, Иркутск, Россия

Аннотация

В работе представлена система мониторинга узлов гетерогенной распределенной вычислительной среды, основанная на микросервисной архитектуре. Она обеспечивает сбор, обработку, агрегирование и анализ данных мониторинга вычислительных процессов и узлов в среды. Цель работы системы мониторинга заключается в предоставлении актуальной информации диспетчеру научных рабочих процессов, необходимой ему для принятия решений по планированию вычислений и распределению вычислительных ресурсов среды.

Ключевые слова

Гетерогенная распределенная вычислительная среда, вычислительные процессы, мониторинг, управление вычислениями

1. Введение

Мониторинг программно-аппаратной инфраструктуры является важной составляющей проблемы управления вычислениями. Как правило он включает: сбор данных о загрузке процессора и оперативной памяти на узлах; мониторинг потребления ресурсов процессами; мониторинг доступности сервисов и оборудования. Общая схема мониторинга программно-аппаратной инфраструктуры представлена на рис. 1.

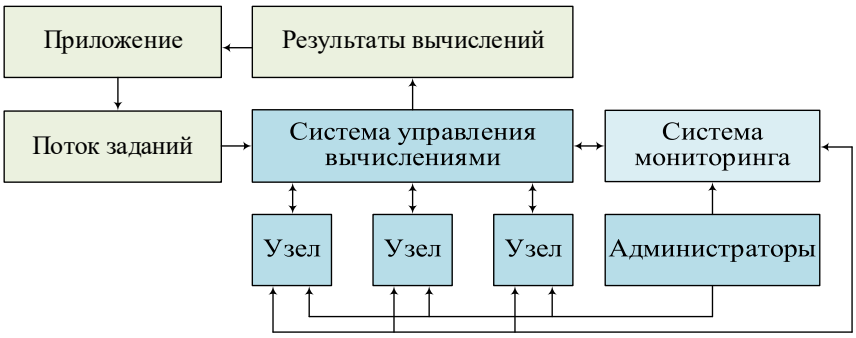


Рисунок 1: Общая схема мониторинга программно-аппаратной инфраструктуры

Известен широкий спектр средств мониторинга вычислительных систем [1]. Однако при использовании подобных систем зачастую возникает проблема интеграции предопределенного набора данных, получаемых, от контрольно-измерительных систем, с дополнительной необходимой информацией от служб операционных систем и администраторов узлов гетерогенной распределенной вычислительной среды (ГРВС). К сожалению, получение и обработка такой информации является проблемой для известных средств мониторинга. В этом

7th International Workshop on Information, Computation, and Control Systems for Distributed Environments (ICCS-DE 2025), July 7–11, 2025, Irkutsk, Russia
EMAIL: dm1try.y4kovlev@yandex.ru (A. 1); agf@icc.ru (A. 2)
ORCID: 0009-0009-7973-1685 (A. 1); 0000-0002-9127-6162 (A. 2)
© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).
ICCS-DE 2025 Workshop Proceedings
DOI: 10.47350/ICCS-DE.2025.38

контексте в рамках данного исследования предложена система мониторинга (СМ) узлов ГРВС с целью сбора, обработки, агрегирования и анализа данных мониторинга и их интеграции с информацией от служб операционных систем и администраторов среды. СМ разработана на основе микросервисного подхода [2].

2. Результаты исследования

Разработана новая СМ узлов ГРВС с целью сбора, обработки, агрегирования и анализа данных о вычислительных процессах и узлах среды. СМ предоставляет обработанные, агрегированные и проанализированные данные диспетчеру научных рабочих процессов инструментального комплекса FDE-SWFs [3]. Диспетчер использует эти данные для принятия решений по планированию вычислений и распределению вычислительных ресурсов среды при выполнении потоков заданий пользовательских приложений. Отличительная особенность СМ заключается в интегрированном использовании данных от контрольно-измерительных систем, служб операционных систем и администраторов узлов ГРВС. Архитектура СМ представлена на рис. 2.

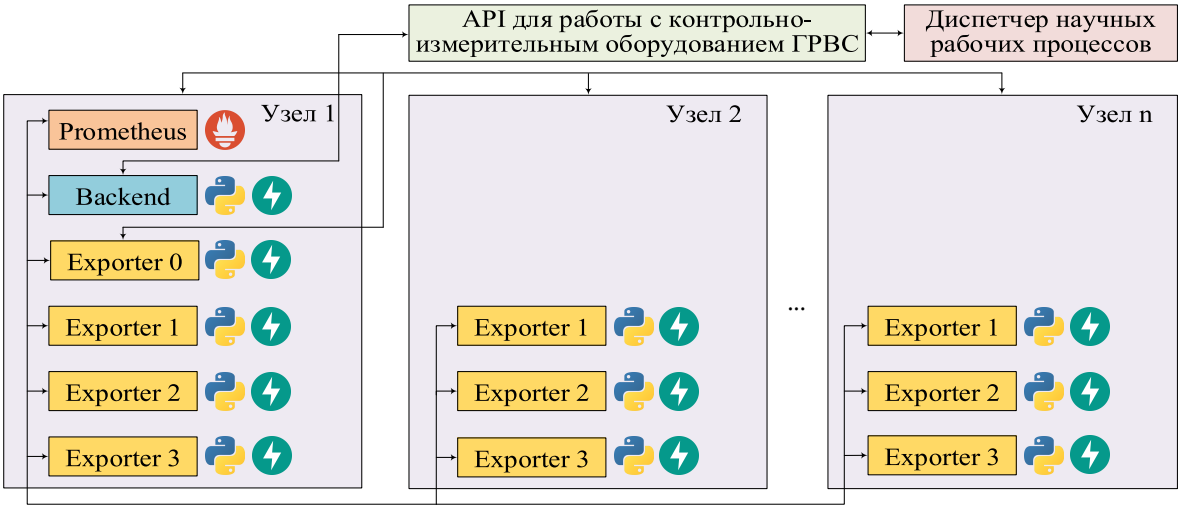


Рисунок 2: Архитектура СМ

СМ включает модуль управления, а также три набора модулей для сбора информации об узлах ГРВС и мониторинга вычислительных процессов. Взаимодействие диспетчера рабочих процессов с данной системой осуществляется с помощью специализированного API для работы с контрольно-измерительным оборудованием ГРВС. Один из узлов ГРВС назначается в качестве основного узла. Два компонента модуля управления (сервисы Backend и Exporter 0) размещаются на основном узле. На этом же узле развертывается система Prometheus [4], предназначенная для сбора и хранения информации об узлах ГРВС. Сервис Exporter 1 используется для получения характеристик об узле, таких как объем оперативной памяти, производительность процессора и др. Сервис Exporter 2 выполняет мониторинг ресурсов по списку заданных процессов. Сервис Exporter 3 выполняет мониторинг всех пользовательских процессов операционной системы. В рамках данной работы дополнительно разработаны новые функции СМ для расчета коэффициентов надежности вычислительного узла и расчета надежности схемы решения задач. Эти функции реализованы на языке Python с использованием библиотеки SymPy [5]. В настоящее время выполняется развитие разработанных средств СМ и их использование в инструментальном комплексе FDE-SWFs для определения надежности выполнения научных рабочих процессов в выбранных узлах ГРВС.

В таблице 1 приведено сравнение функциональных возможностей СМ с возможностями известных контрольно-измерительных систем [1]. Символы '+' и '–' означают соответственно наличие и отсутствие возможности. Базируясь на результатах сравнительного анализа функциональных возможностей систем в таблице 1, можно видеть, что СМ предоставляет полный набор возможностей, требуемых для эффективного управления вычислениями

диспетчером научных рабочих процессов, включая новую дополнительную возможность анализа показателей надежности процессов и узлов.

Таблица 1. Функциональные возможности систем

Функциональная возможность	Lapta	Disparity	ClustrX	Zabbix	Ganglia	Nagios	Ovis2	Xymon	СМ
Задание новых метрик	–	–	–	+	+	–	–	–	+
Мониторинг процессов и узлов	+	+	+	+	+	+	+	+	+
Сбор данных об использовании ресурсов процессами	+	+	+	+	+	–	–	–	+
Агрегирование данных	–	+	+	+	+	–	+	–	+
Анализ показателей надежности процессов и узлов	–	–	+	–	–	–	–	–	+

СМ успешно применена на практике в процессе выбора алгоритмов оптимизации инфраструктуры энергетических комплексов для определения эффективности использования вычислительных ресурсов исследуемыми алгоритмами [6]. Применение СМ позволило осуществить выбор наиболее подходящих алгоритмов для разных классов моделей энергетических комплексов с точки зрения критериев владельцев ресурсов (загрузки процессоров, использования оперативной памяти и надежности работы узлов).

3. Заключение

Разработана СМ с целью сбора, обработки, агрегирования и анализа данных мониторинга вычислительных процессов и узлов ГРВС. Ее отличительная особенность заключается в интегрированном использовании данных от контрольно-измерительных систем, служб операционных систем и администраторов узлов ГРВС.

4. Благодарности

Работа выполнена в рамках гранта № 075-15-2024-533 Минобрнауки РФ на выполнение крупного научного проекта по приоритетным направлениям научно-технологического развития (№ гос. рег. 124052100088-3).

5. Список литературы

[1] Р. О. Костромин, И. А. Сидоров, Системы мониторинга смешанной вычислительной инфраструктуры, Материалы международной конференции «Ляпуновские чтения» (г. Иркутск, Россия, 5–7 декабря 2022 г.), Иркутск, Изд-во ИДСТУ СО РАН, 2022, с. 79.
[2] Д. А. Кравченко, Микросервисная архитектура, Интерактивная наука, 2022, № 4, с. 43–44. doi:10.21661/r-556565.
[3] М. Л. Воскобойников, А. Г. Феоктистов, А. Н. Черных, Разработка и применение сервис-ориентированных научных приложений в инструментальном комплексе FDE-SWFs, Труды ИСП РАН, 2024, т. 36, № 6, 195–214. doi:10.15514/ISPRAS-2024-36(6)-11D.
[4] Prometheus [Электронный ресурс]. URL: https://prometheus.io/.
[5] SymPy [Электронный ресурс]. URL: https://www.sympy.org/en/index.html.
[6] O. Edeleva, A. Edelev, M. Voskoboinikov, A. Feoktistov, Scientific Workflow-Based Synthesis of Optimal Microgrid Configurations, Energies, 2024, vol. 17, no. 23, p. 6138. doi:10.3390/en17236138.

Образовательный чат-бот в Telegram с AI-генерацией тестов

Владимир В. Лапин[1], Оксана А. Естафиади[1], Алексей С. Смирнов[2], Алина Р. Багаутдинова[3]

[1] *ГБОУ СК «Гимназия №25», Ставрополь*
[2] *ЦДО «Поиск», Ставрополь*
[3] *Северо-Кавказского федерального университета, Ставрополь*

Аннотация

В современном мире, где доступ к информации и обучению становится все более мобильным и персонализированным, возникает потребность в инструментах, способных адаптироваться к индивидуальным запросам пользователей. Чат-боты в мессенджерах, таких как Telegram, уже доказали свою эффективность в различных областях, включая образование. Они позволяют автоматизировать процессы обучения, предоставляя пользователям возможность получать знания в удобное время и в интерактивной форме. Разработан образовательный чат-бот для Telegram, использующий искусственный интеллект (AI) для генерации тестов, управления классами и анализа результатов. Бот автоматизирует процесс создания учебных материалов, адаптирует задания под уровень знаний учащихся и предоставляет учителям аналитику. Проект направлен на повышение эффективности и доступности образовательного процесса.

Ключевые слова

чат-бот, Telegram, искусственный интеллект, генерация тестов, образование, персонализированное обучение

1. Введение

Использование искусственного интеллекта для генерации тестов добавляет новое измерение в образовательные технологии. AI способен анализировать уровень знаний пользователя, подбирать задания соответствующей сложности и создавать уникальные тесты, что делает процесс обучения более гибким и эффективным. Это особенно важно в условиях, когда традиционные методы обучения не всегда могут удовлетворить потребности современных учащихся, стремящихся к самостоятельному и непрерывному образованию.

Кроме того, Telegram как платформа для образовательного бота обладает рядом преимуществ: высокая популярность среди пользователей, кроссплатформенность, удобство использования и возможность интеграции с другими сервисами. Это делает его идеальным инструментом для внедрения инновационных образовательных решений. Таким образом, разработка образовательного чат-бота с функцией AI-генерации тестов не только отвечает текущим трендам цифровизации образования, но и открывает новые возможности для повышения качества и доступности обучения.

Внедрение образовательных чат-ботов представляет собой многоэтапный процесс, который требует учета технических, педагогических и организационных аспектов. Одной из ключевых особенностей является интеграция бота в существующую образовательную экосистему, что подразумевает не только техническую настройку, но и адаптацию пользователей к новому инструменту. На начальном этапе важно определить целевую аудиторию и задачи, которые будет решать бот. Например, это может быть автоматизация проверки знаний, генерация

7th International Workshop on Information, Computation, and Control Systems for Distributed Environments (ICCS-DE 2025), July 7–11, 2025, Irkutsk, Russia

© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

ICCS-DE 2025 Workshop Proceedings
DOI: 10.47350/ICCS-DE.2025.39

индивидуальных заданий или предоставление справочной информации. После определения целей необходимо разработать архитектуру бота, которая включает в себя модули для обработки запросов, хранения данных и взаимодействия с пользователями. Важным элементом является выбор платформы для разработки, такой как Telegram, которая предоставляет удобный API и широкие возможности для интеграции.

Следующий этап — это создание контента и алгоритмов работы бота. Для образовательных чат-ботов особенно важно обеспечить качественную генерацию тестов и заданий, что требует использования специализированных алгоритмов и баз данных. Например, можно использовать шаблоны для создания вопросов разных типов: множественный выбор, открытые вопросы или задания на соответствие. Кроме того, необходимо предусмотреть возможность адаптации заданий под уровень знаний пользователя, что может быть реализовано с помощью анализа предыдущих ответов и результатов.

Особое внимание следует уделить тестированию и доработке бота перед его запуском. На этом этапе важно проверить работу всех функций и выявить возможные ошибки. Например, можно провести тестовый запуск среди небольшой группы пользователей и собрать их отзывы для улучшения функционала. После запуска необходимо регулярно обновлять контент и алгоритмы работы бота, чтобы он оставался актуальным и полезным для пользователей.

Таким образом, схема архитектуры чат-бота демонстрирует последовательность взаимодействия между пользователем, платформой Telegram, сервером, базой данных и AI-модулем. Каждый элемент системы играет важную роль в обеспечении функциональности и удобства использования образовательного чат-бота.

2. Архитектура Телеграмм-бота

Архитектура Telegram-бота состоит из нескольких ключевых компонентов:

Telegram API: предоставляет API для взаимодействия с ботами. Через API бот получает сообщения от пользователей и отправляет ответы. Для работы с API используется библиотека python-telegram-bot, которая упрощает взаимодействие.

Ядро бота (логика): это основной модуль, который обрабатывает входящие сообщения, выполняет команды и генерирует ответы.Логика может включать обработку текста, вызов внешних API (например, для генерации тестов), работу с базами данных и т.д.

База данных (опционально): Для хранения данных пользователей, результатов тестов или других данных используется база данных (например, SQLite, PostgreSQL или MongoDB).

Внешние сервисы (опционально: бот может взаимодействовать с внешними сервисами, такими как AI-модели для генерации контента, платежные системы или облачные хранилища.

Сервер: Бот должен быть размещен на сервере, чтобы работать 24/7. Для этого можно использовать облачные платформы, такие как Heroku, AWS или VPS.

Ниже на рисунке 1 приведен пример простого Telegram-бота, который генерирует тесты на основе входящих сообщений.

Для работы с Telegram API и генерацией текста установим библиотеки: pip install python-telegram-bot transformers

```python
# Инициализация модели для генерации вопросов
question_generator = pipeline("text2text-generation", model="valhalla/t5-small-qa-qg-hl")
# Функция для генерации вопросов
def generate_questions(text, num_questions=3):
    questions = []
    for _ in range(num_questions):
        result = question_generator(text, max_length=50)
        question = result[0]['generated_text'].strip()
        questions.append(question)
    return questions
# Команда /start
async def start(update: Update, context: CallbackContext):
    await update.message.reply_text("Привет! Отправь мне текст, и я создам тест.")
# Обработка текстовых сообщений
async def handle_message(update: Update, context: CallbackContext):
    text = update.message.text
    questions = generate_questions(text)
    # Формируем тест
    test = "Вот твой тест:\n\n"
    for i, question in enumerate(questions):
        test += f"{i + 1}. {question}\n"
    await update.message.reply_text(test)
# Основная функция
def main():
    # Создаем приложение бота
    application = ApplicationBuilder().token("YOUR_TELEGRAM_BOT_TOKEN").build()
    # Регистрируем обработчики команд и сообщений
    application.add_handler(CommandHandler("start", start))
    application.add_handler(MessageHandler(filters.TEXT & ~filters.COMMAND, handle_message))
    # Запускаем бота
    application.run_polling()
if __name__ == "__main__":
    main()
```

Рисунок 1: Пример простого Telegram-бота, который генерирует тесты на основе входящих сообщений.

Этот код представляет собой реализацию Telegram-бота на Python, который использует библиотеку python-telegram-bot для взаимодействия с Telegram API и модель transformers для генерации вопросов на основе текста. Бот позволяет пользователю отправлять текст, после чего автоматически генерирует тест с вопросами и отправляет его обратно пользователю.

Каждый компонент системы выполняет определенные функции:

Пользователь: Взаимодействует с ботом через Telegram, отправляет команды и получает обратную связь.

Telegram Bot: Обрабатывает команды, управляет взаимодействием между пользователем и API, осуществляет подключение к Yandex GPT.

Yandex GPT API: Создает тестовые задания и управляет их генерацией.

Классы (bot_data): Хранит информацию о классах, их студентах и кодах доступа.

Тесты (bot_data): Хранит информацию о созданных тестах и их параметрах.

Результаты (bot_data): Хранит результаты тестирования, включая детализированные данные по каждому ученику.

3. Используемые технологии

Для реализации проекта используются следующие технологии:

1. Python 3.10+: Основной язык разработки, выбранный за его простоту и мощные библиотеки для работы с API и асинхронного программирования.
2. pyTelegramBotAPI: Библиотека для работы с Telegram Bot API, позволяющая легко управлять сообщениями и командами.
3. aiohttp: Асинхронная библиотека для выполнения HTTP-запросов. Обеспечивает эффективность и позволяет обрабатывать несколько запросов одновременно, что критично для работы с API.
4. JSON: Формат для хранения данных — используется локально для хранения информации между сессиями работы.

На рисунке 2 представлена работа Телеграмм-бота

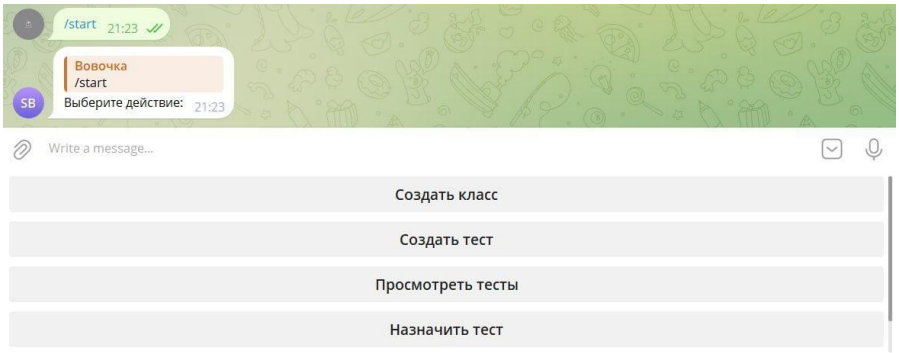


Рисунок 2: Начало работы Телеграмм-бота

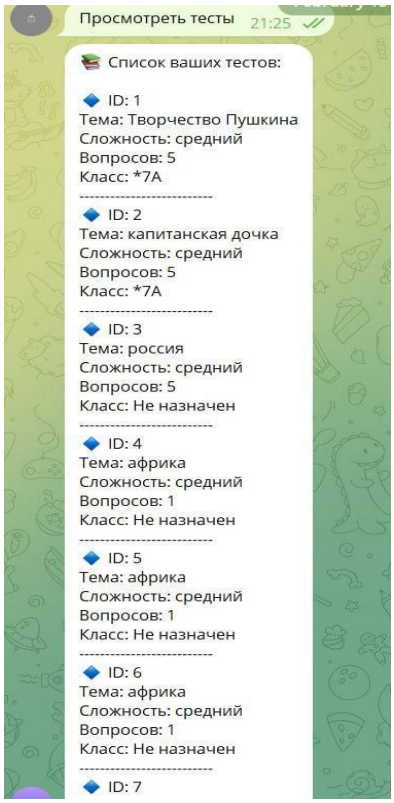


Рисунок 3: Просмотр тестов чат-бота

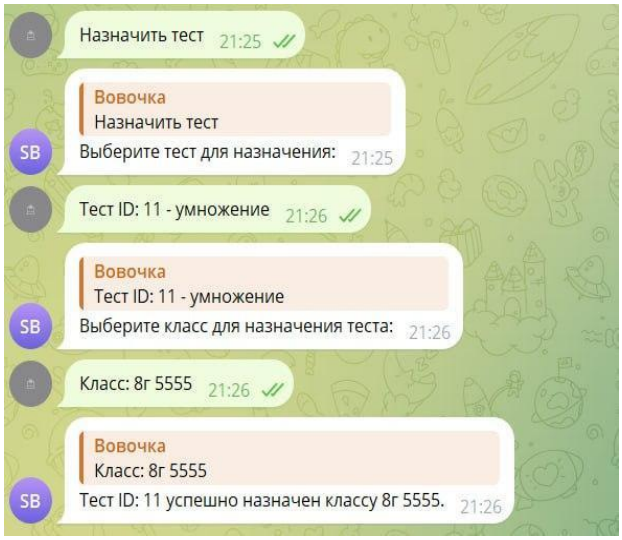


Рисунок 4: Работа чат-бота

4. Вывод

Проект демонстрирует эффективное использование современных технологий, таких как Telegram Bot API и Yandex GPT, для решения задач в сфере образования. Разработка бота, ориентированного на автоматизацию учебных процессов, создает новые возможности для оптимизации работы учителей, повышения вовлеченности учеников и улучшения качества образовательного процесса в целом. Telegram Bot API предоставляет удобный и мощный инструмент для создания ботов, которые могут взаимодействовать с пользователями в режиме реального времени, делая образовательный процесс более доступным и интерактивным. Бот выступает в роли посредника между учениками и учебными ресурсами, позволяя автоматизировать такие задачи, как генерация тестов, проверка ответов и анализ результатов. Интеграция Yandex GPT, мощной языковой модели, позволяет боту генерировать осмысленные и релевантные вопросы на основе предоставленного текста, что значительно экономит время учителей и повышает качество учебных материалов. Учителя получают возможность сосредоточиться на индивидуальной работе с учениками, в то время как бот берет на себя рутинные задачи, такие как создание тестов и проверка ответов. Для учеников бот становится удобным инструментом, который адаптирует тесты под их уровень знаний, предоставляет мгновенную обратную связь и делает обучение более персонализированным и интересным. Использование элементов геймификации, таких как баллы и рейтинги, повышает мотивацию учеников и делает учебный процесс более увлекательным. Кроме того, бот собирает данные о результатах тестирования, что позволяет учителям анализировать прогресс учеников и корректировать учебный процесс в соответствии с их потребностями. Таким образом, проект не только оптимизирует работу учителей, но и создает условия для более эффективного и вовлеченного обучения, демонстрируя, как современные технологии могут быть использованы для улучшения качества образования.

5. Благодарность

Проект реализован при поддержке ГБОУ СК «Гимназия №25» и «Центра Поиск».

6. Список литературы

[1] Байкова, К. Д. "Анализ и разработка функционального Telegram-бота." *Молодой исследователь Дона* 6 (33) (2021): 15-20.

[2] Гатулин, Р. Р., and Д. А. Колупаева. "Использование мессенджера Telegram для реализации технологии электронного обучения в вузе." *Санкт-Петербургский образовательный вестник* 11-12 (15-16) (2017): 31-33.

[3] Филонов, Дмитрий Русланович, and Владислав Игоревич Тупикин. "Чат-бот для Telegram для помощи абитуриентам." *Заметки по информатике и математике*. 2017. 152-156.

[4] Валинурова, Анна Александровна, Наталья Владимировна Балабанова, and Иван Алексеевич Маценков. "АЛГОРИТМ РАЗРАБОТКИ TELEGRAM-БОТА-ПРОДУКТИВНОГО ПОМОЩНИКА СОВРЕМЕННОГО БИЗНЕСА." *Современные наукоемкие технологии. Региональное приложение* 2 (74) (2023): 60-67.

[5] Вяткин, Александр Николаевич, and Татьяна Инакентиевна Русских. "Telegram-боты и их разнообразие." *Наука XXI века: технологии, управление, безопасность: Материалы II национальной научной конференции, Курган, 21 апреля 2022 года*. 2022.

[6] Вяткин, Александр Николаевич, and Татьяна Инакентиевна Русских. "Telegram-боты и их разнообразие." *Наука XXI века: технологии, управление, безопасность: Материалы II национальной научной конференции, Курган, 21 апреля 2022 года*. 2022.

Научно-организационный отдел
Федерального государственного бюджетного учреждения науки
Института динамики систем и теории управления имени В.М. Матросова
Сибирского отделения Российской академии наук
664033, Иркутск, ул. Лермонтова, д. 134
E-mail: rio@icc.ru

Подписано к печати 26.06.2025 г.
Формат бумаги 60×84 1/16, объем 13,63 п.л.
Заказ 1. Тираж 1 экз.

Отпечатано в ИДСТУ СО РАН