

**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 IV Международного семинара
по информационным, вычислительным и управляющим системам
для распределенных сред
(ICCS-DE 2022)**

4 – 8 июля 2022 г., Иркутск, Россия

**Иркутск: Издательство ИДСТУ СО РАН
2022**

УДК: 004.7/.9

ББК: 32.96 + 32.97

Редакционная коллегия:

Бычков И.В., Черных А.Н., Феокистов А.Г.

Материалы IV Международного семинара по информационным, вычислительным и управляющим системам для распределенных сред (ICCS-DE 2022), 4-8 июля, 2022, Иркутск, Россия. Иркутск: Изд-во ИДСТУ СО РАН, 2022. 134 с.

Proceedings of the 4th International Workshop on Information, Computation, and Control Systems for Distributed Environments (ICCS-DE 2022), July 4-8, 2022, Irkutsk, Russia. Irkutsk: ISDCT Publisher, 2022. 134 p.

ISBN 978-5-6041814-3-0

Материалы научного сборника включают избранные статьи и тезисы IV Международного семинара по информационным, вычислительным и управляющим системам для распределенных сред (ICCS-DE 2022), 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) 4-8 июля, 2022 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. Особое внимание в них уделяется системам, функционирующим в различных распределенных средах.

Официальный сайт семинара ICCS-DE 2022: <https://iccs-de.icc.ru/>

ISBN 978-5-6041814-3-0



© Институт динамики систем и теории управления им. В.М. Матросова СО РАН, 2022

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

Сопредседатель программного комитета

Andrei Tchernykh, CICESE Research Center, Ensenada, Mexico

Члены программного комитета

Jorge Mario Cortés-Mendoza, Кафедра электронных вычислительных машин, Южно-Уральский государственный университет, Челябинск, Россия;

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;

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

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

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

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

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

Федоров Р.Е., 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

Башарина О.Ю., Иркутский государственный университет, г. Иркутск, Россия;

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

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

Кензин М.Ю., 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;

Костромин Р.О., 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;

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

Толстихин А.А., 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;

Чекан М.А., 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. Software Architectures in Complex Socio-Technical Systems: Review and Experiences	8
С. Валеев, Н. Кондратьева. Многоуровневые эталонные модели бизнес-процессов	19

СЕКЦИЯ: ИНФОРМАЦИОННЫЕ И ТЕЛЕКОММУНИКАЦИОННЫЕ СИСТЕМЫ

O. Basharina. Methodology for Collecting, Processing, and Analyzing Socio-Economic Information	25
Aleksandra Stojkov and Zeljko Stojanov. Software Reengineering: A Qualitative Study in two Software Companies	30
Е. Барахтенко, В. Стенников, Д. Соколов, Г. Майоров. Принципы построения цифрового двойника для решения задач проектирования интегрированных энергетических систем	36
Н. Береснева, А. Еделев. Риск-ориентированный подход к поиску критических элементов системы энергетики	43
Д. Соколов, В. Стенников, Е. Барахтенко, Г. Майоров. Разработка программной платформы для создания цифровых двойников интегрированных энергетических систем	50
А. Спасибо, А. Ветров, Е. Фереферов. Информационные технологии для поддержки археологических исследований	56

СЕКЦИЯ: ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ

A. Reznik, A. Soloviev. Intelligent software system APP-MNIT for analytical calculations using parallel computing.....	62
A. Stolbov, A. Lempert, A. Pavlov. A Web-based Software System for Intelligent Support in Modeling and Forecasting the Water Resources	68

СЕКЦИЯ: ПАРАЛЛЕЛЬНЫЕ И РАСПРЕДЕЛЕННЫЕ ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ

A. Feoktistov, A. Tchernykh, V. Dmitriev. Mitigating Uncertainty of Resource Allocation in Heterogeneous Computing Environment	74
O. Sukhoroslov, M. Gorokhovskiy, R. Ilgovskiy, T. Kuskarov, Y. Semenov, A. Vetrov. Towards a General Framework for Studying Resource Management in Large Scale Distributed Systems.....	79
В. Девятков, Т. Девятков, М. Федотов, Р. Минниханов, В. Шестюк. Среда имитационного моделирования GPSS Studio и моделирующее ядро GPSS World Core.....	97
Д. Романова. Проектирование комбинационных схем на функционально-поточковом параллельном языке программирования.....	105

СПЕЦИАЛЬНАЯ СЕКЦИЯ ДЛЯ МОЛОДЫХ УЧЕНЫХ

M. Chekan. Comparative Analysis of Tools for Automating Microgrid Modeling	111
I. Ignatev. BigData technologies: opportunities and prospects for import substitution in the Russian Federation	114
Е. Безуглова, Е. Ширяев, Н. Кучеров, В. Кучуков. Коды исправления ошибок, как модификация методики стеганографии	117
С. Горский, Р. Костромин, И. Сидоров. Интеграция средств контейнеризации для автоматизации разработки и выполнения пакетов прикладных программ.....	121
В. Ушаков. Построение областей достижимости в пространстве возмущений	127
Е. Ширяев, Е. Безуглова, Н. Кучеров, В. Кучуков. Исследование математической модели метода сравнения чисел в СОК для четных диапазонов	130

Software Architectures in Complex Socio-Technical Systems: Review and Experiences

Zeljko Stojanov¹

¹University of Novi Sad, Technical Faculty "Mihajlo Pupin" Zrenjanin, Djure Djakovica BB, 23000 Zrenjanin, Serbia

Abstract

Socio-technical systems are in general enterprise systems composed of social and technical elements, intended to achieve specific business objectives. Software elements are usually located in the central part of systems and used for integration of services and data. In addition, they are also spread over various equipment in other segments of the system to provide specific functionalities. Deployment of software elements in various layers and parts of complex systems based on their functionalities, requires careful design of software architecture that will enable robust and reliable functioning of the system. This paper presents a short review of software architecture patterns used in complex socio-technical systems, followed with the author's experiences with data-driven, layered and service-oriented architectures. Based on literature review and description of experiences, challenges in considering software architecture in complex socio-technical systems are stated.

Keywords

software architecture, software architecture patterns, socio-technical system, data-driven architecture, service-oriented architecture, layered architecture, architecture quality attributes

1. Introduction

Socio-technical systems are composed of social (people and society) and technical (machines and technology) elements, aimed at achieving specific behaviour and purpose [1]. This leads to the need for a broad knowledge of both technical and social disciplines to understand, design, operate and maintain them. In addition, due to the complexity of socio-technical systems, social and technical part should be inquired together. Design of complex socio-technical systems requires understanding complexity of each particular element (social or technical) and their relationships (logical, functional, structural) [2]. Baxter and Sommerville [3] proposed a framework for socio-technical systems engineering which is based on research of work design, information systems, computer-supported cooperative work, and cognitive systems engineering. The framework links organisational change processes and technical systems development to cover all socio-technical issues.

Software and IT components are essential for functioning and business performance of contemporary organizations in changing and volatile market. Software components provide meaningful and efficient communication of humans and equipment, as well as data process-

^{4th} International Workshop on Information, Computation, and Control Systems for Distributed Environments (ICCS-DE 2022), July 04–08, 2022, Irkutsk, Russia

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

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

© 2022 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

ICCS-DE 2022 Workshop Proceedings (iccs-de.icc.ru)

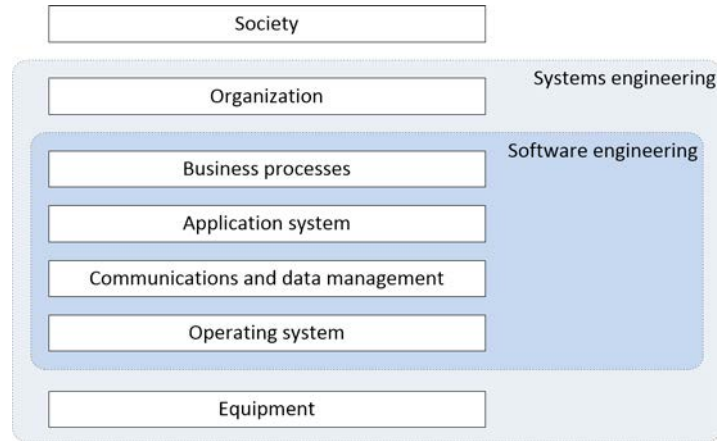


Figure 1: Layered structure of complex socio-technical systems

ing, which points out the importance of software systems in complex socio-technical systems in contemporary organizations [4]. For better understanding of complex socio-technical systems, and insight in to system engineering and software engineering parts in designing these systems, Sommerville [5] proposed layered structure of socio-technical systems presented in Figure 1. Interactions occur between neighbouring layers. Software part of the system cannot be observed in isolation, but rather as a part of complex structure composed of different types of equipment. Software part of the system enable functioning of variety of equipment, but also serves for integration of system parts and data.

Based on the above observations, the main objective of this paper is to propose challenges in designing software architecture patterns in complex socio-technical systems based on literature review and personal experience presented in the author’s previous research. The paper is structured in the following way. The second section provides a short review of software architectures used in complex socio-technical systems. The third section presents the author’s experiences with three types of software architectures in different socio-technical systems. The last section presents proposed challenges for selecting software architecture styles in complex socio-technical systems, as well as concluding remarks.

2. Literature review

The role of software architecture evolved as the way of business changed in the globally connected and ever-changing world [6]. From the mostly monolithic architectures in 1980s and distributed monoliths in 1990s, software architectures evolved towards to Internet connected architectures in 2000s, Internet based services in 2010s, and finally to intelligently connected architectures in 2020s. The presented trend point out the new architectural focus on algorithms and data, leading to new movement to architectures that are created at runtime (intelligent dynamic composition, distributed cloud deployment, connection of ”things”). Software architectures of modern enterprise systems are closely connected to business models used in contemporary organizations [7].

Software architecture encompasses early design decisions, that are hard to change, and therefore it deserves careful consideration especially for development and maintenance of large-scale, long-living software systems [8]. The main features of modern software architectures are [9]:

- *Changeability*. The dynamics of the system in which the software is integrated causes a change in software requirements, which leads to the use of flexible architectures.
- *Integrability*. Possibility to integrate with variety of other software systems, equipments and social environments.
- *Scalability*. It reflects the capacity to scale in size, processing capacity and performances to adapt to changes in the environment.
- *Loosely coupled*. This characteristics is based on abstraction, separation of concerns, and information hiding, leading to highly modifiable and flexible systems that can reconfigure based on changed demands.

Technical part of complex systems is composed of different equipment and heterogeneous software elements. System architecture is the top level issue influencing strategic decisions, engineering trade-off and associated rationales regarding how system will satisfy stated demands. The complexity of the systems mainly depends on the integration of different technical components, and heterogeneity of components [9]. There are two main paradigms related to the essence of software architecture [10]:

- *Pure technical paradigm*. It relates to the system structure and behaviour, components and connectors, description languages, design methods, patterns and reference architectures.
- *Socio-technical paradigm*. It consider architecture from the stakeholders' points of view, including they reasoning and decisions.

Variety of architecture design patterns are used in designing software architecture in complex socio-technical systems, which is essential for facilitating specific organization and communication of distributed software components [11]. Analysis of developers posts at Stack Overflow indicate that the most frequently used architecture patterns are MVC, Layered, SOA, Pipe and Filter, and Reflection [12]. Each architecture pattern has specific characteristics suitable for particular business and socio-technical requirements. In large and complex socio-technical systems multiple patterns are used together. The benefit of using well-known design patterns is that their advantages and drawbacks are known. Review of contemporary architecture patterns, such as layered, event-driven, service-oriented or microservices architecture is presented in [13][14].

Based on a systematic literature review of architecture and design patterns for IoT systems [15], the authors indicate that 57% of totally 143 identified patterns are non-IoT patterns, designed by using conventional software and system architecture and design patterns. Since more domains adopt IoT systems, it is expected that the number of specific IoT design patterns will increase in the future. In some specific cases, such as wearable mobile sensors used for well-being and health of people, well-known architecture patterns are extended, and new patterns named "Sampling Package" and "Data Transformer" are created, while their robustness and feasibility are tested in a small case study [16].

When selecting software architecture patterns for complex systems, it is very important to assess quality attributes of the proposed patterns. Based on the analysis of software developers posts at Stack Overflow, Bi et al. [12] indicate that developers consider different quality attributes for used software architectures. The most frequently used quality attributes are maintainability and performance. Despite the proven importance of software architecture quality attributes when selecting architecture design pattern, their influence and trade-offs are usually neglected in the practice [12]. Developers usually avoid considering quality attributes because there are a number of existing patterns, new patterns regularly emerge, heterogeneity of pattern descriptions, and the lack of tools for automatic pattern analysis [17]. Quality attributes (scalability, maintainability, security, availability, and portability) of a layered software architecture in a sensor-based IoT system are presented in [18].

3. Experiences

In this section the author's experiences with data-driven [19], service-oriented [20], and layered [18][21] software architectures are presented.

3.1. Data-driven architecture

Data-driven model of software architecture was proposed and used for development of a software application for managing computer networks' scenarios in IT engineering education [22, 19]. The data-driven model of software architecture is based on a generic model of software applications that enables direct manipulation of multiple entities associated to current visual form and selection of the most appropriate path for using software application [23]. Two main objectives of the proposed data-driven architecture are to provide the basis for model-driven development, and to improve software maintenance activities. Fulfilment of the proposed objectives ensures that software product reflects the real user requirement through the whole software life-cycle.

Data-driven approach to application development starts with data model, which is modelled with XML Schema in order to enable easy transformation of the model to UML suitable for object oriented development, and later to relational model suitable for database design. Initial data model is described with Network Node Description Language (NNDL) [24], which is XML based language for modelling network nodes in network scenarios used for teaching computer networks' concepts at university. Transformation of data models is presented in Figure 2. In addition, initial NNDL data model is used for defining the content of XML configuration files for nodes in network scenarios.

The main element in the presented data-driven architecture design is `Entity` that is used to define software elements in the different layers. All entity related elements are grouped in logic packages that contain: visual forms for manipulating entity or set of entities in tabular form, a Java class `Entity` in the middle business logic layer, and a table `tblEntity` in database layer. Architecture of a software application based on packages derived from data model is presented in Figure 3. Packages are organized in subsystems, while relationships between packages are derived from relationships between data entities in data model. These relationships are re-

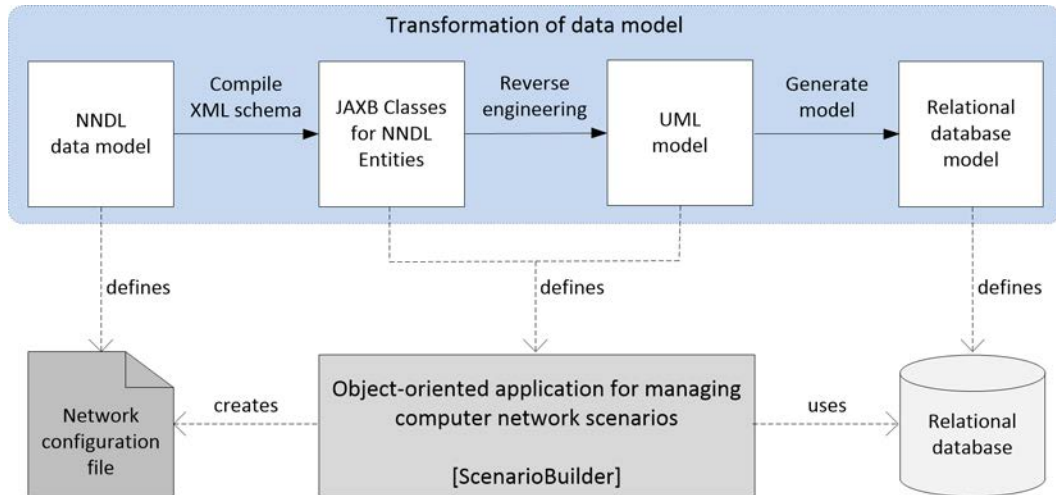


Figure 2: Data model transformations in data-driven software architecture design

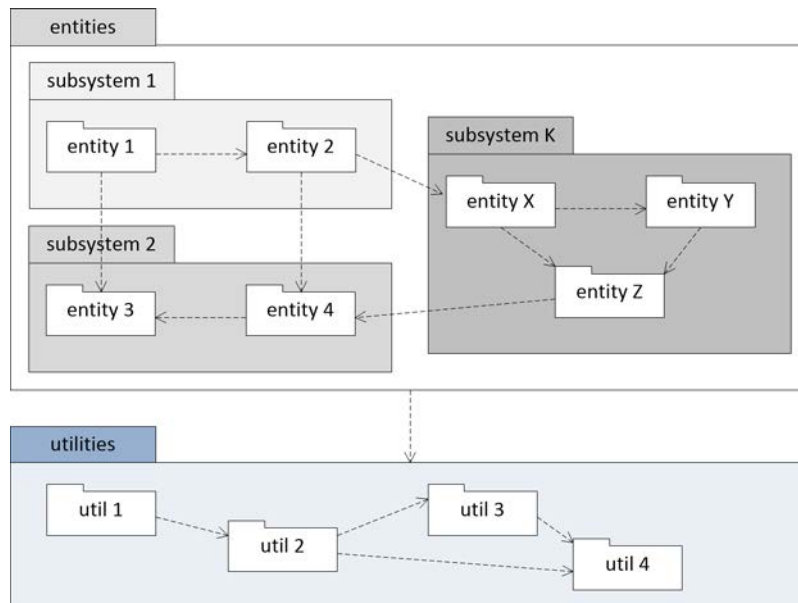


Figure 3: Data-driven software architecture based on data entity packages

flected also in user interface of software application. All entity packages use utilities organized in similar packages.

Navigation in user interface also reflects relationships between data defined in the data model. This is supported through uniform design of visual forms for handling single entity (FormSingleRecord), and visual forms for handling multiple entities in tabular form (FormMultipleRecords). All forms implement standard set of functionalities proposed in interface IFormActionHandler.

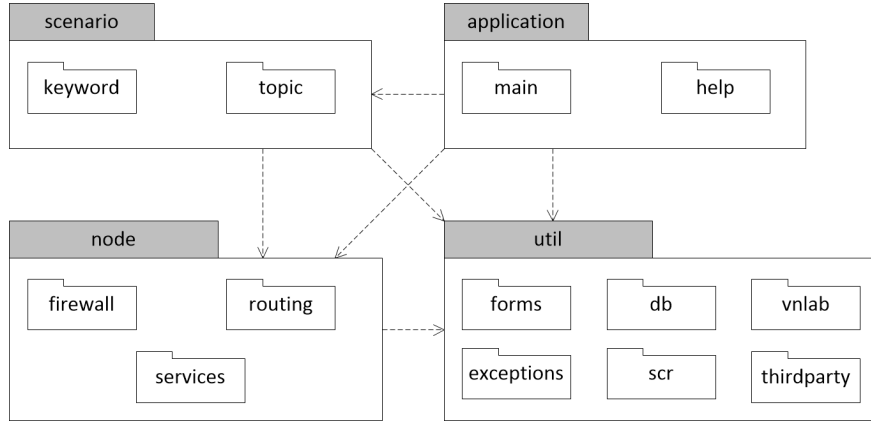


Figure 4: Data-driven software architecture implemented in software application ScenarioBuilder

Implementation of proposed data-driven software architecture in *ScenarioBuilder* software application for managing computer network scenarios is presented in Figure 4. Entity packages are grouped in packages named *scenario* and *node*, while utilities packages are placed in package *util*. Package *application* contains packages for starting main application form and providing help during the use. For development are used Java EE technologies.

3.2. Service-oriented architecture

Service-oriented architecture is implemented in on-demand monitoring system using a drone fleet in a smart city environment [20]. Overall architecture of a system for drone management within smart city environment is presented in Figure 5. The main part of the system is cloud-based drone management center with software services for collecting and integrating data, managing drone routes and providing info for relevant city services (police, ambulance, parking services, public transport, etc.). Data sources are Vehicles Detecting Sensor (VDSs), maps of city, and data collected from drones.

Software services are deployed in cloud and communicate with other part of the system. Architecture of software services in cloud is presented in Figure 6. The main software service is Drone navigation control service which manage navigation of drone fleet based on data collected by service IoT data upload service and city maps collected by Map collecting service and preprocessed by Map analytics service. Drone communication service send control signals to drone fleet, while Drone data upload service collects data from drones and sends them to Drone data analytics service. Based on analysis of data collected from drones Emergency action service will call appropriate city services (police, ambulance, etc.) and/or publish data to community by using Urban traffic info web service. Presented services are partially developed and deployed by using Java EE technologies.

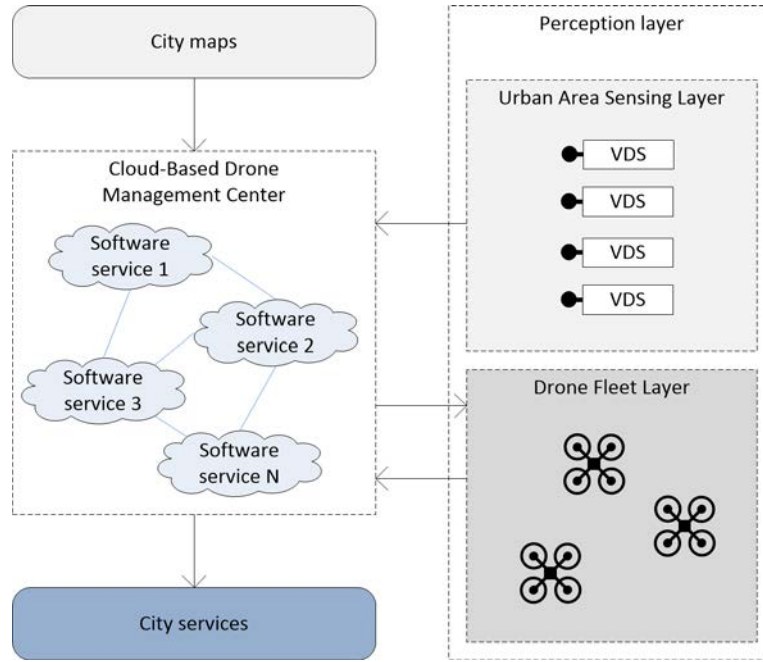


Figure 5: Overall architecture of a cloud-based drone fleet monitoring system

3.3. Layered architecture

Layered software architecture for monitoring industrial environmental conditions in manufacturing settings is presented in [21], while its quality attributes are examined in [18]. Distribution of software components in layers is presented in Figure 7. The whole project is based on open source software and hardware components. Communication between components occur between components in Middleware Layer and between appropriate components deployed in different layers as it is presented in Figure 7.

The main characteristic of the proposed software architecture is its heterogeneity regarding designed software components, which is caused by the use of hardware components produced by different vendors. This resulted in use of variety of technologies and programming languages. Characteristics of software components in layers:

- *Perception layer.* Software components are designed as Firmware Software Components written in C programming language for collecting and processing data from sensors.
- *Transport layer.* Software components are designed as Firmware Software Components written in C programming language for transmitting data from software components in perception layer to middleware layer.
- *Middleware layer.* Software components are designed as services deployed in cloud, and they are the core for integration of the whole system. Data Analytics Software Component receives and processes data. This component is implemented as Blynk Server, an Open-Source Netty based Java server. Web server is implemented as The Apache Tomcat, Java open source web server, on which data visualization service (Web Application

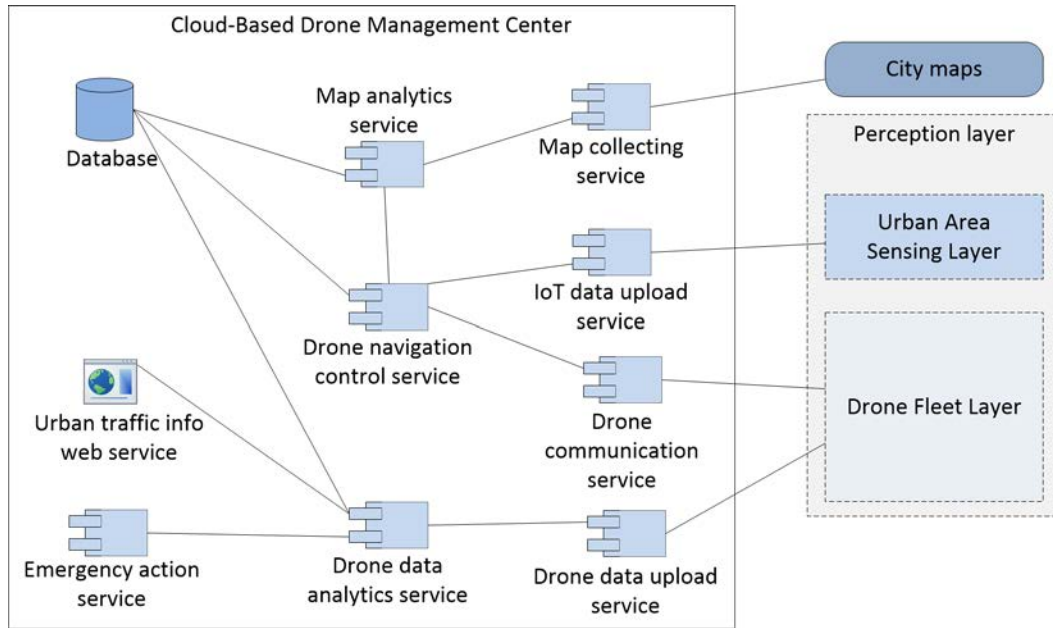


Figure 6: Service-oriented architecture of software part of a cloud-based drone fleet management system

for Data Visualization) is deployed. SQLite Database Server is used for storing and managing data in the cloud.

- *Application layer.* Software component designed as a light web application with responsive design suitable for use at different devices (laptops, tablets, phones).

4. Conclusions

Software architecture plays vital role for integrating heterogeneous processing and hardware components in complex socio-technical systems, and therefore, it should be carefully planned and designed. Use of architecture patterns helps in designing robust systems with desired performances. Selection of the most suitable architecture pattern depends on the purpose and specificity of the system, and in many cases it requires its adaptation. In this paper, review of existing software architecture patterns in socio-technical systems is presented, together with the author's experiences with some specific architecture patterns.

Based on the literature review and the author's personal experiences, some important challenges in designing software architecture for socio-technical systems can be distinguished. These challenges are:

1. Understanding the purpose and complexity of the whole system, with both technical and social issues.
2. Selecting the most suitable software architecture for the stated problem. If there is no existing pattern, creation of the new one or adaptation of the existing one is acceptable option.

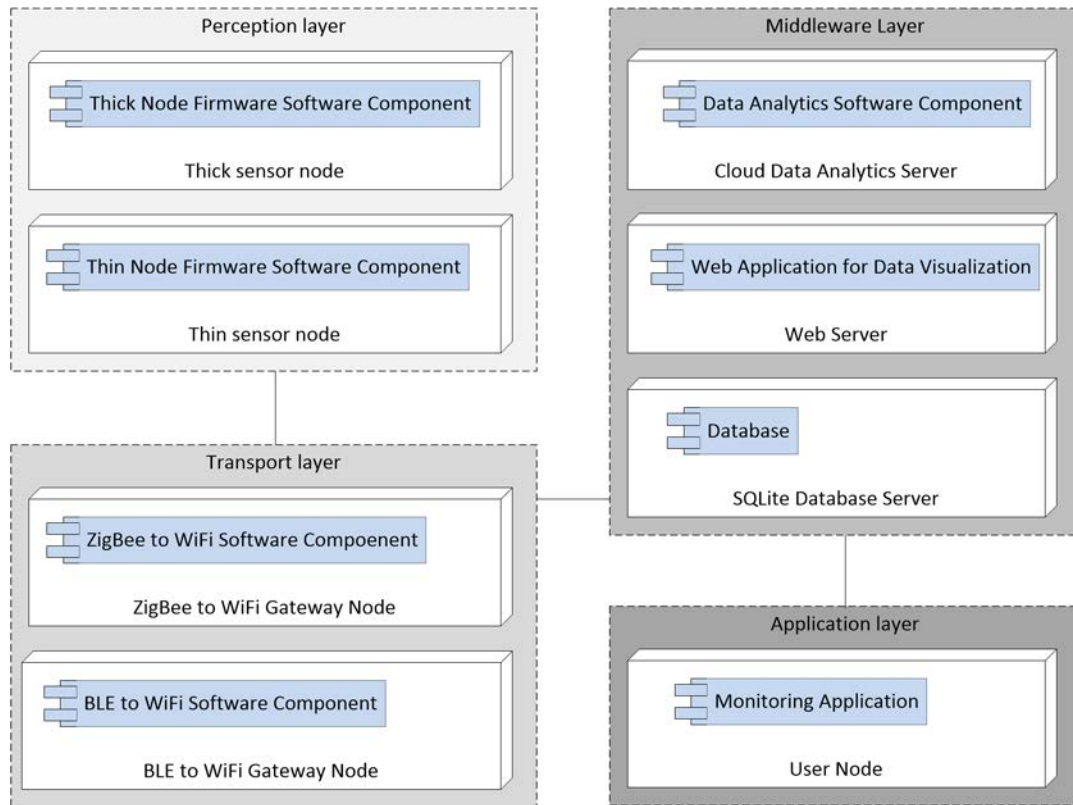


Figure 7: Layered architecture of software part of industrial monitoring system

3. Selecting the optimal set of technologies, methods and tools for creating system architecture, and based on that designing details of all software component and their relationships.
4. Selecting a set of architecture quality attributes to be met. In many cases, it is necessary to have compromises in the manner and scope of satisfying certain attributes.

5. Acknowledgments

Ministry of Education, Science and Technological Development, Republic of Serbia, supports this research under the project "The development of software tools for business process analysis and improvement", project number TR32044.

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] W. B. Rouse, N. Serban, Complex socio-technical systems - understanding and influencing causality of change, *Information Knowledge Systems Management* 10 (2011) 25–49. doi:10.3233/IKS-2012-0184.
- [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] 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.
- [5] I. Sommerville, *Software Engineering*, 9 ed., Addison Wesley, Boston, MA, USA, 2011.
- [6] E. Woods, Software architecture in a changing world, *IEEE Software* 33 (2016) 94–97. doi:10.1109/MS.2016.149.
- [7] S. Hyrynsalmi, S. Rauti, E. Kaila, Bridging the gap between software architecture and business model development: A literature study, in: *Proceedings of the 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, Opatija, Croatia, 2019, pp. 1519–1524. doi:10.23919/MIPRO.2019.8756974.
- [8] W. Hasselbring, Software architecture: Past, present, future, in: V. Gruhn, R. Striemer (Eds.), *The Essence of Software Engineering*, Springer, Cham, Cham, Switzerland, 2018, pp. 169–184. doi:10.1007/978-3-319-73897-0_10.
- [9] L. Khalid, *Software Architecture for Business*, Springer Cham, Cham, Switzerland, 2020. doi:10.1007/978-3-030-13632-1.
- [10] R. Capilla, A. Jansen, A. Tang, P. Avgeriou, M. A. Babar, 10 years of software architecture knowledge management: Practice and future, *Journal of Systems and Software* 116 (2016) 191–205. doi:10.1016/j.jss.2015.08.054.
- [11] I. Gorton, *Essential Software Architecture*, 2nd ed., Springer-Verlag Berlin Heidelberg, 2011. doi:10.1007/978-3-642-19176-3.
- [12] T. Bi, P. Liang, A. Tang, Architecture patterns, quality attributes, and design contexts: How developers design with them, in: *Proceedings of the 25th Asia-Pacific Software Engineering Conference (APSEC)*, Nara, Japan, 2018, pp. 49–58. doi:10.1109/APSEC.2018.00019.
- [13] D. Duggan, *Enterprises software architecture and design: entities, services, and resources*, John Wiley & Sons, Hoboken, NJ, USA, 2012.
- [14] M. Richards, *Software Architecture Patterns: Understanding Common Architecture Patterns and When to Use Them*, 1st edition ed., O'Reilly Media, Sebastopol, CA, USA, 2015.
- [15] H. Washizaki, S. Ogata, A. Hazeyama, T. Okubo, E. B. Fernandez, N. Yoshioka, Landscape of architecture and design patterns for iot systems, *IEEE Internet of Things Journal* 7 (2020) 10091–10101. doi:10.1109/JIOT.2020.3003528.
- [16] J. E. Bardram, Software architecture patterns for extending sensing capabilities and data formatting in mobile sensing, *Sensors* 22 (2022). doi:10.3390/s22072813.
- [17] P. Velasco-Elizondo, R. Marín-Piña, S. Vazquez-Reyes, A. Mora-Soto, J. Mejia, Knowledge representation and information extraction for analysing architectural patterns, *Science*

- of Computer Programming 121 (2016) 176–189. doi:10.1016/j.scico.2015.12.007.
- [18] Z. Stojanov, D. Dobrilovic, Software architecture quality attributes of a layered sensor-based iot system, in: I. V. Bychkov, D. Karastoyanov (Eds.), Proceedings of the 4th Workshop Information Technologies: Algorithms, Models, Systems (ITAMS 2021), volume 2984, CEUR Workshop Proceedings, Irkutsk, Russia, 2021, pp. 66–74.
 - [19] Z. Stojanov, D. Dobrilovic, J. Stojanov, Extending data-driven model of software with software change request service, Enterprise Information Systems 12 (2018) 982–1006. doi:10.1080/17517575.2018.1445296.
 - [20] G. Jotanovic, V. Brtko, J. Stojanov, Z. Stojanov, G. Jausevac, D. Dobrilovic, Smart city iot on-demand monitoring system using a drone fleet, in: Proceedings of International Conference on Future Access Enablers of Ubiquitous and Intelligent Infrastructures (6th EAI FABULOUS 2022), Zagreb, Croatia (online), 2022, pp. 1–18.
 - [21] Z. Stojanov, D. Dobrilovic, G. Jotanovic, D. Perakovic, G. Jausevac, V. Brtko, Software architectures in smart manufacturing: Review and experiences, in: I. Bychkov, Z. Stojanov, A. Tchernykh, A. Feoktistov (Eds.), Proceedings of the 1st International Workshop on Advanced Information and Computation Technologies and Systems (AICTS 2020), volume 2858, CEUR Workshop Proceedings, Irkutsk, Russia, 2020, pp. 160–168.
 - [22] Z. Stojanov, D. Dobrilovic, B. Perisic, Integrating software change request services into virtual laboratory environment: Empirical evaluation, Computer Applications in Engineering Education 22 (2014) 63–71. doi:10.1002/cae.20529.
 - [23] G. Milosavljevic, B. Perisic, A method and a tool for rapid prototyping of large-scale business information systems, Computer Science and Information Systems 1 (2004) 57–82. doi:10.2298/CSIS0402057M.
 - [24] D. Dobrilovic, Z. Stojanov, B. Odadzic, B. Markoski, Using network node description language for modeling networking scenarios, Advances in Engineering Software 43 (2012) 53–64. doi:10.1016/j.advengsoft.2011.08.004.

Многоуровневые эталонные модели бизнес-процессов

Сагит Валеев^{1,2}, Наталья Кондратьева¹

¹ Сочинский государственный университет, ул. Политехническая, 7, г. Сочи, 354053, Россия

² Уфимский государственный авиационный технический университет, ул. Карла Маркса, 12, г. Уфа, 450000, Россия

Аннотация

Современные высокотехнологичные отрасли, такие как авиационная промышленность, могут служить примером сверхбольшой системы, в которой бизнес-процессы отражают все этапы жизненного цикла сложных технических систем, включая этапы проектирования, испытаний, производства, эксплуатации, сбыта и сервисного обслуживания. Для обеспечения эффективности основных этапов жизненного цикла авиационной техники используются различные информационные платформы предприятий, объединенных в корпорацию, в которых в качестве базовых, наиболее важных компонентов используются модели бизнес-процессов. Для описания бизнес-процессов применяются различные объектно-ориентированные языки и средства их реализации. В настоящее время накоплен большой опыт использования моделей бизнес-процессов на различных предприятиях и технологий внедрения их в информационно-управляющие системы. Представляется актуальной разработка обобщенных многоуровневых эталонных моделей для оценки эффективности бизнес-процессов на различных этапах жизненного цикла авиационной техники.

Ключевые слова

Бизнес-процессы, эффективность, эталонные модели, иерархическая архитектура

1. Введение

В рамках концепции цифровой экономики авиационная отрасль может рассматриваться как сверхбольшая система. Сверхбольшие системы относят к классу «система систем». Система систем – это система, которая включает в себя функционально независимые подсистемы, независимые компоненты и подсистемы управления, а также характеризуется эволюционным развитием, эмерджентным поведением и географически распределенным пространственным расположением ее подсистем.

Сверхбольшие системы децентрализованы, разрабатываются и используются широким кругом заинтересованных сторон с конфликтующими потребностями, постоянно развиваются и состоят из разнородных частей. Сопровождение сверхбольшой системы обычно происходит одновременно с ее эксплуатацией и требует разработки новых методов управления [1-3].

Как известно, информационно-управляющие системы являются важным фактором, определяющим успешность эффективного функционирования промышленной отрасли. Решение, принятое руководством компании, должно эффективно управлять огромным объемом информации, чтобы максимизировать ее полезность. Управление большими объемами информации, генерируемой на различных этапах жизненного цикла продукции, использование информации с датчиков и данных о состоянии бизнес-процессов различных промышленных предприятий сегодня основаны на технологиях IoT и BigData [4-6].

4th International Workshop on Information, Computation, and Control Systems for Distributed Environments (ICCS-DE 2022), July 04–08, 2022, Irkutsk, Russia

EMAIL: vss2000@mail.ru (A. 1); knv24@mail.ru (A. 2)

ORCID: 0000-0002-0249-7087 (A. 1); 0000-0001-8699-1578 (A. 2)



© 2022 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

ICCS-DE 2022 Workshop Proceedings

Концепция больших данных объединяет приемы и технологии, позволяющие извлекать знания из огромных потоков данных в режиме реального времени. Активный интерес исследователей и практиков в области авиации к технологии BigData обусловлен реализацией высокоскоростных каналов передачи данных для динамических объектов, а также внедрением технологий NoSQL и Cloud Computing [7-9]. Генераторами потоков данных в авиации, например, являются различные датчики и устройства силовой установки и планера. При решении задачи группового управления воздушными судами обрабатывается значительный поток данных [10, 11]. Отмечается, что система управления газотурбинными двигателями может формировать во время полета массив данных объемом до 20 ТБ в час.

При анализе эффективности бизнес-процессов используются различные методы и технологии, основанные на различных математических моделях и методах искусственного интеллекта [12-18]

В качестве основных выводов по результатам анализа, проведенного в этой области, следует отметить отсутствие стандартов, определяющих требования к моделям бизнес-процессов, используемых для решения задач управления применительно к авиационной отрасли. Это не позволяет рассматривать авиационную отрасль как сложную систему, управляемую данными. Таким образом, в настоящее время целостно управлять и развивать авиационную отрасль невозможно.

2. Архитектурный подход к реализации бизнес-процессов в распределенных системах

На рис. 1 представлена обобщенная архитектура системы управления компанией и архитектура бизнес-процессов в компании. Архитектура бизнес-процессов отражает архитектурную организацию компании и включает в себя наборы бизнес-процессов, обеспечивающих эффективность жизненного цикла продукта. Основными элементами этой архитектуры являются бизнес-процессы, модели которых реализованы для различных уровней управления, интерфейсы между уровнями управления и базами данных.

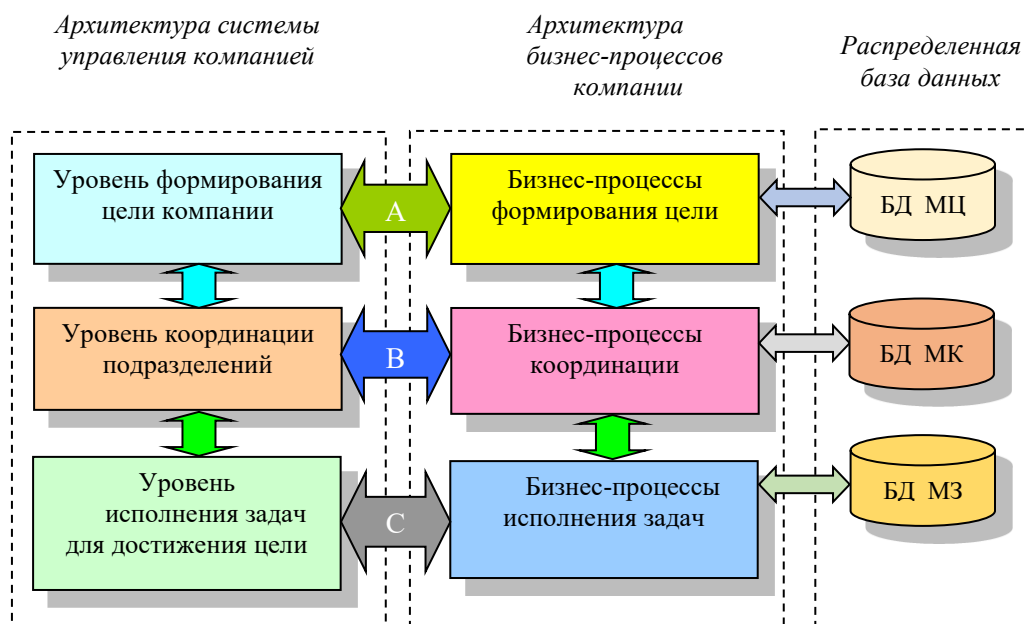


Рисунок 1: Архитектурный подход к организации бизнес-процессов и многоуровневая база данных эталонных моделей бизнес-процессов

В базах данных моделей хранится вся информация об истории выполнения бизнес-процессов формирования цели (БД_МЦ), бизнес-процессов координации подсистем (БД_МК) и текущем состоянии бизнес-процессов (БД_МЗ), что позволяет решать задачи анализа состояния производства и принимать эффективные управленческие решения.

Для каждого предприятия разрабатываются модели бизнес-процессов, которые являются эталонными для данного предприятия. Традиционно база моделей является важнейшим активом предприятия. Использование эталонных моделей бизнес-процессов для различных уровней позволяет оценить эффективность выполнения процессов и решить задачу выявления узких мест. На сегодняшний день нет обобщенных эталонных моделей, которые подходили бы для любого предприятия. Каждое предприятие отрасли использует свои модели, технологии их интеграции в информационно-управляющую систему.

3. Разработка обобщённых эталонных моделей бизнес-процессов отрасли

Как известно, идея стандартизации обмена информацией в открытых системах основана на следующих базовых принципах:

- Принцип снижения энтропии, т.е. приведения к единообразию и стандартизации формы объектов и информации об этих объектах.
- Принцип инвариантности, т. е. выявление требований, которые необходимы с учетом изменения свойств стандартизованных объектов.
- Принцип сохранения жизнеспособности стандартизованных требований, устанавливающий разумные пределы стандартизации.

Важную роль в рамках стандартизации проектных решений, основанных на идее открытых систем, играет внедрение CALS-технологий. Это позволяет сформировать основу для создания и использования единой информационной среды при проектировании, производстве, испытаниях и эксплуатации. Данные технологии успешно внедряются в авиационных холдингах АО «Компания «Сухой», ПАО «Туполев», ПАО «Корпорация «Иркут», ПАО «ОДК-Уфимское моторостроительное производственное объединение», ПАО «ОДК-САТУРН» и др.

В то же время применение больших объемов данных для проектирования, испытаний и эксплуатации систем управления и силовых установок самолетов сегодня находится только на начальном этапе. Причина кроется как в сложности проектирования систем управления силовыми установками и самолетами и неоднородности циркулирующих информационных потоков, так и в отсутствии современных стандартов, регламентирующих процессы разработки перспективных информационно-управляющих систем силовых установок и летательных аппаратов.

Как известно, распределенные социально-технические системы относятся к классу иерархических человеко-машинных систем, анализ состояния бизнес-процессов в которых является сложной задачей. Для ее решения применяются методы системного анализа, а именно система декомпозируется на подсистемы и бизнес-процессы, реализуемые на различных иерархических уровнях для достижения цели.

На рисунке 2 представлен процесс создания эталонной модели бизнес-процессов, включающий построение формализованных моделей бизнес-процессов, внедрение их в архитектуру предприятия и интеграцию их с архитектурой компании. Как следует из рассматриваемой схемы, для создания эталонной модели необходима обработка больших массивов данных, интегрированных с бизнес моделями на всех уровнях предприятия.

При построении моделей бизнес-процессов используются различные методологии, например, на основе спецификации BPMN (Business Process Model and Notation). Разработка, сопровождение, анализ и интеграция бизнес моделей осуществляется на основе различных платформ, например, CAMUNDA.

Управление бизнес-процессами в рассматриваемых системах реализовано на основе иерархически организованной системы управления. Каждый уровень управления использует информацию из различных источников данных: датчики, отчеты, устные сообщения и т. д.

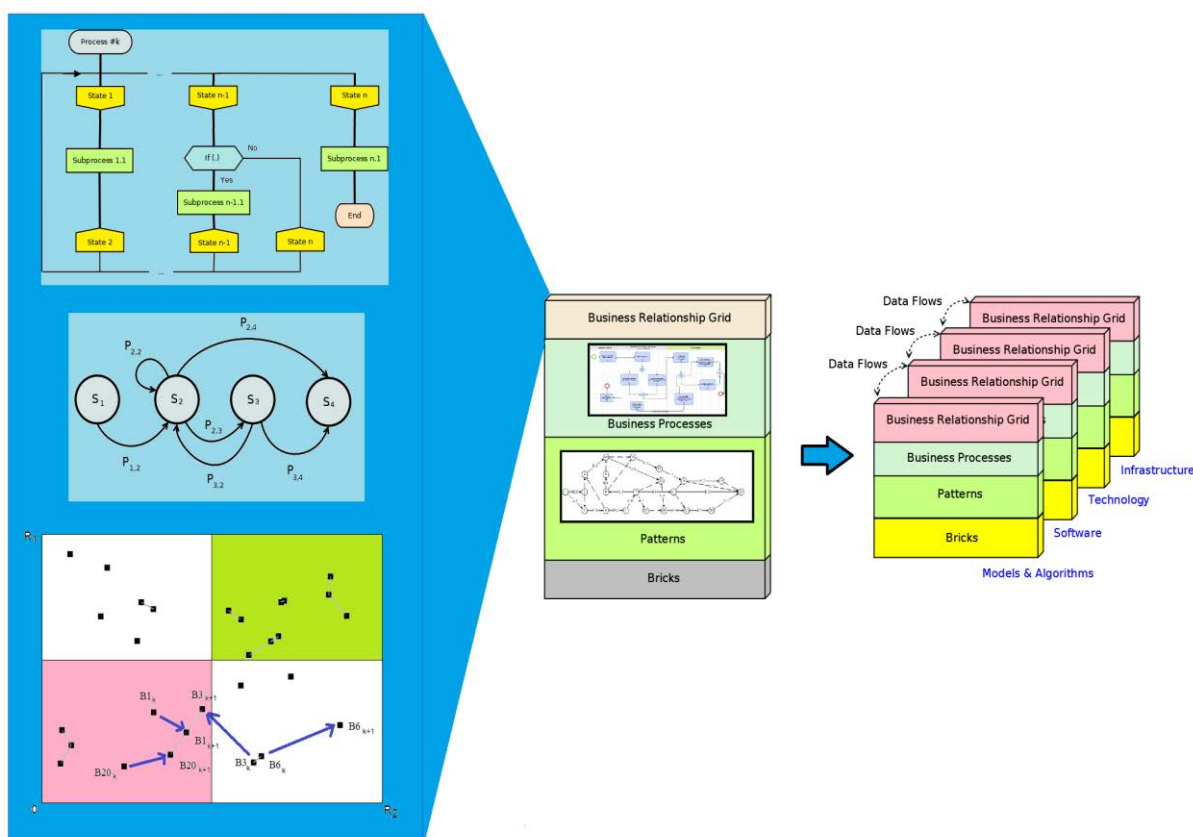


Рисунок 2: Процесс построения эталонной модели бизнес-процессов

Эта информация позволяет решать текущие задачи анализа состояния системы и принимать управленческие решения по повышению эффективности распределенной системы. Для анализа состояния распределенной организационно-технической системы необходима информация о текущем состоянии бизнес-процессов.

Каждый уровень иерархической системы управления содержит описание бизнес-процессов с разной степенью детализации, что требует формирования различных смысловых показателей, отражающих состояние бизнес-процесса. К таким показателям могут быть отнесены данные об основных характеристиках бизнес-процессов, представленные, например, в виде графиков, таблиц и т.п.

Сбор, передача, хранение и обработка больших потоков данных в организационно-технических системах является сложной технико-организационной задачей. Форма представления большого объема данных о текущем состоянии системы для повышения эффективности принятия решений должна отражать качественные характеристики бизнес-процессов.

Одним из подходов, используемых при анализе состояния распределенных информационных систем, является формирование снимков состояния. Пространство состояний распределенной организационно-технической системы отражает в зависимости от степени детализации производственных процессов их обобщенные показатели.

Сложность оценки состояния производственных процессов связана с необходимостью формирования наборов данных, характеризующих эти процессы, что для больших систем является нетривиальной задачей. Для оценки текущего состояния организационно-технической системы, разработки механизмов формирования снимков состояния системы требуется разработка протоколов обмена сообщениями между бизнес-процессами.

При анализе состояния вычислительных распределенных систем предприятий используется понятие системного снимка, отражающего состояние вычислительной системы и вычислительных процессов. Сложность создания снимка вызвана необходимостью обхода всех узлов системы и сбора данных об их состоянии. Для решения этой задачи используются

различные алгоритмы формирования массива данных. Для формирования моментального снимка состояния бизнес-процессов необходимо определить множество различных состояний, разработать протоколы обмена данными о состоянии процессов и решить задачу визуализации бизнес-процессов в заданном пространстве состояний.

4. Заключение

Рассматривается проблема разработки многоуровневых моделей эталонных бизнес-процессов для высокотехнологичных отраслей, которые можно отнести к классу сверхбольших систем, в которых бизнес-процессы отражают этапы жизненного цикла сложных технических систем, включая этапы проектирования, испытаний, производства, эксплуатации, сбыта и сервисного обслуживания. Для обеспечения эффективности основных этапов жизненного цикла используются различные информационные платформы предприятий, в которых в качестве наиболее важных составляющих используются модели бизнес-процессов.

В настоящее время накоплен большой опыт применения моделей бизнес-процессов на различных предприятиях, а также технологий их внедрения в информационно-управляющие системы. Анализ состояния бизнес-процессов распределенной организационно-технической системы в пространстве состояний позволяет фиксировать траектории бизнес-процессов. Для решения этой проблемы предлагается алгоритм формирования снимка системы бизнес-процессов. Визуализация бизнес-процессов в пространстве состояний позволяет повысить эффективность принятия решений.

Обсуждается архитектура бизнес-процессов отрасли. Отмечается, что она имеет многоуровневую архитектуру, отражающую сложность задачи управления, необходимость формирования бизнес-процессов для различных уровней управления, разработки интерфейсов и протоколов обмена информацией между уровнями.

Рассматривается обобщенная процедура формирования многоуровневой модели бизнес-процессов, включающая процессы построения формализованных моделей бизнес-процессов, анализ эффективности бизнес-процессов в пространстве состояний и внедрение моделей в архитектуру предприятия.

5. Литература

- [1] I. Bider, A. Jalali, J. Ohlsson, Adaptive case management as a process of construction of and movement in a state space, in: Y.T. Demey, H. Panetto (Eds.), *On the Move to Meaningful Internet Systems: OTM 2013 Workshops*, volume 8186 of *Lecture Notes in Computer Science*, Springer, Berlin, Heidelberg, 2013. doi:10.1007/978-3-642-41033-8_22.
- [2] S. Ghilardi, A. Gianola, M. Montali, A. Rivkin, *Delta-BPMN: A Concrete Language and Verifier for Data-Aware BPMN*, volume 1287 of *Lecture Notes in Computer Science*, Springer, Berlin, Heidelberg, 2021. doi:10.1007/978-3-030-85469-0_13.
- [3] F. Corradini, F. Fornari, A. Polini, B. Re, F. Tiezzi, A. Vandin, A formal approach for the analysis of BPMN collaboration models, *J. of Systems and Software* 180 (2021):111007. doi:10.1016/j.jss.2021.111007.
- [4] M. Ramos Merino, J. Santos, L. Alvarez Sabucedo, V. M. Rorís, J. Sanz-Valero, BPMN-E2: a BPMN extension for an enhanced workflow description, *Softw Syst Model* 18 (2019):2399–2419. doi:10.1007/s10270-018-0669-2.
- [5] Y. Ren, Y. Ding, Y. Zhang, P. Christofides, A three-level hierarchical framework for additive manufacturing, *Digital Chemical Engineering*, 8 (2021):100001. doi:10.1016/j.dche.2021.100001.
- [6] A. Kopp, D. Orlovskiy, D. Kuka, An approach to forming dashboards for business processes state analysis, in: volume 51 of *Bulletin of National Technical University KhPI, Series: System Analysis Control and Information Technologies*, 2017, pp. 44-52. doi:10.20998/2079-0023.2017.51.07.
- [7] K. Cox, *Business process modelling*, in: K. Cox, *Business Analysis, Requirements, and Project Management: A Guide for Computing Students*, CRC Press, Boca Raton, Florida, the U.S.A., 2021, pp. 19-39. doi: 10.1201/9781003168119-2.

- [8] S. Guerreiro, Locating and visualizing the most valuable business processes states in large control policy graphs produced by POMDP, in: Proceedings of 2017 IEEE 19th Conference on Business Informatics (CBI), Thessaloniki, Greece, 2017, pp. 89-98. doi:10.1109/CBI.2017.35.
- [9] S. Valeev, N. Kondratyeva, Process Safety and Big Data, Elsevier, Amsterdam, Netherlands, 2021. doi:10.1016/C2019-0-03546-7.
- [10] S. Valeev, N. Kondratyeva, R. Karimov, M. Verkhoturov, T. Islamgulov, L. Shekhtman, Production planning in a construction company as an element of Gartner enterprise architecture, in: I. Bychkov, A. Tchernykh, A. Feoktistov (Eds.), Proceedings of the 3rd International Workshop on Information, Computation, and Control Systems for Distributed Environments, Irkutsk, Russia, July 5-9, 2021, pp. 198–208. URL: <http://ceur-ws.org/Vol-2913/>.
- [11] A. Tchernykh, I. Bychkov, A. Feoktistov, S. Gorsky, I. Sidorov, R. Kostromin, A. Edelev, V. Zorkaltsev, A. Avetisyan, Mitigating uncertainty in developing scientific applications in integrated environment,” in: Proceedings of the Institute for System Programming of the RAS, 33(1), 2021:151–172. doi:10.15514/ISPRAS-2021-33(1)-11.
- [12] M. de Leoni, W. Aalst, M. Dees, A general framework for correlating business process characteristics, in: S. Sadiq, P. Soffer, H. Völzer (Eds.), Business Process Management BPM 2014, volume 8659 of Lecture Notes in Computer Science, Springer, Cham, 2014. doi:10.1007/978-3-319-10172-9_16.
- [13] M. Saikia, S. Majumder, S. J. Borah, Snapshot in distributed system: a brief survey, in: Proceedings of National Conference on Emerging Trends and Applications in Computer Science – 2010, St. Antony’s College, Shillong, 9-10 April 2010, pp. 256–259.
- [14] I. Skopin, Modelling purposeful processes based on the geometric representation of their trajectories, in: I. Bychkov, A. Tchernykh, A. Feoktistov (Eds.), Proceedings of the 3rd International Workshop on Information, Computation, and Control Systems for Distributed Environments, Irkutsk, Russia, July 5-9, 2021, pp. 164–181. URL: <http://ceur-ws.org/Vol-2913/>.
- [15] V.D. Parondzhanov, Visual syntax of the DRAKON language, Programming and Computer Software. Official English Translation of Programmirovaniye 21(3) (1995):142–153.
- [16] F. M. Maggi, C. D. Francescomarino, M. Dumas, and C. Ghidini, “Predictive monitoring of business processes,” in Proceedings of the 26th International Conference on Advanced Information Systems Engineering (CAISE 2014), ser. Lecture Notes in Computer Science, vol. 8484. Springer, 2014, pp. 457–472.
- [17] J. Becker, D. Breuker, P. Delfmann, and M. Matzner, “Designing and implementing a framework for event-based predictive modelling of business processes,” in Proceedings of the Enterprise Modelling and Information Systems Architectures (EMISA 2014), ser. LNI, vol. P-234. GI, 2014, pp. 71–84.
- [18] D. Breuker, , M. Matzner, P. Delfmann, J. Becker, , and and, “Comprehensible predictive models for business processes,” MIS Quarterly, vol. 40, no. 4, pp. 1009–1034, 2016.

Methodology for Collecting, Processing, and Analyzing Socio-Economic Information

Olga Basharina ^{1,2}

¹ Matrosov Institute for System Dynamics and Control Theory of SB RAS, Lermontov St., 134, Irkutsk, 664033, Russia

² Ural State University of Economics, March 8 St., 62, 620144, Yekaterinburg, Russia

Abstract

The article deals with topical issues of ecological monitoring of the protected Baikal natural territory. A methodology is presented that collects and analyzes socio-economic indicators for modeling the functioning of existing and planned infrastructure facilities on the coast of Lake Baikal. This methodology adapts the known methods and approaches for collecting, recording, and analyzing the anthropogenic impact on the ecology of the region. It was effectively used in the system of mathematical and information modeling of the energy complex.

Keywords

Ecology, Baikal natural territory, infrastructure objects, energy supply, mathematical and information modeling

1. Introduction

Currently, the tasks of environmental monitoring are particularly relevant for protected natural areas, in particular the Baikal Natural Territory (BNT), which has unique water, land, biological and other important resources. Scientists and researchers offer various methods for collecting and recording socio-economic indicators and assessing their impact on the environment. One of the pressing environmental problems is the reduction of emissions from the operation of heat and power supply systems [1, 2, 3].

The purpose of this study is to develop a methodology for collecting, processing and analyzing socio-economic data, adapting known methods and approaches in relation to electric power complexes. The environmental monitoring data obtained in this way are used in the construction of mathematical models of both existing and planned energy infrastructure facilities. The collection and analysis of socio-economic indicators makes it possible to calculate not only the anthropogenic impact on the ecological situation of the BNT, but also to investigate, predict and optimize the technological, economic and environmental performance of the studied infrastructure facilities [4].

2. Materials and methods

This technique is included in the multilevel system of mathematical and information modeling of BNT infrastructure facilities and is based on the vulnerability analysis of the energy complex. A global vulnerability analysis is carried out on the basis of an assessment of the performance of the object under study in critical conditions: adverse weather conditions, fluctuations in energy prices, a significant increase in demand for heat and electricity, etc. the construction of which will have the most adverse consequences for the functioning of the entire complex [5].

4th International Workshop on Information, Computation, and Control Systems for Distributed Environments (ICCS-DE 2022), July 04–08, 2022, Irkutsk, Russia

EMAIL: basharinaolga@mail.ru

ORCID: 0000-0002-7151-782X



2022 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

ICCS-DE 2022 Workshop Proceedings

The methodology for collecting, processing, and analyzing socio-economic data includes the following steps:

1. Determination of socio-economic indicators, the change in the values of which may affect the vulnerability of the energy complex.
2. Collection, reduction, and correction of retrospective data. Checking the correctness of the information, eliminating anomalous data? and filling in their gaps.
3. Statistical analysis of data.
4. Modeling and forecasting of socio-economic indicators.
5. Assessment of the quality of forecast values.

2.1. Tourist flow modeling

The concept of the configuration of the energy complex, which is one of the components of the analysis of its vulnerability, also includes socio-economic indicators of the functioning of infrastructure facilities. The values of these indicators directly depend on the number of consumers of the services of these objects: the local population and various categories of tourists.

The population of the BNT as of January 1, 2021, is 1.43 million people, and the number of tourists who visited the Irkutsk region and the Republic of Buryatia in 2021, according to regional tourism agencies, was about 1.5 million people. (excluding self-organized tourists). The intra-annual dynamics of tourist flows are subject to seasonal fluctuations associated with the natural and climatic features of the territory and periods of business activity and leisure of potential tourists. Political and social events also affect their number.

Due to the absence of a strict system for recording tourists visiting the BNT, the collection and analysis of data on tourist flows can be implemented based on the approach developed under the leadership of Evstropieva at the Institute of Geography of SB RAS [6]. It takes into account the heterogeneity of the tourist flow in the context of recreational areas, uses different methods of data collection depending on the category of tourists (Table 1), and allows modeling the flow of tourists in a certain period (day, month, season, year).

Table 1

Methods for collecting data by categories of tourist flow

Tourist flow category	Methods and main sources of data collection
Organized rest in collective accommodation facilities	Statistical data from official and Internet sources
Amateur tent vacation	Field observations; interviewing
Route tourism	Data on registration of tourist groups in the Ministry of Emergency Situations
Ecotourism in specially protected natural areas	Data on issued permits to visit objects; field observations; interviewing
Winter sports holidays at ski resorts	Data on the number of purchased subscriptions
Excursion tourism	Collection of official statistics of attendance of excursion objects; field observations; interviewing
Holidays with accommodation in the households of the local population	Field observations; interviewing

The complexity of obtaining the full amount of data based on the methods presented above determines the need to calculate tourism marketing indicators, for example, the maximum and average daily number of tourists, the number of tourists staying at a time, the average length of stay, etc. Complementing the initial and calculated data allows you to perform an expert assessment of the tourist flow indicators.

The use of this technique makes it possible to calculate various indicators of the flow of tourists, for example, the load factor for an individual or complex of tourist facilities or recreational areas. The intra-

annual dynamics of this indicator makes it possible to analyze and predict peak loads on various infrastructure facilities. In the framework of the study, not only abnormally low air temperatures, but also high values of tourist flow volumes can be a disturbing factor for the functioning of the energy system.

The calculation of the peak load of heating networks under these disturbances can be carried out on the basis of statistical data on the number of heated objects and their characteristics. Within the studied natural area, there are quite a lot of small heat sources, such as electric boilers and electric heating devices and furnaces. They use wood and coal as fuel. These sources are widely used in individual residential and commercial premises, tourist and recreation centers, but, unfortunately, there is no data on accounting for their fuel needs.

Thus, it is necessary to be able to assess the load on the electrical networks due to the increased consumption of electricity for heating during the cold season and with the influx of tourists. Average per capita electricity consumption for heating purposes can be calculated using the following formula [8]:

$$E = \frac{(P_w - P_s) \cdot t}{k}, \quad (1)$$

where P_w – maximum power consumption in winter, kWh; P_s – maximum power consumption in summer, kWh; t – time of using the maximum power consumption, hour; k – number of consumers, pers.

This formula allows you to determine the average per capita electricity consumption for heating both for an individual private house and for a campsite or village, taking into account the local population and the number of tourists.

2.2. Assessment of anthropogenic impact on the ecology of the region

Modeling the tourist flow makes it possible to assess not only the vulnerability of the energy complex but also its impact on the ecology of the region because it is one of the main sources of environmental pollution. Thus, the transition to alternative sources of renewable energy is today a top-priority and urgent task at the federal and regional levels to protect the BNT.

A wide range of scientific research has been carried out to assess the impact of energy facilities on the environment and the possibility of switching to more energy-efficient and environmentally friendly renewable energy sources [2, 9]. For the production of electricity at the BNT, you can use photo and wind power plants, small hydroelectric power plants, and gas-generating power plants; and for the production of thermal energy: solar heat supply systems, systems with heat pumps, heat supply systems with wind turbines, wood-fired boilers and other sources [10].

Table 2 presents the results of replacing traditional energy sources with renewable ones in the form of coefficients of change in fuel consumption and pollutant emissions in relation to the current situation. The expected reduction in the mass of emissions in the atmosphere when replacing sources of electricity and heat with renewable ones will be 9.2 t/year and 170 t/year, respectively [2].

The calculation of the tourist flow indicators also makes it possible to assess the impact of road transport on the ecological situation of the BNT and the load on the transport infrastructure of the region [7, 11]. In addition to polluting the soil, water, and air with fuels and lubricants, and exhaust gases, cars destroy natural landscapes, vegetation, and the habitat of rare living organisms. An analysis of traffic flow indicators is necessary for the implementation of the program for the development of a tourist and recreational zone and a set of environmental measures aimed at reducing the negative impact of vehicles on the environment on the coast of Lake Baikal.

Estimating the number of tourists makes it possible to determine the volumes of medical, transport, and other social services necessary for the local population and vacationers. These data can be used in the construction of various infrastructure facilities, predicting the anthropogenic impact on the ecology of the region, etc. [7].

Table 2

Factors for fuel consumption and pollutant emissions

Energy production / number of facilities	Mass of emissions and waste					
	Fuel consumption	Solid particles	Sulfur oxides	Nitrogen oxides	Oxides of carbon	Ash and slag waste
Electricity generation using renewable energy sources / 5 objects	0.534	–	0.501	0.572	–	–
Production of thermal energy using heat pumps instead of coal- fired boilers / 6 objects	0	0.989	0.996	0.998	0.995	0.994

The standards of wastewater treatment are determined in the order of the Ministry of Natural Resources of Russia No. 63 of March 5, 2010. But according to the analytical report on the development of a regional concept for the development of a drainage system in the central ecological zone of the Baikal natural territory [12], none of the five existing sewer treatment plants ensures the cleaning of waste Water to these requirements. Half of the wastewater is discharged on the ground without cleaning. In addition, the volume of wastewater is significantly affected by the steadily increasing number of organized and amateur tourists every year (Table 3).

Table 3

Dynamics of indicators of 2018 compared to 2008

Area	Population, people	Number of tourists, people	Volume of waste water, m ³ / day
Olkhonsky	-6	+5648	+1114.4
Irkutsk	+409	+2488	+941.4
Slyudyansky	-1473	+1324	+3007.7

The consequences of environmental pollution are reflected in the social and industrial sectors of the economy as financial and material costs for their prevention or elimination. These costs can be estimated using the methodology for determining the prevented environmental damage, according to which specific cost estimates of damage from the release of 1 conventional ton of the reduced mass of pollutants are calculated for the economic region. Thus, it is possible to calculate the possible anthropogenic consequences on the ecology of the region [3].

3. Conclusions

A methodology for the collection, processing, storage and analysis of socio-economic data is proposed, which is used to monitor the functioning of the BNT infrastructure facilities. The application of this technique is one of the stages of conceptual modeling of energy complex facilities. The technique was used in the implementation of the service of simulation modeling of environmentally friendly equipment of BPT infrastructure facilities [13], which operates in a heterogeneous distributed environment with multi-agent control of computing [14]. It is possible to develop and use the developed methodology in other subject areas.

4. Acknowledgements

The study was supported by the Russian Foundation for Basic Research and the Government of the Irkutsk Region, project No. 20 47 380002 r_a “Mathematical and Information Modeling of Infrastructural Facilities of the Baikal Natural Territory”, as well as the Ministry of Science and Higher Education of the Russian Federation, project No. FWEW 2021 0005 “Technologies for the Development and Analysis of Subject oriented intelligent group control systems in non-deterministic distributed environments” (Reg. No. 121032400051-9).

5. References

- [1] B.G. Saneev, E.P. Maysyuk, I.Yu. Ivanova, Assessment of energy-related environmental impacts during the implementation of promising projects for the development of deposits in the Arctic territories of the Russian eastern regions, *Arctic: Ecology and Economy* 11-4 (2021) 466–480. doi: 10.25283/2223-4594-2021-4-466-480.
- [2] B.G. Saneev, I.Yu. Ivanova, Plan of measures for the development of environmentally friendly energy supply of the central ecological zone of the Baikal natural territory, ISEM SB RAS, 2020. URL: http://www.sbras.ru/files/news/docs/126_po_energetike_14.12.2020.pdf.
- [3] N.B. Botoeva, Audit of anthropogenic air pollution in the central ecological zone of the Baikal natural area, *Management of economic systems: (electronic scientific journal)* 115 (2018).
- [4] A.G. Feoktistov, R.O. Kostromin, S.A. Gorsky, Development of intelligent technology for decision making support in queuing systems based on their simulation modeling on supercomputers, *Modern high technologies* 2 (2020) 76–80. doi: 10.17513/snt.37918.
- [5] A.V. Edelev, N.M. Beresneva, S.A. Gorsky, Vulnerability Analysis of Energy Infrastructure and its Implementation, *Modern high technologies* 1 (2022) 47–52. doi: 10.17513/snt.39008.
- [6] O.V. Evstropieva, A.Yu. Bibaeva, E.D. Sanzheev, Modeling tourist flows at the regional and local levels. Experience of implementation in the central ecological zone of the Baikal Natural Territory, *Service and Tourism: Current Challenges* 13-1 (2019) 85–97. doi: 10.24411/1995-0411-2019-10110.
- [7] O.V. Evstropieva, Regional effects of the development of the international and national tourism system (on the example of the Baikal region), Ph.D. thesis, Institute of Geography named after V.B. Sochavy SB RAS, Irkutsk, 2021.
- [8] G.O. Borisov, Options of heat and energy supply of the central ecological zone of the Baikal natural territory, *Bulletin of the BSC SB RAS* 26-2 (2017) 103–109.
- [9] G.O. Borisov, D.Z. Ubonova, Problems and perspectives of power supply of the central ecological zone of the baikal natural territory, *Economic Bulletin of the East Siberian State University of Technology and Management* 8 (2019) 25–32. doi: 10.38028/ESI.2020.17.1.007.
- [10] D.N. Karamov, A.V. Edelev, A.G. Feoktistov, Modeling energy supply of objects located in the protected Baikal natural territory on the basis of renewable energy sources, *International technical-economic journal* 5 (2020) 7–24. doi: 10.34286/1995-4646-2020-74-5-7-24.
- [11] S.Z. Vologzhina, S.A. Novikova, F.M. Yasko, Air Pollution by Auto Transport Emissions on the Territory of Tourism-Recreation Coastal Zone of Lake Baikal, *The bulletin of Irkutsk State University. Ser. Earth Sciences* 22 (2017) 15–29.
- [12] Analytical report on the development of a regional concept for the development of a wastewater system in the central ecological zone of the Baikal natural territory, Autonomous non-profit organization "Information Center in Design", Moscow, 2019. URL: <https://irkobl.ru/sites/gkh/documents/komhoz/Аналитический%20отчет%20АНО%20ИЦД.pdf>
- [13] A.G. Feoktistov, R.O. Kostromin, I.A. Sidorov, S.A. Gorsky, O.Yu. Basharina, Digital twins of operation processes for environmentally friendly equipment of infrastructure object, *Modern high technologies* 1 (2021) 57–62. doi: 10.17513/snt.38471.
- [14] A.G. Feoktistov, R.O. Kostromin, S.A. Gorsky, I.V. Bychkov, A.N. Tchernykh, O.Yu. Basharina, Redundancy and Uncertainty-Based Algorithms for Computation Planning, in: *Proc. ISP RAS*, vol. 34, issue 1, 2021, pp. 123-140. doi: 10.15514/ISPRAS-2022-34(1)-9.

Software Reengineering: A Qualitative Study in two Software Companies

Aleksandra Stojkov¹, Zeljko Stojanov¹

¹University of Novi Sad, Technical Faculty “Mihajlo Pupin”, Zrenjanin, Republic of Serbia

Abstract

Software reengineering is part of the software life cycle and plays a major role for business companies whose business depends on software quality. The quality of software is to be able to follow all business processes and the possibility to grow together with the company it supports. Over the years, maintenance has become more and more complicated and expensive, so reengineering is needed. To find out whether reengineering is done and how often, what are the reasons and whether it is more optimal than making a new application, we organized a qualitative research based on the interviews with two employees in two software companies in Serbia. They shared their experiences, opinions, and knowledge, after which analysis was performed. The results reveal three important segments of the reengineering process: Project Scope, Reengineering Process and Software System. The existence of multiple segments indicates that the process is complex and requires detailed analysis before execution. The results can be useful to those who want to continue research in this direction or to conduct similar research. In addition, experts from industry can find systematized facts on software reengineering and adapt them to their practice.

Key words

Software reengineering, Software maintenance, Reengineering process, Qualitative research

1. Introduction

Reengineering can be considered as the last step in software modification. When so many changes and modifications are made to the software and when it becomes difficult and expensive to maintain, reengineering occurs. Modifications are all changes that are performed on software. This can be updates, resolving bugs, improving features, launching a new version of the program. It can be grouped into four forms: *Corrective* – correction of identified problems in software functioning, *Adaptive* - adapting the program to a new technical environment and a new business, *Perfective* - improving the system in terms of performance and sustainability, and *Preventive* - modifications and adjustments to the software to extend its life cycle [1].

Reengineering is a process of changing software in way to change their shape, while retaining their functionality. Usually it is a set of process activities which result in a new form of the application. These processes are reverse and forward engineering, restructuring, re-documentation and translation [2].

In order to do reengineering, an existing application is required as input, to create a new version of software. Before starting the reengineering, it is necessary to analyze the software in detail and recreate it and update with a newer version [3]. With clearly defined goals before the process begins, reengineering would be cost-effective. Cost-effectiveness is also affected by the structure of the application, and documentation of the existing software [4]. The cost of reengineering is lower than building a new software, and the risk during reengineering is also lower than redeveloping a new software. The main difference between reengineering and building new software is that when writing a new software it starts from scratch, and for reengineering the starting point is an old software. The old software has some architecture that consists of interconnected structures that provide functionality,

4th International Workshop on Information, Computation, and Control Systems for Distributed Environments (ICCS-DE 2022), July 04–08, 2022, Irkutsk, Russia

EMAIL: aleksandra.stojkov@tfzr.rs (A. Stojkov); zeljko.stojanov@uns.ac.rs (Z. Stojanov);

ORCID: 0000-0002-1078-7068 (A. Stojkov); 0000-0001-6930-5337 (Z. Stojanov);



© 2022 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

ICCS-DE 2022 Workshop Proceedings

proper operation of the system and response to user requirements [5]. Software also includes definitions of metrics, patterns, visualizations, soft skills, integration architecture, features, characteristic, evolutionary architecture, styles, and more [6]. Elements of which the software consists of are software artifacts. The role of artifacts in reengineering has been described in our previous research [7]. During the reengineering process data are also reengineered [8].

There are many factors facilitating reengineering. These factors can be business growth, outdated technologies, increasing complexity or just following trends [2].

The purpose of this paper is to investigate the real situation with software reengineering in software companies in Serbia. The results were obtained by questioning two experienced developers about their experience in software reengineering. The methodologies described in the reference [9] were used to collect, analyze, and present the research results. A detailed description of the methods is given in the methodology section.

The paper consists of several sections. The related work section lists papers that have been engaged in research and used elements of the same methodology as applied in this paper. The research method section describes the process of conducting research and process of analyzing research data. The results section shows results emerged during the research. The conclusion provides summarized results and further research.

2. Related Work

Software reengineering is a common method for extending software life cycle, and a lot of research has been done on this topic. This section will present some papers dealing with software reengineering in which an empirical research was conducted.

In the paper [10] the authors performed a quantitative analysis in two steps. The first step was to compile an interview guide followed with interviewing 5 participants. In the second step, with the help of a previous interviews' answers, a survey was compiled and was sent to eighteen participants. All participants were from IT companies. The research results show the activities and challenges in software reengineering towards microservice architecture.

Qualitative analysis is presented in the reference [11]. The authors conducted six interviews with respondents working in software development companies. The questions referred to the software reuse strategy and the results of the research will be used to improve their approach described in the paper.

A study in paper [12] investigates how much the software reuse process costs. They combine qualitative and quantitative analyses. Data were collected with the help of 26 interviews from various organizations and 57 scientific literature sources. The analysis provides a software reuse strategy and provides evidence that confirms, but also, that refutes the hypothesis. Reuse is considered as an important support for software reengineering.

3. Research Methods

Research methods are following by methodological guidelines describes in reference [9]. The process contains three main steps, as it is presented in Figure 1.

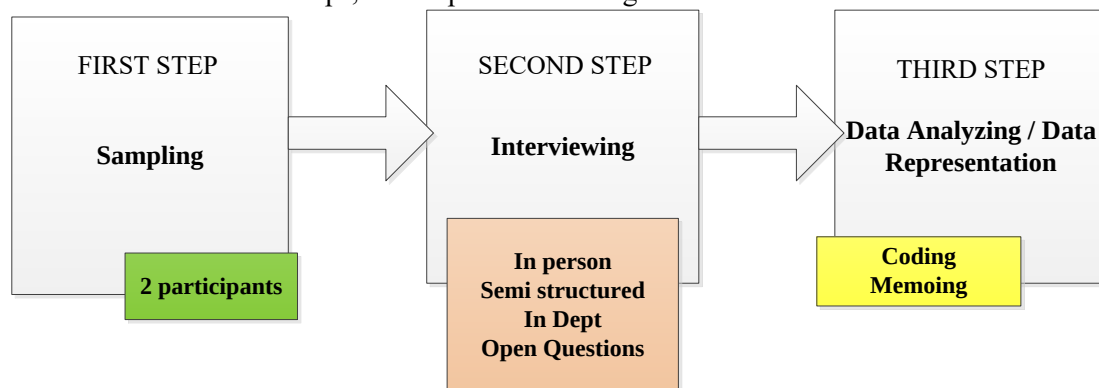


Figure 1: Steps of qualitative research using interviewing

Sampling. The first step was to define the sample population for research. Software companies in Serbia were searched and a list of potential participants was made. We decided to use qualitative analysis, which is an analysis of a small number of carefully selected cases. In addition, qualitative methods can provide depth insight into the real practice based on the participants experience. This ensures that research findings emerge from the participants words and their experience.

We selected two software experts from two companies in Serbia for this study. After that, one employee from each company was selected and general questions were asked. It was the second round of selection. Based on the answers, we concluded that two people were eligible to participate in further research, because their experience and knowledge are closely related to the topic. The interviews were conducted with the selected respondents.

Table 1 shows data of respondents and the companies they come from. Table has columns on position and experience for participants and size and industries for companies where they work. The size of the company is written based on the number of employees. The range and ranking criteria for company size can be seen in [13].

Table 1

Participants and company information

Participant			Company	
	Position	Experience	Size	Industries
p1	Senior Software Engineer	7	small	Software Development
p2	Software Developer	8	micro	Systems and Services Design

Interviewing. The interviews that were made were semi-structured, consisted of open-ended questions. Also, these interviews can be classified as In-Depth and were performed in person. This structure was chosen because respondents had more freedom to explain the answer, and it encourages discussion and inclusion of additional questions. From the face-to-face interview you can learn more about participant's views and thoughts. The interview process took approximately 45-60 minutes. In order to collect as much data as possible, the whole conversation was recorded with an audio recorder, and later the transcription was performed and saved as a word document.

Analyzing/Representation. After the data were collected, it was necessary to analyze them. The first step was reading and coding. The method of In-Vivo, Initial-Open Coding was used. This type of coding meant that by reading the text, the key facts are extracted and marked with short labels using words and phrases from that text (participants words). With a combination of thematic and focus coding all codes were later categorized into two major groups. This categorization shows more general data, but also contains definitions of all individual cases. The last step is to summarize the results by writing a memos. A memo is a representation of what is inferred from the data and in what way they are relate to each other.

4. Results

The whole research process took a couple of months. After choosing the topic and method of research, it was necessary to devise questions that will cover the entire research area and encourage the respondents to talk. The selection of participants was also important, because not everyone deals with this topic. The interviews were conducted in a pleasant atmosphere and the conversations lasted from 45-60 minutes, during which the respondents gave answers and theirs opinions on the questions asked. After the recorded interview, the transcription and analysis of the results were performed. The results are presented with the help of coding after which the codes are grouped by topics. In Figure 2 all Initial codes and topics to which they belong are presented. The diagram shows that some codes belong to both areas, because there is no clear boundary between them.

Project Scope consists of two initial codes. It defines the type of process [ic1] and the business activities/decisions [ic2]. It considers how the work related to the software system will be performed.

Reengineering process was created by generalizing 10 initial codes. All codes relate to the field of reengineering process. The reengineering process begins by defining the most common types of

software for which reengineering is performed [ic8], lists the reasons why reengineering should be done [ic3], and shows the frequency of the process in practice [ic5]. It also shows how reengineering requirements are monitored [ic6], what changes are made [ic4], what technologies are used [ic7], and defines the architecture of the new system [ic9].

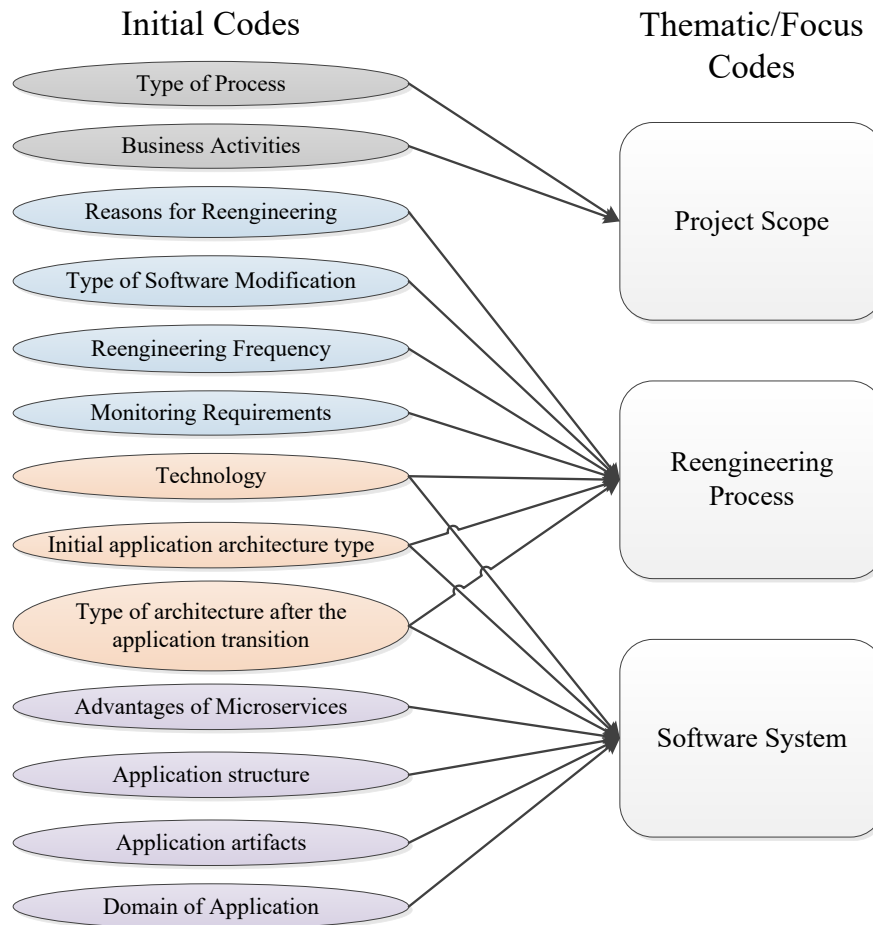


Figure 2: Initial Codes grouped on the basis of Theme

Software System is a group of 7 initial codes. It contains information of system characteristics. It describes what is the type of architecture [ic8, ic9] and its benefits [ic10], what is the structure [ic11] and what system artifacts [ic12] does it contain. Also it defines in what technology [ic7] software product was built and for which software application domain [ic13].

In the Table 2 all initial codes, their memos and text excerpts are listed. Written text in the form of a memo is an explanation and meaning of the initial code. The text excerpt is a set of key elements taken from the text of the interview. This is what the respondent said in response to the question. Based on these values, the initial coding was performed.

Table 2
Defining Initial Codes

	Initial Codes (ic)	Memo	Text Excerpt
ic1	Type of Process	Defines processes of software development and maintenance.	Software Engineering [p1], Reengineering [p1], upgrade [p1]
ic2	Business Activities	Decides what work will be performed and in what way	Building a new application [p1, p2], Involvement in existing projects [p1, p2]

ic3	Reasons for Reengineering	Gives the reasons why it is necessary to reengineer the application	Complexity [p1], [p2] Finding a better way to do the same thing [p1], Badly written code [p1], Client's needs [p2], amendments to laws and regulations [p2], change the data type [p2], monitoring market demands [p2], technology advancement [p2]
ic4	Type of Software Modification	Determines what changes can be made to the software to improve its quality	adding new features [p1, p2], Application improvement [p1], Upgrade library version [p1], Modify existing functions [p1], Bug fixing [p1], Defining architecture [p1], writing tests [p2]
ic5	Reengineering Frequency	Shows the extent to which reengineering is performed	Not often [p1] Often [p2]
ic6	Monitoring Requirements	Shows which tools help guide the reengineering process	Jira for task tracking [p1]
ic7	Technology	Explains which technology is best for creating software	It is not always good to adopt the latest technologies. The latest proven and tested technologies should be used. [p2] Best to use open source technologies [p2]
ic8	Initial application architecture type	Displays which is an architecture system that needs to be reengineered	Monolithic architecture [p1], Domain service architecture [p2]
ic9	Type of architecture after the application transition	Shows the architecture of the system after reengineering	Monolithic architecture web application [p1], Microservices architecture [p1], [p2] Micro-frontend [p2]
ic10	Advantages of Microservices	Explains the reasons why microservices are better to use	Each microservice in charge of one function. [p2] Language agnostic. [p2] Division of responsibilities. [p2] Reduced possibility of errors [p2] Easily replaceable [p2]
ic11	Application structure	For which platform was the software system created	Web application [p1], Console application [p1]
ic12	Application artifacts	Which building elements make up the application	Frontend and backend [p1], Load balancer [p1], Server [p1], Specification requirements [p1]
ic13	Domain of Application	For what purpose is the software system used	Medicine [p1], Transport [p1], Statistics [p1], Dashboard [p1], GDPR [p1], Marketing [p1], Reservation [p1], Telecommunications [p1], Banking systems [p1], accounting and billing for small business owners [p2], Human Resource Management [p2] management of automatic and manual tests [p2]

5. Conclusion

In this paper qualitative analysis was performed in a small-scale study on software reengineering. Two respondents were interviewed and their responses were represented. The presented results indicate that the reengineering process contains 3 segments. The first segment, Software System, examines the software, analyzes its advantages and disadvantages. It provides answers to questions: What is the function of software, what is it intended for, how it was built and what elements does it contain. The second segment is Project Scope. It determines in which direction the existing software will be developed. The options are to start over (creating new software with the same functions), upgrading and modifying an old application or reengineering. The third segment, Reengineering Process, occurs if it is estimated that the application can no longer be upgraded and changed, when it grows so much and becomes complicated and expensive to maintain. The results also show that the situations are diverse, and that the respondents responded differently to how often reengineering is done.

The results presented in the paper, as well as the described method can be useful to anyone who wants to do similar research, or wants to get information or do research that builds on reengineering software. In addition, research results can be useful for software experts from industry who can find systematization of topics related to software reengineering.

The limitation of this research is that it was done on a small sample with two software experts. However, small scale qualitative studies are very important and useful, because each case could be analyzed in more detail, which is the point of qualitative analysis.

In the next research, the sample will be expanded as well as the group of questions that will be analyzed. The focus will also be on reengineering towards microservice architectures.

6. References

- [1] E. Burton Swanson (1976) The dimensions of maintenance. In Proceedings of the 2nd international conference on Software engineering (ICSE '76). IEEE Computer Society Press, Washington, DC, USA, 492–497.
- [2] S. Valenti (2002). Successful Software Reengineering. IRM PRESS. ISBN 1-931777-12-8, eISBN: 1-931777-33-0
- [3] A. Alam, T. Padenga (2010). Application Software Reengineering. PEARSON, ISBN-10: 8131731855, ISBN-13: 9788131731857
- [4] M. K. Ruhl, M. T. Gunn (1991) Software Reengineering: A Case Study and Lessons Learned. Computer Systems Laboratory National Institute of Standards and Technology Gaithersburg, MD 20899
- [5] L. Bass, P. Clements, R. Kazman (2021). Software Architecture in Practice, 4th Edition (Rough Cut). Addison-Wesley Professional, ISBN-10: 0136885977, ISBN-13: 9780136885979
- [6] M. Richards, N. Ford (2020). Software Architecture: An Engineering Approach. O'REILLY. ISBN: 978-1-492-04345-4
- [7] A. Stojkov, Z. Stojanov (2021). Review of challenges in identifying microservices from software artifacts. 11th International Conference on Applied Information and Internet Technologies - AIIT 2021 October 15th, 2021, Zrenjanin, Serbia, pp: 185-189, ISBN 978-86-7672-352-2
- [8] I. Sommerville (2000). Software Engineering 6th Edition
- [9] L. M. Given (2008). The Sage encyclopedia of qualitative research methods. Los Angeles, Calif: Sage Publications.
- [10] P. Di Francesco, P. Lago, I. Malavolta (2018) "Migrating Towards Microservice Architectures: An Industrial Survey," 2018 IEEE International Conference on Software Architecture (ICSA), 2018, pp. 29-2909, doi: 10.1109/ICSA.2018.00012.
- [11] J. Fischer, S. Bougouffa, A. Schlie, I. Schaefer, B. Vogel-Heuser (2018) "A Qualitative Study of Variability Management of Control Software for Industrial Automation Systems," 2018 IEEE International Conference on Software Maintenance and Evolution (ICSME), 2018, pp. 615-624, doi: 10.1109/ICSME.2018.00071.
- [12] J. Krüger, T. Berger (2020) An empirical analysis of the costs of clone- and platform-oriented software reuse. ESEC/FSE 2020: Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, November 2020, Pages 432–444, <https://doi.org/10.1145/3368089.3409684>
- [13] ECHA, How to determine the company size category. <https://echa.europa.eu/support/small-and-medium-sized-enterprises-smes/how-to-determine-the-company-size-category/step-5>

Принципы построения цифрового двойника для решения задач проектирования интегрированных энергетических систем

Евгений Барахтенко¹, Валерий Стенников¹, Дмитрий Соколов¹, Глеб Майоров¹

¹ Федеральное государственное бюджетное учреждение науки Институт систем энергетики им. Л.А. Мелентьева Сибирского отделения Российской академии наук, ул. Лермонтова, д. 130, Иркутск, 664033, Россия

Аннотация

Интегрированные энергетические системы обладают все возрастающим значением в экономике, энергетике, промышленности, коммунально-бытовом секторе и других сферах жизни общества. Проектирование интегрированных энергетических систем представляет собой сложную задачу, которая обусловлена высокой сложностью конфигурации этих систем, широким набором применяемого оборудования и используемого для его моделирования разнообразного набора математических моделей и специализированного программного обеспечения. Применение цифровых двойников позволит повысить эффективность проектирования, качество принимаемых проектных решений и автоматизировать трудоемкие вычислительные операции.

Ключевые слова

Цифровой двойник, интегрированная энергетическая система, онтология, программная платформа

1. Введение

Создание на базе нескольких отдельно функционирующих систем энергоснабжения (электро-, тепло/холодо-, газоснабжения и других) новой энерготехнологической конструкции в виде интегрированной энергетической системы (ИЭС) значительно расширяет их функциональные возможности, обеспечивает взаимозаменяемость энергоносителей и реализует синергетический эффект по обеспечению надежного, безопасного, экономичного и экологичного энергоснабжения [1-3]. Проектирование ИЭС представляет собой сложную задачу, которая обусловлена высокой сложностью конфигурации этих систем, широким набором применяемого оборудования и используемого для его моделирования разнообразного набора математических моделей и специализированного программного обеспечения. Применение цифровых двойников позволит повысить эффективность проектирования, качество принимаемых проектных решений и автоматизировать трудоемкие вычислительные операции.

По мнению специалистов цифровой двойник является одним из перспективных подходов для реализации интеллектуального производства и Индустрии 4.0 [4-7]. Цифровой двойник – это модель реального физического объекта, которая при этом является динамически развивающейся вместе с физическим объектом [8]. Применительно к энергетике цифровой двойник можно рассматривать как реальное отображение всех компонентов системы в жизненном цикле с использованием физических данных, виртуальных данных и данных взаимодействия между ними, то есть цифровой двойник создает виртуальный прототип реальной системы, с помощью

4th International Workshop on Information, Computation, and Control Systems for Distributed Environments (ICCS-DE 2022), July 04–08, 2022, Irkutsk, Russia

EMAIL: sva@isem.irk.ru (A. 1); barakhtenko@isem.irk.ru (A. 2); sokolov_dy@isem.irk.ru (A. 3); mayorovgs@isem.irk.ru (A. 4)
ORCID: 0000-0001-6219-0354 (A. 1); 0000-0002-6934-0025 (A. 2); 0000-0002-4068-7770 (A. 3); 0000-0002-7405-1965 (A. 4)



© 2022 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

ICCS-DE 2022 Workshop Proceedings

которого можно проводить эксперименты и проверять гипотезы, прогнозировать поведение системы и решать задачи управления ее развитием и функционированием [9, 10]. С использованием цифрового двойника можно исследовать поведение и вновь устанавливаемого оборудования еще до его установки на реальных энергообъектах, его совместимость и влияние на другие системы; моделировать самые сложные технические ситуации, в том числе отказы оборудования, изменения в выработке энергии от ВИЭ и поведении потребителей, внешнее воздействие и неправильные действия персонала [11-15 и др.]. В настоящее время отсутствуют работы, направленные на создание универсальной технологии построения цифровых двойников для решения задач проектирования ИЭС.

2. Принципы построения цифровых двойников для решения задач проектирования интегрированных энергетических систем

Цифровой двойник ИЭС представляет собой имитацию реальной физической ИЭС. Поэтому при решении задач проектирования ИЭС или развития (оптимизации) существующей ИЭС ее цифровой двойник должен отражать текущее состояние системы, а также возможные состояния в будущем в различные временные периоды, для которых определяется состав и последовательность реализации проектных решений с целью достижения требуемых ее показателей. Полученные проектные решения основаны на расчетах по поиску рационального изменения ИЭС, ее структуры и параметров с учетом заданных условий, ограничений, а также рассматриваемых сценариев. Цифровые двойники ИЭС должны обеспечить возможность отслеживания динамики развития ИЭС во времени, формирование проектных решений для различных временных периодов и рассмотрение возможных сценариев развития системы.

При решении задач проектирования ИЭС с использованием цифрового двойника необходимо учитывать тот факт, что ИЭС представляют собой сложные технические системы и при их исследовании применяются подходы на основе иерархического моделирования, что обеспечивает возможность выполнять исследования различного типа и масштаба систем, входящих в состав ИЭС. ИЭС состоят из различного типа систем энергоснабжения, которые являются подсистемами в таких системах. Каждая из подсистем содержит свой набор элементов. Эти элементы можно сгруппировать в соответствии со следующими выполняемыми энергетическими функциями: генерация, транспорт, распределение и потребление. В свою очередь, каждому элементу присущ свой набор оборудования в соответствии с выполняемыми энергетическими функциями и принадлежностью к типу системы энергоснабжения. Моделирование выполняется для всех подсистем, их набора элементов и состава оборудования, что соответствует моделированию отдельных систем энергетики. При этом существуют характерные особенности моделирования при совместном рассмотрении систем различного типа в рамках ИЭС. Эти особенности связаны с техническими и технологическими решениями по интеграции, поэтому необходимо выполнять моделирование отдельных подсистем и учитывать технические и технологические решения по интеграции с целью обеспечения координации подсистем и выполнения общесистемных целевых установок. Применение иерархического моделирования также обеспечивает возможность совместного исследования различного по масштабам систем, путем агрегирования информации по отдельным системам меньшего масштаба и ее представление для координации более крупных по масштабу систем, или наоборот, ее дезагрегирование для координации работы крупных систем с меньшими по масштабу системами.

Системы энергоснабжения определенного типа обладают характерными для этого типа свойствами, составом прикладных задач и используемым для их решения специализированным ПО. Знания о них необходимо сохранить в форме, пригодной для обработки программными системами, и использовать при решении различных задач проектирования. Для этого необходимо организовать хранение этих знаний в виде онтологий [16-18]. В процессе построения цифровых двойников и их применения используются онтологии, которые позволяют формализовано описать объекты предметной области, их свойства и взаимосвязи между этими объектами.

С учетом всего вышесказанного определены следующие принципы построения цифровых двойников для решения задач проектирования ИЭС:

- Выполнение сложных инженерных расчетов по поиску оптимальных путей преобразования ИЭС с целью повышения эффективности и надежности их функционирования.
- Моделирование ИЭС, включающей различные типы систем энергоснабжения со своими индивидуальными особенностями, с учетом иерархического принципа ее построения.
- Организация единого информационного пространства при решении задач проектирования ИЭС.
- Двухнаправленная связь между внешней средой и цифровым двойником ИЭС для отслеживания изменений и актуализации информации в цифровом двойнике и формирования проектных решений по ИЭС.
- Отслеживание динамики развития ИЭС во времени.
- Учет при решении задач проектирования ИЭС наличия множества центров принятия решения по снабжению энергией различного вида с возможностью преобразования из одного вида в другой.
- Необходимость рассмотрения большого количества элементов ИЭС со сложным поведением.
- Использование знаний о ИЭС, ее подсистемах, используемом программном обеспечении, особенностях решения задач проектирования ИЭС.

3. Парадигма проектирования интегрированных энергетических систем на базе их цифровых двойников

Предлагается новая парадигма проектирования ИЭС на базе ее цифрового двойника. В рамках предложенной парадигмы инженером создается компьютерная модель ИЭС, на основе которой в автоматизированном режиме строится цифровой двойник, результаты моделирования на котором используются для подготовки реальных проектных решений и, в конечном счете, построения реальной ИЭС (рис. 1). В свою очередь, на основе данных с характеристиками реальной ИЭС можно построить ее компьютерную модель, на основе которой будет построен цифровой двойник в автоматизированном режиме. Рассмотрим основные этапы в предлагаемой парадигме.

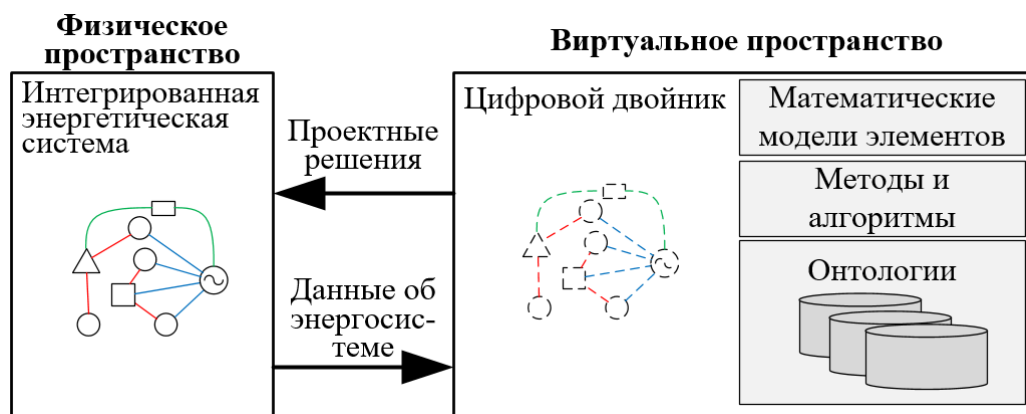


Рисунок 1: Проектирование интегрированной энергетической системы на базе цифрового двойника

Этап I. Подготовка компьютерной модели ИЭС.

Этап II. Автоматизированное интеллектуальное построение модели цифрового двойника ИЭС на базе ее компьютерной модели.

Этап III. Автоматизированное интеллектуальное построение цифрового двойника ИЭС на базе его модели.

Этап IV. Компьютерное и математическое моделирование ИЭС для получения данных, которые будут использоваться для построения вновь проектируемой ИЭС или развития (оптимизации) существующей ИЭС.

Этап V. Реализация разработанных решений.

Этап VI. Сбор данных о функционировании ИЭС и переход на этап I для уточнения компьютерной модели ИЭС.

На этапе I создается компьютерная модель ИЭС. Компьютерная модель ИЭС может быть представлена как совокупность графа, описывающего конфигурацию этой системы, и набора графических и математических моделей, описывающих свойства ее подсистем и элементов.

На этапе II в автоматизированном режиме создается модель цифрового двойника ИЭС на основе знаний, хранение которых организовано в виде системы онтологий. Эта модель включает следующие составляющие:

- Описание структур данных ИЭС и ее подсистем.
- Описание программных компонентов, реализующих математические модели и методы для решения прикладной задачи.
- Граф, описывающий ход вычислительного процесса.
- Совокупность структур данных, описывающая входные и выходные параметры.

На этапе III выполняется автоматизированное построение цифрового двойника на основе его модели с использованием знаний, хранение которых организовано в виде системы онтологий. Цифровой двойник – это способ организации пространства решения набора прикладных задач, включающий следующий набор составляющих:

- компьютерная модель ИЭС (структуры данных, описывающих конфигурацию системы и свойства ее составных элементов);
- математический аппарат (включающий математические модели и алгоритмы);
- программные компоненты (графические, системные, прикладные математические, типовые модели элементов подсистем ИЭС), интегрированные в единую программную систему;
- интерактивный графический интерфейс пользователя, включающий графическую компьютерную модель;
- хранилище данных, содержащее значение свойств системы и ее элементов в различные моменты времени;
- система обмена данными с внешней средой.

Построение цифрового двойника – это автоматизированная интеграция перечисленных его составляющих в единое информационное пространство.

На этапе IV выполняется компьютерное и математическое моделирование ИЭС с использованием мультиагентного подхода [19-21], математических моделей и алгоритмов, реализованных в виде программных компонентов. Вычислительный процесс управляется знаниями из системы онтологий и проходит в автоматизированном режиме. В результате проведенного моделирования будут получены данные, на основе которых будут подготовлены проектные решения для вновь проектируемых ИЭС или оптимизируемых существующих ИЭС.

На этапе V полученные проектные решения реализуются на практике для построения ИЭС.

На этапе VI осуществляется сбор данных о характеристиках ИЭС при функционировании этих систем в разные периоды времени. Далее происходит переход на этап I для уточнения компьютерной модели ИЭС с использованием собранных данных.

4. Реализация цифрового двойника интегрированной энергетической системы

Выполнена программная реализация элементов цифрового двойника ИЭС. В программной среде AnyLogic сформирована мультиагентная модель ИЭС. С использованием разработанной мультиагентной модели выполнен вычислительный эксперимент по развитию ИЭС. Решение задачи развития моделируемой ИЭС выполняется с использованием разработанных авторами программных компонентов. Интеграция этих программных компонентов в рамках единого

вычислительного процесса выполняется динамически на основе знаний из онтологий. В результате взаимодействия агентов и подключенных программных компонентов в мультиагентной модели было найдено оптимальное решение, в соответствии с которым сформированы и реализованы необходимые меры по строительству сетевого и генерирующего оборудования. Наглядное представление полученного решения отображено на рис. 2, задействованные в энергоснабжении участки сетей выделены соответствующим цветом. Проведенный вычислительный эксперимент показал возможность применения цифрового двойника для решения задачи развития ИЭС. В результате выполненного эксперимента на разработанной мультиагентной модели удалось сформировать оптимальную схему ИЭС для энергоснабжения потребителей с учетом системных условий и ограничений.

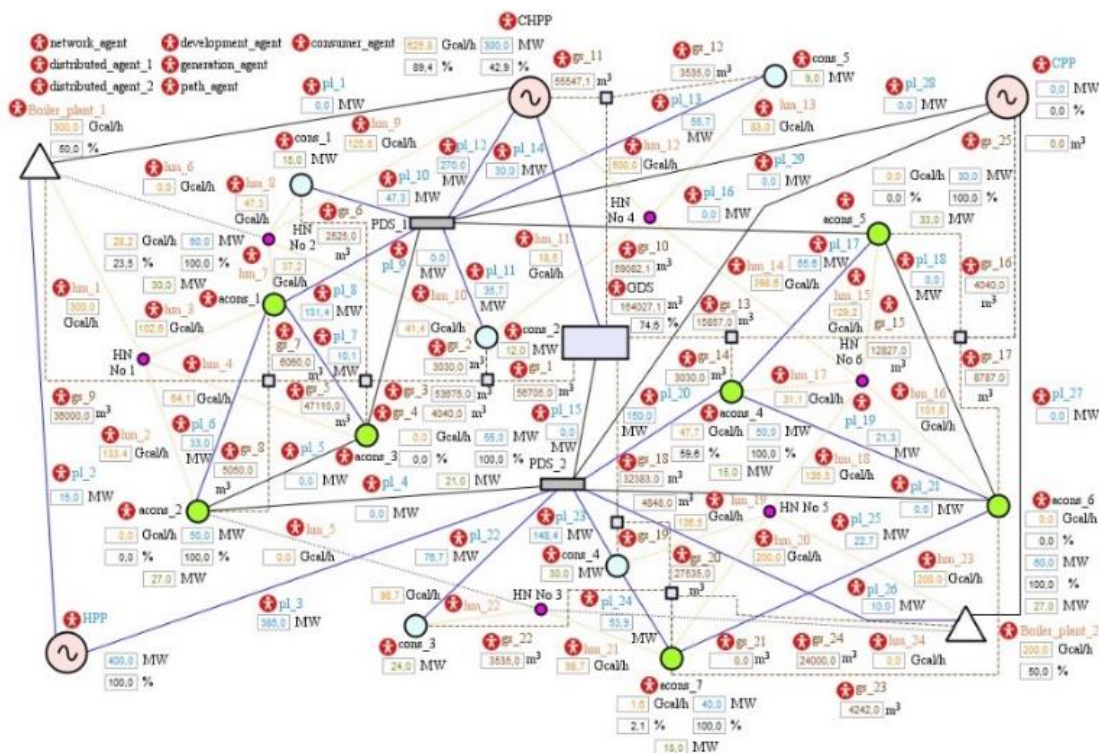


Рисунок 2: Результаты расчета интегрированной энергетической системы

5. Заключение

Интегрированные энергетические системы обладают все возрастающим значением в экономике, энергетике, промышленности, коммунально-бытовом секторе и других сферах жизни общества. В современных условиях становится актуальным оптимальное проектирование этих систем, имеющее под собой научное, инженерное и экономическое обоснование. Проектирование ИЭС представляет собой сложную задачу, которая обусловлена высокой сложностью конфигурации этих систем, широким набором применяемого оборудования и используемого для его моделирования разнообразного набора математических моделей и специализированного программного обеспечения. Применение цифровых двойников позволяет повысить эффективность проектирования, качество принимаемых проектных решений и автоматизировать трудоемкие вычислительные операции. В статье рассмотрены принципы построения цифровых двойников для решения задач проектирования ИЭС. Предложена парадигма проектирования ИЭС на базе ее цифрового двойника. Представлены результаты вычислительного эксперимента, полученные с использованием выполненной программной реализации элементов цифрового двойника ИЭС для тестовой схемы энергоснабжения.

6. Источник финансирования исследования

Исследования выполнены в Институте систем энергетики им. Л.А. Мелентьева Сибирского отделения Российской академии наук при поддержке Российского научного фонда (проект № 22-29-01611).

7. Список литературы

- [1] Н. И. Воропай, В. А. Стенников, Интегрированные интеллектуальные энергетические системы, Изв. РАН. Энергетика 1 (2014) 64–73.
- [2] N. I. Voropai, V. A. Stennikov, E. A. Barakhtenko, Integrated energy systems: challenges, trends, philosophy, Stud. Russ. Econ. Dev. 28(5) (2017) 492–499.
- [3] N. I. Voropai, V. A. Stennikov, E. A. Barakhtenko, Methodological principles of constructing the integrated energy supply systems and their technological architecture, J. Phys. Conf. Ser. 1111 (2018) 012001. doi:10.1007/s10845-019-01512-w.
- [4] K. Y. H. Lim, P. Zheng, C. Chen, A state-of-the-art survey of Digital Twin: techniques, engineering product lifecycle management and business innovation perspectives, J. of Intelligent Manufacturing 31 (2020) 1313–1337.
- [5] A. M. M. Sharif Ullah, Modeling and simulation of complex manufacturing phenomena using sensor signals from the perspective of Industry 4.0, Advanced Engineering Informatics 39 (2019) 1–13. doi:10.1016/j.aei.2018.11.003.
- [6] P. Pileggi, J. Verriet, J. Broekhuijsen, C. van Leeuwen, W. Wijbrandi and M. Konsman, A Digital Twin for Cyber-Physical Energy Systems, in: Proceedings of the 7th Workshop on Modeling and Simulation of Cyber-Physical Energy Systems, MSCPES 2019, IEEE, 2019, pp. 1–6. doi:10.1109/MSCPES.2019.8738792.
- [7] X. Xu, Machine Tool 4.0 for the New Era of Manufacturing, The International Journal of Advanced Manufacturing Technology 92 (2017) 1893–1900.
- [8] В. А. Стенников, Цифровые двойники в нефтегазовой отрасли – драйвер отечественной науки, Текстовый вариант интервью Брилеву С.Б. URL: <https://globalenergyprize.org/ru/2021/05/14/valerij-stennikov-cifrovye-dvojniki-v-neftegazovoj-otrasli-drajver-otechestvennoj-nauki/>.
- [9] F. Tao, F. Sui, A. Liu, Q. Qi, M. Zhang, B. Song, Z. Guo, S. C.-Y. Lu, A. Y. C. Nee, Digital twin-driven product design framework, International Journal of Production Research 57 (2019) 3935–3953. doi: 10.1080/00207543.2018.1443229.
- [10] Е. Никитина, Попали в сети: как работают цифровые двойники в электроэнергетике, URL: <https://pro.rbc.ru/news/5db1b59a9a79474bb142a3fe>.
- [11] С. А. Ерошенко, А. И. Хальясмаа, Технологии цифровых двойников в энергетике, статья в сборнике трудов конференции «Электроэнергетика глазами молодежи-2019», Изд-во ИРНИТУ, 2019, с. 55–58
- [12] N. Karanjkar, A. Joglekar, S. Mohanty, V. Prabhu, D. Raghunath, R. Sundaresan, Digital Twin for Energy Optimization in an SMT-PCB Assembly Line, in: Proceedings of the 2018 IEEE International Conference on Internet of Things and Intelligence System, IOTAIS, IEEE, 2018, pp. 85–89.
- [13] Программно-технический комплекс «Виртуально-цифровая АЭС с ВВЭР», URL: <http://www.ibrae.ac.ru/contents/362>.
- [14] С. К. Андрияшечкин, С. П. Ковалев, Е. И. Нефедов, Подходы к разработке и применению цифровых двойников энергетических систем, Цифровая подстанция 12 (2019) 38–43.
- [15] С. П. Ковалев, Проектирование информационного обеспечения цифровых двойников энергетических систем, Системы и средства информатики, 30 (2020) 66–81.
- [16] T. R. Gruber, A translation approach to portable ontology specifications, Knowledge Acquisition 5 (1993) 199–220.
- [17] S. Staab, R. Studer, Handbook on Ontologies, 2nd ed., Springer-Verlag, Heidelberg, 2009.
- [18] S. Staab, T. Walter, G. Gröner, F. S. Parreiras, Model driven engineering with ontology technologies, Reasoning Web. Semantic Tech. for Soft. Engin. Lect. Notes Comput. Sci. (2010) 62–98.

- [19] Y. Ren, D. Fan, Q. Feng, Z. Wang, B. Sun, D. Yang, Agent-based restoration approach for reliability with load balancing on smart grids, *Appl. Energy* 249 (2019) 46–57.
- [20] V. Stennikov, E. Barakhtenko, G. Mayorov, D. Sokolov, B. Zhou, Coordinated management of centralized and distributed generation in an integrated energy system using a multi-agent approach *Appl. Energy* 309 (2022) 118487.
- [21] Z. Zhang, R. Jing, J. Lin, X. Wang, K. H. van Dam, M. Wang, C. Meng, S. Xie, Y. Zhao, Combining agent-based residential demand modeling with design optimization for integrated energy systems planning and operation, *Appl. Energy* 263 (2020) 114623.

РИСК-ОРИЕНТИРОВАННЫЙ ПОДХОД К ПОИСКУ КРИТИЧЕСКИХ ЭЛЕМЕНТОВ СИСТЕМЫ ЭНЕРГЕТИКИ

Н.М. Береснева¹, А.В. Еделев¹

¹ Институт систем энергетики им. Л.А. Мелентьева СО РАН, Лермонтова 130, Иркутск, 664033, Россия

Abstract

Приводится описание риск-ориентированного подхода к поиску наиболее важных элементов системы энергетики. Определение критичности элементов строится на анализе рисков недопоставок энергоресурсов потребителям различных категорий. Случайные отказы элементов систем энергетики моделируются с помощью библиотеки PARMONC.

Keywords

Система энергетики, живучесть, уязвимость, критические элементы.

1. Введение

Каждая система энергетики, включающая в себя подсистему передачи (транспортирования) и распределения энергоресурсов, относится к ключевым критическим инфраструктурам, нарушение функционирования которых отрицательно влияет на государство, экономику и общество [1].

Живучесть системы энергетики определяет её способность функционировать в экстремальных ситуациях, таких как стихийные бедствия и техногенные катастрофы [2].

Одной из составляющих живучести является уязвимость, которая имеет две взаимодополняющие интерпретации [3]. В глобальном контексте уязвимость отражает "пассивную" реакцию системы (падение производительности) на возникновение экстремальных ситуаций. В локальном контексте уязвимость характеризует восприимчивость компонент систем к отдельным видам крупных возмущений. Компоненты, отказ которых приводит к масштабным негативным последствиям для потребителей энергоресурсов, называются критическими.

На самом верхнем уровне иерархии моделей систем энергетики [3] критические компоненты представляют собой сложные технические территориально-распределённые объекты. К таким критически важным объектам (КВО) относятся, например, крупные электрические станции или узловые компрессорные станции Единой системы газоснабжения (ЕСГ) России [4]. На нижележащих уровнях иерархии моделей системы энергетики детально рассматриваются составляющие объектов энергетической инфраструктуры, например, конкретное энергетическое оборудование, имеющее точечную пространственную привязку. Чтобы уйти от зависимости к рассматриваемому территориальному уровню далее объекты энергетической инфраструктуры, либо их составляющие будут называться критическими элементами.

Поиск критических элементов [5] строится на базе оценки реализации сценариев крупных возмущений, полученных в результате проведения многовариантных вычислительных экспериментов [6]. Ниже представлен риск-ориентированный подход к нахождению критических элементов систем энергетики на методологическом и практическом уровнях.

4th International Workshop on Information, Computation, and Control Systems for Distributed Environments (ICCS-DE 2022), July 04–08, 2022, Irkutsk, Russia

EMAIL: beresneva@isem.irk.ru (A. 1); flower@isem.irk.ru (A. 2)

ORCID: 0000-0002-0003-947X (A. 1); 0000-0003-2219-9754 (A. 2)



2022 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

ICCS-DE 2022 Workshop Proceedings

2. Риск-ориентированная методика определения критических элементов систем энергетики

Предлагаемая в данной статье методика анализирует последствия совместных целенаправленных и случайных отключений элементов. Под целенаправленные отключения в этом случае попадают группы предварительно выбранных отраслевых объектов, из множества которых впоследствии выбираются критические элементы [7]. Случайные отказы моделируются для оставшихся элементов систем. Таким образом достигается более реалистичное моделирование поведения систем энергетики в экстремальных условиях. Определение критических элементов в данном случае строится на анализе рисков недопоставки энергоресурсов различным категориям потребителей [8]. В совокупности эти возможности методики формируют риск-ориентированный характер задания сценариев экстремальных ситуаций, риск-ориентированный выбор на базе этих сценариев критических элементов.

Обобщенная схема рассматриваемой методики представлена на рис. 1. Она включает в себя следующие основные этапы:

- определение параметров сценариев возмущений;
- проведение сценарных расчетов;
- обработка результатов сценарных расчетов;
- определение перечня критических элементов.

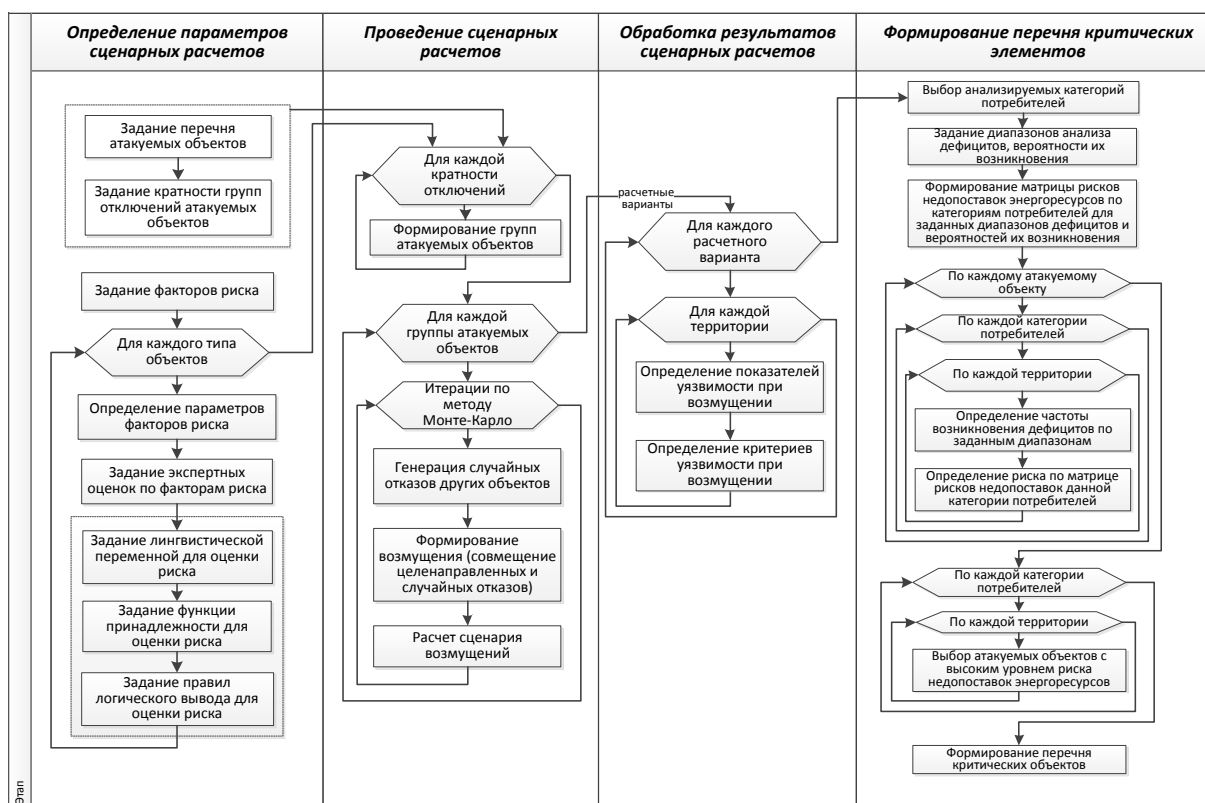


Рисунок 1: Этапы риск-ориентированной методики поиска критических элементов

В этой схеме на первом этапе в случае отсутствия достоверной статистической информации об авариях на объектах энергетики используются экспертные знания. Аппарат нечеткой логики в этом случае преобразует экспертные оценки таких факторов риска, как характер и интенсивность проявления экстремальных ситуаций, степень их влияния на объект энергетики, в вероятность случайного отключения данного объекта.

На втором этапе на каждой из N итераций цикла Монте-Карло реализуется один сценарий возмущения, который представляет собой комбинацию целенаправленной атаки на заданные элементы системы энергетики и случайные отказы прочих элементов.

Целевое предназначение этих двух этапов – получение расчетных данных для оценки последствий крупных возмущений на базе многовариантных вычислительных экспериментов. Поэтому данные этапы можно считать вспомогательными для последующей комплексной оценки критичности отраслевых объектов на третьем и четвертом шаге рассматриваемой схемы.

На третьем этапе риск-ориентированной методики поиска критических элементов формируются и реализуются цепочки преобразований «результаты сценарных расчетов - показатели уязвимости - критерии уязвимости» [6]. Показатели уязвимости в этой цепочке определяются для каждого сценария возмущения, характеризуя с необходимых точек зрения последствия реализации заданных возмущений. Критерии уязвимости являются обобщенной оценкой показателей на множестве сценариев случайных отключений для каждого атакуемого объекта. При анализе недопоставок энергоресурсов показатели уязвимости описывают в относительной форме ущерба потребителей от совместного воздействия целенаправленной атаки и случайных отказов прочих элементов на итерации $l = \overline{1 \dots N}$ цикла Монте-Карло:

$$z_l^k = 1 - \frac{y_l^k}{R_l^k}, \quad (1)$$

где z_l^k – дефицит энергоресурса k , R_l^k – потребность в энергоресурсе k , y_l^k – поставка энергоресурса k . Параметры R_l^k и y_l^k являются результатами сценарных расчетов на итерации l цикла Монте-Карло [4].

Критичность атакуемых элементов определяется на четвертом этапе представленной методики с помощью матрицы риска A , m столбцов которой соответствуют диапазонам ущерба от недопоставки какого-либо энергоресурса, а n строк – диапазонам вероятности этих недопоставок. Значением элемента a_{ij} матрицы риска A является взвешенное произведение значений оценок диапазонов ущерба $i = \overline{1 \dots m}$ и вероятности недопоставки $j = \overline{1 \dots n}$. Матрицы риска в методике могут быть универсальными, либо формироваться отдельно для разных категорий потребителей какого-либо энергоресурса. Ущерб от недопоставок k -го энергоресурса, вероятности этих недопоставок – ранее определяемые на множестве сценариев возмущений критерии уязвимости.

Если область значений ущерба от недопоставки какого-либо энергоресурса k разбивается на m равных диапазонов длиной ΔZ^k , то вероятность недопоставки энергоресурса k в диапазоне дефицита с номером $i = \overline{1 \dots m}$ рассчитывается следующим образом:

$$p_i^k = \frac{\sum_{l=1}^N F_i(z_l^k)}{N}, \quad (2)$$

где F_i – функция, которая равна 1 при выполнении условия $(i-1)\Delta Z^k < z_l^k \leq i\Delta Z^k$ и 0 в других случаях.

Если область значений вероятности недопоставки энергоресурса k разбита на n равных частей, то при попадании значения p_i^k в диапазон $j = \overline{1 \dots n}$ риск недопоставки энергоресурса k – будет равен значению элемента a_{ij} матрицы риска A . Величина риска по более, чем одному диапазону ущербов или вероятности недопоставки определяется как сумма значений соответствующих элементов a_{ij} .

Логика идентификации рисков (на количественном и качественном уровнях) с помощью матрицы A на четвертом этапе методики поиска критических элементов формируется исходя из целей проводимого анализа с учетом критичности анализируемых диапазонов, их численных порогов. Комплексная оценка критичности атакуемых элементов для группы потребителей определяется взвешенной сверткой балльных оценок рисков отдельных потребителей.

3. Оценка критичности объектов Единой системы газоснабжения России

Апробация представленной риск-ориентированной методики поиска критических элементов выполнена на примере ЕСГ России. Практическая значимость данной апробации – оценка влияния учета вероятностей недопоставок газа по системе в целом при выборе критических элементов на конечный перечень газовых объектов. Данная оценка проводилась по ранее сформированному для единичных отключений атакуемых объектов списку из 61 элемента (компрессорные станции, подземные хранилища газа, источники, дуги газовой сети) [9].

Принципиальными отличиями текущих исследований стали моделирование случайных отказов, выбор критических объектов с позиций рисков недопоставок. Допустимые вероятности отказов компонент транспортной составляющей Единой системы газоснабжения России и подземных хранилищ газа, расположенные на её территории, задавались исходя из данных, приведённых в табл. 1 [10, 11].

Таблица 1

Интенсивность аварий объектов газовой отрасли

Тип объекта	Интенсивность аварий
Компрессорная станция, аварий/год	0,25
Подземное хранилище, аварий/год	0,1
Источник, аварий/год	0,1
Участок магистрального газопровода, аварий/км год	$1,3 \cdot 10^{-5}$

Матрица рисков в исследовании формировалась в диапазоне максимального объема недопоставок (в пределах 30% относительного дефицита), исходя из соображений кризисности дефицитов, переступивших 60% порог от полученного максимального дефицита. Вероятности недопоставок оценивались десятизначной интервальной шкалой.

Количественная и качественная оценка рисков в матрице рисков (табл. 2) формировалась с учетом кризисности диапазонов недопоставок газа, вероятности их возникновения. В обоих случаях была введена единая качественная шкала, включающая низкий, средний и высокий уровни кризисности с уникальными пороговыми значениями анализируемых характеристик. Логика идентификации уровней риска, заданных в этой же шкале кризисности, была выстроена исходя из следующих утверждений:

Таблица 2

Матрица рисков недопоставок газа в текущей апробации

Вероятность недопоставок газа	Диапазоны недопоставок газа, %						
	0 – 5	5 - 10	10 - 15	15 - 20	20 - 25	25 - 30	30 - 35
0,9-1	10	20	45	60	75	90	105
0,8-0,9	9	18	40,5	54	67,5	81	94,5
0,7-0,8	8	16	36	48	60	72	84
0,6-0,7	7	14	31,5	42	52,5	63	73,5
0,5-0,6	6	12	18	36	45	54	63
0,4-0,5	2,5	10	15	30	37,5	45	52,5
0,3-0,4	2	8	12	24	30	36	42
0,2-0,3	1,5	6	9	12	15	27	31,5
0,1-0,2	1	2	3	8	10	12	14
0-0,1	0,5	1	1,5	4	5	6	7

- ключевым показателем оценки уровня риска является уровень недопоставок; чем выше уровень недопоставки, тем более критичен риск, не зависимо от вероятности возникновения дефицита;

- вероятность возникновения недопоставок корректирует принадлежность риска к определенному качественному уровню; в случае минимальных значений вероятности критичность рисков недопоставок считается незначительной;
- недопустимость пересечений балльных оценок принятых уровней рисков недопоставок в сформированной матрице, их максимальность для кризисной ситуации.

Анализ полученных в проведенных исследованиях результатов показал следующее.

Наблюдается схожесть сформированного перечня объектов со списком КВО ЕСГ России, полученным в [9]. В полученном перечне в числе первых 61 элементов присутствует 58% исходных объектов из перечня КВО. При этом наблюдается 97% соответствие на первой тридцатке объектов, полное соответствие состава по первым семи элементам. В последнем случае отмечается близость балльных оценок рисков недопоставок, как следствие несколько иная ранжированность элементов.

Анализ категорирования кризисности объектов по заданной матрице рисков дал схожую с перечнем газовых КВО картину кризисности, показал возможность попадания одних и тех же объектов в смежные категории. Категорированность наглядно показала зависимость получаемых результатов от логики формирования матрицы рисков, от правил балльной оценки рисков на рассматриваемых численных интервалах, особенно в части регламентации вероятностей недопоставок. В данном исследовании при десятизонной шкале вероятностей ущербов с допущением о незначительности любых недопоставок с минимальной вероятностью (в пределах 0,2) сложилась следующая картина (табл. 3). Наиболее кризисными стали объекты 1-7, 9-10, 14-16 из перечня КВО ЕСГ России с преимущественным уровнем недопоставок в районе 17,3-17,7%, с высоким уровнем вероятности (0,8-1) их возникновения. Пять объектов данного множества одновременно попали в соседнюю, менее кризисную категорию с более высокими оценками дефицитов (19,7-22%) с минимальным уровнем вероятности (0-0,2). Суммарно в среднюю категорию кризисности вошло 11 газовых КВО (объекты 1-2, 4-5, 8, 10, 12, 13, 15) с высокими относительными дефицитами (17,8-23%) и с низкими величинами вероятности (в пределах 0,1).

Таблица 3

Анализ выбранных объектов ЕСГ России по наиболее кризисным уровням риска

Уровень кризисности объектов	Число объектов	% от газовых КВО (61 объект)	% от общего числа (57 объектов)	Повторяемость объектов по группам	Номера газовых КВО	Диапазон создаваемых дефицитов, %
Высокий	11	18 %	31 %	6 из 11 (54%)	1-7, 9-10, 14-16	15.97 % - 21.18 %
Средний	29	48 %	83 %	6 из 29 (21%)	1-2, 4-5, 8, 10, 12, 13, 15	15.97 %- 21.18 %

4. Заключение

Заявленная методика выбора критичности элементов позволяет более реалистично анализировать функционирование систем энергетики в экстремальных ситуациях, позволяет учитывать потенциальные риски недопоставок энергоресурсов различным категориям потребителей. Преимуществом методики является ее универсальность в отношении СЭ различных уровней территориальной иерархии. Текущая апробация методики на примере ЕСГ России выявила следующие ключевые особенности:

- показала существенное влияние вероятностного фактора, зависящего от количества циклов Монте-Карло, на выбор критических объектов;
- показала важность корректного задания матриц рисков недопоставок энергоресурсов, важность учета фактических объемов недопоставок, повторяемости вероятностей возникновения этих недопоставок при выборе численных диапазонов матриц;
- доказала правдоподобность формируемых с помощью методики результирующих перечней критических объектов.

Дальнейшая апробация методики планируется на уровне ТЭК страны, где существенную роль играет фактор межсистемного взаимодействия энергетических отраслей.

5. Благодарности

Работа выполнена при финансовой поддержке Министерства науки и высшего образования Российской Федерации в рамках проекта государственного задания FWEU-2021-0003 № АААА-А21-121012090014-5 программы фундаментальных исследований РФ на 2021-2030 гг. с использованием ресурсов ЦКП "Высокотемпературный контур" (Минобрнауки России, проект № 13.ЦКП.21.0038) и ЦКП Сибирский Суперкомпьютерный Центр ИВМиМГ СО РАН, а также РФФИ в рамках научных проектов № 20-47-380002-р_а и № 20-08-00367.

Авторы выражают благодарность Иркутскому суперкомпьютерному центру СО РАН за предоставление доступа к высокопроизводительному кластеру «Академик В.М. Матросов» (Иркутский суперкомпьютерный центр СО РАН, Иркутск: ИДСТУ СО РАН; <http://hpc.icc.ru>, дата обращения 18.04.2022).

6. Литература

- [1] Воропай Н.И., Направления и проблемы трансформации электроэнергетических систем, *Электричество*, 7 (2020): 12-21.
- [2] Wan C., Yang Z., Zhang D., Yan X., Resilience in transportation systems: a systematic review and future directions, *Transport reviews*, 38.4 (2018): 479-498.
- [3] Н.И. Воропай, А. В. Стенников, Иерархическое моделирование систем энергетики, Академическое изд-во «Гео», Новосибирск, 2020.
- [4] С.В. Воробьев, Еделев, А.В., Применение метода определения критических элементов в сетях технических инфраструктур для поиска критически важных объектов газотранспортной сети России, *"Энергетическая политика"*, 1 (2018): 45-51.
- [5] Jonsson H., Johansson J., Johansson H., Identifying critical components in technical infrastructure networks, *Proceedings of the Institution of Mechanical Engineers. Part O: Journal of Risk and Reliability*. 2008. Vol. 222. No. 2. P. 235-243.
- [6] Еделев А.В., Береснева Н.М., Горский С.А., Анализ уязвимости энергетической инфраструктуры и его реализация, *Современные наукоемкие технологии*, 1 (2022): 47-52.
- [7] Edelev, A. V., Marchenko, M. A., Basharina, O. Y., Identification of the energy system critical elements using the PARMONC library, *Journal of Physics: Conference Series*, 1715. 1 (2021): 012064.
- [8] Marino A., Zio E., A framework for the resilience analysis of complex natural gas pipeline networks from a cyber-physical system perspective., *Computers & Industrial Engineering*, 162 (2021): 107727
- [9] Еделев А.В., Сендеров С.М., Сидоров И.А., Применение распределенных вычислений для выявления критически важных объектов газотранспортной сети России, *Информационные и математические технологии в науке и управлении*, 1 (2016): 55–62.
- [10] Praks Pavel, Vytis Kopustinskas, Marcelo Masera, Probabilistic modelling of security of supply in gas networks and evaluation of new infrastructure, *Reliability Engineering & System Safety*, 144 (2015): 254-264.

- [11] Гостинин И.А., Вирясов А.Н., Семенова М.А., Анализ аварийных ситуаций на линейной части магистральных газопроводов, Инженерный вестник Дона, 2(2013), URL: ivdon.ru/ru/magazine/archive/n2y2013/1618.

Разработка программной платформы для создания цифровых двойников интегрированных энергетических систем

Дмитрий Соколов¹, Валерий Стенников¹, Евгений Барахтенко¹, Глеб Майоров¹

¹ Институт систем энергетики им. Л.А. Мелентьева СО РАН (ИСЭМ СО РАН), Лермонтова, 130, Иркутск, 664033, Россия

Аннотация

Интегрированные энергетические системы (ИЭС) являются сложными техническими объектами, технологии цифровых двойников позволяют повысить качество их моделирования и проектирования. В статье предложен методический подход к разработке программной платформы для создания цифровых двойников ИЭС и решения комплекса задач, возникающих при проектировании этих систем. Для автоматизации этапов построения программной платформы реализуется концепция Model-Driven Engineering. Предложена трехуровневая архитектура сетевой многопользовательской программной системы. Предложен подход к автоматизированному построению цифрового двойника, представлен алгоритм, в соответствии с которым выполняется автоматизированное построение.

Ключевые слова

Интегрированная энергетическая система, программная платформа, автоматизация программирования, программная инженерия, применение онтологий

1. Введение

В последнее время развивается комплексный подход, рассматривающий совместное развитие и функционирование нескольких отдельных систем (электро-, тепло-, газо-, хладоснабжения и др.) в составе интегрированных энергетических систем (ИЭС) [1, 2]. При этом под ИЭС понимается система, включающая совокупность индивидуальных систем энергетики, взаимосвязанных посредством общих взаимозависимых режимов функционирования при производстве, потреблении и транспортировке соответствующих видов энергии, а также комплексном развитии этих систем.

Объединение индивидуальных систем энергетики в единый технологический комплекс обеспечит реализацию новых функциональных возможностей, применение более совершенных технологий в эксплуатации и создание интегрированных централизованно-распределенных систем с координированным управлением их режимами и активным участием потребителей в процессе энергоснабжения. Переход к построению ИЭС обеспечивает более высокую экономическую эффективность перечисленных систем, высокую надежность и бесперебойность энергоснабжения потребителей.

Проблема оптимального проектирования ИЭС является очень важной, а ее эффективное решение обеспечивает значительное снижение экономических затрат и приемлемое качество снабжения потребителя необходимой ему энергией [1, 3, 4]. Применение современных подходов на основе концепции цифровых двойников позволяет повысить адекватность

4th International Workshop on Information, Computation, and Control Systems for Distributed Environments (ICCS-DE 2022), July 04–08, 2022, Irkutsk, Russia

EMAIL: sokolov_dy@isem.irk.ru (A. 1); sva@isem.irk.ru (A. 2); barakhtenko@isem.irk.ru (A. 3); mayorovgs@isem.irk.ru (A. 4)

ORCID: 0000-0002-4068-7770 (A. 1); 0000-0001-6219-0354 (A. 2); 0000-0002-6934-0025 (A. 3); 0000-0002-7405-1965 (A. 4)



© 2022 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

ICCS-DE 2022 Workshop Proceedings

математических и компьютерных моделей, обеспечивает возможность решения проектных задач в рамках единого программного обеспечения, что позволяет в результате получить решение проектной задачи более высокого качества в сравнении с подходами предыдущих поколений. В настоящее время подходы к проектированию сложных технических систем на основе концепции цифровых двойников [5] получают широкое распространение, как в энергетике [6], так и в других отраслях [7-9].

В ИСЭМ СО РАН сформулирована новая научная задача разработки единой и универсальной технологии построения цифровых двойников ИЭС и разработки программного инструмента для проектирования этих систем. В настоящей статье представлен оригинальный методический подход к построению программной платформы для создания цифровых двойников ИЭС и решения задач их проектирования.

2. Формулировка научной проблемы и подход к ее решению

На основе выполненной структуризации проблематики и исследований ИЭС сделан вывод о необходимости разработки единой программной платформы для моделирования и проектирования ИЭС. Платформа должна выступать в качестве базы для автоматизированного построения цифровых двойников ИЭС и решения всего комплекса задач компьютерного моделирования ИЭС при их проектировании.

Новая платформа должна отвечать следующим требованиям:

- Обеспечивать решение всех подзадач в рамках единой программной системы.
- Связать решение отдельных подзадач в единое решение.
- Позволять гибко настраивать вычислительную процедуру под особенности оборудования и математические модели, описывающие его работу.
- Гибко настраивать вычислительный процесс, реализуя различные алгоритмы решения задачи.

Элементы платформы должны быть построены в соответствии с парадигмой компонентного программирования, что обеспечит возможность представления методов, алгоритмов и моделей в виде элементов программного фонда, пригодных для многократного использования. Программные компоненты должны обладать стандартизированными в рамках единого программного инструмента программными интерфейсами с описанием подпрограмм, форматов входных и выходных данных. В свою очередь практическое применение компонентного подхода приводит к возникновению сложной проблемы, вызванной необходимостью интеграции в программную систему большого количества программных компонентов, используемых для моделирования различных элементов систем, которые могут быть представлены широким набором оборудования и находиться в различных состояниях. Высокая трудность поддержания в актуальном состоянии всего набора необходимых программных компонентов позволяет сделать вывод, что для реализации программной платформы необходимо привлечение парадигм программирования, ориентированных на автоматизацию этапов построения программного обеспечения.

Современные подходы к разработке программного обеспечения являются предметом исследования многих специалистов и представлены в литературных источниках достаточно широко. Основой моделирования современного программного обеспечения является объектно-ориентированный подход [10, 11]. В работах [12-15] для автоматизации этапов построения программного обеспечения предлагается концепция Model-Driven Engineering (MDE). Концепция MDE представляет собой совокупность методических подходов к автоматизированному построению сложных программных систем на основе предварительно разработанных моделей. Эта концепция разработки программного обеспечения в настоящее время активно развивается, подходы на ее основе позволяют успешно разрабатывать сложные программные системы [16-18]. Подходы к автоматизации этапов построения программного обеспечения, основанные на применении метапрограммирования, изложены в работах [19, 20].

3. Новый методический подход

В ИСЭМ СО РАН разрабатывается оригинальный методологический подход к построению программной платформы для создания цифровых двойников ИЭС и решения практических задач, возникающих при проектировании этих систем энергетики. Концепция MDE является основой автоматизированного построения программного обеспечения на базе этой платформы. Особенность реализации этой концепции в рамках разработанного подхода состоит в широком использовании современных технологий метапрограммирования [19, 20], которые обеспечивают высокую гибкость для динамического конструирования элементов программной системы и внесения в нее необходимых изменений.

Основные составляющие разрабатываемого методического подхода:

1. Состав программных компонентов и их стандартизированные программные интерфейсы.
2. Состав и методы построения онтологий.
3. Трехуровневая архитектура многопользовательской программной системы.
4. Оригинальный подход к автоматизированному построению цифрового двойника ИЭС.

Основными составляющими разрабатываемой программной платформы являются следующие: компоненты, управляющие работой других программных компонентов; компоненты, реализующие математические методы и алгоритмы; компоненты для доступа к базам данных, онтологии (онтология ИЭС, включающая в себя онтологии индивидуальных систем энергетики, онтология программного обеспечения (онтология ПО), онтология задач). Java является основным языком программирования для реализации платформы. Отдельные программные компоненты могут быть реализованы на C/C++, Fortran, Python и других языках. Для хранения онтологий используется предметно-ориентированный язык, разработанный на основе языка описания данных XML (eXtensible Markup Language).

Предложенная для реализации программной платформы трехуровневая архитектура сетевой многопользовательской системы представлена на рис. 1. Приведем описание каждого из ее уровней.

1. Web-приложение, работающее на сервере. Эта подсистема отвечает за взаимодействие с пользователями в рамках компьютерной сети и для решения расчетно-аналитических задач на стороне сервера.
2. Клиентская часть сетевой системы. Она предназначена для работы с графическими компьютерными моделями ИЭС и данными по их элементам. Эта подсистема используется для подготовки исходных данных и просмотра результатов вычислений.
3. Подсистема хранения данных. Она предназначена для хранения и многократного использования компьютерной модели ИЭС, исходных данных и результатов расчетов, графических данных о городской застройке.

Серверная часть программной системы разрабатывается на языке Java и работает под управлением контейнера сервлетов Apache Tomcat. Для хранения модели ИЭС и передачи ее конфигурации между подсистемами программной платформы используется текстовый формат обмена данными JSON (JavaScript Object Notation). Созданные пользователями компьютерные модели ИЭС сохраняются платформой в базу данных и могут быть многократно использованы при проведении расчетов. Для взаимодействия с базами данных используется технология JDBC (Java DataBase Connectivity), обеспечивающая единый интерфейс взаимодействия платформы с системами управления базами данных (СУБД) при помощи языка SQL (Structured Query Language). В зависимости от масштабов решаемых задач, количества одновременных пользователей и особенностей сетевой нагрузки на web-приложение JDBC обеспечивает возможность гибкой интеграции в платформу различных многопользовательских реляционных СУБД, таких как Firebird, PostgreSQL, MS SQL Server и др.

Взаимодействие пользователя программной платформы с компьютерной моделью ИЭС организовано как работа с набором слоев, каждый из которых соответствует одному из уровней иерархической модели ИЭС. При этом, в зависимости от целей моделирования и особенностей решаемой задачи, один слой может представлять как отдельную энергетическую систему (в целом) в составе ИЭС, так и один из уровней иерархической модели какой-либо

подсистемы ИЭС. Иерархическая модель ИЭС может состоять из моделей отдельных энергетических систем разных масштабов (межрегиональные сети, крупные городские магистральные сети, районные распределительные сети, сети отдельных населенных пунктов и т.д.). Так, например, ИЭС небольшого населенного пункта может быть подключена к крупной газотранспортной системе и городской системе электроснабжения, при этом теплоснабжающая система в составе этой ИЭС может быть небольшого масштаба и работать на газе или угле.

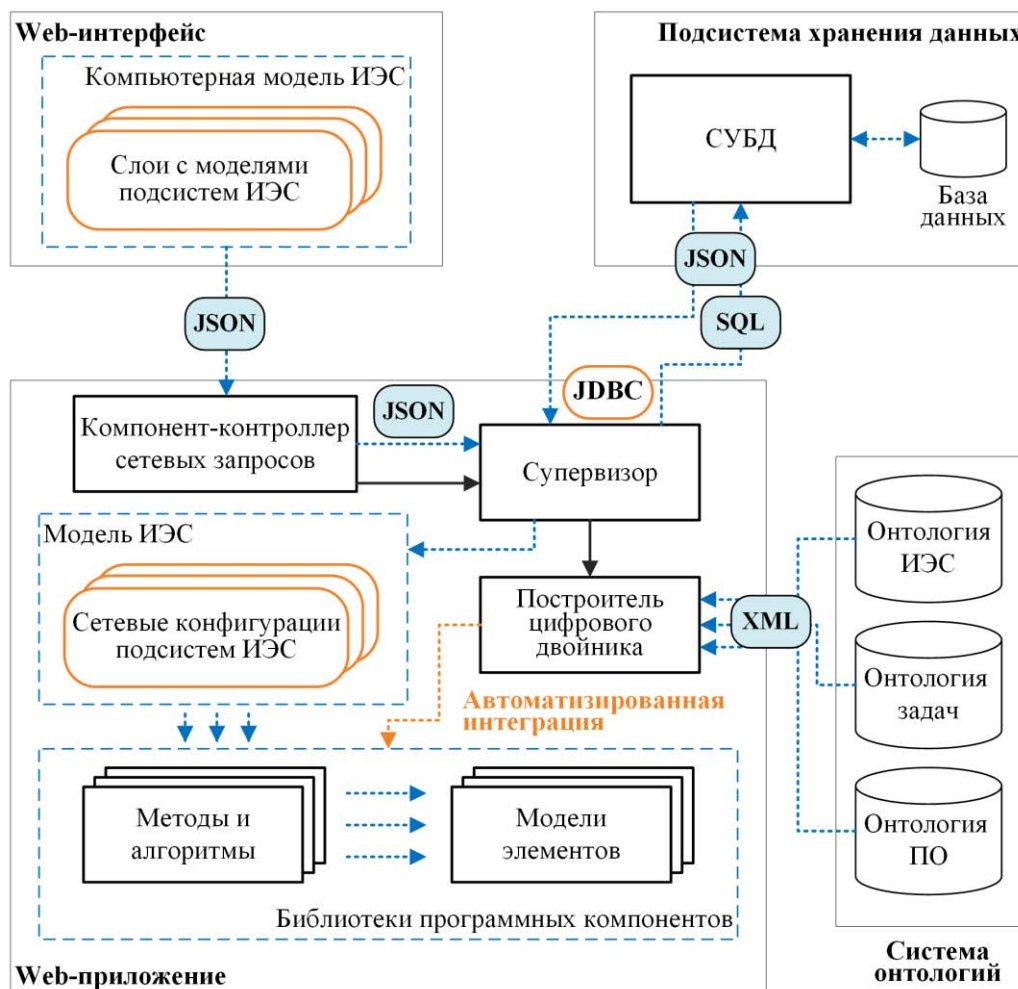


Рисунок 1: Многопользовательская сетевая программная система. Трехуровневая архитектура

Авторами был разработан оригинальный подход к построению цифрового двойника и организации вычислительного процесса. В соответствии с этим подходом конструирование цифрового двойника и вычислительной подсистемы выполняется в автоматизированном режиме. Подход предполагает, что автоматизированное конструирование выполняется в соответствии с алгоритмом, шаги которого представляются в следующем виде.

1. Сформировать список используемого оборудования на основе созданной пользователем компьютерной модели ИЭС.
2. Сформировать список математических моделей используемого оборудования на основе компьютерной модели ИЭС.
3. Сформировать список необходимых программных компонентов на основе созданных на предыдущих шагах списков.
4. Сформировать описание программного компонента, реализующего алгоритм решения прикладной проблемы.
5. Построить модель конструируемой подсистемы программной системы.
6. Загрузить описание необходимых программных компонентов из онтологии ПО.

7. Подключить к программной системе программные компоненты, реализующие алгоритмы и математические модели оборудования.
8. Сформировать структуры данных, необходимые для функционирования подключенных программных компонентов.

4. Выводы

Переход к моделированию и проектированию современных систем энергетики как ИЭС позволяет значительно повысить эффективность их работы. ИЭС являются сложными техническими объектами, требующими структуризации больших объемов данных, применения сложных математических моделей и методов, организации сложных вычислительных процессов. Технологии цифровых двойников позволяют преодолеть перечисленные трудности. В статье предложен методический подход к разработке программной платформы для создания цифровых двойников ИЭС и решения комплекса задач, возникающих при проектировании этих систем. Для автоматизации этапов построения программной платформы реализуется концепция MDE. Предложена трехуровневая архитектура сетевой многопользовательской программной системы. Предложен подход к автоматизированному построению цифрового двойника, представлен алгоритм, в соответствии с которым выполняется автоматизированное построение.

Предложенное в статье методологическое обеспечение может применяться в организациях, занимающихся разработкой программного обеспечения для анализа, проектирования и оптимизации ИЭС различного назначения. Реализованное программное обеспечение при его внедрении в проектные организации повысит качество получаемых проектных рекомендаций и снизит трудоемкость их получения.

5. Источник финансирования исследования

Исследования выполнены в Институте систем энергетики им. Л.А. Мелентьева Сибирского отделения Российской академии наук при поддержке Российского научного фонда (грант № 22-29-01611).

6. Список литературы

- [1] Н. И. Воропай, В. А. Стенников, Интегрированные интеллектуальные энергетические системы, Известия РАН. Энергетика 1 (2014) 64-73.
- [2] N. Voropai, V. Stennikov, S. Senderov, E. Barakhtenko, O. Voitov, A. Ustinov, Modeling of integrated energy supply systems: main principles, model, and applications, J. Energy Eng. 143 (2017) 1-11. doi: 10.1061/(ASCE)EY.1943-7897.0000443.
- [3] V. Stennikov, G. Mayorov, Research of a multiagent model of an integrated energy supply system developed in the anylogic software environment, E3S Web Conf. 209 (2020). doi: 10.1051/e3sconf/202020903024.
- [4] V. Stennikov, E. Barakhtenko, G. Mayorov, D. Sokolov, B. Zhou, Coordinated management of centralized and distributed generation in an integrated energy system using a multi-agent approach, Appl. Energy 309 (2022). doi: 10.1016/j.apenergy.2021.118487.
- [5] M. Grieves, Digital twin: manufacturing excellence through virtual factory replication: White paper 1, Florida Institute of Technology (2014), pp. 1-7.
- [6] A. K. Sleiti, J. S. Kapat, L. Vesely, Digital twin in energy industry: Proposed robust digital twin for power plant and other complex capital-intensive large engineering systems, Energy Reports 8 (2022) 3704-3726, doi: 10.1016/j.egyr.2022.02.305.
- [7] D. Jones, Ch. Snider, A. Nassehi, J. Yon, B. Hicks, Characterising the Digital Twin: A systematic literature review, CIRP J. Manuf. Sci. Technol. 29 (2020) 36-52. doi: 10.1016/j.cirpj.2020.02.002.

- [8] M. Liu, Sh. Fang, H. Dong, C. Xu, Review of digital twin about concepts, technologies, and industrial applications, *J. Manuf. Syst.* 58 (2021) 346-361, doi: 10.1016/j.jmsy.2020.06.017.
- [9] V. Zambrano, J. Mueller-Roemer, M. Sandberg, P. Talasila, D. Zanin, P. G. Larsen, E. Loeschner, W. Thronicke, D. Pietraroia, G. Landolfi, A. Fontana, M. Laspalas, J. Antony, V. Poser, T. Kiss, S. Bergweiler, S. P. Serna, S. Izquierdo, I. Viejo, A. Juan, F. Serrano, A. Stork, Industrial digitalization in the industry 4.0 era: Classification, reuse and authoring of digital models on Digital Twin platforms, *Array* 14 (2022), doi: 10.1016/j.array.2022.100176.
- [10] G. Booch, J. Rumbaugh, I. Jacobson, *The Unified Modeling Language User Guide*, Addison Wesley, Boston, 1999.
- [11] G. Booch, *Object-Oriented Analysis and Design with Applications*, 2nd. ed., Addison Wesley, Boston, 1994.
- [12] A. R. da Silva, Model-driven engineering: A survey supported by the unified conceptual model, *Computer Languages, Systems & Structures* 43 (2015) 139-155. doi: 10.1016/j.cl.2015.06.001.
- [13] M. Brambilla, J. Cabot, M. Wimmer, *Model-driven software engineering in practice. Synthesis Lectures on Software Engineering*, Morgan & Claypool, Kentfield, CA, 2012.
- [14] J. Seixas, A. Ribeiro, A. Rodrigues da Silva, A Model-Driven Approach for Developing Responsive Web Apps, in: *Proceedings of the 14th International Conference ENASE 2019*, SciTePress, Setubal, pp. 257–264. doi: 10.5220/0007678302570264.
- [15] D. Akdur, V. Garousi, O. Demirörs, A survey on modeling and model-driven engineering practices in the embedded software industry, *J. Syst. Archit.* 91 (2018) 62-82. doi: 10.1016/j.sysarc.2018.09.007.
- [16] I. Boussaïd, P. Siarry, M. Ahmed-Nacer, A survey on search-based model-driven engineering, *Autom Softw Eng* 24 (2017) 233–294. doi: 10.1007/s10515-017-0215-4.
- [17] I. Al-Azzoni, J. Blank, N. Petrović, A Model-Driven Approach for Solving the Software Component Allocation Problem, *Algorithms* 14 (2021). doi: 10.3390/a14120354.
- [18] E. de Araújo Silva, E. Valentin, J. R. H. Carvalho, R. da Silva Barreto, A survey of Model Driven Engineering in robotics, *J. Comput. Lang.* 62 (2021). doi: 10.1016/j.cola.2020.101021.
- [19] K. Hazzard, J. Bock, *Metaprogramming in .NET*, Manning Publications, NY, 2013.
- [20] R. Lämmel, *Software Languages: Syntax, Semantics, and Metaprogramming*, Springer, Cham, 2018.

Информационные технологии для поддержки археологических исследований

А. Б. Спасибо¹, А. А. Ветров¹, Е. С. Фереферов^{1,2}

¹Иркутский государственный университет, Иркутск, Россия

²Институт динамики систем и теории управления имени В.М. Матросова Сибирского отделения Российской академии наук, Иркутск, Россия

Аннотация.

В работе рассматриваются способы применения современных информационных технологий для различных решения задач археологических исследований. Описывается возможность и перспективы создания археологических информационных систем, обеспечивающих учет и анализ собранных археологических данных на все этапах исследования. Затрагиваются возможные трудности при создания таких информационных систем. Рассматривается одно из наиболее перспективных направлений информационных технологий в археологии – трехмерное моделирование. Приводятся примеры применения 3D-технологий (3D-сканирования, 3D-печати и 3D-моделирования) для проведения археологических исследований в различных направлениях и на разных этапах работы. В частности: создание высокодетализированных реплик реальных объектов, виртуальные экспозиции и выставки, реконструкция внешнего облика человека (например, по черепу), сохранение и восстановление первоначального облика изделия.

Ключевые слова.

Информационные технологии, информационная система, 3D-моделирование, 3D-сканирование, археология.

1. Введение

Первостепенной задачей любого археолога является изучение объекта (находки) и получение исторической информации о нем. Современные археологи всегда работают с большим количеством информации, фотоматериалами и результатами сканирования объектов или местности. На данный момент только в России обнаружено более ста тысяч археологических памятников. Лишь небольшая часть из них находится на учете и исследуется. Ежегодно обрабатываются большие объемы новых материалов, систематизируются старые архивы и исследования.

В последние несколько лет для поддержки и систематизации археологических исследований в работу археологов активно внедряются современные информационные технологии. Уже сейчас задачи, решаемые ИТ в области археологии, составляют довольно большой список:

- Хранение и обработка данных (атрибутивные данные, графические данные, пространственные карты, модели)
- Визуализация
- Публикация

4th International Workshop on Information, Computation, and Control Systems for Distributed Environments (ICCS-DE 2022), July 04–08, 2022, Irkutsk, Russia

EMAIL: aleksbor@gmail.com (A. 1); vetrov@icc.ru (A. 2); fereferov@icc.ru (A. 3)

ORCID: 0000-0002-3715-2513 (A. 1); 1111-2222-3333-4444 (A. 2); 0000-0002-7316-444X (A. 3)



© 2022 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

ICCS-DE 2022 Workshop Proceedings

- Анализ
- Моделирование
- Извлечение данных
- Работа с большими данными (Big Data)

Применение современных IT технологий в археологических исследованиях позволяет значительно автоматизировать процесс проведения исследований и сбора данных, повысить качество получаемых результатов.

2. Информационные системы в археологии

Работы по разработке, созданию и внедрению информационных систем для решения различных задач археологии ведутся давно и в России, и зарубежом.

Одна из первых археологических информационных систем (АИС) в России – это «Археограф» [1]. Она была разработана в 2008 году в Институте истории материальной культуры РАН. На сегодняшний день информационная система более не развивается, так как с течением времени устарела. АИС «Археограф» была реализована в виде настольной информационной системы и обеспечивала учет и хранение базовой информации о памятниках и находках. При помощи средств ГИС MapInfo осуществлялся научный анализ пространственных данных. Взаимодействие с ГИС было реализовано через технологию DDE (Dynamic Data Exchange).

В силу своей спецификации, информация, получаемая на каждом из этапов археологических исследований, применяется большое количество классификаторов. Кроме того, необходимо фиксировать как послойное описание процесса работ, так и общую информацию о месте проведения раскопок (историю открытия, географическое положение). Все это составляет большой объем информации, который нужно обрабатывать и систематизировать для дальнейшего изучения или хранения. Отсюда вытекает одна из главных проблем внедрения информационных технологий в археологические исследования – это отсутствие единых стандартов организации исследований в целом, а также последующего описания полученных результатов на каждом из этапов того исследования. Из-за этого становится сложно провести относительные оценки степени изученности имеющихся археологических объектов.

Все археологические исследования тесно связаны с использованием пространственных данных. Археологические карты в России составлялись и издавались еще до революции. Правила составления археологических карт наиболее полно были описаны в работе "Методика составления археологической карты" [2] в 1939 году. Данную методичку используют до сих пор при составлении современных археологических карт, например с использованием ГИС [3] (GIS). Однако четкого стандарта, как и при описании полученных результатов исследований, для создания археологических карт нет. Поэтому, фактически, при составлении карт археологических находок часто выполняется лишь несколько этапов, описанных в методичке, или эти этапы не соблюдаются вовсе.

Все это делает практически невозможным сравнение имеющихся результатов научных исследований в области археологии и создание единой стандартизированной информационной системы для записи всей существующей информации. Структура проектируемой АИС также должна учитывать количество содержащейся в ней информации, спецификации этой информации и конечного пользователя информационной системы.

Условно все АИС в мире можно разделить на две категории: служебные и научные.

Служебные АИС создаются на уровне муниципалитета, района или страны. То есть в любом случае ограничены территориально. Основная задача таких информационных систем заключается в охране памятников культурного наследия и мониторинга их состояния. Данные, содержащиеся в служебной АИС, имеют общий характер. Реализуются такие АИС в виде реестров, баз данных или геопорталов.

Научные АИС создаются на уровне проектов, институтов, музеев. Они решают задачи в рамках определенных исследований. Данные, хранящиеся в научных АИС, имеют детальный

характер описания и часто имеют ограниченный доступ. Реализуются в виде реестров, баз данных и геопорталов с активным использованием технологий ГИС.

3. 3D-технологии

Большой скачок в развитии компьютерных технологий в археологии связан с внедрением и развитием такой сферы, как 3D-технологии. Возможности применения трехмерного моделирования для археологических исследований очень обширны. Однако, в настоящее время не все археологические исследования проводятся с использованием трехмерных технологий.

В первую очередь слабое распространение 3D-технологии связано с трудоемкостью процесса. Археологические находки являются сложными объектами с точки зрения геометрии. Сказывается отсутствие специалистов. Без специальной подготовки и соответствующего образования в сфере истории, археологии и IT невозможно качественно овладеть 3D-технологиями для проведения археологических исследований.

В настоящее время в археологических исследованиях распространено 3D-моделирование методами фотограмметрии и сканированием [4]. Такие технологии позволяют сохранять и реконструировать любые археологические объекты с высокой точностью и детализацией. В ранних исследованиях фотограмметрия использовалась чаще, чем сканирование. Так как из-за высокой стоимости оборудования для 3D-сканирования оно не было широко распространено. Однако в настоящее время для археологических исследований доступны различные бюджетные модели 3D-сканеров и такой метод сканирования становится все более популярным. Выбор сканера обусловлен характеристиками желаемого оборудования: размером сканируемого объекта, требуемым качеством модели, предполагаемым местом использования и ценовым диапазоном.

Например, для сканирования большие археологических объектов (памятников, скульптур) могут быть использованы такие сканеры как 3D-Scanner Creaform Go!SCAN Spark или 3D-Scanner Shining 3D EinScan. Для сканирования и моделирования рельефа местности используются специальные сканеры, такие как Ground-based laser scanner Trimble SX10.

Наиболее популярными являются сканеры, способные сканировать мелкие объекты с высокой точностью. Такие сканеры помогают сохранить информацию и внешний облик археологического объекта с высокой точностью, показывая результат отклонения менее 1 мм. Таких результатов можно добиться используя, например, сканер 3D-Scanner RangeVision Spectrum [5]. Пример результата сканирования объекта с текстурой сканером Scanner RangeVision Spectrum приведен на Рисунке 1.



Рисунок 1: 3D-модель черепа человека, погребение Шумилиха 1999 г.

3.1. Трехмерные модели высокой точности

Высокодетализированные 3D-модели археологических объектов позволяют воспользоваться услугами специалистов по всему миру, не подвергая хрупкий и светочувствительный оригинал транспортировке. Благодаря 3D-технологиям можно детально изучить реплику археологического объекта в любой точке мира, скачать ее или при необходимости распечатать на 3D-принтере. Работа с высокодетализированными 3D-сканами дает возможность исследовать объекты, которые во время их изучения могут быть полностью утрачены из-за своей хрупкости, плохой сохранности или по другим причинам. Таким образом, использование 3D-моделей и реплик вместо оригинала не только помогает сохранить первоначальный внешний облик объекта, но сводит к минимуму риск утраты ценной археологической находки.

Кроме того, методы 3D-сканирования с текстурированием позволяют изучать объекты, требующие цветопередачи. Такой способ, например, использовался при изучении наскальных рисунков Горного Алтая [6].

Высокодетализированные 3D-модели могут использоваться в качестве учебных пособий при дистанционном формате обучения и при традиционном (очном). Виртуальные модели археологических объектов можно демонстрировать на уроках в любом помещении, где есть доступ к компьютеру или проектору. Для какого-то одного объекта можно сделать несколько реплик из пластика на 3D-принтере, чтобы обучающиеся могли тренироваться на них проводить измерения, реконструкции и другие задачи, связанные с археологическими исследованиями. Для изучения большого количества археологических объектов больше не требуется специальное оборудование, отдельные помещения для хранения находок, где бы поддерживалась постоянная влажность и температура.

Применение и использование 3D-технологий в современном мире открывает доступ к виртуальным выставкам и музеям. Современные технологии позволяют предоставить для выставок как небольшие объекты, так и крупные архитектурные памятники [7]. Таким образом, возможность виртуального представления различных археологических находок обеспечивает доступность объектов культурного наследия, в том числе тщательно оберегаемых или представленных только в виде экспонатов конкретных музеев, широкому кругу лиц.

3.2. Реконструкция

С помощью 3D-сканирования, специального программного обеспечения и определенных навыков специалисты могут проводить реконструкции виртуальных предметов археологических находок. Большое количество находок представляют из себя не целые предметы, а лишь их части, которые сохранились до наших дней.

Например, расколотый керамический сосуд можно восстановить практически полностью собрав и соединив вместе кусочки, на которые он был разбит. Но никто не может гарантировать, что все керамические осколки будут в идеальном состоянии, никакие их части не были утрачены и на реконструкции не останется дефектов. В то же время, с помощью современных программ и алгоритмов в перспективе можно будет виртуально восстановить первоначальный облик любого предмета: разбитого керамического сосуда, железного орудия с большой долей коррозии, частично сохранившейся фигурки животного.

Отдельным направлением в реконструкции археологических находок можно выделить восстановление облика исторических личностей по их останкам. Восстановление лица по черепу или метод антропологической реконструкции облика на краниологической основе позволяет по внешним данным черепа, по его рельефу и структуре реконструировать портретное лицо (облик) человека, которому этот череп принадлежал. Реконструкция внешнего облика человека на основе его черепа давно является самостоятельным направлением науки и широко используется не только в археологии.

В привычном понимании реконструкции на подлинном черепе при помощи скульптурного пластилина восстанавливаются основные мускулы и черты лица. Это крайне негативно сказывается на сохранности важной археологической находки.

Альтернативой этому, современные программы и алгоритмы позволяют виртуально восстановить внешний облик человека по его останкам, то есть построить его 3D-модель. Это никак не вредит подлинным останкам, так как все этапы реконструкции происходят с использованием 3D-сканов костей. Такой способ позволяет не только воссоздать внешний вид лица человека, но и создать полную модель скелета исторической личности (если сохранились достаточное количество костей скелета). Пример виртуального «соединения» костей продемонстрирован на Рисунке 2 на примере длинных костей правой руки человека.



Рисунок 2: Соединение 3D-моделей по анатомическому атласу (правая рука).

Не так давно были предложены полностью автоматизированные методы реконструкции лица на основе 3D-модели черепа [8]. Такие методы не требуют вмешательства человека в процесс реконструкции, но полученные результаты пока еще не отличаются высоким качеством и достоверностью.

В настоящее время активно также ведутся работы над алгоритмами, которые бы позволили анимировать полученные трехмерные реконструкции.

3.3. Моделирование рельефа

В археологии большое внимание уделяют картам и рельефу местности.

Особый интерес представляют специализированные геоинформационные системы [9]. Такие системы обеспечивают удобство работы с интерактивными картами и возможность пространственного анализа. Важным направлением является также 3D-моделирование раскопов, комплексов, городищ и других архитектурных комплексов. Для моделирования рельефа можно использовать несколько способов.

В случае, например, использования лазерного сканера результатом съёмки является облако точек, которое представляет собой подробную точечную 3D-модель объекта съёмки. На основе данных 3D сканирования можно как выполнять подготовку традиционных плоских чертежей, так и строить трёхмерные модели объектов или их отдельных частей. А используя инструментальные средства ГИС Карта, строиться 3D-модель поверхности, которая строится с учетом рельефа местности, на которую будет наложено изображение матричной, векторной или растровой карты, также включающее объекты с 3D-метрикой.

4. Выводы

Анализируя ситуацию в российской науке, можно сказать, что внедрение современных информационных технологий в археологические исследования находится на начальном этапе. Не все связанные с археологией научные центры, музеи и учреждения в России используют хотя бы несколько из видов современных информационных технологий для проведения исследований, хотя общепризнанно, что они весьма перспективны и могут применяться для многих работ.

Некоторые технологии, такие как информационные системы, используются в археологии уже давно и продолжают успешно развиваться. Однако многие технологии, например 3D-моделирование археологических объектов, только начинают развиваться.

5. Благодарности

Данная работа выполнена по гранту Правительства РФ, проект No 075-15-2021-631 от 08.06.21: «Байкальская Сибирь в каменном веке: на перекрестке миров».

Список литературы

- [1] Васильев Ст. А. АИС «Археограф»: система описания археологических памятников и вывода данных в ГИС // Археология и компьютерные технологии: представление и анализ археологических материалов. – Ижевск: УдмИИЯЛ УрО РАН, 2005. – С. 13–21.
- [2] Мансуров А. А. Методика составления археологической карты // Н.-и. ин-т краевед. и музейной работы. – Москва : [б. и.], 1939. – 64 с.
- [3] Кениг А. В, Зайцева Е. А. Применение технологии ГИС и методов ДЗЗ в системе выявления, мониторинга и охраны объектов археологического наследия // Интерэкспо Гео-Сибирь. 2013. №. URL: <https://cyberleninka.ru/article/n/primenenie-tehnologii-gis-i-metodov-dzz-v-sisteme-vyyavleniya-monitoringa-i-ohrany-obektov-arheologicheskogo-naslediya>.
- [4] Вавулин М. В. Технологии трехмерной оцифровки крупных автономных археологических объектов // Вестник Томского государственного университета. – Томск, 2016, N. 407 – С. 55–60.
- [5] Чистяков П. В., Ковалев В. С., Колобова К. А., Шалагина А. В., Кривошапкин А. И. 3D моделирование археологических артефактов при помощи сканеров структурированного подсвета // Теория и практика археологических исследований. – 2019. N. 3 (27). – С. 102–112.
- [6] Мартынов А.И., Сивина К.Н., Мещерский Р.Д. Гора Туру-Алты в Горном Алтае и проблема идентификации древних святилищ // Вестник Кемеровского государственного университета. – Кемерово, 2019 – N. 3 (79). – С. 626-634.
- [7] Вавулин М. В. Технологии трехмерной оцифровки крупных автономных археологических объектов // Вестник Томского государственного университета. – Томск, 2016. N. 407. – С. 55–60.
- [8] Turner WD, Brown RE, Kelliher TP, Tu PH, Taister MA, Miller KW // A novel method of automated skull registration for forensic facial approximation. – Forensic Sci Int. 2005 Nov 25; 154(2-3):149-58. DOI: 10.1016/j.forsciint.2004.10.003.
- [9] Акашеев А. А. Пространственный анализ данных в исторических науках. Применение геоинформационных технологий. – Нижний Новгород: Нижегородский госуниверситет, 2011. – С. 79.

Intelligent software system APP-MNIT for analytical calculations using parallel computing

Aleksandr Reznik ¹, Aleksandr Soloviev ¹

¹*Institute of Automation and Electrometry of the Siberian Branch of the Russian Academy of Sciences (IA&E SB RAS), Academician Koptyug ave. 1, Novosibirsk, 630090, Russia*

Abstract

The paper presents the software system APP-MNIT, which is capable of performing complex analytical calculations using parallel computing. The task that was the stimulus for the development of this complex software is described. The algorithm underlying the operation of the presented system is given. The results obtained with the use of APP-MNIT are presented.

Keywords

Software systems, analytical calculations, parallel computing.

1. Introduction

In scientific practice, there are cases when scientists are faced with problems that look simple in formulation, but turn out to be extremely difficult to solve. Among such examples, we can note the still unsolved Collatz conjecture problem [1] or the Inscribed square problem [2].

A similar situation arose in our work. When calculating exact analytical formulas for estimating the parameters of error-free reading of digital images with multi-threshold integrators [3], we are faced with the need to solve the following problem:

"Let n points $x_1, x_2, \dots, x_n$ be randomly thrown onto the interval $(0, L)$. It is necessary to find the probability $P_{n,k}(\varepsilon, L)$ that the interval $(0, L)$ does not contain any subinterval of length ε containing more than k points".

It turned out that at present the exact analytical solution is known only for the case $k=1$ [4-5]. To advance in solving this problem, we have created several specialized algorithms based on analytical transformations on a computer [6]. In the next Section, one of the algorithms that formed the basis for the development of the APP-MNIT software system using parallel computing will be described.

We call APP-MNIT software system "intelligent", because in addition to computing power, it has all the tools for carrying out complex and branched symbolic-analytical calculations with obtaining formulas in an analytical form.

⁴th International Workshop on Information, Computation, and Control Systems for Distributed Environments (ICCS-DE 2022), July 04–08, 2022, Irkutsk, Russia

EMAIL: reznik@iae.nsk.su (A. 1); soloviev@iae.nsk.su (A. 2);

ORCID: 0000-0003-3466-5433 (A. 1); 0000-0002-5352-222X (A. 2);



© 2022 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

ICCS-DE 2022 Workshop Proceedings

2. Algorithm for Analytical Computer Computation of Multidimensional Integrals over Convex Polyhedra in n -Dimensional Space

So, we have developed a specialized algorithm for calculating the formulas $P_{n,k}(\varepsilon, L)$ using direct integration over domains, which are parametrically specified polyhedra in n -dimensional space. The idea of the method is as follows. At the initial stage, in the scheme for calculating the formulas $P_{n,k}(\varepsilon, L)$, using the usual normalization, the interval $(0, L)$ is replaced by the interval $(0, 1)$. Thus, the parameter L is excluded from further consideration. Further, the probability $P_{n,k}(\varepsilon)$ is presented in the following form:

$$P_{n,k}(\varepsilon) = n! \int_{D_{n,k}(\varepsilon)} \dots \int dx_1 \dots dx_n, \quad (1)$$

where the area of integration $D_{n,k}(\varepsilon)$ is described by a system of linear inequalities:

$$\begin{cases} 0 < x_1 < x_2 < \dots < x_{n-1} < x_n < 1, \\ x_{k+1} - x_1 > \varepsilon, \\ x_{k+2} - x_2 > \varepsilon, \\ \vdots \\ x_n - x_{n-k} > \varepsilon. \end{cases} \quad (2)$$

At the next stage, integral (1) over domain (2) is rewritten in the equivalent form:

$$P_{n,k}(\varepsilon) = n! \int \dots \int 1[x_1]1[x_2] \dots 1[x_n - x_{n-1}]1[1 - x_n]1[x_{k+1} - x_1 - \varepsilon] \times \\ \times 1[x_{k+2} - x_2 - \varepsilon] \times \dots \times 1[x_n - x_{n-k} - \varepsilon] dx_1 \dots dx_n, \quad (3)$$

thus, the integration is already carried out over the entire space. The indicator function of the integration area $D_{n,k}(\varepsilon)$ is transformed into a product of unit functions

$$1[z] = \begin{cases} 0, & z \leq 0, \\ 1, & z > 0, \end{cases}$$

that are included in the integrand of the integral.

After that, the n -dimensional integral (1) with the help of successive application of the relation

$$\left(\prod_{j=1}^l 1[x_r - \alpha_j] \right) \left(\prod_{i=1}^m 1[\beta_i - x_r] \right) = \sum_{j=1}^l \sum_{i=1}^m 1[x_r - \alpha_j] 1[\beta_i - x_r] \left\{ 1[\beta_i - \alpha_j] \left(\prod_{\substack{q=1 \\ q \neq j}}^l 1[\alpha_j - \alpha_q] \right) \left(\prod_{\substack{s=1 \\ s \neq i}}^m 1[\beta_s - \beta_i] \right) \right\} \quad (4)$$

reduces to a set of iterated integrals with the limits of integration already placed. If, using the algorithmic scheme (4), we calculate the limits of integration with respect to the variable x_l in the original integral (1), then it will split into two integrals:

$$\begin{aligned}
P_{n,k}(\varepsilon) = & n! \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} \left[\int_0^{x_2} dx_1 \right] 1[x_{k+1} - x_2 - \varepsilon] 1[x_2] \left(\prod_{m=3}^n 1[x_m - x_{m-1}] \right) \times \\
& \times 1[1 - x_n] \left(\prod_{m=2}^{n-k} 1[x_{m+k} - x_m - \varepsilon] \right) dx_2 \dots dx_n + \\
& + n! \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} \left[\int_0^{x_{k+1} - \varepsilon} dx_1 \right] 1[x_2 + \varepsilon - x_{k+1}] 1[x_{k+1} - \varepsilon] \left(\prod_{m=3}^n 1[x_m - x_{m-1}] \right) \times \\
& \times 1[1 - x_n] \left(\prod_{m=2}^{n-k} 1[x_{m+k} - x_m - \varepsilon] \right) dx_2 \dots dx_n. \tag{5}
\end{aligned}$$

After carrying out in each of the expressions into which the original n -fold integral decomposes, setting the limits of integration over the variables x_2 , then x_3 , and so on, as a result, we obtain a complete set of iterated integrals, to the sum of which the original multiple integral (1) is reduced, and the limits integrations are already set in it. At the final stage, the following is done:

1. Sequential integration of each of the N iterated integrals into which the original integral (1) has decomposed is performed.
2. The results obtained are combined (the limits of change of the free parameter ε are taken into account).

The presented algorithm allows constructively calculating the formulas $P_{n,k}(\varepsilon)$, which completely solves the problem of finding the probability $P_{n,k}(\varepsilon)$ for fixed values of n and k . The main difficulty in using this method of getting formulas $P_{n,k}(\varepsilon)$ for cases $n > 4$ is associated with a huge amount of calculations that need to be done (it is required to set the limits of integration, check all intermediate systems of inequalities for compatibility and directly calculate iterated integrals). The table shows an example of manual results obtained using the above method for $n=3$ and $n=4$.

Table 1

Exact analytical formulas $P_{n,k}(\varepsilon)$ for fixed values of n and k (manually calculated)

n	k	change area of ε	$P_{n,k}(\varepsilon)$
3	2	(0,1)	$1-3\varepsilon^2+2\varepsilon^3$
4	2	(0,1/2)	$1-12\varepsilon^2+24\varepsilon^3-14\varepsilon^4$
4	2	(1/2,1)	$2-8\varepsilon+12\varepsilon^2-8\varepsilon^3+2\varepsilon^4$
4	3	(0,1)	$1-4\varepsilon^3+3\varepsilon^4$

However, for $n > 4$, the volume of necessary calculations increases so much that it becomes unrealistic to do manually all the necessary work. Therefore, we have created a special software package for computer calculation of the necessary dependencies (using the scheme presented above). The algorithms were implemented in Matlab and C++. With their help, the formulas $P_{n,k}(\varepsilon)$ up to $n=11$ were calculated.

3. Increasing the speed of the algorithm using parallel computing

Unfortunately, the computer programs based on the algorithm in Section 2 still did not allow us to obtain enough data to make further progress in solving the problem. The solution was the development of a specialized software system for a supercomputer (cluster) using parallel computing.

To create such a system, a cluster of the Information and Computing Center of Novosibirsk State University (NUSC NSU) was used. C++ was selected as the programming language; the program used both OpenMP and MPI programming interfaces. The developed software system was named "APP-MNIT". In printed form, the code of the software system occupies more than 70 sheets.

At the beginning of software system's development in 2010, the peak (theoretical) performance of the computing complex was 5447.7 Gflops, and the one obtained on the Highly Parallel Linpack test was 4085 Gflops [7]. The complex was on 18th place in the 9th edition of the TOP 50 CIS supercomputers rating [8]. Due to the high workload of the complex, there were restrictions on the number of cores used by users. Most of the time APP-MNIT system used 16 cores. This, as will be shown below, made it possible to make significant progress in solving the problem posed.

It is known that not every problem can be effectively solved using parallel computing [9-10]. The algorithm of direct integration over areas, described in the Section 2, can be efficiently parallelized. The brief essence of the parallel algorithm is as follows. Recall that it is necessary to calculate the integral (1), where the area of integration $D_{n,k}(\varepsilon)$ is described by a system of linear inequalities (2). The parallel algorithm allows to significantly reduce the time for checking and transformation of the system of equations (2):

$$\left\{ \begin{array}{l} x_2 > \alpha, \\ x_2 > \beta, \\ \vdots \\ \vdots \\ \vdots \\ x_2 < x_3. \end{array} \right. \Rightarrow \left\{ \begin{array}{l} x_2 > \alpha, \\ \vdots \\ x_2 < x_3 \\ \alpha > \beta. \end{array} \right. \Rightarrow \text{CPU - 1}$$

$$\left\{ \begin{array}{l} x_2 > \beta, \\ \vdots \\ \vdots \\ \vdots \\ x_2 < x_3, \\ \beta > \alpha. \end{array} \right. \Rightarrow \text{CPU - 2}$$

Figure 1: Schematic diagram for the parallel verification of the system of inequalities.

As can be seen in fig. 4, there are two alternative lower limits in the system of inequalities, α and β . The algorithm needs to check both options. The parallel algorithm allows checking these options simultaneously by sending data to different processors (due to this, a significant acceleration of the calculation process is obtained).

Further, the calculation is carried out in accordance with the scheme given in Section 2. In fact, all the cores are “equal” and do the same work (it was found that for large n , the loading of the cores occurs fairly equally). Due to the absence of time-consuming interprocessor interactions, the software system scales well and can be run on a large number of cores. At the final stage, it only necessary to combine the results obtained, taking into account the boundaries of the free parameter ε (similar to the sequential algorithm).

With the help of a parallel algorithm, it was possible to obtain polynomial values up to $n=14$ (this result was obtained on the NUSC NSU computing cluster in 27 hours on 16 processors). An example of an analytical solution obtained by the software system is presented in Table 2 below.

Table 2

Exact analytical formulas $P_{n,k}(\varepsilon)$ for fixed values of $n=13$ and $k=2$

n	k	change area of ε	$P_{n,k}(\varepsilon)$
13	2	(0,1/8)	$1-858\varepsilon^2+12012\varepsilon^3+80080\varepsilon^4-3294720\varepsilon^5+30086628\varepsilon^6-62925720\varepsilon^7-909369747\varepsilon^8+607341600\varepsilon^9-34250637564\varepsilon^{10}+8277827280\varepsilon^{11}-54680029443\varepsilon^{12}-3365355508\varepsilon^{13}$
13	2	(1/8,1/7)	$104\varepsilon-5850\varepsilon^2+158444\varepsilon^3-2848560\varepsilon^4+38877696\varepsilon^5-419752476\varepsilon^6+3535787112\varepsilon^7-22501646739\varepsilon^8+104573017120\varepsilon^9-341340799228\varepsilon^{10}+738292725456\varepsilon^{11}-948033227011\varepsilon^{12}+546390458380\varepsilon^{13}$
13	2	(1/7,1/6)	$-143+11440\varepsilon-417846\varepsilon^2+9241804\varepsilon^3-138278855\varepsilon^4+1479488868\varepsilon^5-11654720220\varepsilon^6+68485295736\varepsilon^7-300346819344\varepsilon^8+971335553760\varepsilon^9-2252457979200\varepsilon^{10}+3547733904000\varepsilon^{11}-3402622080000\varepsilon^{12}+1501156800000\varepsilon^{13}$

Despite the fact that the parallel algorithm moved only 3 steps further than the sequential one, this result was very important, since it made possible to discover new dependencies and helped in the development of a rigorous mathematical proof of the formulas obtained using a computer. The following is an example of a formula that was obtained with the help of the presented software system. For odd values $n = 2m + 1$ on the segment $1/(m+1) < \varepsilon < 1/m$, new formula was established:

$$P_{2m+1,2}(\varepsilon) = C_{2m+1}^{m+1}(1 - m\varepsilon)^{m+1}(1 - (m-1)\varepsilon)^m - 2C_{2m+1}^{m+2}(1 - m\varepsilon)^{m+2}(1 - (m-1)\varepsilon)^{m-1} + C_{2m+1}^{m+3}(1 - m\varepsilon)^{m+3}(1 - (m-1)\varepsilon)^{m-2}.$$

Detailed information about finding this formula and its proof can be found in [11].

4. Conclusion

This work is an example of how the solution of tasks that seem very simple to formulate and solve turns into a complex scientific problems that requires not only the development of specialized algorithms, but also the creation of systems using parallel computing. We called the APP-MNIT system "intelligent", since it is not a primitive "calculator", but has the ability to carry out complex and branched analytical transformations and obtain formulas in an analytical form that a person is not

able to get due to the gigantic volume of required calculations. We hope that in the future we will be able to expand the capabilities of the developed algorithms and program systems to solve similar more difficult problematic tasks.

5. Acknowledgements

This work was supported by the Ministry of Science and Higher Education of the Russian Federation (project No. AAA-A17-117052410034-6).

6. References

- [1] E. Watson, D. McCue, The Stabilization of Sequences from the Collatz Conjecture, *International Journal of Mathematics Trends and Technology* 68 (2022), doi:10.14445/22315373/IJMTT-V68I2P507.
- [2] R. Schwartz, Rectangles, curves, and Klein bottles, *Bulletin of the American Mathematical Society* 59 (2021): 1–17. doi:10.1090/bull/1755.
- [3] A. Reznik, V. Efimov, A. Torgov, A. Soloviev, Analytical computer calculations in problems with random division of the interval, *Pattern Recognition and Image Analysis (Advances in Mathematical Theory and Applications)* 22 (2012), doi: 10.1134/S1054661812020125.
- [4] D. Darling, On Class Problems Related to the Random Division of an Interval, *Ann. Math. Stat.*, 24 (1953) 239-253.
- [5] J. Naus, Some Probabilities, Expectations, and Variances for the Size of Largest Clusters and Smallest Intervals, *J. Am. Statist. Assoc.* 61 (1966) 1191-1199.
- [6] A. Reznik, A. Tuzikov, A. Soloviev, A. Torgov, Analysis of random point images with the use of symbolic computation codes and generalized Catalan numbers. *Optoelectronics, Instrumentation and Data Processing* 52 (2016) 529-536, doi: 10.3103/S8756699016060017.
- [7] Official information of the Information and Computing Center of NSU, 2022. URL: <http://nusc.nsu.ru/wiki/doku.php/doc/nusc/about>.
- [8] CIS-supercomputers rating, 2010. URL: <http://top50.supercomputers.ru>.
- [9] R. Robey, Y. Zamora, *Parallel and High Performance Computing*, Manning Publications, New-York, 2021.
- [10] H. Casanova, A. Legrand, Y. Robert, *Parallel Algorithms*, Chapman and Hall, London, 2008.
- [11] A. Reznik, A. Soloviev, A. Torgov. On the probability of the formation of local groups in random point images, *Pattern Recognition and Image Analysis (Advances in Mathematical Theory and Applications)* 26 (2016) 714-719, doi: 10.1134/S1054661816040155.

A Web-based Software System for Intelligent Support in Modeling and Forecasting the Water Resources

Aleksandr Stolbov¹, Anna Lempert¹ and Aleksandr Pavlov¹

¹ *Matrosov Institute for System Dynamics and Control Theory, Siberian Branch of Russian Academy of Sciences (ISDCT SB RAS), Lermontov St. 134, Irkutsk, Russia*

Abstract

The paper deals with the problems of complex model coupling. It is suggested that joint use of component-based and ontological approaches facilitates the processes of technical, conceptual and methodological coupling. As a methodological tool for solving semantic interoperability problems a component-based modeling ontology and related knowledge base are considered. As a software tool the knowledge-based systems development platform is applied. As a mathematical tool a hydrological-ecological balance model is taken. At the current stage of study mostly imputed data is utilized for testing. The proposed software system inherits from the platform general-purpose components for editing a conceptual model, designing knowledge base rules and forming a complex model in the form of a workflow and provides new modules specific to the modeling of water resources.

Keywords

Knowledge-Based Systems, Component-based Modeling, Ontology, Water Resources

1. Introduction

Monitoring, modeling, forecasting and quality estimating of the water resources faces a lot of challenges to be solved. For instance, a lot of diverse techniques related to different levels of coordination and various stakeholders leads to methodological chaos; insufficient competence or lack of proper measure and analytic tools in specialized laboratories to select adequate solutions; difficulties in organizing sampling during mass analysis in the control and monitoring system; at last, the complexity of the considered object itself. Traditionally, mathematical models and scenario analysis are among the most effective and popular tools that allow you to bind and coordinate information of different contents in the process of studying complex objects. In recent years, mathematical and simulation models related to the water resources studies have shifted from the category of monodisciplinarity to interdisciplinary. This trend has increased the complexity of models not only from the mathematical viewpoint, but also from the perspective of software required for designing and simulating models. In this context, the key problems are the issues of organizing interaction between subjects and objects of the modeling process, in particular:

- communication between research groups: both from different knowledge domains and geographically remote;
- combining models into a single united resulting model related to the considered complex object;
- analysis and transformation of data obtained from various sources to process in computations.

The paper makes contributions for answering these problems by applying the component-based and ontological approaches: one to handle with decentralizations and coupling issues, another for semantic interoperability. In the component-based approach a complex model consists of two or more

4th International Workshop on Information, Computation, and Control Systems for Distributed Environments (ICCS-DE 2022), July 04–08, 2022, Irkutsk, Russia

EMAIL: stolboff@icc.ru (A. 1); lempert@icc.ru (A. 2); asd@icc.ru (A. 3)

ORCID: 0000–0001–6513–7030 (A. 1); 0000–0001–9562–7903 (A. 2); 0000–0002–7753–7514 (A. 3)



2022 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

ICCS-DE 2022 Workshop Proceedings

separate models connected by a common computational framework to ensure automatic interaction between them. So, a complex model can be mono disciplinarian, i.e. it can cover only a few processes from one area. But most benefits from component-based approach one can achieve when carrying out a comprehensive assessment of the phenomenon with the help of many complementary models from different domains.

In the course of this interdisciplinary research complex model can be considered as system of systems [1] – a hierarchical structure with different levels of abstraction, where systems and indicators at lower levels can scale to higher levels. Then as complex model and number of related elements is growing, the developer needs to store and process more and more heterogeneous information about units and their proper coupling. And it is vital to take into consideration possible inconsistency in semantic interoperability. That's where ontological approach can suit. By creating an ontology of a certain subject area (the standard form includes such elements as a concept, relation, property, instance), the developer formalizes and structures knowledge in a special way (for example, using patterns). As a result, it is created a single expandable conceptual space, that can be interpreted by both researchers and computer programs. So, the joint usage of component and ontological approaches along with related web-based software system is relevant and makes contribution to the application of modern information and communication capabilities towards the system analysis, modeling and simulation problems.

In the next section of the paper there is a discussion about the principles, pros, cons and typical problems of the component-based modeling in the general way with examples regarding to environmental studies. Then to handle coupling problems a software tool and information models is proposed. At last an illustrative hydrological-ecological example is considered.

2. A scenario modeling in the knowledge-based systems development platform

2.1. The component-based modeling approach overview

The ideas of component-based modeling and design are widely in demand in various fields of knowledge and are based on general concepts of analysis, synthesis, decomposition, etc. In the context of current research component-based modeling is considered for problems of mathematical and simulation modeling related to water resources, in particular, and environment modeling in general. For this reason, Integrated Environmental Modeling (IEM) [2, 3] research area was chosen as one of the main source of inspiration for carrying out our studies.

IEM is rather new scientific discipline that studies the relationship of the main characteristics of society and the economy along with the biosphere and the atmosphere within a single model structure. A distinctive feature of IEM is focusing on achieving practical results and obtaining ready-to-use recommendations that could be implemented by specialists from government agencies to solve environmental problems. In this regard, it is important to find a balance between the computational possibilities to carry out multivariate, efficient and accurate calculations with interpreted results and a complex coverage of the studied problem considering all the diversity of interacting objects and subjects. Thus, the necessity to involve local groups with different competencies stimulates the active use of the component-based approach in solving IEM related problems.

IEM community has proposed a set of various protocols (OpenMI, OMS, BMI) to solve the problem of modularity and connectivity. Current IEM frameworks utilize modern software engineering principles and contain a set of queries and functions for model coupling as well as integration with other common software infrastructure (data bases and CRUD operations, visualization, etc.). The main purpose of existing IEM standards is technical integration, i.e. supporting model chain implementations: automatic data exchange between models; proper execution of model sequence; issues regarding hardware performance. For instance, the OpenMI (Open Modeling Interface) standard is actively used for water resources modeling (Delft3D, InfoWorks, MASCARET, SOBEK, RFSM-EDA). Key OpenMI features confirming its technical integration capabilities are the following:

- two types of component implementation: facade pattern or direct support of programming interface from model code;
- unidirectional and bidirectional connections between components: various options for querying and returning data;
- two types of events: a change in the status of a component or a change in the element value involved in the exchange data process between components.

To overcome mentioned semantic interoperability problems it is necessary to extend coupling process with two more integration stages: conceptual and methodological ones. Conceptual integration deals with semantic interpretation of components inputs and outputs and related coupling problems. Concept consistency, concept-based constraint checks, unit conversion are in the scope here. Methodological integration handles more complex coupling difficulties in the system of systems context: methods consistency, model scaling, meaningful model chain interpretation and validation.

So to support conceptual and methodological integration the explicit knowledge processing methods and tools should be utilized, including ontology and knowledge bases. In the next sections a discussion about this topic is done in more details.

2.2. Software system architecture overview

A web-based software system for intelligent support in modeling and forecasting the water resources is being developed on the base of knowledge-based systems development platform (KBSDP) [4]. The key platform characteristics are the following: a web-oriented component-based open client-server architecture; the ability to integrate external problem-oriented tools into the applied KBS; the central role of the conceptual model in the development process; the application of the workflow approach for the components assembling [5].

In order to take into account the specifics related to the scenario modeling process (SMP) of water resources, including the computational modules, new problem-oriented components (W-T and W-C) have been developed as well as existing ones have been modified.

The W-T component is a unified user terminal through which interaction with other components is carried out. The W-T concerns with: control over the sequence of the SMP; storing all necessary information about the state of the SMP, adapting the functionality of other components to the current context and calling appropriate methods.

The W-C component utilizes the benefits of component-based architecture of KBSDP. The W-C contains a set of modules for solving emerging computational problems: calculations by formula or function, differential equations solving, extreme problem solutions, etc. The W-C implementation is carried out both on the basis of open libraries and original programs developed by the authors. The W-C modules are coded in Java, C++, PHP. The launch of executable modules is based on the software and technical characteristics described during registration in the KBSDP: URI, access methods, including authentication parameters, etc.

Additional meaningful information about models and methods needed for conceptual and methodological integration is described in the form of ontology.

2.3. A component-based modeling ontology for coupling

Currently, the prototype of component-based modeling ontology for KBSDP is being developed. Mainly this ontology is compilation of published ontologies [6-10] with adoption for KBSDP specific. The main contribution was made by water resource ontology [6] and following its ideas component-based modeling ontology for KBSDP is presented in multilevel form:

- The resource level deals with general information about researchers, affiliations, project history and versioning, etc.
- The technical level deals with information about underling computational infrastructure, including target operating systems, common purpose software and hardware facilities, supported programming languages, etc.

- The coupling level deals with information about underlying modeling and simulation infrastructure, including parameters of frameworks for model executions, specific software and hardware facilities, supported programming interfaces, etc.
- The formal level deals with information about equations, models, solutions, numerical methods in terms of mathematical theory.
- The domain level currently relates to water resources. This level was created based on different sources of information: reference literature; official sources of information (registers and cadastres, funds and data banks); publications (monographs, journals) and web resources containing the results of surveys and monitoring data; water resources ontologies.
- The unit measure level deals with quantity types ontology.
- The scenario level deals with information about multivariate calculations and analysis.

2.4. A component-based modeling knowledge base for coupling

The KBSDP has components for creating rules on the top of ontology. Thus it has functionality for generating knowledge base fact templates from the ontology elements. Then these templates can be organized in rules with the help of special inbuilt graphical web editor. When constructing the knowledge base rules, the following conditions are used: the presence of a fact of the specified template, a restriction on the value of the fact slot (text, number), a comparison of the slot value with an existing fact, a comparison with the slot value of another fact. To form the right side of the rule, the following types of actions are used: creating a new fact, modifying an existing fact, deleting a fact. The KBSDP generates a knowledge base code on DROOLS.

In the context of a component-based modeling the KBSDP provides three groups of users (mathematicians, domain experts and knowledge engineers) jointly form recommendations in the form of logical rules for solving problems of the following types:

- identification of the process regarding to the research objectives, depending on the conditions in which the complex object is functioning;
- identification of algorithms for processes, depending on the spatial scale, the time step, the availability of observational data and expert assessments, the numerical solution method;
- selection of computational components for algorithms: matching of input and output interfaces of components; validation of the model chain execution.

The result of performing logical inference is the specification of the complex model in terms of the proposed component-based modeling ontology.

3. An illustrative example

As an illustrative model, a hydrological-ecological balance model in the form of system of linear ordinary differential equations was considered [11]. It was divided into a set of elementary component-models and described according to proposed ontology. The domain ontology level deals with different sources of water intake/expenditure and different sources of substance intake/expenditure. The mathematical layer corresponds to the elements of the water balance equation and the mass balance equation of the substance.

From the computational viewpoint, a model component is a software module that implements calculation algorithm and supports program interface of one or more modeling frameworks. For testing purposes, the components of the model [11] were coded in Java and C++ along with providing OpenMI standard support. According to the model [11], the following components have to be developed: discharge from the main source for the water body; flow rate of tributaries; waste-water discharges; rainfall water; water intake; evaporation from the surface; water flow to an adjacent objects; substance consuming due to its decomposition or by organisms consumption; loss or intake of substance to/from the bottom; intake of a substance as a transformation of other substances; intake of a substance as a product of biochemical processes; input/loss of a characteristic substance along with water discharge.

Ontology instances related to these components describe specific situations in their properties and relations. The rules for determining the composition of the complex model depending on the research objectives and the availability of information were formed. To perform computational experiments, in addition to the components of balance models, other model components to generate necessary input data were developed. In turn, available statistical information about the hydrological characteristics of a water body and expert assessments were used as input data for data generation components.

Currently the testing process was carried out with regards to the following principal algorithm:

1. The user creates instances of the ontology (paper section 2.3) that characterize the ecological and economic situation in the territory.
2. The user forms the objectives of the current study in terms of ontology instances.
3. Based on the knowledge base prototype (paper section 2.4) during interactive inference the *specification* of the complex model is formed.
4. If all the necessary data has been obtained, the user performs the assembly of the complex model in accordance with the specification.
5. The user launches a complex model for computation.

The computational experiments have shown the fundamental operability of the models, methods and tools applied in the web-based software system for intelligent support in modeling and forecasting the water resources. The obtained results confirmed the idea that the combination of component-based and ontological approaches allow quickly prototyping complex models with technical, conceptual and methodological integration support.

4. Conclusions

The web-based software system for intelligent support in modeling and forecasting the water resources it is being developed in the form of a component-based modeling knowledge-based system. As a tool for its creation the authors' knowledge-based system development platform (KBSDP) is applied. Simultaneously the set of KBSDP components is refined and extended.

Based on the review of the component-based modeling methods and tools applied in the integrated environmental modeling domain three classes of model coupling problems have been identified: technical integration (matching data to run a chain of models), conceptual integration (matching the conceptual apparatus), methodological integration (matching modeling methods). It is suggested to apply ontological and knowledge-based approaches for model coupling process. As a result a component-based modeling ontology along with related component-based modeling knowledge base are considered.

To perform computational experiments, as an illustrative example of a complex model for testing proposed web-based software system a well-known balance model for calculating changes in the hydrological and ecological characteristics of water object is utilized.

5. Acknowledgements

The present study was supported by the Ministry of Education and Science of the Russian Federation Project no. 121030500071-2 "Methods and technologies of a cloud-based service-oriented platform for collecting, storing and processing large volumes of multi-format interdisciplinary data and knowledge based upon the use of artificial intelligence, model-driven approach and machine learning", no. 121041300065-9 "Theoretical foundations, methods and high-performance algorithms for continuous and discrete optimization to support interdisciplinary research" and the Russian Foundation for Basic Research and the Government of the Irkutsk Region as part of a scientific project No. 20-47-380001.

6. References

- [1] T. Iwanaga, H.H. Wang, S.H. Hamilton, V. Grimm, T.E. Koralewski, A. Salado, S. Elsayah, 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).
- [2] 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.
 - [3] R. Moore, A. Hughes, Integrated environmental modelling: Achieving the vision, *Geological Society London Special Publications* 408 (2016) 17–34. doi:10.1144/SP408.12.
 - [4] O. A. Nikolaychuk, A. I. Pavlov, A. B. Stolbov, The software platform architecture for the component-oriented development of knowledge-based systems, in: *Proceedings of the 41st International Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO, IEEE, 2018*, pp. 1234–1239. doi:10.23919/MIPRO.2018.8400194.
 - [5] A. I. Pavlov, A. B. Stolbov, A. S. Dorofeev, The workflow component of the knowledge-based systems development platform, in: *Proceedings for 2nd Scientific-practical Workshop Information Technologies: Algorithms, Models, Systems, ITAMS*, volume 2463 of *CEUR Workshop Proceedings*, 2019, pp. 47–58.
 - [6] M. Elag, J. L. Goodall, An ontology for component-based models of water resource systems, *Water Resour. Res.* 49 (2013). doi:5077–5091.10.1002/wrcr.20401.
 - [7] Dunlap, L. Mark, S. Rugaber, V. Balaji, J. Chastang, L. Cinquini, C. DeLuca, D. Middleton, S. Murphy, Earth system curator: Metadata infrastructure for climate modeling, *Earth Science Informatics* 1 (2008) 131–149. doi:10.1007/s12145-008-0016-1.
 - [8] O. A. Nevzorova, N. Zhiltsov, A. Kirillovich, E. Lipachev, OntoMathPRO Ontology: A Linked Data Hub for Mathematics, *Communications in Computer and Information Science* 468 (2014) 105–119. doi:10.1007/978-3-319-11716-4_9.
 - [9] R. Raskin, M. Pan, Knowledge representation in the Semantic Web for Earth and Environmental Terminology (SWEET), *Computers & Geosciences* 31 (2005) 1119–1125. doi:10.1016/j.cageo.2004.12.004.
 - [10] J. Syvitski, E. Hutton, S. Peckham, R. Slingerland, CSDMS - A modeling system to aid sedimentary research, *Sedimentary Geology* 9 (2011) 4–9. doi:10.2110/sedred.2011.1.4.
 - [11] V. N. Mihajlov, *Hydrology of river mouths*, Izd-vo MGU, Moscow, 1998 (in Russian).

Mitigating Uncertainty of Resource Allocation in Heterogeneous Computing Environment

Alexander Feoktistov¹, Andrei Tchernykh^{2,3,4} и Vladimir Dmitriev⁵

¹ Matrosov Institute for System Dynamics and Control Theory of SB RAS, Lermontov St. 134, Irkutsk, Russia, 664033

² CICESE Research Center, Ensenada-Tijuana Highway, Ensenada, Mexico, 22860, 3918

³ Ivannikov Institute for System Programming of RAS, Alexander Solzhenitsyn st. 25, Moscow, Russia, 109004

⁴ South Ural State University, Lenin prospect 76, Chelyabinsk, Russia, 454080

⁵ Irkutsk State University, Karl Marx St. 1, Irkutsk, Russia, 664003

Abstract

In the paper, we propose an approach to mitigating uncertainty in the correspondence of computational tasks and resources of a heterogeneous environment. The aforementioned uncertainty has a negative impact on such objectives of distributed computing as the problem-solving time, the reliability of computations, and the efficiency of resource use. The results of the experimental analysis demonstrate the advantages of using the proposed approach to improve above listed parameters due to a classification of computational jobs.

Key words

Distributed computing, heterogeneous environment, job classification, mitigating uncertainty

1. Introduction

Nowadays, job scheduling and resource management are among the challenges in providing distributed computing in heterogeneous environments [1]. When allocating resources to execute jobs, the following questions arise:

- What properties does a job have?
- What are the computational characteristics of the resource?
- What resource is more matching for a rational job execution?

With respect to the aforementioned questions, there is a need in studying job properties and determining their correlation with the resource characteristics. In this regard, one of the advanced approaches for allocating resources in such a way as to ensure their efficiency and rationally execute jobs is the job classification [2]. Therefore, let us consider some recent approaches to job classification-based scheduling that improve the results of traditional strategies of resource allocation such as the computer resource requirements-based, QoS-based, evolutionary algorithms-based ones, etc.

Zuo et al. [3] propose a job classification-based method of computation scheduling in a cloud environment. The proposed job classification is carried out by taking into account the intensity in the use of various components of a computational resource by jobs. The following three classes of jobs are distinguished: jobs with CPU intensive use, jobs with the I/O intensive use, and jobs with the memory intensive use.

The greedy-based algorithm for cloud computing scheduling was proposed in [4]. Within the proposed algorithm, jobs are classified based on QoS with respect to the following two criteria: job completion time and bandwidth. Corresponding to the jobs classes, the most suitable branch of the algorithm is selected, the required evaluations are computed, and the obtained results are used as

⁴th International Workshop on Information, Computation, and Control Systems for Distributed Environments (ICCS-DE 2022), July 04–08, 2022, Irkutsk, Russia

EMAIL: agf@icc.ru (Alexander Feoktistov); chernykh@cicese.mx (Andrei Tchernykh); dvi7@yandex.ru (Vladimir Dmitriev)

ORCID: 0000-0002-9127-6162 (Alexander Feoktistov); 0000-0001-5029-5212 (Andrei Tchernykh)



2022 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

ICCS-DE 2022 Workshop Proceedings

feedback with the scheduler. The proposed algorithm allows the scheduler to decrease the job completion time and increases user satisfaction.

The fulfillment results of evolutionary algorithms can also be improved by applying additional task classification. User characteristic-based analysis is applied by Kaur et al. [5] for allocating the right resources to execute submitted jobs through the genetic algorithm. Such a classification was useful in improving the scheduling efficiency and load balancing of cloud resources.

Examples of improving scheduling results in other well-known strategies through job classification are shown in [6].

A common problem of the considered approaches to job classification within computation scheduling is the selection of a limited number of predefined job characteristics. This leads to the effective applying these approaches only in the specific cases for which they were developed. In this regard, we proposed an approach to a job classification that gets rid of this limitation.

The rest of the paper is organized as follows. Section 2 provides a brief description of techniques used for evaluating workflow makespan. The main aspects of the job classification are represented in Section 3. In Section 4, we give an illustrative example to show the essence of our approach to mitigating uncertainty and the results in its application. Finally, the last section concludes our study.

2. Evaluating program execution time

Program execution time is one of the key factors in resource allocation. Depending on the available information about computing process, various methods for predicting program execution time are used. We consider jobs that are formed in distributed applied software packages developed in the Orlando Tools framework [7]. A job represent the execution specification for one of programs (modules) of scientific workflow.

Workflow management systems (Pegasus [8], Taverna [9], etc.) similar Orlando Tools often implement only one or two methods for predicting the module execution time. In comparison with such systems, Orlando Tools provides a large set of capabilities for determining the module execution time. In particular, it supports the following evaluations of the module runtime:

- User's evaluation,
- Dynamic analysis-based predictive evaluation,
- Test-based predictive evaluation within a continuous integration of applied software,
- Evaluation based on the statistical information of job executions.

Methods for predicting the module execution time used in Orlando Tools are discussed in detail in [10].

3. Job classification

We define the finite set $H = \{h_1, h_2, \dots, h_k\}$ of job characteristics. For each characteristic, the following attributes are set:

- Domain D_i of values for i th characteristic,
- Integer rank $r_i \geq 1$ indicating the importance extent for i th characteristic,
- Weight $w_i \geq 0$ reflecting a numerical expression of the importance for i th characteristic.

If a domain of values for the i th characteristic is not defined, then $D_i = \{\theta\}$, where θ is a symbol of an uncertainty.

We use the set H to form the finite set $C = \{c_1, c_2, \dots, c_m\}$ of job classes. Each class is defined by the following two sets of characteristics: basic (mandatory) set and additional (optional) set.

We use the Boolean matrices A and B of dimension $k \times m$ to determine the characteristics of the basic and additional sets of class characteristics. The element $a_{ij} = 1$ ($a_{ij} = 0$) of the matrix A means that the i th characteristic belongs (does not belong) to the basic set of characteristics of the j th class. Likewise, the element $b_{ij} = 1$ ($b_{ij} = 0$) of the matrix B means that the i th characteristic belongs (does not belong) to the additional set of characteristics of the j th class. Each i th characteristic included in the basic or additional sets of the j th class has the specified domain $D_{ij}^* \subseteq D_i \setminus \{\theta\}$ of valid values for this

class. Note that if $a_{ij} \vee b_{ij} = 0$, then $D_{ij}^* \equiv \{\emptyset\}$. Matrices A and B must satisfy the following constraints:

$$\begin{aligned} \bigvee_{j=1}^m \bigwedge_{i=1}^k \bar{a}_{ij} &= 0, \\ \bigvee_{i=1}^k \bigvee_{j=1}^m (a_{ij} \wedge b_{ij}) &= 0. \end{aligned}$$

A job is represented by the Boolean vector x of dimension k . A one-to-one correspondence is established between the element indices of the vector x and the indices of characteristics from H . The value of the i th element is determined as follows:

$$x_i = \begin{cases} 0, & \text{if } D_i' \equiv \{\emptyset\}, \\ 1, & \text{otherwise,} \end{cases}$$

where D_i' is the domain of the i th characteristics required in the job. The vector x must satisfy the constraint:

$$\bigwedge_{i=1}^k \bar{x}_i = 0.$$

Within a primary classification, we use the following characteristic function to determine the accordance between the characteristic domains given in the job and domains of characteristic required in the j th class:

$$\chi_j = \begin{cases} 0, & \text{if } \exists i: (a_{ij} \vee b_{ij} = 1) \wedge (D_i' \cap \bar{D}_{ij}^* \neq \emptyset), \\ 1, & \text{otherwise,} \end{cases}$$

where $i \in \overline{1, k}$, $j \in \overline{1, m}$. The results of applying the primary classification of the job are represented by the Boolean vector y as follows:

$$y_j = \chi_j, j \in \overline{1, m}.$$

However, as the result of the primary classification, the job can be correlated with several classes at once. Therefore, in order to further mitigate the uncertainty, a secondary classification using additional knowledge is required. To this end, we define the additional characteristic functions of the secondary classification as follows:

$$\chi_j^k(x, y, z) = \begin{cases} 0, & \text{if } \max_{\forall l: y_l = 1} \{f_l^k(x, z)\} - f_j^k(x, z) > e_k, \\ 1, & \text{otherwise,} \end{cases}$$

where $f_j^k(x)$ is the k th function intended to evaluate the extent of belonging of the job to the i th class, e_k is an upper limit on the equivalence for the function values, z is an additional information, $k = 1, 2, \dots$, and $l \in \overline{1, m}$. For example, the functions $f_j^k(x)$ can take into account the following knowledge: the number of optional characteristics matching the class, an aggregated rank of characteristics, a summary weight of characteristics, and statistical information about job executions.

Often different resources may be suitable for executing the same job, but with a different efficiency. Based on experience in managing the computing environment, resource administrators can pre-assign resources to execute the most appropriate classes of jobs, thereby mitigating the uncertainty in matching resources to jobs. In many cases, this can significantly increase the efficiency of resource use and improve the quality criteria for executing jobs. We use the Boolean matrix R of dimension $t \times k$ to determine the relations between resources and job classes, where t is a number of resources. The element $r_{ij} = 1$ ($r_{ij} = 0$) of the matrix R means that the i th resource matches (does not matches) to the j th class.

4. Example of mitigating uncertainty

Within estimating the extent of mitigating uncertainty of the resource allocation, we use the information entropy also named Shannon entropy. In information theory, the entropy of some random variable (for example, a resource's match for a specific job) reflects the average level of uncertainty

inherent to the possible outcomes of this variable. We consider the information entropy in the following way assuming that the decrease in entropy provides mitigating uncertainty of the resource allocation:

$$H(v) = \sum_{i=1}^n p_i(v)g(v),$$

where v is a random variable, n is a number of variable outcomes, $p_i(v) \in [0,1]$ is the probability of the i th outcome ($\sum_{i=1}^n p_i(v) = 1$), and $g(v)$ is calculated as follows:

$$g(v) = \begin{cases} 0, & \text{if } p_i(v) = 0, \\ -\log_2 p_i(v) & \text{otherwise.} \end{cases}$$

Let us consider the environment that includes three resources (HPC clusters) with different computational characteristics. The environment administrator has defined eight job classes. These classes have different basic sets of job characteristics and one optional characteristic, which represents the required job execution time. The matching of resources to job classes, based on administrator expertise, is reflected in the matrix

$$R = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

A job arrives in the environment. In this job, all mandatory characteristics are specified. The optional characteristic has not been given. We consider two random variables: a job's match for defined classes (v_1) and a resource's match for a specific job (v_2). For these variables, $p_1(v_1)$ – $p_8(v_1)$, $p_1(v_2)$ – $p_3(v_2)$, $H(v_1)$, and $H(v_2)$ are calculated in mitigating uncertainty of the resource allocation.

Table 1 demonstrates the results in mitigating uncertainty. Its first row shows the initial probability of the v_1 and v_2 outcomes, as well as corresponding them $H(v_1)$ and $H(v_2)$. The next three rows of Table 1 show the results of sequentially applying primary classification, secondary classification using computational history, and secondary classification using the results of the module testing. Each stage of the classification is accompanied by clarifying the matching of resources to job classes using the matrix R . In this example, we have achieved mitigating uncertainty for $H(v_1)$ from 3.00 to 0.34 and for $H(v_2)$ from 1.06 to 0. Decreasing in $H(v_1)$ and $H(v_2)$ is shown in Figure 1.

Table 1
The results in mitigating uncertainty

$p_1(v_1)$	$p_2(v_1)$	$p_3(v_1)$	$p_4(v_1)$	$p_5(v_1)$	$p_6(v_1)$	$p_7(v_1)$	$p_8(v_1)$	$p_1(v_2)$	$p_2(v_2)$	$p_3(v_2)$	c	$H(v_2)$
0.125	0.125	0.125	0.125	0.125	0.125	0.125	0.125	0.333	0.333	0.333	3.00	1.06
0.250	0.250	0.250	0.250	0	0	0	0	0.444	0.556	0	2.00	0.99
0.115	0.741	0.132	0.012	0	0	0	0	0.988	0.012	0	1.14	0.09
0	0.936	0.064	0	0	0	0	0	1.000	0	0	0.34	0

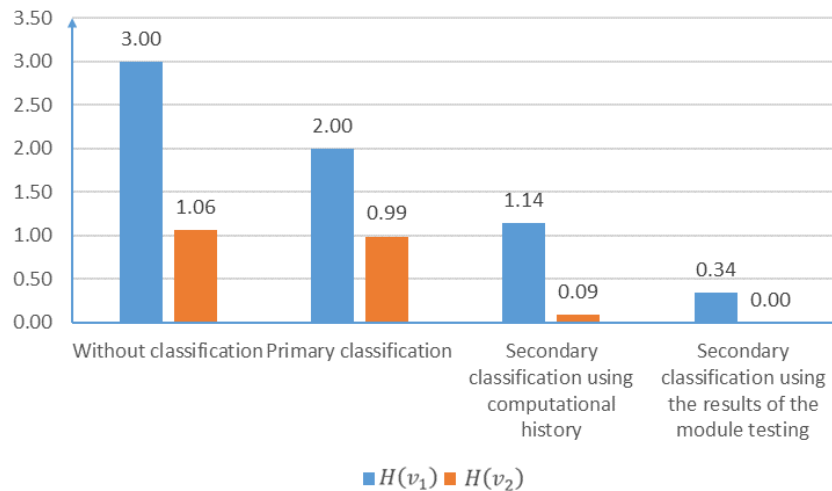


Figure 1: Decreasing in $H(v_1)$ and $H(v_2)$

5. Conclusions

The presence of uncertainty in scheduling distributed computing is a persistent challenge. Therefore, the study of this uncertainty in its various forms is an important direction of research.

Our contribution in this field of research lies in developing a heuristic approach for mitigating uncertainty in the matching of computational resources and jobs, which are launching in the heterogeneous distributed computing environment. The proposed approach is based on job classification using a set of characteristic functions. In comparison with other approaches to a study of such uncertainty, we use information entropy to estimate the uncertainty mitigation extent.

An illustrative example shows the essence of our approach and some useful results in its application.

Acknowledgments

The study was supported by the Ministry of Science and Higher Education of the Russian Federation, project no. FWEW-2021-0005 «Technologies for the development and analysis of subject-oriented intelligent group control systems in non-deterministic distributed environments» (registration no. 121032400051-9).

References

- [1] M. K. Mishra, Y. S. Patel, M. Ghosh, G. B. Mund, A review and classification of grid computing systems, *International Journal of Computational Intelligence Research* 13 (2017) 369–402.
- [2] B. K. Dhanalakshmi, K. C. Srikantaiah, K. R. Venugopal, Dynamic Computation of Threshold Value for Classifying Jobs in Cloud Computing for Efficient Resource Utilization, *Journal of Computational and Theoretical Nanoscience* 17 (2020) 4458–4461. doi:10.1166/jctn.2020.9097.
- [3] L. Zuo, S. Dong, L. Shu, C. Zhu, G. Han, A multiqueue interlacing peak scheduling method based on tasks' classification in cloud computing, *IEEE Systems Journal* 12 (2016) 1518–1530. doi:10.1109/JSYST.2016.2542251.
- [4] J. Li, L. Feng, S. Fang, An greedy-based job scheduling algorithm in cloud computing, *Journal of Software* 9 (2014) 921–925. doi:10.4304/jsw.9.4.921-925.
- [5] K. Kaur, N. Kaur, K. Kaur, A novel context and load-aware family genetic algorithm based task scheduling in cloud computing, *Advances in Intelligent Systems and Computing* 542 (2018) 521–531. doi:10.1007/978-981-10-3223-3_51.
- [6] S. Rajasoundaran, A. V. Prabu, S. Routray, S. S. Kumar, P. P. Malla, S. Maloji, A. Mukherjee, U. Ghosh, Machine learning based deep job exploration and secure transactions in virtual private cloud systems, *Computers and Security* 109 (2021) 102379. doi:10.1016/j.cose.2021.102379.
- [7] A. Feoktistov, S. Gorsky, I. Sidorov, I. Bychkov, A. Tchernykh, A. Edelev, Collaborative Development and Use of Scientific Applications in Orlando Tools: Integration, Delivery, and Deployment, *Communications in Computer and Information Science* 1087 (2020) 18–32. doi:10.1007/978-3-030-41005-6_2.
- [8] E. Deelman, R. F. da Silva, K. Vahi, M. Rynge, R. Mayani, R. Tanaka, W. Whitcup, M. Livny, The Pegasus workflow management system: translational computer science in practice, *Journal of Computational Science* 52 (2021) 101200. doi:10.1016/j.jocs.2020.101200.
- [9] K. Damkhang, P. Tandayya, Middleware for running and debugging Taverna workflows utilising RESTful web services, *International Journal of Simulation and Process Modelling* 15 (2020) 546–561. doi:10.1504/IJSPM.2020.112464.
- [10] Feoktistov A.G., Basharina O.Yu. Predicting runtime of computational jobs in distributed computing environment, in *Proceedings of the 2nd International Workshop on Information, Computation, and Control Systems for Distributed Environments (ICCS-DE 2022)*, CEUR-WS Proceedings, 2020, 2638 (2020), pp. 109–117. doi:10.47350/ICCS-DE.2020.10.

Towards a General Framework for Studying Resource Management in Large Scale Distributed Systems

Oleg Sukhoroslov^{1,2}, Maxim Gorokhovskii², Roman Ilgovskiy², Tagir Kuskarov², Yury Semenov³ and Andrey Vetrov²

¹*Institute for Information Transmission Problems of the Russian Academy of Sciences, 19 build.1 Bolshoy Karetny per., Moscow, 127051, Russian Federation*

²*HSE University, 11 Pokrovsky Bulvar, Moscow, 109028, Russian Federation*

³*Moscow Institute of Physics and Technology, 9 Institutskiy per., Dolgoprudny, Moscow Region, 141701, Russian Federation*

Abstract

Resource management is a critical part of modern large-scale distributed systems and an active area of research. The diversity of systems, workloads and requirements have lead to many domain-specific problem statements, solutions and simulation tools proposed by researchers, which are hard to compare and reuse. In this paper, an overview of related resource management problems is made and a general simulation framework to support studying such problems is presented. The framework has modular architecture, and is built around the generic simulation core and a set of basic resource models, which can serve as a common foundation for development of domain-specific models, components and simulators. The framework prototype has been evaluated and proved to be flexible and adaptable to different problem domains, while achieving performance and scalability needed to support the modeling of modern large scale problems.

Keywords

distributed systems, resource management, scheduling, simulation, cloud computing, IaaS, FaaS, workflow

1. Introduction

Modern systems and applications are increasingly distributed. This is due to the growing computational and data processing demands, and the need to serve and connect large numbers of clients and resources while meeting high performance, availability and scalability requirements. Grid computing infrastructures, peer-to-peer networks, Internet-scale applications and cloud computing platforms are examples of modern large-scale distributed systems that can consist of millions of machines spread across many locations around the world, processing hundreds thousands requests per second from billions of clients.

Distributed computing technologies allow to flexibly aggregate the resources of individual machines into scalable high-performance computing infrastructures with required characteristics. Resource management, including resource allocation and task scheduling, is a critical part

^{4th} International Workshop on Information, Computation, and Control Systems for Distributed Environments (ICCS-DE 2022), July 04–08, 2022, Irkutsk, Russia

✉ sukhoroslov@iitp.ru (O. Sukhoroslov)

ORCID [0000-0003-2836-2524](https://orcid.org/0000-0003-2836-2524) (O. Sukhoroslov)



© 2022 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

ICCS-DE 2022 Workshop Proceedings (iccs-de.icc.ru)

of operating such large-scale infrastructures due to their significant financial and environmental impact. The energy used by the world's data centers has doubled over the past decade and is estimated to account for around 1% of worldwide electricity use [1]. Even a small reduction of resource fragmentation rate can lead to substantial cost savings for public cloud providers [2].

Resource management and scheduling in distributed systems is an active area of research aimed at optimization of resource utilization, application performance, energy consumption and other operational aspects of such systems. While this area of research is established decades ago and has accumulated many results, the constantly increasing system scales, emergence of new computing paradigms and technologies, and changing requirements introduce new problem statements and challenges. The diversity of systems, workloads and requirements have lead to many specific problem statements and solutions proposed by researchers, which are often hard to compare, reuse and combine.

The proposed resource management methods should be subject to thorough evaluation before using them in real systems. Due to the large scale of considered systems, it is generally not feasible or time-consuming and expensive to conduct experiments and evaluate the proposed methods in a full-scale system, even for system developers. Also, due to the client behavior, dynamicity and non-determinism of real environments, the experimental conditions and results are not reproducible, which is unsuitable for comparison of several methods. Building a copy of real system for running experiments is also economically infeasible. Therefore, the initial evaluation is usually performed by means of simulation. The quality of obtained results depends on the quality of used simulation models of the studied system and its workload. Thorough evaluation also implies running the studied methods on diverse sets of system configurations and workloads by conducting thousands of simulations. This and the large scale of considered systems places high demands on performance and scalability of the used simulation tools in order to obtain experimental results in a reasonable time.

Currently there is a lack of generally accepted frameworks for studying and evaluation of resource management methods, including ready-to-use simulation models and benchmarks. Researchers often had to implement and evaluate the proposed methods using their own purpose-built simulators and benchmarks. This complicates the comparison of methods proposed by different authors and reproduction of results. While there are some established simulation frameworks used by researchers, e.g. [3, 4, 5, 6], they are intended to be used as libraries and do not provide ready-to-use simulators. This has lead to development of many domain-specific and ad-hoc simulators based on these frameworks by different authors [7, 8, 9, 10, 11].

While the problem specialization is inevitable, we argue that it is possible to improve research efficiency by reducing the diversity of used tools and duplication of efforts. First, we propose to establish common benchmarks for method evaluation, including ready-to-use simulators, datasets, evaluation scenarios and result processing tools. A researcher needs only to implement a new method using the provided API, and then can immediately run and compare it with existing methods. This will reduce the researcher effort and the need to develop ad-hoc solutions. Second, we propose to systematically study the related problem statements, identify similar problems and generalize them where possible. This will provide a basis for unified framework and reuse of tools between researchers studying different application domains. We propose to use optimization modeling and simulation as foundations for this framework. The increasing scale of modern systems also requires the improvement of performance and scalability of current

simulation frameworks to tackle the new challenges.

This paper is a first step towards implementing this vision. Section 2 provides a broad overview of resource management problems in modern large-scale distributed systems. Section 3 presents the proposed general simulation framework for studying such kind of problems. Section 4 evaluates the framework implementation and compares it with state-of-the-art solutions. Section 5 concludes and outlines the future work.

2. Resource Management in Large Scale Distributed Systems

Resource management and scheduling in distributed systems is an active area of research since 1990s. In a general form, it considers how to efficiently execute a given *workload* on a given *system* subject to specific requirements and constraints in order to optimize the metrics such as application performance, resource utilization, cost, etc. Due to a variety of existing workloads and systems, there are many domain-specific problem statements. In this section, we briefly characterize some of them relevant to the large-scale systems.

Large-scale distributed computing systems, such as *volunteer computing projects* [12, 13] and *grid infrastructures* [14, 15], are aimed at aggregation of autonomous resources, such as desktops and compute centers, into large resource pools to support compute- and data-intensive research. Volunteer computing projects like Folding@Home can attract millions of volunteers and achieve peak performance surpassing the world's fastest supercomputers. Grid infrastructures like EGI federate resources of hundreds of compute centers amounting together millions of CPU cores and hundreds of petabytes of storage.

The main workload in these systems are *high-throughput computing jobs* [16] consisting of a large number of independent tasks. Due to heterogeneity and dynamic availability of resources, scheduling tasks in these systems is not a trivial problem. For example, in volunteer computing projects it is needed to minimize the total execution time of large bag-of-tasks experiments while avoiding slow and misbehaving machines. In grid computing infrastructures, due to their cooperative nature, it is important to optimize the average job throughput and resource utilization while ensuring fair resource sharing across user communities and respecting compute centers' policies. Other types of workloads executed in these systems include *workflows* [17, 18] consisting of tasks with data dependencies, *parallel jobs* consisting of tasks communicating with each other, or more complex *many-task applications* [19]. In general, such workload can be represented as a set of multi-task jobs. The subjects of study are job and task scheduling, resource allocation, load balancing etc.

Popular *internet-scale applications*, such as search engines, social networks and video streaming services, are aimed at serving a large number of clients while providing the desired quality of service in terms of response time, throughput, availability and consistency [20, 21]. To achieve these goals, service providers build and operate geo-distributed infrastructures consisting of many data centers across the globe [22]. The workloads running in these systems can be divided mainly into services and batch processing parts [23, 24].

Services workload is related to processing of requests from clients, where response time is critical, by a set of long-running backend processes. The challenges are how to select the number of processes and allocate resources to them, distribute requests and balance the load between

them such as to optimize resource utilization and operational costs while achieving the desired performance and fault-tolerance objectives [25]. The requests here are mostly independent, however in some cases dependencies may exist, e.g. in the form of transactions. Due to faults, which is a norm on a large scale, and fluctuating user load with sudden spikes, the used methods should be able to quickly adapt to such changes.

Batch workload is related to performing auxiliary data processing and computation jobs, such as building a search index or training a deep learning model. These jobs are usually split into many parallel tasks that are executed in background on dynamically allocated resources of clusters [23, 24]. The challenges on the jobs level are how to allocate resources between multiple jobs to optimize the cluster resource utilization while respecting job resource requirements, priorities and deadlines, providing fairness, etc. The challenges on the job level are how many tasks to use, how to schedule them and distribute processed data between them, how to avoid stragglers, recover from failures, etc.

Another more recently emerged type of data center workloads is *stream processing* which consists in near real-time parallel processing of large data streams by a set of continuously running processes [26, 27, 28]. The processes are organized into the acyclic directed graph, which edges correspond to data streams between the processes. The challenge is how to map the processes to available resources to optimize resource utilization while reaching the desired throughput and latency objectives.

The diversity of described data center workloads poses additional resource management challenges. Since operating separate resource pools for each workload leads to wasted resources, the same machines are used to run different workloads [23, 24]. The challenge is how to keep the desired performance guarantees for critical workloads while efficiently using the remaining resources for running other workloads.

Cloud computing has emerged as a new paradigm for on-demand access to a vast pool of computing resources providing a modern alternative to on-premises deployments [29]. Clouds can significantly lower time-to-solution via quick resource provision, skipping the lengthy process of building a new cluster on-premises or avoiding long queue wait times on shared computing facilities. By providing a wide range of possible virtual machine configurations, clouds allow to easily adapt to changing workloads. Clouds can also reduce the total cost of ownership by allowing dynamic auto-scaling of computing resources depending on the current load, or by using spot instances that represent excess cloud capacity. These advantages lead to migration of many workloads, including the described above, to public clouds. At the same time, the new paradigm has posed new resource management challenges, both from the point of view of cloud provider and user.

Public cloud providers operate many virtualized data centers worldwide, arranged in a hierarchy of regions and availability zones. In Azure, larger zones have over a hundred thousand machines, spread over more than a hundred clusters [2]. A cloud provider offers computing resources in the form of virtual resources (machines, disks, networks) on a pay-per-usage basis. This forms the core Infrastructure as a Service (IaaS) layer upon which all higher cloud layers (PaaS, SaaS, FaaS) are built. Therefore, allocation of resources of physical machines (PMs) to virtual machines (VMs) is a crucial aspect of cloud resource management [2].

VM allocation brings several resource management challenges. First, the initial assignment of VM request to PM should be computed quickly in online fashion, i.e. without knowing the next

requests, while avoiding the resource fragmentation. It should also take into account customer requirements in the form of VM placement constraints, e.g. spread VMs across different racks to avoid correlated failures. Second, migration of VMs between PMs is required to avoid hot spots, reduce the number of active PMs and perform PM maintenance [30]. Third, as VMs often do not utilize all allocated resources, such resources as CPU are oversubscribed to avoid their underutilization [31]. However, since VM resource demand can be highly dynamic, static allocation can lead to resource contention and VM performance degradation. Therefore, dynamic allocation methods are needed that adapt to changes in VM load, control allocations and migrate VMs from overloaded PMs. Since VMs can run various customer workloads unknown to the cloud provider, predicting the future VM load and the impact of oversubscription is much more challenging [32].

Cloud computing also brings new resource management challenges from the perspective of a cloud user related to careful selection and dynamic management of allocated resources according to specific workload needs and objectives. First, the diversity of VM types leads to the problem of finding the optimal resource configuration for running a particular workload, e.g. HPC job or web application [33]. Depending on the situation, the objective can be the cost of running workload or the execution time, subject to additional constraints like performance, deadline or budget. Second, for workloads with dynamic resource needs it is essential to leverage the cloud elasticity by dynamically allocating and releasing resources in order to reduce costs. This requires additional methods, such as horizontal and vertical auto-scaling strategies [34].

Serverless computing model [35, 36] enables the execution of user workloads in a cloud without explicit resource allocation by the user, as it is done previously with VMs. The workload is described in the form of *functions*, which code is executed on demand on resources dynamically allocated by the cloud provider. While this approach is currently applicable to a limited set of workloads, it removes a burden of managing resource allocations from the cloud users. However, the related challenge is just shifted to the provider, which is now responsible for dynamic resource allocation of resources for running user functions [37]. The objective is to minimize the related resource overhead, e.g. time the function code is idle when being loaded in PM memory, while meeting the desired invocation latency and performance targets. In contrast to the user-side problem, the provider has less information about the function workload, which also brings additional complexity.

The complexity and diversity of modern applications and workloads, the shift towards resource virtualization and the mentioned resource management challenges have lead to development of new platforms, such as OpenStack and Kubernetes. These platforms play a role of operating systems for distributed resource pools and provide resource management components that address some of the mentioned challenges. For example, OpenStack includes a VM scheduler for placement of VMs, while Kubernetes includes a scheduler for management of pods, groups of containers. While these platforms support sophisticated resource management policies, their core algorithms are simple and has a potential for improvement [38]. Besides that, existing solutions are often designed for specific types of workload, which complicates the use of unified resource pools for running heterogeneous workloads.

The described diversity of systems, workloads and requirements have lead to many specific problem statements and solutions proposed by researchers, which are hard to compare, reuse and combine. This and the trend towards running diverse workloads across large distributed

resource pools requires describing and generalizing these problems within a common framework. We propose to use optimization modeling and simulation as foundations for such framework. First, all mentioned problems can be formulated as optimization problems, and mathematical models can provide a clear and succinct form for their description and generalization. Second, as was explained in Section 1, simulation is an important tool for initial evaluation and comparison of proposed resource management methods. In this paper we focus on the simulation part, while the optimization modeling will be covered in future work.

3. Simulation Framework

This section describes the proposed simulation framework called *DSL*ab, which is intended to support the research of resource management in large-scale distributed systems. We describe the requirements to such framework, its architecture and current implementation.

3.1. Requirements

From the overview of resource management problems in Section 2 we can identify several important requirements that should be met by the proposed simulation framework.

First, in order to support the modeling of diverse system configurations and workloads, it should be generic, modular and flexible, with ability to extend and adapt it to various domains and use cases. That means that the framework should be built as a set of loosely-coupled modules around the generic core and application programming interfaces.

Second, it should be performant and scalable in order to support modeling of modern large-scale systems consisting of up to millions of entities and generating millions of events per second. It should also be resource efficient to support simulating such systems on an average personal computer, and should facilitate parallel execution of multiple simulations for conducting large experiments and parameter studies.

Third, the framework should support the modeling of basic system resources (compute, storage and network) with adequate levels of accuracy. For considered problems, it is generally not needed to model the CPU or network usage on the level of instructions or packets. However, it can be important to realistically model the resource sharing, contention, performance degradation and other similar effects.

Finally, the framework should be easy to use and extend. This includes the ability to precisely control simulation and create custom scenarios, deterministic execution for reproducibility, support for logging and visualization, ability to implement own extensions or methods with minimal effort, support for description and import of system configurations and workloads in common formats, and a rich set of examples.

3.2. Architecture

The developed DSLab framework has modular architecture presented on Figure 1. It consists of three layers. *Simulation core* layer includes the generic discrete-event simulation engine and application programming interface (API) for interaction with it. *Resource models* layer includes the extensible set of ready-to-use models of basic system resources, such as compute, storage

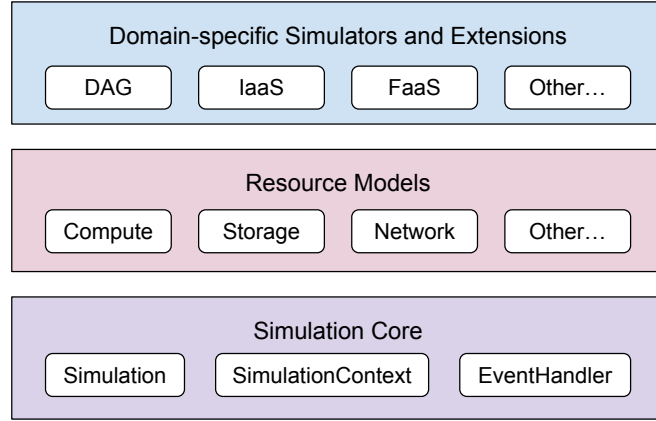


Figure 1: Architecture of DSLab framework.

and network. *Simulators and extensions* layer includes the extensible set of domain-specific models, components and complete solutions, such as ready-to-use simulators, built upon the previous two layers.

Despite this layered logical architecture, the framework is actually developed as a set of loosely-coupled modules. The only mandatory dependence is the simulation core, which is small and easy to understand. Therefore a user can choose how deep to dive into the framework and which modules to use for particular simulation needs. Note also that the framework core is not specifically tied to resource management and can be used in other research and educational domains.

The framework prototype has been implemented in Rust programming language. The choice of Rust was motivated by performance requirements, rich set of modern features and vibrant community. While being as efficient as C++, Rust also provides memory safety guarantees and avoids related bugs. On the downside, Rust is not so widely popular among the researchers as C++, Java or Python, and has a rather steep learning curve. We plan to address this issue by supporting integration with components written in other languages in the future. Below we provide an overview of current implementations of the mentioned layers.

3.3. Simulation Core

The simulation core is implemented as a compact library providing a generic discrete-event simulation engine. This engine can be used to implement arbitrary deterministic simulations consisting of user-defined *components* producing and consuming *events*. The engine manages simulation state, which includes clock, event queue and random number generator. The latter is initialized with user-defined seed to ensure deterministic execution and reproduction of results.

The simulation is configured and managed via the *Simulation* interface, which includes methods for registering simulation components, stepping through the simulation, obtaining the current simulation time, etc. It is possible to use any user-defined Rust types as simulation components, and future work will address integration with other languages. The components access simulation state and produce events via the *SimulationContext* interface. Each component

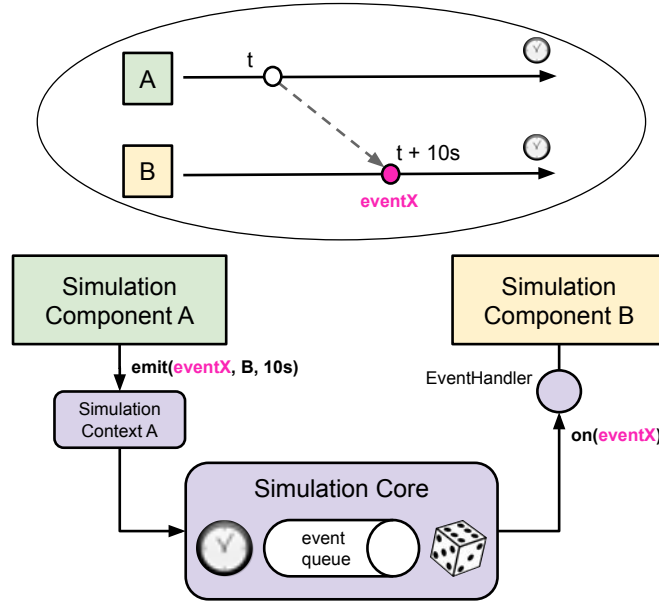


Figure 2: Event-driven simulation in DSLab.

typically uses a unique simulation context, which allows to differentiate events produced by different components. To be able to consume events, the component should implement the *EventHandler* interface, which is invoked by the engine to pass events to the component. Each simulation component is registered with unique name and identifier, which can be used for specifying the event source or destination, logging purposes, etc.

The simulation is implemented as a sequence of events (see Figure 2). Each event has a unique identifier, timestamp, source, destination and user-defined payload. The simulation engine supports using arbitrary serializable types as event payloads, the structure of payload is opaque to the engine. The events are processed in their timestamp order by retrieving the next event from the event queue, advancing the simulation clock to event time and invoking the *EventHandler* of component specified as the event destination. When processing the event, the component can create and emit new events with arbitrary future timestamps via its *SimulationContext*. It is also possible to cancel the previously emitted events before they are processed, e.g. to change the completion time of some pending activities.

The simulation core also provides convenient facilities for logging of events or arbitrary messages during the simulation with inclusion of component names, logging levels, etc.

The described simulation core functionality proved to be sufficient for currently implemented use cases. The simulation execution is performed sequentially using a single thread. While using multiple threads can potentially speed up simulation, the efficient parallel implementation of discrete-event simulation is a challenging problem. Since in practice researchers often need to run a large number of simulations, it is more reasonable to run many independent single-threaded simulations in parallel than to try to parallelize a single simulation. The performance of current implementation of simulation core is evaluated in Section 4.

3.4. Resource Models

This layer should provide a set of ready-to-use models of basic system resources, which are needed for simulating the considered distributed systems. The current prototype includes several implementations of generic compute, storage and network components, which can be used in simulations.

The *Compute* component represents an abstract model of computing resource, such as a single machine or a cluster, which can perform computations represented as compute tasks. The resource is characterized by the provided number of CPU cores, their speed in flop/s and amount of memory. The compute task is characterized by the amount of computations in flops, number of used cores, and amount of used memory. Two models of compute resource are currently implemented.

The *singlecore* model implements a resource with single core, which resources are evenly shared between the currently running tasks. Each time some task is completed or a new task is submitted, the resource shares and completion times of all running tasks should be updated. A naive method of iterating over all tasks and updating their completion times on each such event can lead to a poor performance. An efficient method is implemented which supports addition of a new task and removal of the task with the smallest completion time in $O(\log n)$, where n is the number of concurrent tasks. The same method is used for modeling equal bandwidth sharing in storage and network models.

The *multicore* model implements a resource with multiple cores, which supports execution of parallel tasks. In this model, the compute task can specify the minimum and maximum number of used cores, and provide a function which defines the dependence of parallel speedup on the number of used cores. The cores allocation for each task is computed upon task arrival and, in contrast to previous model, is not changed during the task execution. This model also supports manual allocation and release of cores and memory, which can be used to model other use cases.

The *Storage* component represents an abstract model of raw data storage, such as disk, which can perform data read and write operations. The storage resource is characterized by its capacity, read and write bandwidth. The operation is characterized by its type (read or write) and the amount of data. The implemented *disk* model supports computing completion times of operations based on a user-defined *bandwidth model*, a function which returns the bandwidth for the given data size to be read or written. Three models are included currently: constant, random with given distribution and empirical with given weighted values set. In addition, there is support for modeling bandwidth sharing and degradation. The *file_system* model implemented on top of the *disk* model supports modeling storage system consisting of multiple disks on the level of file system operations, such as file read and write.

Modeling of network communication is essential for simulation of distributed systems and applications. The study of considered resource management and scheduling problems does not generally require the use of sophisticated packet-level network models. Moreover, such fine-grained models are prohibitively expensive on large system and time scales. Therefore, the cheaper models that simulate only the aspects of network behavior important to the studied problem are used. Since these requirements may depend on the use case, the network modeling is not embedded in the simulation core and instead is implemented on this layer.

In general, the network can be characterized by its topology (link graph), including the

latency and bandwidth of individual links. The network communication is characterized by the data size, source and destination. In cases where communications correspond to small messages, it is sufficient to model only the network latency. This does not require the use of dedicated network model, as the simulation core directly supports emitting events with arbitrary delays. However, in cases that involve large data transfers it is necessary to model the network bandwidth. Depending on whether the network contention can occur or not, it can also be necessary to model the bandwidth sharing between the contending data transfers. Finally, the network topology modeling needs can range from a single backbone link to a complex link graph of a wide-area network.

Currently two network models are implemented that cover the basic modeling requirements. The network topology corresponds to a single backbone link with specified latency and bandwidth. The simulation components are assigned to the network hosts, and the network is bypassed for communications between components on the same host. The models support message passing and data transfer operations. The message passing operation corresponds to emitting an event with delay equal to the network latency between the event source and destination. The data transfer operation corresponds to transferring the specified amount of data between the specified source and destination, with notification of the specified component after the transfer is completed. The *constant_bandwidth* model assumes that all data transfer operations use the full link bandwidth. The *shared_bandwidth* model implements dynamic equal sharing of link bandwidth between concurrent data transfers. The implementation of support for modeling arbitrary network topologies is in progress.

3.5. Domain-specific Simulators and Extensions

The described simulation core and resource models can serve as a common foundation for development of domain-specific models, components and complete simulators for different use cases. To verify this assumption, test the flexibility of the current prototype and provide examples of its use, the following simulators and extensions have been implemented.

The *DAG* extension is dedicated for studying the scheduling of workloads represented as directed acyclic graphs (DAG), such as scientific workflows or data-parallel jobs. It includes the model of heterogeneous distributed computing system comprised of several resources, the model of DAG job consisting of multiple tasks with data dependencies, the runner which executes a DAG job using the specified scheduling algorithm, the interface for implementation of such algorithms and several ready-to-use implementations. The extension also provides the tools for bootstrapping simulation, importing DAGs in common formats and visualization of DAG execution traces. The extension was successfully used to reproduce results of experiments from [39].

The *IaaS* extension implements a model of cloud computing infrastructure for studying the cloud resource management on Infrastructure as a Service (IaaS) level, such as virtual machine placement and migration. It includes the models of basic cloud components such as physical machine (host), virtual machine, VM scheduler, placement store, monitoring service, etc. The extension allows to provide user-defined models for host power consumption and VM load, and implement custom VM placement and migration algorithms. The unique feature of this extension is support for modeling of multiple concurrent VM schedulers operating on local,

asynchronously updated copies of resource pool state.

The *FaaS* extension is aimed at modeling the Function as a Service (FaaS) layer of clouds, including the scheduling and execution of cloud functions in dynamically deployed containers. It includes the models of serverless application, cloud function, container, host, container manager, function registry, invocation scheduler, etc. The extension allows to implement and test custom container management policies, function invocation scheduling and container deployment strategies. The extension was successfully used to reproduce the results of experiments from [37].

4. Evaluation

The evaluation of the developed simulation framework can be performed based on the formulated requirements. As presented in Section 3, the DSLab framework is modular and is built around the generic simulation core, it supports the modeling of basic system resources on adequate level and is proved to be flexible and adaptable to different use cases. Therefore, this section focuses on evaluating the performance and scalability of the framework required for modeling the large-scale systems. The detailed evaluation of the accuracy of implemented models will be made in a separate work.

The evaluation is based on implementation of several benchmarks and examples from different domains. In each case, the performance of implementation using DSLab is compared to implementation using one of the state-of-the-art solutions.

4.1. Ping-Pong

This example simulates the interaction between N processes. Each process communicates with P other processes, which are called process peers. On each iteration a process sends a *Ping* message to one of its peers and waits for reply from the peer. Upon receiving a *Ping* message, the process replies to the sender with the *Pong* message. Upon receiving a *Pong* from the peer, the process starts the next iteration, and so on, until it completes the specified number of iterations. This simple example allows to benchmark the performance and scalability of the simulation core by creating a large number of simulated entities and events.

Two implementations are created: one using DSLab, and the other using SimGrid [5] (version 3.31), a popular framework for developing simulators of distributed applications written in C++. Both implementations can be configured to use either a fixed message delay or a simple network model with no bandwidth sharing. The obtained simulation times for various numbers of processes and iterations are presented in Table 1. The developed framework is significantly (20-30 times) faster than SimGrid, being able to process up to 10 millions of events per second. DSLab also demonstrates better scalability and allows to simulate a system with million processes. The use of network model (bottom half of the table) noticeably impacts the performance and scalability of both implementations. We plan to investigate this issue in the future. Overall, the results without network model demonstrate that the developed simulation core provides high performance and scalability.

Table 1
Results for Ping-Pong benchmark

Processes	Peers	Iterations	Network Model	Simulation Time (s)	
				DSLab	SimGrid
2	1	1000000	N	0.17	5.27
2	1	10000000	N	1.61	52.29
1000	10	1000	N	0.35	5.93
10000	100	1000	N	4.27	159.81
100000	100	1000	N	82.88	-
1000000	100	1000	N	946.20	-
2	1	1000000	Y	0.26	7.19
2	1	10000000	Y	2.52	72.48s
1000	10	1000	Y	0.53	71.20
10000	100	1000	Y	8.49	7739.58
100000	100	1000	Y	253.94	-
1000000	100	1000	Y	4202.09	-

4.2. Master-Workers

This example simulates a typical distributed computing system consisting of heterogeneous hosts. A host provides computing resources in the form of CPU cores and memory for execution of compute tasks. Each host runs a *worker* process which manages execution of tasks on the host. The *master* process, which runs on one of the hosts, is responsible for accepting new tasks, scheduling tasks to workers and collecting task results. Each task has varying complexity, resource requirements, input and output data sizes. When a task is assigned to a worker, the worker downloads the task input data from the master to the local disk, executes the task, stores the result on the local disk and uploads it to the master. This more complex example uses several resource models (compute, storage, network) allowing to test their implementations and evaluate their impact on performance.

Two implementations are created: one using DSLab, and the other using SimGrid (version 3.31). Both implementations can be configured to use either a network model without or with bandwidth sharing. The obtained simulation times for various numbers of workers and tasks are presented in Table 2. Similar to the previous example, the implementation using DSLab significantly outperforms the one using SimGrid. The framework scalability allows to simulate a system with million workers. However, the use of more complex network model with bandwidth sharing (bottom half of the table) also noticeably impacts the simulation performance.

4.3. Scientific Workflow Execution

This example simulates the execution of scientific workflow in distributed computing system. Several realistic workflow instances with thousands of tasks are produced using the workflow generator from the WfCommons project. The system consists of 10 computing resources with 10 cores each.

Table 2
Results for Master-Workers example

Workers	Tasks	Bandwidth Sharing	Simulation Time (s)	
			DSLAb	SimGrid
100	10000	N	0.08	0.63
1000	10000	N	0.08	7.47
1000	100000	N	0.97	41.05
10000	1000000	N	16.31	14015.21
100000	1000000	N	21.53	-
1000000	1000000	N	26.11	-
1000000	10000000	N	320.32	-
100	10000	Y	0.13	0.60
1000	10000	Y	0.17	20.60
1000	100000	Y	5.97	335.53
10000	1000000	Y	977.40	-

Table 3
Results for Workflow Execution example

Workflow	Tasks	Simulation Time (s)		Memory (MB)	
		DSLAb	WRENCH	DSLAb	WRENCH
Montage	4991	0.20	9.08	16	459
Epigenomics	9995	0.44	12.46	27	649
Srsearch	9997	0.66	38.88	42	2444
Genome	9998	0.64	48.40	34	790

Two implementations are created: one using DSLAb with *DAG* extension, and the other using WRENCH [11] (version 2.0), a workflow management system simulation framework based on SimGrid. The obtained simulation times and memory consumption are presented in Table 3. The implementation using DSLAb is 30-70 times faster and consumes significantly less memory than the one using WRENCH. The simulated workflow execution times produced by our implementation and WRENCH differ only by 5-6 %. This difference can be explained by more detailed workflow execution modeling in WRENCH (including the workflow management system, storage access, etc.) and other simulation peculiarities. However, the observed performance difference cannot be explained only by the use of more detailed models.

4.4. Virtual Machine Placement

This example simulates the placement of virtual machines on the hosts of a cloud data center. The specified number of virtual machines with varying characteristics are created in the beginning of the simulation and passed to the cloud scheduler. The scheduler uses BestFit algorithm to assign virtual machines to hosts. The simulation proceeds until all virtual machines are

Table 4
Results for Virtual Machine Placement example

Hosts	VMs	Simulation Time (s)	
		DSLlab	CloudSim Plus
200	6000	0.11	8.07
2000	60000	18.11	472.61
5000	150000	123.51	2704.86
10000	300000	511.69	18625.51

Table 5
Results for Cloud Function Invocation example

Invocations	Simulation Time (s)	
	DSLlab	OpenDC
40000	0.109	4.965
400000	0.952	26.865
4000000	11.028	231.706
40000000	169.720	2410.613

scheduled and started on hosts.

Two implementations are created: one using DSLab with *IaaS* extension, and the other using CloudSim Plus [6] (version 7.2.0), a popular cloud computing simulation framework written in Java. The obtained simulation times for various numbers of hosts and VMs are presented in Table 4. The implementation based on DSLab is 20-70 faster. This can be partially explained by the performance difference between Rust and Java runtimes, and simulation peculiarities. However, as in the previous case, the performance overhead of the state-of-the-art framework is significant and can be prohibitive for simulating large scale cases.

4.5. Cloud Function Invocation

This example implements the trace-driven simulation of cloud function invocations in FaaS environment. The traces with different number of invocations of two functions are used. Two implementations are created: one using DSLab with *FaaS* extension, and the other using OpenDC [40] (version 2.0), a new platform for datacenter simulation. The obtained simulation times are presented in Table 5. Again, the implementation based on DSLab is significantly faster, though the gap is decreasing for the larger number of invocations.

5. Conclusion

Resource management is a critical part of operating modern large-scale distributed systems and an active area of research. The diversity of systems, workloads and requirements have lead to

many problem statements, solutions and simulation tools proposed by researchers, which are hard to compare and reuse. We propose to improve research efficiency by reducing the diversity of used tools and duplication of efforts. As a first step in this direction, a general simulation framework for studying such problems is developed.

The presented DSLab framework has modular architecture, built around the generic simulation core and a set of basic resource models, which can serve as a common foundation for development of domain-specific extensions. The framework prototype has been evaluated and proved to be flexible and adaptable to different problem domains, while achieving the performance and scalability levels needed to support the modeling of modern large scale problems. The framework along with the described examples is published as an open source project¹.

Future work will address the remaining issues, such as integration with other languages and model validation. It is also planned to investigate the other aspects of the proposed vision, such as generalized problem statements and the use of optimization modeling. Finally, we plan to further improve and extend DSLab, including investigation of its use in other research and educational domains.

Acknowledgments

This work is supported by the Russian Science Foundation (project 22-21-00812).

References

- [1] E. Masanet, A. Shehabi, N. Lei, S. Smith, J. Koomey, Recalibrating global data center energy-use estimates, *Science* 367 (2020) 984–986.
- [2] O. Hadary, L. Marshall, I. Menache, A. Pan, E. E. Greeff, D. Dion, S. Dorminey, S. Joshi, Y. Chen, M. Russinovich, et al., Protean:{VM} allocation service at scale, in: 14th USENIX Symposium on Operating Systems Design and Implementation (OSDI 20), 2020, pp. 845–861.
- [3] R. Buyya, M. Murshed, Gridsim: A toolkit for the modeling and simulation of distributed resource management and scheduling for grid computing, *Concurrency and computation: practice and experience* 14 (2002) 1175–1220.
- [4] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. De Rose, R. Buyya, Cloudsim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms, *Software: Practice and experience* 41 (2011) 23–50.
- [5] H. Casanova, A. Giersch, A. Legrand, M. Quinson, F. Suter, Versatile, scalable, and accurate simulation of distributed applications and platforms, *Journal of Parallel and Distributed Computing* 74 (2014) 2899–2917.
- [6] M. C. Silva Filho, R. L. Oliveira, C. C. Monteiro, P. R. Inácio, M. M. Freire, Cloudsim plus: A cloud computing simulation framework pursuing software engineering principles for improved modularity, extensibility and correctness, in: 2017 IFIP/IEEE symposium on integrated network and service management (IM), IEEE, 2017, pp. 400–406.

¹<https://github.com/osukhoroslov/dslab>

- [7] S. K. Garg, R. Buyya, Networkcloudsim: Modelling parallel applications in cloud simulations, in: 2011 Fourth IEEE International Conference on Utility and Cloud Computing, IEEE, 2011, pp. 105–113.
- [8] W. Chen, E. Deelman, Workflowsim: A toolkit for simulating scientific workflows in distributed environments, in: 2012 IEEE 8th international conference on E-science, IEEE, 2012, pp. 1–8.
- [9] P.-F. Dutot, M. Mercier, M. Poquet, O. Richard, Batsim: a realistic language-independent resources and jobs management systems simulator, in: Job Scheduling Strategies for Parallel Processing, Springer, 2015, pp. 178–197.
- [10] Z. Cai, Q. Li, X. Li, Elasticsim: A toolkit for simulating workflows with cloud resource runtime auto-scaling and stochastic task execution times, Journal of Grid Computing 15 (2017) 257–272.
- [11] H. Casanova, R. F. da Silva, R. Tanaka, S. Pandey, G. Jethwani, W. Koch, S. Albrecht, J. Oeth, F. Suter, Developing accurate and scalable simulators of production workflow management systems with wrench, Future Generation Computer Systems 112 (2020) 162–175.
- [12] D. P. Anderson, J. Cobb, E. Korpela, M. Lebofsky, D. Werthimer, Seti@ home: an experiment in public-resource computing, Communications of the ACM 45 (2002) 56–61.
- [13] D. P. Anderson, G. Fedak, The computational and storage potential of volunteer computing, in: Sixth IEEE International Symposium on Cluster Computing and the Grid (CCGRID’06), volume 1, IEEE, 2006, pp. 73–80.
- [14] I. Foster, C. Kesselman, The Grid 2: Blueprint for a new computing infrastructure, Elsevier, 2003.
- [15] I. Bird, Computing for the large hadron collider, Annual Review of Nuclear and Particle Science 61 (2011) 99–118.
- [16] M. Livny, J. Basney, R. Raman, T. Tannenbaum, et al., Mechanisms for high throughput computing, SPEEDUP journal 11 (1997) 36–40.
- [17] I. J. Taylor, E. Deelman, D. B. Gannon, M. Shields, et al., Workflows for e-Science: scientific workflows for grids, volume 1, Springer, 2007.
- [18] E. Deelman, K. Vahi, G. Juve, M. Rynge, S. Callaghan, P. J. Maechling, R. Mayani, W. Chen, R. F. Da Silva, M. Livny, et al., Pegasus, a workflow management system for science automation, Future Generation Computer Systems 46 (2015) 17–35.
- [19] I. Raicu, I. T. Foster, Y. Zhao, Many-task computing for grids and supercomputers, in: 2008 workshop on many-task computing on grids and supercomputers, IEEE, 2008, pp. 1–11.
- [20] E. A. Brewer, Lessons from giant-scale services, IEEE Internet computing 5 (2001) 46–55.
- [21] J. R. Hamilton, et al., On designing and deploying internet-scale services., in: LISA, volume 18, 2007, pp. 1–18.
- [22] L. A. Barroso, U. Hölzle, The datacenter as a computer: An introduction to the design of warehouse-scale machines, Synthesis lectures on computer architecture 4 (2009) 1–108.
- [23] M. Schwarzkopf, A. Konwinski, M. Abd-El-Malek, J. Wilkes, Omega: flexible, scalable schedulers for large compute clusters, in: Proceedings of the 8th ACM European Conference on Computer Systems, 2013, pp. 351–364.
- [24] A. Verma, L. Pedrosa, M. Korupolu, D. Oppenheimer, E. Tune, J. Wilkes, Large-scale cluster management at google with borg, in: Proceedings of the Tenth European Conference on Computer Systems, 2015, pp. 1–17.

- [25] J. Dean, L. A. Barroso, The tail at scale, *Communications of the ACM* 56 (2013) 74–80.
- [26] T. Akidau, A. Balikov, K. Bekiroğlu, S. Chernyak, J. Haberman, R. Lax, S. McVeety, D. Mills, P. Nordstrom, S. Whittle, Millwheel: Fault-tolerant stream processing at internet scale, *Proceedings of the VLDB Endowment* 6 (2013) 1033–1044.
- [27] M. Zaharia, T. Das, H. Li, T. Hunter, S. Shenker, I. Stoica, Discretized streams: Fault-tolerant streaming computation at scale, in: *Proceedings of the twenty-fourth ACM symposium on operating systems principles*, 2013, pp. 423–438.
- [28] Y. Mei, L. Cheng, V. Talwar, M. Y. Levin, G. Jacques-Silva, N. Simha, A. Banerjee, B. Smith, T. Williamson, S. Yilmaz, et al., Turbine: Facebook’s service management platform for stream processing, in: *2020 IEEE 36th International Conference on Data Engineering (ICDE)*, IEEE, 2020, pp. 1591–1602.
- [29] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, et al., A view of cloud computing, *Communications of the ACM* 53 (2010) 50–58.
- [30] A. Ruprecht, D. Jones, D. Shiraev, G. Harmon, M. Spivak, M. Krebs, M. Baker-Harvey, T. Sanderson, Vm live migration at scale, *ACM SIGPLAN Notices* 53 (2018) 45–56.
- [31] S. A. Baset, L. Wang, C. Tang, Towards an understanding of oversubscription in cloud, in: *2nd USENIX Workshop on Hot Topics in Management of Internet, Cloud, and Enterprise Networks and Services (Hot-ICE 12)*, 2012.
- [32] E. Cortez, A. Bonde, A. Muzio, M. Russinovich, M. Fontoura, R. Bianchini, Resource central: Understanding and predicting workloads for improved resource management in large cloud platforms, in: *Proceedings of the 26th Symposium on Operating Systems Principles*, 2017, pp. 153–167.
- [33] N. J. Yadwadkar, B. Hariharan, J. E. Gonzalez, B. Smith, R. H. Katz, Selecting the best vm across multiple public clouds: A data-driven performance modeling approach, in: *Proceedings of the 2017 Symposium on Cloud Computing*, 2017, pp. 452–465.
- [34] T. Llorido-Botran, J. Miguel-Alonso, J. A. Lozano, A review of auto-scaling techniques for elastic applications in cloud environments, *Journal of grid computing* 12 (2014) 559–592.
- [35] I. Baldini, P. Castro, K. Chang, P. Cheng, S. Fink, V. Ishakian, N. Mitchell, V. Muthusamy, R. Rabbah, A. Slominski, et al., Serverless computing: Current trends and open problems, in: *Research advances in cloud computing*, Springer, 2017, pp. 1–20.
- [36] E. Jonas, J. Schleier-Smith, V. Sreekanti, C.-C. Tsai, A. Khandelwal, Q. Pu, V. Shankar, J. Carreira, K. Krauth, N. Yadwadkar, et al., Cloud programming simplified: A berkeley view on serverless computing, *arXiv preprint arXiv:1902.03383* (2019).
- [37] M. Shahradd, R. Fonseca, Í. Goiri, G. Chaudhry, P. Batum, J. Cooke, E. Laureano, C. Tresness, M. Russinovich, R. Bianchini, Serverless in the wild: Characterizing and optimizing the serverless workload at a large cloud provider, in: *2020 USENIX Annual Technical Conference (USENIX ATC 20)*, 2020, pp. 205–218.
- [38] L. Suresh, J. Loff, F. Kalim, S. A. Jyothi, N. Narodytska, L. Ryzhyk, S. Gamage, B. Oki, P. Jain, M. Gasch, Building scalable and flexible cluster managers using declarative programming, in: *14th USENIX Symposium on Operating Systems Design and Implementation (OSDI 20)*, 2020, pp. 827–844.
- [39] H. Casanova, Y. C. Wong, L. Pottier, R. F. da Silva, On the feasibility of simulation-driven portfolio scheduling for cyberinfrastructure runtime systems, in: *Job Scheduling Strategies*

- for Parallel Processing, 2022.
- [40] F. Mastenbroek, G. Andreadis, S. Jounaid, W. Lai, J. Burley, J. Bosch, E. Van Eyk, L. Versluis, V. Van Beek, A. Iosup, *Opencdc 2.0: Convenient modeling and simulation of emerging technologies in cloud datacenters*, in: 2021 IEEE/ACM 21st International Symposium on Cluster, Cloud and Internet Computing (CCGrid), IEEE, 2021, pp. 455–464.

Среда имитационного моделирования GPSS Studio и моделирующее ядро GPSS World Core

Владимир В. Девятков^{1,2,3}, Тимур В. Девятков³, Максим В. Федотов³, Расуль Р. Минниханов¹, Василий М. Шестюк¹

¹ Академия наук Республики Татарстан, ул. Баумана 20, Казань, 420111, Россия

² КНИТУ-КАИ, ул. К. Маркса д. 10, Казань, 420015, Россия

³ ООО «Элина – Компьютер», ул. Толстого д. 16а, 24, Казань, 420015, Россия

Аннотация

Описывается развитие среды моделирования GPSS Studio и ее применение для проведения имитационного исследования сложных дискретных систем. Показаны особенности и преимущества применения среды на всех этапах имитационного исследования. Представлено моделирующее ядро среды – российская система моделирования GPSS World Core и описаны новые функциональные возможности ядра в части использования в модели динамического вызова программ на других языках программирования. Приведены примеры применения среды при проведения практических имитационных исследований.

Ключевые слова

Сложные системы, имитационное моделирование, структурная схема, приложение, исследование

1. Введение

Существует множество языков и систем имитационного моделирования (далее ИМ). За время существования данной технологии (с 60-х годов прошлого века) создано свыше 500 языков ИМ. Не многим была уготована длинная жизнь и популярность у пользователей. В настоящее время мировыми лидерами среди них являются достаточно «молодые» языки, такие как SIMIO [1], AnyLogic [2], Extend [3], Simul8 [4].

Но не отстает от них и язык GPSS, которому уже свыше 60 лет. В этом смысле язык GPSS является мировым рекордсменом. На его основе было создано целое семейство языков. Начиная с первой версии Дж. Гордона в 1961 году [5] и заканчивая GPSS World [6], GPSS/H [7], SLX [8] и aGPSS [9]. Параллельно с западным миром, язык GPSS применялся и развивался также в СССР, а затем и в России. В 70-е годы он признавался одним из ведущих языков программирования и его развитием занималась компания IBM. После перехода с mainframe на персональные компьютеры, компания IBM прекратила поддержку GPSS и появилось большое количество университетских и коммерческих разработок. Среди них, одним из наиболее удачных и функциональных стал язык GPSS World разработки компании Minuteman Software, который в 2001 году был локализован в России и получил широкое распространение.

Но постепенно, информационные технологии уходили вперед, а GPSS World, начиная с 2012 года не совершенствовался. Хотя по своей моделирующей мощи, простоте разработки моделей, производительности вычислений и популярности у пользователей находится на высочайшем уровне. Поэтому, встал вопрос – либо необходимо совершенствовать технологию

4th International Workshop on Information, Computation, and Control Systems for Distributed Environments (ICCS-DE 2022), July 04–08, 2022, Irkutsk, Russia

EMAIL: vladimir@elina-computer.ru (A. 1); the-9th@yandex.ru (A. 2); f.maxmax@yandex.ru (A. 3), r@sumus.fund (A. 4), v@sumus.fund (A. 5),

ORCID: 0000-0003-1570-8004 (A. 1); 0000-0002-6419-0208 (A. 2); 0000-0001-9625-5427 (A. 3), 0000-0002-6539-6806 (A. 4), 0000-0003-0850-779X (A. 5),



2022 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

ICCS-DE 2022 Workshop Proceedings

использования языка и привести ее к требованиям и возможностям современных технологий, либо язык умрет. Начиная с 2011 в компании «Элина – Компьютер» (г. Казань) начались работы по модернизации использования языка GPSS World, посредством создания внешней оболочки для управления имитационным исследованием. Сначала, в 2013 году появился расширенный редактор GPSS World [10], а затем в 2018 году – среда моделирования GPSS Studio. Схематично, история развития GPSS показана на рис. 1.

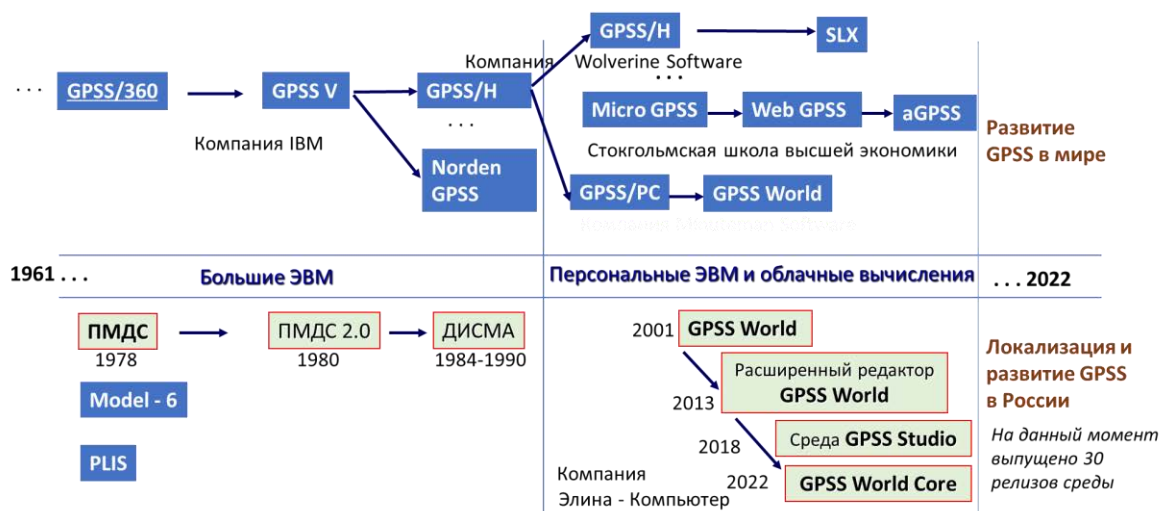


Рисунок 1: История развития семейства языков GPSS в России (СССР) и в мире

Концептуальной основой создания GPSS Studio стала усовершенствованная методология имитационных исследований [11]. Использование методологии позволило не просто автоматизировать процесс имитационного исследования - было создано единое исследовательское пространство в рамках одного программного инструмента, удалось согласовать и синхронизировать интерфейсы обмена данными и результатами между этапами, унифицировать язык взаимодействия пользователя с моделью в процессе исследования.

Рассмотрим возможности GPSS Studio более подробно.

2. Разработка модели и исследование в среде моделирования GPSS Studio

Основным элементом, объединяющим имитационное исследование в GPSS Studio, стало понятие проекта. Проект интегрирует, в виде структурированного дерева, все что связано с конкретным имитационным исследованием: исходные данные, варианты моделей, структурные схемы, результаты отладки, результаты экспериментирования и формируемые пользователем отчеты об исследовании. Внутри проекта может быть создано несколько вариантов модели и для каждой из них создается своя ветка в дереве проекта.

Создание любой модели начинается с разработки иерархической структурной схемы. Схема – это графическое отображение структуры моделируемой системы в виде элементов, связей между этими элементами и при необходимости «детализации» каждого элемента новым уровнем в виде дочерней схемы. Существует три вида элементов схемы (ТЭБов) – неделимый элементарный ТЭБ, который не содержит дочерних схем, но имеет логику в виде GPSS модели, композитный ТЭБ, не имеющий своей логики, но содержащий в себе дочернюю схему и ТЭБы для описания данных. Данные, описанные на корневом уровне, имеют действие по всей схеме, а данные, описанные на дочернем уровне – действуют только на нем и других уровнях, дочерних по отношению к этому уровню. Согласно логике модели, элементы внутри уровня и дочерние схемы между собой соединяются связью, в случае если между ними движутся транзакты.

После завершения разработки схемы, осуществляется автоматическая генерация модели на языке GPSS World Core.

Пример структурной схемы и автоматически сгенерированного текста показан на рис. 2.

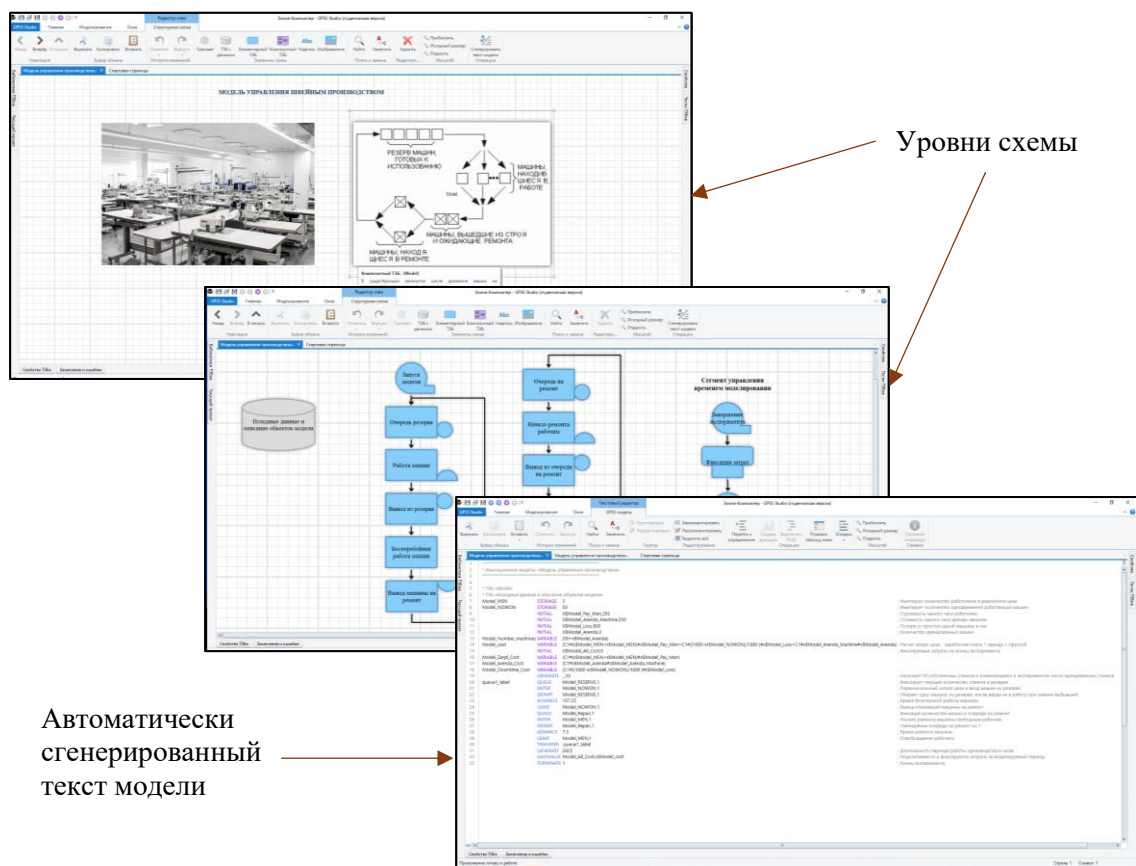
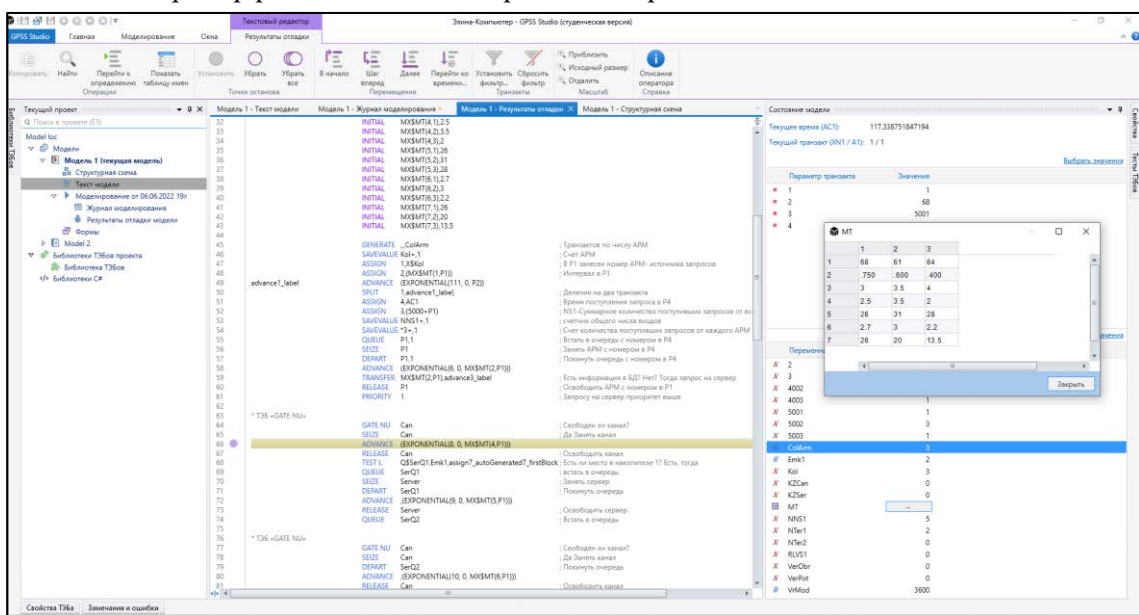


Рисунок 2: Пример схемы и текста простейшей модели из «Красной книги» Шрайбера

Далее, осуществляется отладка модели. Для упрощения и ускорения отладки, в рамках среды создан специальный интерактивный отладчик с возможностями пошаговой отладки, контрольными точками, с динамической визуализацией параметров транзактов, матриц и других СЧА модели. Пример работы отладчика приведен на рис. 3.



По результатам успешного выполнения модели формируется типовой для GPSS стандартный отчет, разделенный на отдельные разделы по объектам модели (блоки, устройства и т.д.) и представляющий диалоги перехода из одного раздела в другой для анализа результатов.

Важнейшим преимуществом среды моделирования является возможность создания вокруг модели имитационного приложения. Для этого имеются специальные конструкторы форм для создания диалогов ввода исходных данных в модель, планирования экспериментов и вывода результатов (текста модели, структурной схемы, введенных данных эксперимента, динамики результатов, таблиц, графиков серий экспериментов и др.). С помощью средств конструирования приложений удастся максимально приблизить язык ввода и язык вывода к предметной области исследования. Разработчик даже может создавать свои, наиболее привычные специалистам предметной области формы ввода данных и вывода результатов. Например, используя наработанные на предприятиях MS Excel таблицы. Пример формирования нестандартных диалогов вывода результатов в табличном редакторе, показан на рис. 4.

	Время прибытия баржа в акваторию	Длительность ожидания освобождения причала	Время начала передвижения из акватории к причалу	Длительность передвижения до причала	Время начала заправки причала (швартовке работы)	Длительность швартовых работ	Время начала разгрузки	Длительность разгрузки	Время завершения разгрузки	Длительность очистки	Время начала отбоя баржа	Длительность швартовых работ	Время начала передвижения в акваторию	Длительность передвижения в акваторию	Время отбоя баржа (уничтожение)	Общее время жизненного цикла (часы)	Загрузка причала (место 1)	Загрузка причала (место 2)
1	0:10:00	0:00:00	0:10:00	0:14:46	0:24:46	0:30:00	0:54:46	10:57:44	11:52:30	0:40:00	12:32:30	0:30:00	13:02:30	0:18:26	13:20:56	13:10:56	68,96%	99,31%
2	0:10:00	0:00:00	0:10:00	0:16:20	0:26:20	0:30:00	0:56:20	10:59:10	11:55:30	0:40:00	12:35:30	0:30:00	13:05:30	0:20:39	13:26:09	13:16:09		
3	0:45:00	12:17:30	13:02:30	0:14:46	13:17:16	0:30:00	13:47:16	0:00:00	0:00:00	0:00:00	0:00:00	0:00:00	0:00:00	0:00:00	0:00:00	#####		
4	0:45:00	12:20:30	13:05:30	0:16:20	13:21:50	0:30:00	13:51:50	0:00:00	0:00:00	0:00:00	0:00:00	0:00:00	0:00:00	0:00:00	0:00:00	#####		
5	5:10:00	0:00:00	0:00:00	0:00:00	0:00:00	0:00:00	0:00:00	0:00:00	0:00:00	0:00:00	0:00:00	0:00:00	0:00:00	0:00:00	0:00:00	#####		
6	5:10:00	0:00:00	0:00:00	0:00:00	0:00:00	0:00:00	0:00:00	0:00:00	0:00:00	0:00:00	0:00:00	0:00:00	0:00:00	0:00:00	0:00:00	#####		
7	5:45:00	0:00:00	0:00:00	0:00:00	0:00:00	0:00:00	0:00:00	0:00:00	0:00:00	0:00:00	0:00:00	0:00:00	0:00:00	0:00:00	0:00:00	#####		
8	5:45:00	0:00:00	0:00:00	0:00:00	0:00:00	0:00:00	0:00:00	0:00:00	0:00:00	0:00:00	0:00:00	0:00:00	0:00:00	0:00:00	0:00:00	#####		
9	6:50:00	0:00:00	0:00:00	0:00:00	0:00:00	0:00:00	0:00:00	0:00:00	0:00:00	0:00:00	0:00:00	0:00:00	0:00:00	0:00:00	0:00:00	#####		
10	6:50:00	0:00:00	0:00:00	0:00:00	0:00:00	0:00:00	0:00:00	0:00:00	0:00:00	0:00:00	0:00:00	0:00:00	0:00:00	0:00:00	0:00:00	#####		
11	7:25:00	0:00:00	0:00:00	0:00:00	0:00:00	0:00:00	0:00:00	0:00:00	0:00:00	0:00:00	0:00:00	0:00:00	0:00:00	0:00:00	0:00:00	#####		
12	7:25:00	0:00:00	0:00:00	0:00:00	0:00:00	0:00:00	0:00:00	0:00:00	0:00:00	0:00:00	0:00:00	0:00:00	0:00:00	0:00:00	0:00:00	#####		

Рисунок 4: Вывод жизненного цикла транспортных средств в модели

Создание приложений существенно расширяет круг пользователей, с ним могут работать специалисты предметной области – не профессионалы ИМ. При этом, приложение получается независимым от среды моделирования и может использоваться автономно, даже на другом компьютере, но в любом случае, для проведения экспериментов требуется ядро моделирования. Используя приложение, пользователь может проводить полноценное исследование, реализуя различные сценарии – проводя отдельные эксперименты, серии экспериментов или оптимизационные эксперименты (рис. 5).

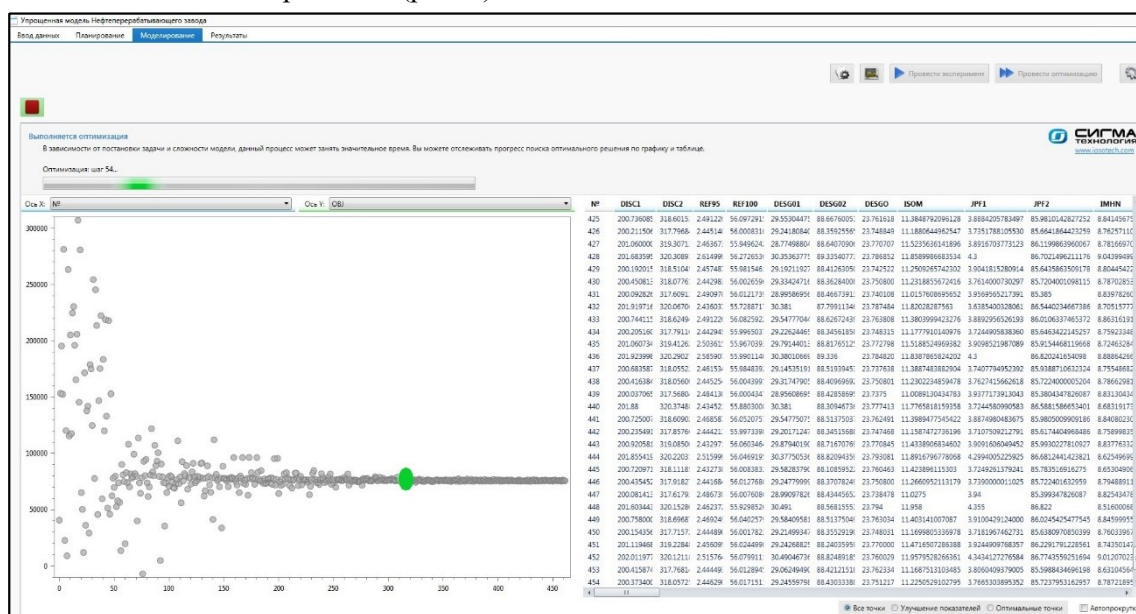


Рисунок 5: Оптимизационный эксперимент в GPSS Studio

Все эксперименты сохраняются в базе данных приложения. К ним можно вернуться во время любого сеанса диалога для детального анализа. Не нужные эксперименты всегда можно удалить.

Важнейшей возможностью, предоставляемой пользователю в имитационном приложении, является накопление виртуального отчета по исследованию в сеансе диалога, куда пользователь «складывает» наиболее важные и значимые результаты. По завершении сеанса, виртуальный отчет превращается в реальный, путем сохранения в нужный MS Word или MS Excel файл, который будет пополняться во время других сеансов. Затем, используя такой отчет-полуфабрикат можно быстро его структурировать, дополнить текстом и более быстро сформулировать выводы и рекомендации по исследованию.

На рис. 6 приведена упрощенная схема проведения имитационного исследования в среде моделирования GPSS Studio.



Рисунок 6: Схема проведения имитационного исследования в GPSS Studio

3. Ядро моделирования GPSS World Core

Базисным элементом в среде моделирования остается его ядро, в котором реализуются все запланированные эксперименты. После завершения развития GPSS World в компании Minuteman Software и передаче его в ООО «Элина – Компьютер», началась новая жизнь симулятора, но не в одиночку, а в тандеме с GPSS Studio. Сейчас система моделирования используется по большей части, как ядро моделирования – для проведения прогонов, созданных в GPSS Studio моделей и реализации спланированных экспериментов.

В настоящее время в язык внесены существенные правки и проведена локализация всех его интерфейсов на русский язык. Важнейшим дополнением языка стало введение блока вызова модулей, написанных на другом языке программирования. Практика показывает, что в любом месте программы может возникнуть необходимость вызова внешней программы - для ввода данных, проведения сложных расчетов или вывода не стандартных результатов. Это дополнение языка GPSS World существенно шире тех возможностей, которые существовали еще в GPSS/360 и GPSS V (блок HELP).

В связи с необходимостью замены иностранного программного обеспечения на отечественные программы и после значительных авторских изменений в программном коде, была осуществлена регистрация нового программного продукта GPSS World Core в Роспатенте РФ [12].

На рис. 7 показан пример экранных форм GPSS World Core.

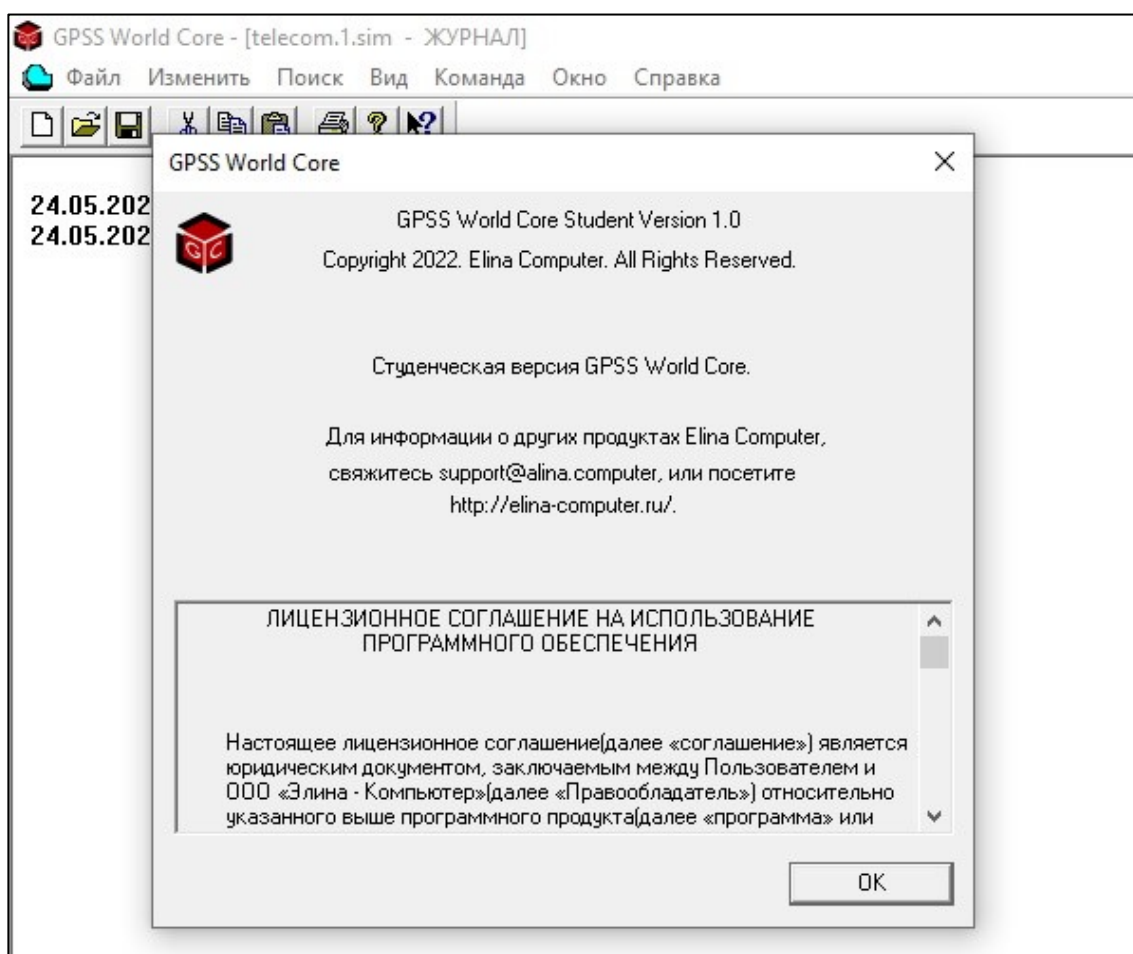


Рисунок 7: Пример работы с GPSS World Core

Планируется дополнение возможностей языка еще целым рядом новых блоков, например, из числа наиболее эффективных блоков, используемых в GPSS/H и которых не было в GPSS World. Также начаты работы по созданию кроссплатформенной версии языка, что особенно важно при переходе многих государственных структур и крупных корпораций на Астра Линукс и другие Linux на подобные операционные системы.

4. Имитационные исследования в GPSS Studio

Процесс создания и развития среды и ядра моделирования сопровождался регулярными практическими применениями в процессе исследования сложных дискретных систем в различных областях экономики - в основном для крупных предприятий и при реализации инфраструктурных объектов. Такое итеративное переключение (разработка/практика) позволило нам с одной стороны проверять новые методические и программные решения, а с другой стороны – ближе знакомиться с проблемами и задачами, которые возникают при проектировании, эксплуатации и модернизации сложных систем. Все это позволило достаточно глубоко провести тестирование продукта и дополнить его целым рядом новых возможностей, необходимость которых подсказала практика

За последние годы, нами и нашими партнерами, был реализован целый ряд больших и важных проектов с использованием GPSS Studio. Например, это моделирование движения грузовых поездов по сети АО РЖД [13], исследование системы производственной логистики проекта крупного целлюлозного завода [14], анализ технологических процессов судосборочного

производства [15] и т.д. На рис. 8 приведен пример разработки в модели стапельного расписания для судостроительного предприятия.

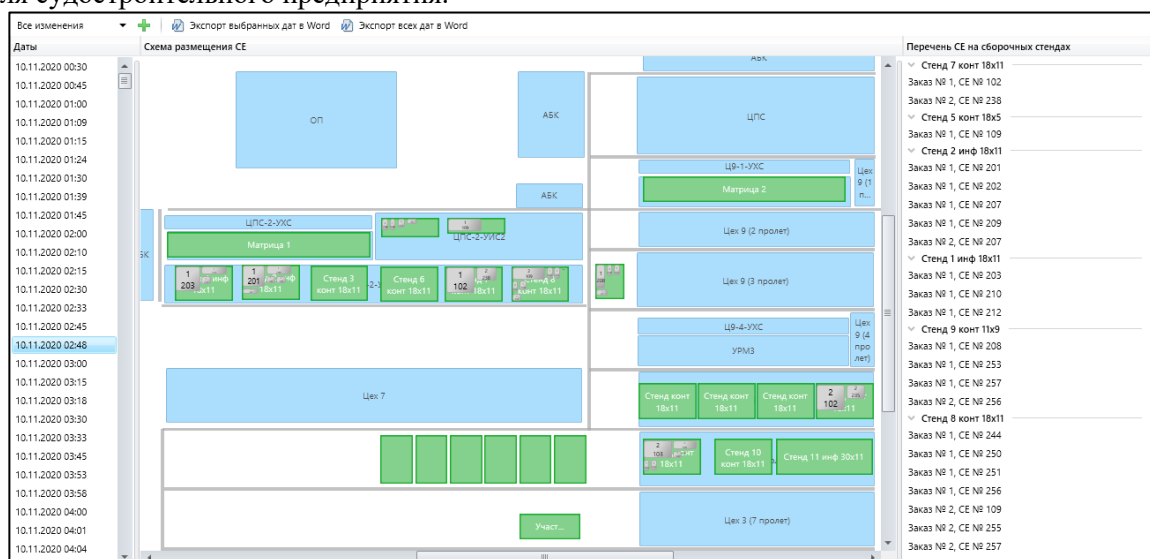


Рисунок 8: Стапельное расписание

Сейчас идет активное и массовое внедрение имитационного моделирования практически во все сферы экономики, но наибольший интерес для нас представляют следующие направления:

- Промышленное моделирование, например, в части реализации национального проекта повышения производительности труда
- Дополнение сетевого планирования новыми возможностями за счет использования графических структурных схем GPSS Studio и моделей
- Реализация платформы управления сложными системами на основе имитационных приложений (цифровых двойников)

5. Заключение

В последние годы наблюдается ренессанс интереса к применению имитационного моделирования. Это связано с целым рядом причин:

- Практической реализацией новых интеграционных подходов к имитационному исследованию
- Появлением новых теоретических методов и алгоритмов в имитационном моделировании
- Качественным и количественным рывком информационных технологий и вычислительных мощностей

В совокупности, все эти факторы позволили многим разработчикам, в том числе и нам в рамках GPSS Studio, приблизить имитационное моделирование к исследователю, значительно ускорить вычисления и синхронизировать запросы заказчиков (по времени и качеству) с возможными темпами разработки модели и проведения исследования.

6. Ссылки

- [1] Jeffrey S. Smith and David T. Sturrock, Simio and Simulation - Modeling, Analysis, Applications 6th Edition, SIMIO LLC, Sewickley, PA, August 1 2021. <https://textbook.simio.com/SASMAA/index.html>
- [2] Карпов, Ю.Г. Имитационное моделирование систем. Введение в моделирование с AnyLogic 5 / Ю.Г. Карпов. – СПб.: БХВ-Петербург, 2006. – 400 с.

- [3] ExtendSim Reference Guide, Imagine That Inc., San Jose, 2022
<https://extendsim.com/learning/documentation>.
- [4] Simul8 Simulation Software Tutorial, Simul8 Corporation, Boston, MA, 2022
<https://www.simul8.com/support/help/doku.php?id=tutorials>
- [5] Gordon, Geoffrey. A General Purpose Simulation Systems Simulation Program / Geoffrey Gordon // Proc. EJCC. – Washington: D.C., Macmillan Publishing Co., Inc., New-York, 1961, P. 87-104.
- [6] Руководство пользователя по GPSS World/ Казань, изд-во "Мастер Лайн", 2002 г., 384 с.
- [7] Варжапетян А.Г. Имитационное моделирование на GPSS/H: учебное пособие / А.Г. Варжапетян. – СПб.: ГУАП, 2007. – 384 с.
- [8] Henriksen, J.O. An introduction to SLX / J.O. Henriksen, S Andradoittir, K. J. Heally, D.H. Withers, and B.L. Nelson // Proceedings of the 1997 Winter Simulation Conference. – P. 559-566.
- [9] Ingolf Ståhl, Richard Born, Business Modeling with a General Purpose Simulation System on Mac OS X and Windows, SSE, Stockholm, 2018, 294 pp.
- [10] Девятков, В.В. Расширенный редактор GPSS World / В.В. Девятков, М.В. Федотов //Пятая всероссийская научно-практическая конференция «Имитационное моделирование, теория и практика»: сборник докладов. – СПб.: ОАО ЦТСС, 2011. – Т 1. – С. 350-354.
- [11] Девятков, В.В. Методология и технология имитационных исследований сложных систем: современное состояние и перспективы развития: монография / В.В. Девятков – М.: Вузовский учебник.: ИНФРА-М, 2013. – 448 с.
- [12] Девятков Владимир Васильевич, Девятков Тимур Владимирович, Федотов Максим Валерьевич Свидетельство о регистрации программы для ЭВМ GPSS World Core 2021680076, 07.12.2021. Заявка № 2021669832 от 07.12.2021.
- [13] Девятков В.В., Девятков Т.В., Уманский В.И., Вдовин А.Н. Оперативное управление поездопотоками на сети железных дорог России с использованием имитационной модели // Материалы восьмой научно-технической конференции «Интеллектуальные системы управления на железнодорожном транспорте. Компьютерное и математическое моделирование – ИСУЖТ-2019» М.: АО НИИАС.
- [14] Маряшина Д.Н., Девятков В.В., Девятков Т.В., Харисов И.З. Сегежа. Исследование системы производственной логистики целлюлозного завода // Десятая всероссийская научно-практическая конференция по имитационному моделированию и его применению в науке и промышленности «Имитационное моделирование. Теория и практика» (ИММОД-2021). Труды конференции (электронное издание), 20–22 октября 2021 г., Санкт-Петербург: АО «ЦТСС», 2021. – 694 с. – ISBN 978-5-905526-05-3. С. 305-313
- [15] Девятков Т.В., Федотов М.В., Долматов М.А. Опыт и перспективы применения методов трехмерной визуализации результатов имитационного моделирования функционирования производственных систем // Шестая международная научно-практическая конференция. «Имитационное и комплексное моделирование морской техники и морских транспортных систем» (ИКМ МТМТС-2021). Труды конференции. (ISBN 978-5-00189-175-8). - М.: Издательство «Перо», 2021. - 165 с. С.51-54.

Проектирование комбинационных схем на функционально-поточном параллельном языке программирования

Дарья Романова^{1,2}

¹ Сибирский федеральный университет, пр. Свободный, 79, г. Красноярск, 660041, Россия

² Красноярский государственный аграрный университет, пр. Мира, 90, г. Красноярск, 660049, Россия

Аннотация

В статье приведен обзор языков программирования и инструментальных средств для проектирования логических схем. Описаны особенности языка функционально-поточного параллельного программирования, позволяющие применять его при проектировании комбинационных схем. Приведены примеры функций, реализующих некоторые комбинационные схемы разными способами.

Ключевые слова

Комбинационная схема, функционально-поточный язык, архитектурная независимость, параллелизм

1. Введение

Сегодня требования к проектированию цифровых интегральных схем (ИС) постоянно возрастают. Одной из главных задач проектирования таких схем является обеспечение переносимости проектных решений между целевыми платформами ИС. Интегральная схема по существу представляется собой систему параллельной обработки информационных потоков, а алгоритмы ее функционирования, на заключительных этапах синтеза, представляются на языках описания аппаратуры (HDL языках). Комбинационные схемы (КС) являются основой при проектировании ИС. Поэтому эффективные решения по обеспечению архитектурной независимости при проектировании КС могут быть найдены в области переносимых параллельных программ.

2. Обзор существующих решений

Традиционно логические схемы проектируют на языках описания аппаратуры, таких как Verilog и VHDL [1]. Эти языки и применяются только на низких уровнях абстракции, где имеется привязка программы к определенной архитектуре.

Существует ряд решений, поддерживающих использование адаптированных, императивных языков программирования высокого уровня (C++) в качестве языков описания аппаратуры, например Handel-C [2] или Impulse-C [3]. Данные языки в первую очередь созданы для решения частных задач и поэтому при построении решений на этих языках происходит жесткая привязка к конкретной архитектуре.

Для конструирования цифровых ИС более эффективно применять функциональные языки, которые обладают более мощным механизмом абстракции по сравнению с императивными языками. Из таких языков для проектирования логических схем подходят встроенные подмножества языка Haskell: Wired [4], Lava [5]. В языке Wired есть большая библиотека функций, позволяющая легко строить комбинационные схемы. Но получаемые решения

4th International Workshop on Information, Computation, and Control Systems for Distributed Environments (ICCS-DE 2022), July 04–08, 2022, Irkutsk, Russia

EMAIL: daryaooo@mail.ru

ORCID: 0000-0002-9020-4802



2022 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

ICCS-DE 2022 Workshop Proceedings

зачастую слишком параметризованы. Lava обладает расширенной системой типов, что упрощает проектирование схем. Однако последовательная структура списков накладывает ограничения на преобразования параллелизма и не позволяет эффективно производить распараллеливание.

Язык ПИФАГОР [6] является функциональным языком, поддерживающим модель потока данных и параллелизм на уровне операций, что является наиболее привлекательным для независимого, сквозного проектирования цифровых схем [7]. В языке отсутствуют циклы, что позволяет избежать конфликтов при использовании одних и тех же данных.

В модели функционально-потокowego параллельного программирования (ФППП) были произведены некоторые модификации, позволяющие адаптировать ее для описания интегральных схем. Динамическая система типов модели была заменена на статическую, поддерживаемую в языках описания аппаратуры. Были введены четыре скалярных типа данных: битовое поле (bits), логический тип (bool), целочисленный тип без знака (uint), целочисленный знаковый тип (int). В видоизмененной модели присутствуют некоторые векторные типы данных, такие как списки данных, асинхронные, параллельные и задержанные списки.

Для обеспечения архитектурной независимости при разработке метода высокоуровневого синтеза интегральных схем используется промежуточное представление программы на языке ПИФАГОР в виде информационного графа (рисунок 1).

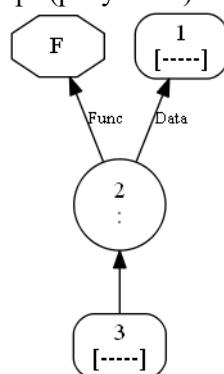


Рисунок 1: Информационный граф

В данном языке задача с максимальным параллелизмом описывается с помощью параллельного и асинхронного списков, а их обработка производится с помощью преобразования этих конструкций.

Переход к конвейерной обработке тоже производится путем трансформации управляющего графа и получением информационного графа в ярусно-параллельной форме.

3. Метод проектирования комбинационных схем на функционально-потокowym языке

Основная идея разрабатываемого метода проектирования комбинационных схем заключается в описании логических схем на языке ФППП и создание их в виде специализированных комбинационных схем, которые не имеют синхронизации и обеспечивают более высокую производительность за счет прямого асинхронного решения без обратной связи.

4. Построение логических схем на языке ФПП программирования

Комбинационные схемы разрабатываются на основе Булевой алгебры, где используется двоичная система счисления. Поэтому основной тип данных для создания логических схем на функционально-потокowym языке – bool. Аргумент принимает значения true или false.

4.1. Проектирование логических элементов

Элементарные логические операции реализуются с помощью логических элементов [8]. Каждый из этих элементов на языке ФППП представляет собой отдельную функцию. Так, был разработан комплект функций, реализующих следующие логические операции: И, ИЛИ, НЕ, И-НЕ, ИЛИ-НЕ, исключающее ИЛИ, И-ИЛИ-НЕ. На рисунках 2 и 3 приведены функции, реализующие некоторые из разработанных элементов.

```
And2Not << funcdef Param
{
    return << Param: *: -;
```

Рисунок 2: Функция на языке ФПП программирования, реализующая логический элемент ИЛИ-НЕ

```
Xor << funcdef Param
{
    x1 << Param:1;
    x2 << Param:2;
    return << ((x1,x2:-):*, (x1:-,x2):*):+ ;
}
```

Рисунок 3: Функция на языке ФПП программирования, реализующая логический элемент И-НЕ

Логические элементы с тремя или более входами реализуются на данном языке программирования путем подачи списка из входных элементов вместо одного аргумента.

4.2. Проектирование комбинационных схем

На основе логических элементов на языке ФППП строятся комбинационные схемы путем вызова соответствующих функций.

На данном языке были разработаны следующие комбинационные схемы: полусумматор, полный двоичный сумматор, многоразрядный сумматоры, двоичные шифраторы и дешифраторы на два, восемь, шестнадцать и тридцать два входа, 4-х, 8-ми, 16-ти- и 32-х канальные мультиплексоры и демультимплексоры.

Существующие в языке конструкции, такие как параллельный список и задержанный список, а также рекурсия, свойственная данному языку, позволяют разрабатывать комбинационные схемы различными способами с разной степенью параллелизма.

Так, если производить ввод, вывод данных или их обработку с помощью параллельных списков можно значительно ускорить работу программы. Например, это продемонстрировано в функции, реализующей прямоугольный дешифратор (рисунок 4), где параллельный список применяется для подсчета выходного сигнала S.

```
Dec_R << funcdef Param // Дешифратор прямоугольный на 8 выходов
{
    D0 << Param:1;
    D1 << Param:2;
    D2 << Param:3;
    D3 << Param:4;
    // подсчет результата по логической функции
    A0 << (D3:Not,D2:Not):And2;
    A1 << (D3:Not,D2):And2;
    A2 << (D3,D2:Not):And2;
    A3 << (D3,D2):And2;
    B0 << (D1:Not,D0:Not):And2;
    B1 << (D1:Not,D0):And2;
    B2 << (D1,D0:Not):And2;
```

```

B3 << (D1,D0):And2;
S<<[(A0,B0):And2, (A0,B1):And2, (A0,B2):And2, (A0,B3):And2,
(A1,B0):And2, (A1,B1):And2, (A1,B2):And2, (A1,B3):And2,
(A2,B0):And2, (A2,B1):And2, (A2,B2):And2, (A2,B3):And2,
(A3,B0):And2, (A3,B1):And2, (A3,B2):And2, (A3,B3):And2 ];
return <<S; // вывод результата
}

```

Рисунок 4: Функция на языке ФППП, реализующая прямоугольный дешифратор

Для работы с большими данными применяются многоразрядные комбинационные схемы. При конструировании многоразрядных комбинационных схем наращивание разрядности и числа каналов производится с помощью пирамидальной структуры на основе уже разработанных более простых схем или с помощью хвостовой рекурсии. В первом случае, на языке ФППП при построении функции N-разрядной (N-канальной) схемы в нее включаются вызовы других функций меньшей разрядности (с меньшим числом каналов).

На рисунке 5 приведена функция 16-тиканального мультиплексора, реализованного на основе функции двухканального мультиплексора и двукратного вызова функции восьмиканального мультиплексора.

```

MUX16_1_1 << funcdef Param // мультиплексор 16 к 1
{
A<<(Param:1,Param:2,Param:3,Param:4,Param:5,Param:6,Param:7,Param:8,Param:1
8,Param:19,Param:20):MUX8_1_1;
B<<(Param:9,Param:10,Param:11,Param:12,Param:13,Param:14,Param:15,Param:16,
Param:18,Param:19,Param:20):MUX8_1_1;
Y << (A,B,Param:17):MUX2_1;
return << Y; // вывод
}

```

Рисунок 5: Функции на языке ФППП, реализующая шестнадцатиканальный мультиплексор наращиванием

В случае построения многоразрядных комбинационных схем с помощью рекурсии на языке ФППП первоначально на основе количества информационных входов считается количество адресных входов по соотношению $n = \log_2 m$, где n – количество адресных входов, а m – количество информационных входов.

Номер активного выходного сигнала рассчитывается по формуле:

$$F = X_i, \text{ если } \sum_{i=0}^{n-1} X_i * 2^i = i$$

Где i – номер текущего элемента, X_i – активный выходной сигнал.

Функция, реализующая такой мультиплексор, приведена на рисунке 6.

```

MUX_N << funcdef Param { // мультиплексор (рекурсия)

D << Param: 1; // информационные входы, подаются массивом // bool list
A << Param: 2; // A- адресные входы // bool list
S << Param: 3; // счетчик // int
i << Param: 4; // номер выходного сигнала // int
F << Param: 5; // выходное значение //bool
E << Param: 6; // разрешение //bool
[(E,o):[!=,=]):?] ^ // цикл проверки если E равно нулю
(
{ "No signal"},
{(D,A,S,i,F):MUX_N1}
):.>> return; // вывод
}
MUX_N1 << funcdef Param // мультиплексор рекурсия (основная функция)
{

```



```

m << D:1; // m – количество информационных входов
n << (m):mmm; подсчет количества адресных входов

[ ((i,n):[>=,<]):? ] ^ // основной цикл программы (рекурсия)
(
    { (D,S):Res1 },
    { (D,A, (S, (A:[(n,i):-]:int, (2,i):math.pow):*):+, (i,1):+,
    D:[(S,1):+]):MUX_N1 }
):.>>return;
}
Res1 << funcdef X // вспомогательная функция, где D и S – аргументы,
поступившие в функцию из MUX_N1
{
    D << X:1;
    S << X:2;
    R << D:[(S,1):+]; // R = Ds+1 помещается в параллельный список
    return << R;
}

```

Рисунок 6: Функция на языке ФППП, реализующая многоканальный мультиплексор рекурсией

5. Практические результаты

По результатам проведенных исследований планируется разработать компилятор, с помощью которого разработанные комбинационные схемы на языке ФППП будут преобразовываться с промежуточного представления (информационного графа программы) напрямую, т.е. без нарушения логики проектирования, в конкретные комбинационные схемы на языке Verilog.

6. Заключение

В ходе проведенного исследования для построения комбинационных схем был выбран язык ФППП программирования, являющийся архитектурно-независимым и, следовательно, позволяющий легко переносить проектируемые решения схем на другие целенные платформы.

На данном языке с помощью функций, реализующих основные логические элементы, были разработаны сумматоры, шифраторы, дешифраторы, мультиплексоры и демультимплексоры различными способами.

На основе разработанных схем разрабатывается метод их переноса на HDL языки.

7. References

- [1] IEEE Std 1800-2012: IEEE Standard for SystemVerilog Unified Hardware Design, Specification, and Verification Language, 2013.
- [2] Handel-C Language Reference Manual. Celoxica Limited, 2005.
- [3] J. Xu, N. Subramanian, A. Alessio and S. Hauck. "Impulse C vs. VHDL for Accelerating Tomographic Reconstruction." 18th IEEE Annual International Symposium on Field-Programmable Custom Computing Machines (2010): 171-174.
- [4] E. Axelsson, K. Claessen, M. Sheeran. "Wired: wire-aware circuit design." Proceedings of the Conference on Correct Hardware Design and Verification Methods (CHARME '05), vol. 3725, 2005, pp. 5—19.
- [5] A. Gill, T. Bull, G. Kimmell. et al. "Introducing Kansas Lava." Implementation and Application of Functional Languages, ser. Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, Sep. 2009, pp. 18—35.
- [6] A. I. Legalov A functional language for creating architecture-independent parallel programs, Vychislitel'nye tehnologii, 2005, vol. 10, no. 1, pp. 71-89 (in Russian)

- [7] O. V. Nepomnyashchy, A. I. Legalov, V. Tyapkin et al. “Methods and algorithms for a high-level synthesis of the verylarge-scale integration.” WSEAS Transactions on Computers, No. 15, 2016, pp. 239—247.
- [8] Sarah L. Harris, David Harris, Digital Design and Computer Architecture: RISC-V Edition, Elsevier Inc., 2021.

Comparative Analysis of Tools for Automating Microgrid Modeling

Mikhail Chekan

Matrosov Institute for System Dynamics and Control Theory of SB RAS, Lermontov St. 134, 664033, Irkutsk, Russia

Abstract

Nowadays, microgrid modeling is an extremely relevant problem. In solving this problem, the most important properties of such grids are studied, including their survivability, reliability, opportunity to self-organize, ability to regulate the supply and demand of energy resources, use of renewable resources, etc. The paper addresses a comparative analysis of well-known tools for automating microgrid modeling. We discuss the functionality of the tools under consideration. In addition, we highlight the main directions of their development.

Keywords

microgrid, modeling, automation tools, comparative analysis

1. Introduction

At present, the intellectualization and digitalization of technologies for data processing and analyzing in the context of studying environmentally friendly and resource-saving functioning energy systems is one of the key directions of energy development both in Russia and abroad. At the same time, in the last decade, microgrid development is established as a new influential technological trend in the field of energy development [1]. A microgrid is a local energy system with its own energy infrastructure, generally capable of autonomous operation [2]. Such a system is characterized by the use of technologies for distributed generating energy. Such technologies support applying renewable energy sources (solar panels and wind turbines), as well as various energy storage devices. Generally, diversification of energy sources in a microgrid increases its energy independence and reliability of electricity supplies to consumers, as well as improves the distribution of energy resources, environmental friendliness, and other criteria of the microgrid work quality. In particular, microgrids use is of key importance for unique natural territories in terms of supporting nature conservation.

2. Tools for automating microgrid modeling

The study of microgrid as a complex system based on mathematical modeling determines the following features of the microgrid model:

- A large number of model variables and mathematical equations that describe the dependencies between these variables,
- The presence in the model of random variables with different laws of their distribution,
- The variety of relations between model elements,
- The existence of various constraints and dependencies from a model time.

Solving the problem of analyzing microgrid performance parameters considering interaction with other similar systems reasonably requires advanced methods and tools for creating the microgrid

4th International Workshop on Information, Computation, and Control Systems for Distributed Environments, July 4–8, 2022, Irkutsk, Russia

EMAIL: chekoopa@mail.ru

ORCID: 0000-0001-7622-421X



© 2022 Copyright for this paper by its authors.

Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

ICCS-DE 2022 Workshop Proceedings

model, as well as preparing and carrying out large-scale experiments using parallel and distributed computing, including calculations in the heterogeneous distributed computing environment.

Nowadays, in Russia and abroad, there is a trend to develop specialized methods and tools for automating the analysis of microgrid functioning processes, including their interaction with each other. The results of such studies are reflected in a wide range of works by Russian and foreign scientists (e.g. [3, 4, 5]). Based on these results, it can be claimed that the advanced methods and tools in this area of research include the economic regulation of demand and supply of energy resources, multi-criteria analysis of the efficiency and balance of their generation, distribution, and consumption, as well as specialized multi-agent decision-making technologies. Here, in general, multi-agent technologies are designed to automate and intellectualize the decision-making process by delegating the rights and responsibilities of microgrid entities to software entities (agents). Agents dynamically perform the coordination of generally contradictory criteria based on competition and cooperation in the process of self-organization of a multi-agent system, carried out automatically.

Based on the specifics of the problem being solved, the following key characteristics of tools for automating microgrid modeling can be distinguished:

- Means used for describing subject domain (UML, XML, DSL, semantic web, ontology, etc.),
- Parameters of microgrids and their elements that can be determined (for example, structural and functional parameters), as well as criteria used for the microgrid performance evaluation (for example, cost, emissions volume, survivability, reliability, etc.),
- Simulated economic mechanisms for regulating the demand and supply of energy resources,
- Type of modeling (analytical, simulation, agent-based, or combined),
- Application software architecture (monolithic application, software library, software applied package, distributed software applied package), application type (desktop, network, service-oriented, etc.), and licenses for the software distribution, use, and support,
- Computing environment (PC, HPC-cluster, Grid system, cloud environment, fog computing platform, heterogeneous distributed computing environment),
- Computing scalability (low, medium, high),
- Simulation results optimization (one-criteria selection, multicriteria selection).

A large spectrum of specialized models, algorithms, methods, and tools to support the study of microgrids has been developed (e.g. [6, 7, 8]). In particular, Basak et al. [9] demonstrate applying the Matlab system for microgrid modeling considering aspects of the microgrid concept and function. This scientific work highlights the implementation of a control strategy for islanded mode, which proved to be effective in meeting critical demands with the cooperation of microsources and storage devices. However, the possibilities of this system for analysis and prediction are quite limited, as the model is proposed as educational. Rozhentsova et al. [10] provides a multi-agent model of a microgrid designated as an “active consumer”. The considered microgrid consists of a generalized production workshop, represented by three types of machines, and distributed solar-wind generation. The model was designed in the AnyLogic simulation environment. It allows for variation of parameters for conducting experiments and testing hypotheses. The authors emphasize the success of the model launch and conclude the viability of the chosen approach. At the same time, this model is limited to a single subject area. It does not both consider islanded mode (as being focused on economical criteria) and allow expansion by elements with different behavior. In [11], HOMER software is applied to design and analyze the performance of a microgrid consisting of a photovoltaic system, wind system, and a diesel plant. The authors demonstrated the process of model description, which is detailed enough, and optimization, which provides the best possible configuration of the modeled system. At the same time, some analysis cases would require integration with external systems. The HOMER website [12] mentions a feature to link with the Matlab system. However, this feature is intended for the dispatch algorithm. It does not support integration with external models. Moreover, the HOMER software is represented as a monolithic desktop application, whereas many analysis and prediction cases require a larger scale of calculations, which implies the utilization of cloud and fog computing. Thus, the results of a comparative analysis of the considered tools show a number of remaining challenges. Among these challenges are the integration of remaining developments, their flexible adjustment to the specifics of the tasks being solved, adaptation for use in the HPC-environment, and methodological support, which have not been fully resolved.

3. Conclusions

We provide a comparative analysis of tools for automating microgrid modeling. The microgrid concept combines advanced technologies and developments in the fields of energy, telecommunications, and information processing to create self-regulating and more reliable energy systems in comparison with traditional ones. Applying microgrids leads to increased efficiency of energy supply, rational use of existing infrastructure, and integration of renewable energy sources. The results of the comparative analysis of tools for automating microgrid modeling made it possible to formulate key directions for the development of similar tools.

Acknowledgments

The study is supported by the Ministry of Science and Higher Education of the Russian Federation, project no. 121032400051-9 “Technologies for the development and analysis of subject-oriented intelligent group control systems in non-deterministic distributed environments”, as well as the Russian Foundation of Basic Research and Government of Irkutsk Region, project no. 20-47-380002 “Mathematical and information modeling of infrastructural objects and systems of the Baikal natural territory”.

References

- [1] M. I. Alomoush, Microgrid combined power-heat economic-emission dispatch considering stochastic renewable energy resources, power purchase and emission tax, *Energy Conversion and Management* 200 (2019) 112090. doi:10.1016/j.enconman.2019.112090.
- [2] D. T. Ton, M. A. Smith, The US department of energy's microgrid initiative, *The Electricity Journal* 25 (2012), 84-94. doi:10.1016/j.tej.2012.09.013.
- [3] A. Hirsch, Y. Parag, J. Guerrero, Microgrids: A review of technologies, key drivers, and outstanding issues, *Renewable and sustainable Energy reviews* 90 (2018) 402-411. doi:10.1016/j.rser.2018.03.040.
- [4] Y. Han, K. Zhang, H. Li, E. A. A. Coelho, J. M. Guerrero, MAS-based Distributed Coordinated Control and Optimization in Microgrid and Microgrid Clusters: A Comprehensive Overview, *IEEE Transactions on Power Electronics* 33 (2017) 6488-6508. doi:10.1109/TPEL.2017.2761438.
- [5] S. Sharma, Y. R. Sood, Microgrids: A Review of Status, Technologies, Software Tools, and Issues in Indian Power Market, *IETE Technical Review*, 39 (2020), 411-432. doi:10.1080/02564602.2020.1850367.
- [6] G. N. Kariniotakis, N. L. Soutanis, A. I. Tsouchnikas, S. A. Papathanasiou, N. D. Hatziaargyriou, Dynamic modeling of microgrids, In *Proceedings of the International Conference on Future Power Systems*, IEEE, 2005, pp. 1-3. doi: 10.1109/FPS.2005.204227.
- [7] A. Francés, R. Asensi, O. García, R. Prieto, J. Uceda, How to model a dc microgrid: Towards an automated solution, in: *Proceedings of the Second International Conference on DC Microgrids (ICDCM)*, IEEE, 2017, pp. 609-616. doi:10.1109/ICDCM.2017.8001110.
- [8] W. Feng, M. Jin, X. Liu, Y. Bao, C. Marnay, C. Yao, J. Yu, A review of microgrid development in the United States—A decade of progress on policies, demonstrations, controls, and software tools, *Applied energy* 228 (2018) 1656-1668. doi:10.1016/j.apenergy.2018.06.096.
- [9] P. Basak, A. K. Saha, S. Chowdhury, S. P. Chowdhury, Microgrid: Control techniques and modeling, in *Proceedings of the 44th International Universities Power Engineering Conference (UPEC)*, IEEE, 2009, pp. 1-5.
- [10] N. Rozhentsova, O. Regir, A. Kotsubinski, Mulyi-Agent Approach to Modeling Active Comcumers, *Bulletin of the Kazan State Power Engineering University* 2 (2019) 47-55.
- [11] D. Restrepo, B. Restrepo-Cuestas, A. Trejos, Microgrid analysis using HOMER: a case study, *DYNA*, 85 (2018) 129-134. doi:10.15446/dyna.v85n207.69375.
- [12] HOMER Pro. URL: <https://www.homerenergy.com/products/pro/index.html>

Технологии BigData в России: возможности и перспективы импортозамещения

И.Н. Игнатьев¹

¹ Иркутский государственный университет, ул. Карла Маркса, 1, Иркутск, 664003, Россия

Abstract

Приводится описание технологии Big Data, структура и возможности импортозамещения на российских процессорах и серверах. Показаны результаты анализа и тестирования российских процессоров, а также их сравнение с процессорами Intel.

Keywords

Big Data, импортозамещение, процессор «Эльбрус», процессор «Байкал».

1. Введение

Большое количество цифровой информации стало неотъемлемой частью нашей действительности. Человечество ежедневно создаёт квинтиллионы байт информации различной степени важности, которую необходимо хранить и обрабатывать. Именно для этой цели и были созданы Big Data (BD) – структурированные или не структурированные массивы данных большого объема. Их обрабатывают при помощи специальных автоматизированных инструментов, чтобы использовать для статистики, анализа, прогнозов и принятия решений.

BD уже используется повсеместно: в госуправлении, промышленности, медицине, торговле и других сферах. Любая отрасль, в которой нужно анализировать большие массивы информации, уже не может обойтись без BD. Что для этого необходимо? Есть 3 основных элемента BD: серверное хранилище, где хранится информация; процессор, который будет обрабатывать данные и сам алгоритм обработки. До некоторых пор данное оборудование и технологии закупались преимущественно за границей, но после введения санкций крупнейшие западные компании IBM, Intel и AMD объявили о прекращении продаж серверного и пользовательского оборудования на территории РФ. В связи с чем в России возникла необходимость разработки собственных технологий и оборудования BD без использования компонентов иностранных компаний. Таким образом проблема импортозамещения в сфере микроэлектроники, в частности в серверах BD, является актуальной.

2. Отечественные процессоры и их характеристики

Сердце любого вычислительного центра – процессор. В России существует 2 крупных и перспективных разработчика процессоров: Baikal Electronics и компания МЦСТ (Московский Центр Спарк Технологий). Первая компания разрабатывает процессоры по лицензии и на архитектуре компании ARM. Вторая – процессоры «Эльбрус» на собственной архитектуре VLIW (Very Long Instruction Word), отдельные элементы которой используют в своих процессорах компания Qualcomm. Baikal Electronics и МЦСТ уже несколько лет выпускают свои процессоры для российских госслужб и рядовых пользователей.

Главной проблемой данного исследования стал тот факт, что единственным российским серверным процессором является Baikal-S, продажи которого должны начаться только в этом

4th International Workshop on Information, Computation, and Control Systems for Distributed Environments (ICCS-DE 2022), July 04–08, 2022, Irkutsk, Russia

EMAIL: beresneva@isem.irk.ru (A. 1); flower@isem.irk.ru (A. 2)

ORCID: 0000-0002-0003-947X (A. 1); 0000-0003-2219-9754 (A. 2)



2022 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

ICCS-DE 2022 Workshop Proceedings

году. При этом согласно бенчмаркам компании-разработчика он показывает результаты, сопоставимые с лидерами в этой отрасли – Intel Xeon Gold 6148 и Kunpeng 920 [2]. Поэтому было решено провести сравнение российских и иностранных процессоров, предназначенных для персональных компьютеров [3]: Baikal-M, Baikal-S, Эльбрус-8СВ и Эльбрус-16С, Intel Xeon Gold 6148, Intel Atom E3940, Intel Core i3- 7300T (Таблица 1).

Таблица 1
Характеристики процессоров

Характеристика	Baikal-M	Baikal-S, 2.5 ГГц	Эльбрус 8СВ	Эльбрус 16С	Intel Xeon Gold 6148	Intel Atom E3940	Intel Core i3- 7300T
Технологический процесс (нм)	28	16	28	16	14	14	14
Архитектура	ARM	ARM	VLIW	VLIW	x86	x86	x86
Общее количество ядер	8	48	8	16	20	4	2
Количество потоков	8	-	-	-	40	4	4
Тактовая частота ядер при тестировании (ГГц)	1,5	2,5	1,3	2	2,4	1,6	3,6
Максимальный объем оперативной памяти (ГБ)	128	768	256	1024	768	8	64
Энергопотребление (Вт)	< 30	120	110	170	150	9,5	51
Кэш первого уровня L1 в каждом ядре (КБ)	48, 32	64, 64	64, 128	64, 128	32, 32	32, 24	32,32
Кэш второго уровня L2 (КБ)	4x1024	24x512	8x512	16x1024	20x1024	2048	2x256
Кэш третьего уровня L3 (МБ)	24	24	16	32	27,5	8	64
Кэш-память 4-го уровня (МБ)	-	32	-	-	-	-	-
Год выпуска	2018	2022	2018	2021	2017	2016	2017

Данные таблицы 1 позволяют сделать ряд выводов:

- технический процесс отечественных процессоров уступает зарубежным аналогам;
- пользовательские отечественные процессоры сопоставимы по характеристикам с зарубежными;
- процессор Baikal-S действительно не уступает по своим характеристикам одному из лидеров рынка – Xeon Gold 6148.

Далее представим результаты тестирования процессоров (Таблица 2). За основу взято тестирование процессоров Baikal-M и Intel от издания Snews Zoom [4]. Все тесты взяты из открытых источников [5, 6], поэтому их перечень ограничен, а некоторые из представленных процессоров не проходили часть тестов.

Стоит отметить, что процессоры Эльбрус имеют собственную архитектуру, под которую большинство программ просто не приспособлено, поэтому часть тестов проводилась при помощи эмуляции процессора архитектуры x86, из-за чего результаты тестов ниже, чем могли бы быть. Эту проблему можно решить с помощью специально созданного программного обеспечения, которое сможет использовать в полной мере возможности этих процессоров. Например, в современных мобильных процессорах Qualcomm Snapdragon содержат

сопроцессоры на основе VLIW-архитектуры, благодаря которым скорость обработки звука и мультимедиа, а также часть цифровой обработки беспроводных сигналов возросла.

Тем не менее, результаты очень даже выдающиеся. Серверный процессор Baikal-S, как и было показано ранее, действительно способен конкурировать с лучшими серверными процессорами, даже при техническом отставании, а модели Эльбрус выдают достойные значения для использования на персональных компьютерах.

Таблица 2

Результаты тестирования процессоров

Тестирование	Baikal-M	Baikal-S, 2.5 ГГц	Эльбрус 8СВ	Эльбрус 16С	Intel Xeon Gold 6148	Intel Atom E3940	Intel Core i3- 7300T
SpecCPU Int 2017	7,92	88	10,68	24,5	100	-	-
SpecCPU FP 2017	8,01	99	16.55	43	100	-	-
Coremark	66195	799500	43008	76232	455000	39387	72238
Whetstone	16477	282900	16495	43184	162500	-	-
Geekbench 5, st	217	498	159 (x86)	211 (x86)	838	-	-
Geekbench 5, mt	1524	17000	1100 (x86)	2926 (x86)	13389	-	-
SpecCPU Int 2017	7,92	88	10,68	24,5	100	-	-
SpecCPU FP 2017	8,01	99	16.55	43	100	-	-
Coremark	66195	799500	43008	76232	455000	39387	72238
Whetstone	16477	282900	16495	43184	162500	-	-
7zip, Decompress	11557	132840	13638	33490	97000	6360	9851

3. Заключение

В заключение следует отметить, что на территории Российской Федерации производятся так же диски и серверное оборудование. Например, компании Aerodisk и Норси-транс занимаются серверным оборудованием. Норси-транс к тому же имеет соглашение с компанией Huawei по выпуску серверов на чипах Kunpeng (но их можно заменить и отечественными). Компании Микрон и Gs Nanotech занимаются производством твердотельных дисков на территории России, из-за чего не зависят от импорта напрямую, но оборудование для производства преимущественно импортное.

Таким образом, результаты проведенного исследования показали, что технологии для замещения ВД в России на отечественное аппаратное обеспечение есть уже сейчас, но нет достаточных ресурсов для обеспечения нужд государства. Правительство РФ предпринимает необходимые меры для развития ИТ-отрасли, поэтому перспективы ее развития в России, на мой взгляд, вполне оптимистичные.

4. Список литературы

- [1] А. К. Ким., В. И. Перекатов, С. Г. Ермаков, Микропроцессоры и вычислительные комплексы семейства «Эльбрус», Питер, Санкт-Петербург, 2013. URL: http://www.elbrus.ru/files/511cea/886487/1a8f40/000000/book_elbrus.pdf
- [2] Ежегодная итоговая конференция Baikal Electronics. URL: https://www.youtube.com/watch?v=NN2wa3rroaI&ab_channel=%D0%91%D0%B0%D0%B9%D0%BA%D0%B0%D0%BB%D0%AD%D0%BB%D0%B5%D0%BA%D1%82%D1%80%D0%BE%D0%BD%D0%B8%D0%BA%D1%81
- [3] Спецификации продукции Intel. URL: <https://ark.intel.com/content/www/ru/ru/ark.html>
- [4] Первый тест: на что способен российский процессор Baikal-M. URL: <https://zoom.cnews.ru/publication/item/63333>
- [5] Хабр, 2022. URL: <https://habr.com/ru>
- [6] Первые тесты процессора Эльбрус-16С. URL: <https://temofeev.ru/info/articles/pervye-testy-protssessora-elbrus-16s/>

Коды исправления ошибок, как модификация методики стеганографии

Екатерина С. Безуглова^{1,2}, Егор М. Ширяев^{1,3}, Николай Н. Кучеров^{1,3} и Виктор А. Кучуков^{1,2}

¹ Северо-Кавказский Федеральный Университет, ул. Пушкина, 1, Ставрополь, 355017, Россия

² Научно-технологический университет «Сириус», просп. Олимпийский, 1, Сочи, 354340, Россия

³ Северо-Кавказский центр математических исследований Северо-Кавказского федерального университета, г. Ставрополь, 355029, Россия

Аннотация

Всеобъемлющее и стремительное развитие технологий и Интернета актуализировало проблему безопасности данных. В связи с потребностью в обеспечении защиты и сохранности данных появился способ передачи и хранения данных, который скрывает сам факт этого – стеганография. Для сокрытия секретного сообщения в документе-носителе было предложено множество различных методов. В данной работе предложено рассмотрение кодов исправления ошибок для модификации методики стеганографии. В работе представлен код Рида-Соломона.

Ключевые слова

Коды исправления ошибок Рида-Соломона, теория кодирования, стеганография.

1. Введение

Обеспечение безопасности является основной задачей при передаче данных [1]. Одним из множества подходов для обеспечения надежной транспортировки секретной информации, является стеганография. Стеганография – способ передачи и хранения информации, скрывающий факт данного действия. Такой результат получается при помещении секретного сообщения в другое, скрывая само существование передаваемой информации [2]. Главной задачей данного способа является сохранение в тайне информации о существовании сообщения [3]. Как правило, это осуществляется таким образом, чтобы третьему лицу было сложно заметить существующее секретное сообщение среди передаваемой информации, а она представлялась ему открытой, безопасной и адекватной. В информационных технологиях у стеганографии две главные задачи. Первая – гарантия целостности данных, вторая – доказуемость прав принадлежности. Следовательно, данные должны оставаться скрытыми в сигнале хоста, даже если он проходит фильтрацию, повторную выборку, обрезку, сжатие данных с потерями и т. д. [4]. Одним из способов реализации стеганографии является разработка решения, использующего канал связи. Существующие подходы в декодируемых кодах в стеганографии встречаются с различными сложностями, например, размер секретного сообщения, сложность алгоритма декодирования и деформация [5, 6, 7]. Эти вопросы создают спрос для дальнейшего исследования в области применения кодов с исправлением ошибок в стеганографии, нахождении новых методов и алгоритмов.

2. Стеганография

^{14th} International Workshop on Information, Computation, and Control Systems for Distributed Environments (ICCS-DE 2022), July 04–08, 2022, Irkutsk, Russia

EMAIL: eshiriaev@ncfu.ru; bezuglovakaterina@mail.ru; nkuchеров@ncfu.ru; vkuchukov@ncfu.ru

ORCID: 0000-0002-2359-1291; 0000-0002-7608-0452; 0000-0003-0337-0093; 0000-0002-1839-2765



© 2022 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

ICCS-DE 2022 Workshop Proceedings

Основные составляющие для стеганографии:

- Документ-носитель (A), который содержит скрытые данные, например текст, изображение, аудиофайлы, видеофайлы, документы и т. д.;
- Секретное сообщение (B), данные в обычном или зашифрованном виде;
- Стего-функция (Fe) и обратная стего-функция (Fe^{-1});
- Ключ (K) для отображения и скрытия сообщения.

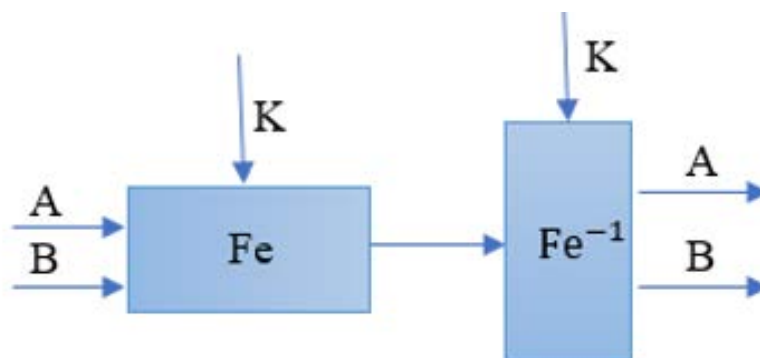


Рисунок 1: Схема типичной стеганографической операции

Существует несколько видов стеганографии: текстовая, аудио, видео, стеганография изображений и протоколов. Если рассмотреть аудио-стеганографию, то в ней секретное сообщение (B) является частью цифрового сигнала, что создает незначительные изменения двоичной аудио-последовательности. Цифровой звук, по сравнению с аналоговым аудио, является дискретным, который создается путем дискретизации непрерывного аналогового сигнала с определенной частотой. Например, в CD-аудио стандартная частота дискретизации около 44 кГц. При применении кодирования с наименьшим значащим битом двоичные образцы цифрового аудио заменяются двоичным эквивалентом секретного сообщения (B). Если задача заключается в том, что нужно вставить букву «В» (100110 в двоичном формате) в mp3-файл с выборкой 16 бит, то наименьший значимый бит 8 последовательных выборок (каждая из которых содержит 16 бит) заменяется битом двоичного кода, который соответствует букве «В».

Таблица 1

Пример встраивания буквы «В» в mp3-файл

Образец mp3-аудио	«В»	Образец mp3-аудио с закодированными данными
1001 1000 0011 1100	0	1001 1000 0011 1100
1101 1011 0011 1000	1	1101 1011 0011 1001
1011 1100 0011 1101	1	1011 1100 0011 1101
1011 1111 0111 1100	0	1011 1111 0011 1100
1011 1010 0111 1111	0	1011 1010 0111 1110
1111 1000 0011 1100	1	1111 1000 0011 1101
1101 1100 0111 1000	1	1101 1100 0111 1001
1000 1000 0001 1111	0	1000 1000 0001 1110

3. Теория кодирования

Теория кодирования – это наука о передаче информации по зашумленному каналу, с конечной целью успешного восстановления полученного сообщения с ошибками и обратного возвращения к оригиналу в безошибочном виде [8]. Передача данных происходит в битах по

каналам связи. Шумы меняют содержание сообщений, отправленных отправителем, поэтому полученное другой стороной сообщение может отличаться от исходного. Коды исправления ошибок были выведены с целью обеспечения передачи данных без ошибок по зашумленным каналам связи. Основная проблема теории кодирования заключается в обеспечении высокого уровня надежности при передаче информации по каналам связи.

3.1. Расстояние Хэмминга

Расстояние Хэмминга [8] применяется для измерения отличий двух строк битов. Расстояние Хэмминга $d_H(x, y)$ между двумя строками, а $x = x_1x_2 \dots x_n$ и $y = y_1y_2 \dots y_n$ – это количество отличных друг от друга позиций в двух строках.

$$d_H(x, y) = \sum_{j=0}^{n-1} W_H(x_j + y_j), \quad (1)$$

где $(x_j + y_j) = \{0, x_j = y_j; 1, x_j \neq y_j\}$, W_H – функция метрики Минковского.

Тогда расстояние Хэмминга можно представить как:

$$d_H(x, y) = W_H(x, y). \quad (2)$$

3.2. Коды Рида-Соломона

Коды Рида-Соломона представляют собой коды с прямым исправлением ошибок, которые были разработаны в 1960 году [8]. Коды Рида-Соломона устроены так, что, если происходит потеря или удаление информационного блока с данными, при наличии достаточного количества оставшихся данных, его можно восстановить. В кодах Рида-Соломона блок из k бит данных кодируется в n бит данных, которые являются кодовым словом, где $n > k$ для передачи данных [9].

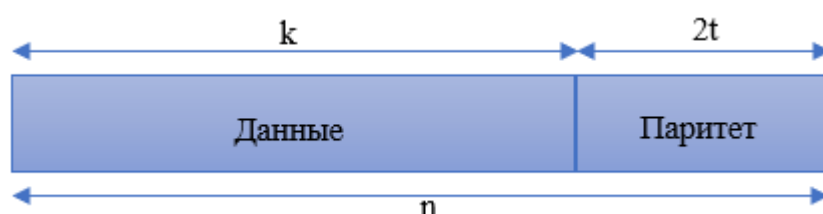


Рисунок 2: Кодовое слово Рида-Соломона в виде (n, k) , где n – длина кодового слова, k – количество бит, а $2t$ – количественный показатель паритета для каждого символа.

4. Заключение

В этой статье было проведено изучение открытой информации по стеганографии и теории кодирования. Продемонстрирован пример встраивания буквы «В» в mp3-файл. Таким образом, используя код Рида-Соломона, можно повысить эффективность стеганографии, путем контроля расстояниями Хэмминга, однако необходимо проанализировать избыточность, которую порождает код Рида-Соломона. Дальнейшая работа будет осуществляться для построения новой модели применения кодов исправления ошибок Рида-Соломона в стеганографическом приложении, а также контроль избыточности полученной системы.

5. Благодарности

Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 20-37-51004, Совета по грантам Президента Российской Федерации в рамках гранта

Президента МК-1203.2022.1.6, Северо-Кавказского центра математических исследований по договору № 075-02-2022-892 с Министерством науки и высшего образования Российской Федерации и стипендии Президента СП-3186.2022.5.

6. Источники

- [1] Кавита К.К., Кошти А., Дунгхав П. Стеганография с использованием алгоритма наименьшего значащего бита //Международный журнал инженерных исследований и приложений (IJERA), вып. – 2012. – С. 2248-9622.
- [2] Сара К. Новый метод стеганографии, основанный на алгоритме НОП (Higher Intensity Of Pixel) и умножении матриц Штрассена // Журнал глобальных исследований в области компьютерных наук. – 2011. – Т. 2. – №. 1.
- [3] Ван Х., Ван С. Кибервойна: стеганография против стегоанализа // Сообщения АСМ. – 2004. – Т. 47. – №. 10. – С. 76-82.
- [4] Бендер В. и соавт. Методы сокрытия данных //Журнал систем IBM. – 1996. – Т. 35. – №. 3.4. – С. 313-336.
- [5] Диоп И. и соавт. Новая стеганографическая схема на основе кодов Рида-Соломона //Международный журнал распределенных и параллельных систем. – 2012. – Т. 3. – №. 2. – С. 81.
- [6] Ли Б. и др. Обзор по стеганографии изображений и стегоанализу //J. Инф. Скрытие Мультим. Сигнальный процесс. – 2011. – Т. 2. – №. 2. – С. 142-172.
- [7] Шенфельд Д., Винклер А. Встраивание с синдромным кодированием на основе кодов ВСН: (исправленная версия этой статьи теперь доступна в виде полного текста. Чтобы просмотреть исходный документ, представленный в печатных материалах (с ошибками), загрузите дополнительный файл) //Материалы 8-го семинара Мультимедиа и безопасность. – 2006. – С. 214-223.
- [8] Рид И.С., Соломон Г. Полиномиальные коды над некоторыми конечными полями //Журнал общества промышленной и прикладной математики. – 1960. – Т. 8. – №. 2. – С. 300-304.
- [9] Ishengoma FR HDFS+: распределенная файловая система Hadoop на основе Erasure coding. - 2013.

Интеграция средств контейнеризации для автоматизации разработки и выполнения пакетов прикладных программ

Сергей Горский, Роман Костромин и Иван Сидоров

ИДСТУ СО РАН, Лермонтова 134, Иркутск, 664033, Россия

Аннотация

Статья посвящена развитию методов поставки конечным пользователям веб-ориентированного инструментального комплекса Orlando Tools, предназначенного для разработки и выполнения пакетов прикладных программ в распределенной вычислительной среде. Особенности эксплуатации доступных ресурсов у пользователей требуют применения различных методов и форматов распространения программного обеспечения. Эффективная разработка веб-ориентированных сервисов и приложений требует применения целого комплекса программных средств, необходимых как для самой разработки, так и для автоматизации процессов, связанных с обслуживанием и сопровождением приложений. В работе проводится сравнение трех вариантов поставки Orlando Tools в виде виртуальной машины, в виде контейнера или с использованием средств классической (ручной) установки программного обеспечения. Полученные результаты показали, что формат контейнера позволяет ускорить и упростить процесс выпуска новых версий Orlando Tools и снизить требования к системным ресурсам необходимым для его работы, облегчая пользователям организацию и проведение научных экспериментов.

Ключевые слова

Контейнеризация, пакеты прикладных программ, распределенные вычисления, автоматизация

1. Введение

Актуальным направлением разработки сложного программного обеспечения (ПО) является переход от монолитной архитектуры к модульной [1]. Каждый модуль выполняет определенную функцию или роль. Композиция модулей образует сложные связи в инструментальных комплексах и позволяет динамически масштабировать приложение в случаях возрастающей нагрузки [2]. Таким образом, общее направление по разработке ПО можно сформулировать следующим образом: большое приложение состоит из независимых модулей.

С другой стороны, на подход к созданию приложений в значительной степени влияет их внутренняя структура и развитие глобальных вычислительных сетей. Так как приложения зачастую не выполняются локально в изолированной среде, а требуют связи с внешними серверами. При этом взаимодействие с ними реализуется через веб-интерфейс.

На данный момент наблюдается большое число средств разработки веб-приложений. Это и множество языков программирования (JavaScript, PHP, Python, Ruby, Java, Perl и др.) и различных средств для них (фреймворков, библиотек, наборов инструментов и т.д.) нацеленных на разработку как веб-интерфейса системы, так и его функционального наполнения (front-end и back-end). При этом выбор средств разработки во многом зависит от предпочтений разработчиков и пользователей. Особенностью современного ПО является ситуация, когда

4th International Workshop on Information, Computation, and Control Systems for Distributed Environments (ICCS-DE 2022), July 04–08, 2022, Irkutsk, Russia

EMAIL: gorsky@icc.ru (A. 1); kostromin@icc.ru (A. 2); ivan.sidorov@icc.ru (A. 3)

ORCID: 0000-0003-0177-9741 (A. 1); 0000-0001-8406-8106 (A. 2); 0000-0002-2398-5426 (A. 3)



2022 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

ICCS-DE 2022 Workshop Proceedings

новые средства или новые версии существующих средств разработки появляются с большой интенсивностью [3].

Не зависимо от выбора средств разработки приходится решать проблему разворачивания (доставки модулей на ресурсы пользователя) такого приложений, так как процесс его установки может быть достаточно сложным. В частности, это актуально, когда у разработчиков нет возможностей протестировать работоспособность системы на широком спектре конфигураций программного обеспечения. В этом случае разрабатываемое приложение может иметь достаточно жесткую привязку к конкретным версиям системного ПО.

В целом, современная разработка ПО требует наличия: изолированной среды для разработки и тестирования; изолированной среды выполнения; средств автоматического обновления приложения; средств автоматического обновления конфигурации среды выполнения; низких накладных расходов на обслуживание среды (использования процессора, дисковой и оперативной памяти).

В рамках исследований, проводимых авторами статьи, активно развивается фреймворк Orlando Tools (OT) [4] для разработки и выполнения распределенных пакетов прикладных программ (РППП) в вычислительной среде. Обратная связь от пользователей ОТ показала необходимость развития системы, с целью соответствия требованиям, упомянутым выше. Таким образом, в статье представлены результаты интеграции средств контейнеризации системы Docker с целью автоматизированного обеспечения пользователей изолированной и легкопереносимой средой выполнения научных приложений.

2. Средства виртуализации

При разработке ПО разработчикам необходимо учитывать значительное число вариантов конфигураций целевой системы. Для этого широко используются средства автоматической конфигурации [5] или виртуализации [6]. Они позволяют оперативно создавать необходимую среду выполнения для ПО. Однако в случае, когда отсутствует возможность гибкой настройки системного ПО могут возникать проблемы совместимости [7].

Потребность создания безопасной, изолированной среды выполнения сформировалась достаточно давно, еще на заре развития UNIX-систем в 1970-ые гг. Продолжительное время только виртуализация позволяла разделить ресурсы хост-системы между различными приложениями, у которых в наличии собственное окружение в виде личной операционной системы. Однако значительные накладные расходы на изменение, миграцию и сопровождение виртуальных машин способствовали развитию иных механизмов создания изолированной среды (песочницы) – виртуализации на уровне операционной системы на основе контрольных групп и пространств имен (cgroups и namespaces соответственно) в системах на базе UNIX.

Эти механизмы позволяли использовать ядро, системные библиотеки, сеть и прочее хост-системы, но изолировать отдельные процессы. Однако их использование было не под силу неподготовленному пользователю. Революцию произвел выход системы Docker [8], которая позволила упростить работу с механизмами виртуализации на уровне операционной системы. В рамках Docker рассматриваются образы и контейнеры (по аналогии как класс и экземпляр класса), на взаимодействии с которыми и строится весь процесс работы с Docker.

Более того, контейнер – это новое осмысление исполняемого файла. Главное условие его выполнения – подходящая версия операционной системы. Контейнер (образ) содержит в себе все необходимое для запуска приложений, настроенные системные библиотеки и т.д. При этом ядро ОС, общие системные компоненты используются из родительской ОС. Благодаря этому образы контейнеров занимают в разы меньше места, чем образы виртуальных машин. Контейнеры гораздо проще переносить между системами – переносится образ и файлы данных или БД – миграция и масштабируемость приложения упрощается в разы. Особое место Docker занимает в средствах непрерывной интеграции программного обеспечения.

При использовании методов непрерывной интеграции выстраиваются цепочки (конвейер), которые сопровождают все шаги, от тестирования и компиляции написанного кода, до сборки установочных файлов и поставки на ресурсы пользователей в автоматическом режиме.

3. Orlando Tools

Orlando Tools – это специализированная среда для разработки и использования распределенных пакетов прикладных программ в гетерогенных распределенных вычислительных средах. ОТ позволяет конечным пользователям описывать вычислительную модель в XML-файле или визуальном редакторе. Вычислительная модель и созданные рабочие процессы хранятся в базе данных знаний. Для автоматизации вышеперечисленных этапов в состав ОТ включена подсистема непрерывной интеграции прикладного ПО. Эта подсистема обеспечивает работу с репозиториями Git. Веб-инструмент с открытым исходным кодом GitLab используется в качестве системы управления репозиторием.

Первая версия ОТ представляла собой централизованную систему. На данный момент ОТ включает в себя следующие компоненты:

- База данных SQL – содержит всю информацию о РППП пользователей, включая информацию о вычислительных ресурсах, репозиториях РППП и информацию о проводимых расчетах, в том числе ссылки на файлы данных.
- Файловая база данных – содержит в себе файлы данных использованные для проведения расчетов пользователями.
- Демоны Orlando Tools обеспечивающие выполнение РППП.
- Веб-интерфейс системы – средства работы пользователя с ОТ.

Эти четыре компонента устанавливаются на один вычислительный ресурс (Рисунок 1) (виртуальную машину) и требуют для своей работы настроенный веб-сервер (PHP, Apache2, Mysql). При развитии пользовательской базы возникла необходимость предоставить возможность пользователям развернуть собственные серверы ОТ в своей вычислительной среде. Причин этого может быть несколько, одна из них доступность вычислительных ресурсов только в локальной сети. Актуальной была и отладка РППП с его модулями на компьютере разработчика с дальнейшим переносом РППП на основной сервер для проведения расчетов. Для передачи РППП его дальнейшим пользователям желательно было предусмотреть публикацию готового РППП с возможностью его быстрого развертывания и использования.

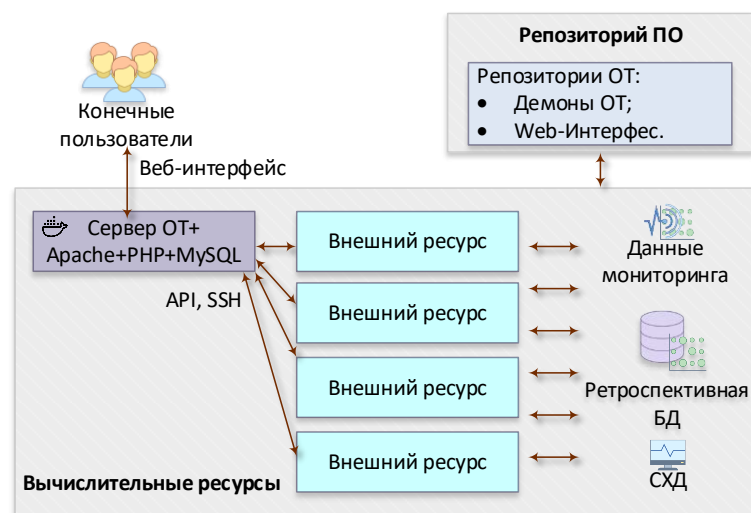


Рисунок 1: Организация систем ОТ.

Так как сервер ОТ выполнялся в виртуализированной среде, поставка новых версий ОТ вызывала ряд трудностей. Из-за того, что размер виртуальной машины превышал 2 Гб, миграция виртуальных машин была затруднительна в силу того, что в процессе работы ее размер мог сильно увеличиваться за счет базы расчетных данных и временных системных файлов. При этом даже чистка данных не позволяла уменьшить уже увеличенный размер виртуальной машины без использования архиваторов из-за особенностей работы виртуальных дисков. Вынос папок базы

расчетных данных за пределы виртуальной машины и монтирование их при запуске решало только часть проблемы, при этом усложняя процесс использования ОТ.

Таким образом, были сформулированы следующие требования к ИК Orlando:

1. быстрое и легкое развертывание системы,
2. малый объем скачиваемых данных,
3. возможность быстрого переноса данных между серверами,
4. возможность публикации РППП.

Второй и третий пункты в случае использования виртуальных машин выполнялись не в полном объеме. Первоначально рассматривалось два варианта создания контейнеров ОТ:

- первый вариант предусматривал использование популярных готовых контейнеров с сайта hub.docker.com. И создания скрипта развертывания ОТ на целевом ресурсе с использованием готовых контейнеров;
- создание своего контейнера, включающего все необходимое для работы ОТ.

В процессе работы проводилось тестирование обоих вариантов. В итоге был выбран второй вариант, как более жизнестойкий и простой для пользователя. Первый вариант был отвергнут следующим причинам:

- проблематично подобрать готовую связку контейнеров с предустановленным ПО и настроить работу ОТ с этими контейнерами;
- при модернизации ОТ может потребоваться подбор новых контейнеров.

Реализация второго варианта имеет вид, представленный на Рисунке 2. Был использован готовый открытый контейнер [9] с системами Apache, PHP, MySQL. Dockerfile использованный для его создания был оптимизирован для ОТ. Файлы баз данных могут быть заменены на внешние и вынесены из контейнера для облегчения их переноса и обновления. Для облегчения процесса разработки тоже самое было сделано с файлами веб-интерфейса и файлами демонов ОТ. То есть все компоненты ОТ могут быть при запуске заменены на внешние или использоваться из контейнера.

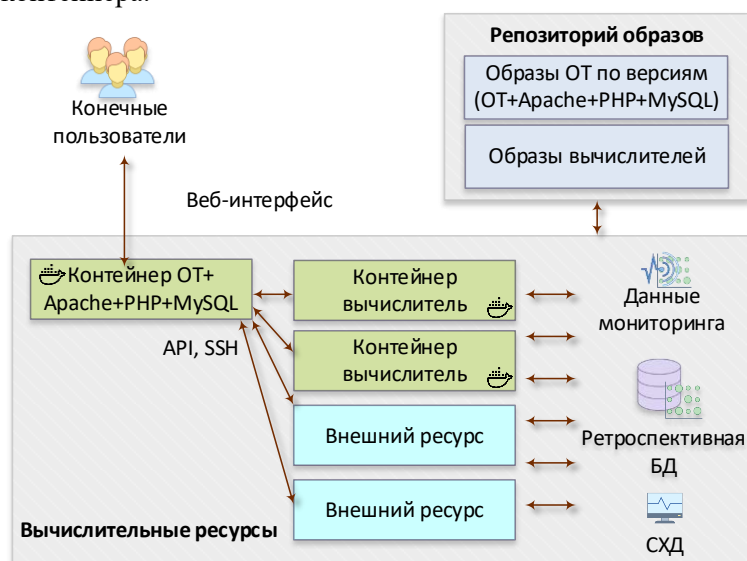


Рисунок 2: Новая композиция компонентов ОТ.

Процесс упаковки ОТ в контейнер содержит следующие этапы:

- разработка Dockerfile;
- подготовка файлов баз данных; подготовка файлов веб-интерфейса; подготовка файлов демонов ОТ.

Структура Dockerfile:

1. Выбор базового образа. Данный шаг облегчат создание базового окружения для выполняемых в контейнере программ.

2. Установка PHP, Apache, MySQL. Мало отличается от установки в обычном режиме. Дополняется созданием переменных окружения и отсутствием интерактивных шагов в установке.
3. Установка демонов Orlando. Исполняемые файлы ОТ были помещены в отдельную папку со всеми необходимыми для их работы динамическими библиотеками. Для их добавления в образ используется команда ADD.
4. Установка веб-интерфейса. Также использовалась команда ADD.

После подготовки файлов для создания образа вызывается команда сборки образа:

```
docker build --build-arg PHP_VERSION=7.4 -t orlando
```

В результате ее выполнения создается образ контейнера с именем orlando. Он может быть опубликован в специализированных репозиториях для дальнейшей загрузки и использования. Для загрузки образа, создания контейнера и его запуска используется команда:

```
sudo docker run -i -t -p "8080:80" --name=my_orlando orlando
```

4. Сравнение подходов

Для объективной оценки различных вариантов установки ОТ, мы определили ряд характеристик готовой системы, требующихся для ее работы. В Таблице 1 приведено сравнение различных вариантов установки ОТ. Практика показала, что использование контейнеров позволяет снизить требования к ресурсам и их количеству.

Таблица 1

Сравнение вариантов поставки ОТ

	Жесткий диск система, GB	Жесткий диск ОТ, GB	Оперативная память ОТ, GB	Скачиваемые данные для установки ОТ, GB
Виртуальная машина	4	5.6	1	2.2
Docker-контейнер	4.8	1.3	0.6	0.6
Установка на выделенный ресурс	5.4	0.3	0.6	0.2

Как видно из Таблицы 1, используемый ранее вариант распространения ОТ в виде виртуальных машин расходует наибольшее число системных ресурсов. Даже в случае использования единичного экземпляра системы на ресурсе. Единственное его преимущество, это облегченная установка и запуск в сравнении с установкой на выделенный ресурс. При наличии выделенного ресурса для ОТ предпочтительным будет использование непосредственной установки ОТ. Однако вариант с контейнерами не сильно ему уступает в плане использования системных ресурсов. При этом, если выделенный ресурс предназначен для запуска нескольких экземпляров ОТ или использования для других целей вариант поставки в виде контейнера Docker становится предпочтительным.

5. Заключение

В статье представлены особенности внедрения контейнеризации для облегчения разворачивания и поставки инструментального комплекса ОТ. Непрерывно продолжается разработка как ОТ, так и РППП. В процессе эксплуатации возникают трудности, которые влияют на изменение архитектуры ОТ, что позволяет сделать этот инструментальный комплекс более универсальным. Различные формы поставки расширяют спектр ресурсов, на которых может быть запущен ОТ. В частности, применение контейнеризации не заменяет существующие методы распространения ОТ, а предоставляет альтернативный подход. Практические эксперименты показали, что использование Docker-образов является достойным дополнением в арсенале разработчиков и пользователей ОТ.

6. Благодарности

Работа выполнена 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».

7. Литература

- [1] Kim G., Debois P., Willis J., Humble J. The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations. IT Revolution Press, 2016. 480 p.
- [2] Shahin M., Babar M.A., Zhu L. Continuous integration, delivery and deployment: a systematic review on approaches, tools, challenges and practices // IEEE Access, 2017. V. 5. P. 3909–3943.
- [3] Morris K. Infrastructure as Code. O'Reilly Media, 2016. 362 p.
- [4] Tchernykh A., Bychkov I., Feoktistov A. et al. Mitigating Uncertainty in Developing and Applying Scientific Applications in an Integrated Computing Environment // Program. Comput. Soft. 2020. V. 46, N 8. P. 483–502.
- [5] Костромин Р.О. Сравнительный обзор средств управления конфигурациями ресурсов вычислительной среды функционирования цифровых двойников, Информационные и математические технологии в науке и управлении. 2021. №. 1(21). С. 132–145. DOI: 10.38028/ESI.2021.21.1.011.
- [6] Eder M. Hypervisor- vs. container-based virtualization // Future Internet and Innovative Internet Technologies and Mobile Communications (IITM). 2016. V. 1. P. 1–7. DOI 10.2313/NET-2016-07-1_01.
- [7] Quigley J.M., Robertson K.L. Configuration Management: Theory, Practice, and Application. Auerbach Publications, 2017. 438 p.
- [8] Docker. URL: <https://www.docker.com> (дата обращения 17.05.2022).
- [9] Docker-lamp. URL: <https://github.com/matrayner/docker-lamp> (дата обращения 17.05.2022).

Построение областей достижимости в пространстве возмущений для оценивания робастности планов информационного взаимодействия группировки подвижных объектов

Виталий Анатольевич Ушаков¹

¹ Санкт-Петербургский государственный университет аэрокосмического приборостроения, ул. Большая Морская, д. 67, лит. А, Санкт-Петербургский, 190000, Россия

Аннотация

Предложен способ оценки робастности (устойчивости) планов информационного взаимодействия группировки подвижных объектов. Для оценки робастности предложено выполнять построение аппроксимированных областей достижимости в пространстве показателей качества планирования, для случая интервально-заданных возмущающих воздействий. Программная реализация выполнена в Matlab.

Ключевые слова

Matlab, область достижимости, оптимальное управление, пространство возмущений, логико-динамическая модель, робастность.

1. Введение

В продолжение исследования, связанного с решением задачи программного управления (планирования) информационными процессами (ИнП), происходящими при взаимодействии группировки подвижных объектов (ПдО). В общем случае ИнП включают в себя процессы получения ПдО информации о контролируемых объектах (КнО), ее обработки, хранения, и процессы передачи и приема информации конечными потребителями. Цель решения данной задачи состоит в повышении оперативности и качества управления указанными ИнП в интересах обеспечения своевременного и обоснованного принятия решения о состоянии КнО. В [1-2] было представлено разработанное и реализованное на программном уровне полимодельное описание оперативного программного управления ИнП при взаимодействии ПдО, которое, в частности, можно использовать для решения задач расчета, анализа и оптимизации показателей качества ИнП, как показано в [3].

В данной статье предлагается оценить робастность полученных планов информационного взаимодействия (ИнВ) группировки ПдО. Для этого выполняется построение аппроксимированных областей достижимости (ОД) [4-7], не по управлению, а по возмущениям на основе логико-динамических моделей (ЛДМ). В качестве исходных исходной ЛДМ может выступать как [1-2], упомянутая во введении, так и, например, ЛДМ программно-математического обеспечения процесса модернизации сложных объектов [8]. Т.е. объектом для ОД будут являться показатели качества ЛДМ. Под возмущающим воздействием будем понимать программу управления группировкой ПдО, составленную до выхода из строя одного из ПдО и как следствие более не пригодная для выполнения поставленных задач. В качестве ПдО может выступать, например, малый космический аппарат, беспилотный летательный аппарат, наземный робототехнический комплекс.

4th International Workshop on Information, Computation, and Control Systems for Distributed Environments (ICCS-DE 2022), July 04–08, 2022, Irkutsk, Russia

EMAIL: mr.vitaly.ushakov@yandex.ru (В.А. Ушаков)

ORCID: 0000-0003-0004-5439 (В.А. Ушаков)



© 2022 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

ICCS-DE 2022 Workshop Proceedings

2. Построение области достижимости по возмущениям на основе логико-динамической модели

Рассмотрим интервальный вариант задания исходных данных. В этом случае будем предполагать, что область допустимых возмущающих воздействий $\Xi(x(t), t)$ (целенаправленных, индифферентных, неизвестных, либо их комбинаций) со стороны внешней среды, будет описываться следующими соотношениями:

$$\tilde{\xi}_j^{(1)}(t) \leq \xi_j(t) \leq \tilde{\xi}_j^{(2)}(t), \quad j = 1, \dots, m, \quad (1)$$

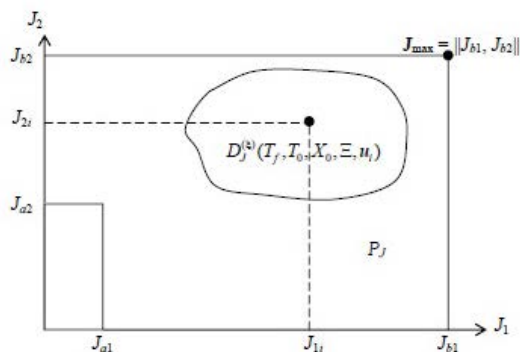
где $\tilde{\xi}_j^{(1)}, \tilde{\xi}_j^{(2)}$ – заданные векторные непрерывные функции (в зависимости от ЛДМ), определяющие возможные минимальные и максимальные значения возмущающих воздействий, которые могут появиться на этапе реализации каждого фиксированного плана функционирования группировки ПдО в рамках того или иного сценария воздействия внешней среды.

Пусть задано некоторое начальное состояние группировки ПдО $\vec{x}(t_0)$ и рассматривается некоторый фиксированный план ее функционирования $\vec{u}_i(t)$. Тогда указанным векторам и области возмущающих воздействий (1) для фиксированного сценария $\tilde{\xi}_j(t)$ соответствует область (множество) возможных значений фазовых переменных ЛДМ. Назовем это множество ОД динамической системы (группировки ПдО) под воздействием возмущений и обозначим ее как (2). Множеству (2) в пространстве состояний соответствует множество точек (область) в пространстве значений показателей (рисунок 1), оценивающих эффективность и устойчивость функционирования группировки ПдО, которое обозначим как (3).

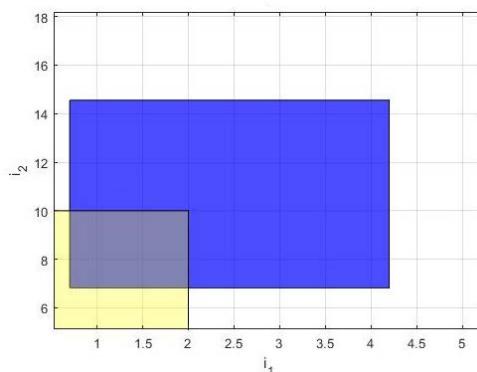
$$D_{\chi}^{(\xi)}(t_f, t_0, X_0, \Xi, \vec{u}_i). \quad (2)$$

$$D_j^{(\xi)}(t_f, t_0, X_0, \Xi, \vec{u}_i). \quad (3)$$

В данной статье рассмотрим только два компонента векторного показателя из разработанной ЛДМ, которые соответствуют показателям, характеризующим суммарное интегральное качество выполненных операций, входящих в соответствующие ИнП (J_1) и стратегию расходования ресурсов (J_p) в процессе ИнВ группировки ПдО.



1: ОД для сокращенного вектора показателей ресурсоемкости и оперативности



2: Результат построение ОД в пространстве возмущений в Matlab

На рисунке 1 показана ОД (3) для сокращённого вектора показателей качества функционирования группировки ПдО (состоящая из J_1 и J_p) вида:

$$\vec{J}' = \|J_1, J_p\|^T. \quad (4)$$

Пусть заданы допустимые границы изменений значений частных показателей функционирования группировки ПдО в виде соотношений (5) и (6), определяющих в пространстве показателей некоторую область P_j (рис. 1).

$$J_{a1} \leq J_1 \leq J_{b1}, \quad (5)$$

$$J_{a2} \leq J_2 \leq J_{b2}, \quad (6)$$

Концептуально оценка робастности планов ИнВ группировки ПдО заключается в варьировании коэффициенты важности показателей качества программного управления для различных комбинаций возмущающих воздействий. В результате получаем наборы точек для заданного количества наборов возмущающих воздействий, которые служат исходными данными для построения аппроксимированных ОД по возмущениям. На основе полученного набора точек строится аппроксимированная ОД, которая формируется как выпуклый четырехугольник. Из-за ограниченности по объему тезисов приведем сразу пример графического представления ОД (рисунок 2), реализованный на программном уровне в Matlab [9]. На рисунке 2 синим цветом показана ОД, а желтым – зона нечувствительности.

3. Заключение

Таким образом, для того чтобы оценить робастность плана ИнВ группировки ПдО к воздействию тех или иных возмущений в случае их интервального задания необходимо построить ОД, описывающую процесс программного управления группировкой ПдО в пространстве показателей качества планирования, для случая интервально-заданных возмущающих воздействий. Очевидно, что существуют различные способы построения ОД, но решения поставленной задачи предложенный метод имеет такие преимущества как геометрическое наглядное построение для лица, принимающего решение, и повышение скорости принятия решения. Предлагается на основе площади пересечения ОД и зоны нечувствительности оценивать робастность плана ИнВ группировки ПдО.

4. Благодарности

Исследования, выполненные по данной тематике, проводились при финансовой поддержке РФФИ в рамках научного проекта № 19-38-90221.

5. Список использованных источников

- [1] Ушаков В.А., Соколов Б.В., Тюгашев А.А. Полимодельное описание и алгоритмы оперативного программного управления информационными процессами на динамических сетях // Системный анализ в проектировании и управлении. – СПб.: Политех-пресс, 2021. Ч. 3. С. 75-81. DOI: [10.18720/SPBPU/2/id21-357](https://doi.org/10.18720/SPBPU/2/id21-357).
- [2] Ушаков В.А. Управление информационными процессами на динамических сетях // Математические методы и модели в высокотехнологичном производстве : сборник докладов (Санкт-Петербург, 10–11 ноября 2021). – СПб.: ГУАП, 2021. С. 168-169.
- [3] Ушаков В.А. Модельно-алгоритмическое обеспечение оперативного оценивания и анализа показателей качества управления информационными процессами // Известия высших учебных заведений. Приборостроение. 2021. Т. 64, № 8. С. 688—692.
- [4] Ушаков В.А. Области достижимости и проектирующие операторы в задачах оптимального управления // XIII Всероссийское совещание по проблемам управления (ВСПУ-2019). М.: ИПУ РАН, 2019. С. 1037-1042. DOI: [10.25728/vspu.2019.1037](https://doi.org/10.25728/vspu.2019.1037).
- [5] Толпегин О.А. Области достижимости летательных аппаратов. СПб: БГТУ ВОЕНМЕХ им. Д.Ф. Устинова, 2013. 141 с.
- [6] Толпегин О.А. Управление ракетами на основе расчёта областей достижимости // Вестник Самарского университета. Аэрокосмическая техника, технологии и машиностроение. 2015. Т. 14, № 1. С. 73-82. DOI: [10.18287/1998-6629-2015-14-1-73-82](https://doi.org/10.18287/1998-6629-2015-14-1-73-82).
- [7] Тимофеев И.С., Толпегин О.А. Приближенный расчет области достижимости космического летательного аппарата // Известия РАН. 2010. № 4(66). С. 34-39.
- [8] Захаров В. В. Программно-математическое обеспечение процесса модернизации сложных объектов // Известия высших учебных заведений. Приборостроение. 2020. Т. 63, № 11. С. 975-984. DOI: [10.17586/0021-3454-2020-63-11-975-984](https://doi.org/10.17586/0021-3454-2020-63-11-975-984).
- [9] MATLAB <https://www.mathworks.com/help/matlab/index.html>

Исследование математической модели метода сравнения чисел в СОК для четных диапазонов

Егор М. Ширяев^{1,2}, Екатерина С. Безуглова^{1,3}, Николай Н. Кучеров^{1,2} и Виктор А. Кучуков^{1,3}

¹ Северо-Кавказский федеральный университет, ул. Пушкина, 1, Ставрополь, 355017, Россия

² Научно-технологический университет «Сириус», просп. Олимпийский, 1, Сочи, 354340, Россия

³ Северо-Кавказский центр математических исследований Северо-Кавказского федерального университета, 355029, г. Ставрополь, Россия

Аннотация

Анализируя методы повышения надежности в распределенных вычислительных системах, зачастую исследователи прибегают к системе остаточных классов (СОК). Определение знака числа и сравнение чисел являются необходимыми операциями при обработке данных. Если для традиционных систем счисления эти операции являются стандартными, то для такой системы, как СОК эти операции являются сложными. И создание эффективного метода, выполняющего эти операции является открытым вопросом. В данной работе рассматривается один из наиболее перспективных методов, его математическая модель и проведен сравнительный анализ с другими моделями.

Ключевые слова

Модулярная арифметика, сравнение чисел, определения знака числа, система остаточных классов, четность числа.

1. Введение

Сравнение чисел является одной из основных операций в вычислительных системах и не представляет большой сложности при обычных вычислениях. Эта операция используется практически в любой области применения вычислительной техники, будь то математические вычисления, рендеринг, хранение данных, передача данных, извлечение данных, шифрование и т.д. В настоящее время многими учеными в их исследованиях было установлено, что применение системы остаточных классов (СОК) [1] в области распределенных вычислений повышает эффективность алгоритмов надежности [2]. В общем случае операцию сравнения можно выполнить довольно простым способом: вычесть одно число из другого и, если число отрицательное, то первое больше второго и наоборот. Многие методы в СОК работают с использованием этого свойства. Однако они часто неэффективны, поскольку для них требуется деление на большие числа или использование сложных в вычислительном отношении алгоритмов [3]. Алгоритмы определяют позиционную характеристику числа [4], которая служит основанием для определения знака числа. В этой статье мы рассмотрим новый алгоритм определения знака числа для чисел в четном диапазоне в СОК.

4th International Workshop on Information, Computation, and Control Systems for Distributed Environments (ICCS-DE 2022), July 04–08, 2022, Irkutsk, Russia

EMAIL: eshiriaev@ncfu.ru; bezuglovakaterina@mail.ru; nkuchеров@ncfu.ru; vkuchukov@ncfu.ru

ORCID: 0000-0002-2359-1291; 0000-0002-7608-0452; 0000-0003-0337-0093; 0000-0002-1839-2765



© 2022 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

ICCS-DE 2022 Workshop Proceedings

2. Система Остаточных Классов

СОК не является традиционной системой счисления [5], которая основана на Китайской теореме об остатках (КТО) [6]. СОК определяется вектором взаимно простых чисел, так называемых модулей, которые можно обозначить как $(p_1, p_2, \dots, p_n)$. Этот вектор называется базисом СОК. Базис также определяется произведением всех модулей в векторе: $P = \prod_{i=1}^n p_i$, где n является длиной вектора модулей p_i . Позиционное число X представляется в СОК в виде вектора остатков $(x_1, x_2, \dots, x_n)$, где

$$\begin{aligned}x_1 &\equiv X(\bmod p_1) \\x_2 &\equiv X(\bmod p_2) \\&\dots\dots\dots \\x_n &\equiv X(\bmod p_n)\end{aligned}$$

В этом случае накладывается ограничение на число X такое, что $X \in [0, P - 1]$, в противном случае остаток от числа X будет получен из числа X' , где $X' = X - P$. Кроме того, следствие из КТО гарантирует уникальность представления неотрицательных целых чисел из интервала $[0, P - 1]$. Для представления отрицательных чисел в СОК выполняется сложение по основанию. То есть, $-X$ будет представлено в виде $X^- = N - |X^-|$. Это свойство используется в методах определения знака числа, основанных на преобразовании числа из СОК в обобщённую позиционную систему счисления (ОПСС) [7].

Главное преимущество СОК для вычислений заключается в процессе выполнения основных арифметических операций (сложение, умножение и вычитание). Операции сложения, умножения и вычитания выполняются в соответствии со следующей общей формулой:

$$\begin{aligned}A * B &= (x_1, x_2, \dots, x_n) * (\beta_1, \beta_2, \dots, \beta_n) = \\&= ((x_1 * \beta_1) \bmod p_1), ((x_2 * \beta_2) \bmod p_2), \dots, ((x_n * \beta_n) \bmod p_n),\end{aligned}\tag{1}$$

где $*$ обозначает такие операции сложения, умножения или вычитания.

Данные операции являются модульными или «простыми» для СОК, другие, как например сравнение и определение знака – «вычислительно сложные» [8].

3. Традиционные методы в СОК

3.1. Метод Китайской Теоремы об Остатках

Существует способ преобразования числа из СОК на основе КТО в ОПСС [5]. Для этого используется следующая формула:

$$X = \left| \sum_{i=1}^n P_i \cdot x_i \cdot |P_i^{-1}|_{p_i} \right|_P,\tag{2}$$

где $P_i = \frac{P}{p_i}$ и $|P_i^{-1}|_{p_i}$ является мультипликативной обратной величиной P_i по модулю p_i . Затем восстановленный номер необходимо сравнить с $P/2$. Если $X < P/2$, тогда число больше нуля, в обратном случае меньше.

Алгоритм этого метода заключается в следующем:

Алгоритм 1

Вход: $(x_1, x_2, \dots, x_n), (p_1, p_2, \dots, p_n)$ $P = \prod_{i=1}^n p_i, P_i = \frac{P}{p_i}$, для $i = \overline{1, n}$.

Выход: S

1. $sum = 0$
2. Для i до n выполнить:
 - 2.1 $sum += x_i \left(\frac{1}{P_i} \right)_{(p_i)P_i}$
3. $X = sum \bmod P$
4. Если $X < P/2$:

5. $S = 1$

6. **Иначе:**

7. $S = 0$

Однако у этого метода есть существенный недостаток. Для того чтобы получить номер X , необходимо выполнить деление на большое число P , что при большом значении чисел значительно увеличит время, необходимое для вычислений.

3.2. Метод четного диапазона

Поскольку диапазон СОК является четным числом, один из модулей СОК является четным числом. Без потери общности мы предполагаем, что модуль p_n является четным, поэтому с помощью свойства $\left\lfloor \frac{\left\lfloor \frac{x}{a} \right\rfloor}{b} \right\rfloor = \left\lfloor \frac{x}{a \cdot b} \right\rfloor$ [9] определение знака сводится к двухэтапному алгоритму: первый этап – деление на целое число $P_n = \frac{P}{p_n}$, второй этап – деление на целое число $\frac{p_n}{2}$, и это формально математически имеет следующий вид:

$$S(X) = \left\lfloor \frac{2 \cdot X}{P} \right\rfloor = \left\lfloor \left\lfloor \frac{X}{P_n} \right\rfloor \cdot \frac{2}{p_n} \right\rfloor. \quad (3)$$

Преобразуем данную формулу, с точки зрения приближенной позиционной характеристики. В качестве позиционной характеристики функция:

$$f(X) = \left\lfloor \sum_{i=1}^n \bar{k}_i x_i \right\rfloor_{2^N} \quad (4)$$

где $\bar{k}_i = \left\lfloor \frac{2^N |P^{-1}|_{p_i}}{p_i} \right\rfloor$.

Доказательство точности приближения представлено в [10].

Алгоритм 2. Сравнение чисел в СОК с четным диапазоном с $f(X)$

Ввод: $X \xrightarrow{\text{СОК}} (x_1, \dots, x_n)$, $Y \xrightarrow{\text{СОК}} (y_1, \dots, y_n)$, $\bar{k}_i = \left\lfloor 2^N \cdot |P_i^{-1}|_{p_i} / p_i \right\rfloor$, для $i = \overline{1, n-1}$, где $N = \lceil \log_2(-n + n \cdot P) \rceil$.

Вывод: $X > Y$ – '10', $X < Y$ – '01', $X = Y$ – '00'.

1. $f_X = 0$; $f_Y = 0$;
2. Для $i := 1$ до n выполнить:
 - 2.1. $f_X = |f_X + \bar{k}_i \cdot x_i|_{2^N}$;
 - 2.2. $f_Y = |f_Y + \bar{k}_i \cdot y_i|_{2^N}$;
3. Если $f_X > f_Y$ тогда вернуть '10'
4. Если $f_X < f_Y$ тогда вернуть '01'
5. Иначе «00»

Алгоритм 2 имеет $2n$ умножений и n сложений.

Предлагаемый подход снижает вычислительную сложность алгоритма сравнения чисел в СОК.

4. Заключение

На основании проведенного исследования математических моделей можно заключить следующее. Во-первых, в предлагаемом методе исключается деление на большое число P . Оба

метода требуют одинаковое количество сложений и умножений. Предлагаемый метод использует приближенную позиционную характеристику. Также предложенный алгоритм позволяет реализовать метод без использования больших чисел при длине модулей от 32 до 64 бит. Из отрицательных характеристик можно выделить то, что необходимо большее количество предвычисленных характеристик.

Используя предложенный метод возможно уменьшить вычислительную нагрузку на систему, тем самым эффективно повысить надежность распределенных систем.

5. Благодарности

Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 20-37-51004, Совета по грантам Президента Российской Федерации в рамках гранта Президента МК-1203.2022.1.6, Северо-Кавказского центра математических исследований по договору № 075-02-2022-892 с Министерством науки и высшего образования Российской Федерации и стипендии Президента СП-3186.2022.5.

6. Источники

- [1] Червяков Н. И. Методы, алгоритмы и техническая реализация основных проблемных операций, выполняемых в системе остаточных классов //Инфокоммуникационные технологии. – 2011. – Т. 9. – №. 4. – С. 4-12.
- [2] Chervyakov N. I., Babenko M. G., Shabalina M. N. Development of a secure system for distributed data storage and processing in the clouds based on the concept of active security in RNS //2017 XX IEEE International Conference on Soft Computing and Measurements (SCM). – IEEE, 2017. – С. 558-560.
- [3] Червяков Н. И. и др. Приближенный метод выполнения немодульных операций в системе остаточных классов //Фундаментальные исследования. – 2012. – Т. 1. – №. 6.
- [4] Исупов К. С. Метод выполнения немодульных операций в системе остаточных классов на основе интервальных позиционных характеристик //Фундаментальные исследования. – 2013. – №. 4-3. – С. 566-570.
- [5] Sousa L., Antao S., Martins P. Combining residue arithmetic to design efficient cryptographic circuits and systems //IEEE Circuits and Systems Magazine. – 2016. – Т. 16. – №. 4. – С. 6-32.
- [6] Chervyakov N. I., Lyakhov P. A., Babenko M. G. Digital filtering of images in a residue number system using finite-field wavelets //Automatic Control and Computer Sciences. – 2014. – Т. 48. – №. 3. – С. 180-189.
- [7] Лебедева А. В., Сплюхин Д. В., Мартынова И. А. Система счисления рядов упорядоченных множеств //МАТЕМАТИКА И МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ. – 2018. – С. 21.
- [8] Амербаев В. М., Малашевич Д. Б. Анализ эффективности реализации модульных операций индексной модулярной арифметики //Известия высших учебных заведений. Электроника. – 2009. – №. 6. – С. 54-57.
- [9] Исупов К. С. Способ уточненного вычисления приближенной позиционной характеристики для выполнения немодульных операций в системе остаточных классов //Фундаментальные исследования. – 2013. – Т. 4. – №. 4.
- [10] Chervyakov N. et al. AR-RRNS: Configurable reliable distributed data storage systems for Internet of Things to ensure security //Future Generation Computer Systems. – 2019. – Т. 92. – С. 1080-1092.

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 15.08.2022 г.
Формат бумаги 60×84 1/16, объем 7,79 п.л.
Заказ 1. Тираж 1 экз.

Отпечатано в ИДСТУ СО РАН