Computing, Telecommunication and Control, 2024, Vol. 17, No. 1, Pp. 10–19.
Информатика, телекоммуникации и управление. 2024. Том 17, № 1. С. 10–19.

Intelligent Systems and Technologies
Интеллектуальные системы и технологии

Research article
DOI: https://doi.org/10.18721/JCSTCS.17101
UDC 62-503.55

PROGRAMMING OF OPEN DISTRIBUTED INDUSTRIAL SYSTEMS BASED ON THE INTERNATIONAL STANDARD IEC 61499

V.V. Potekhin[1] ✉, A.P. Alekseev[1], E.V. Kuklin[1], A.E. Misnik[2], Ya.D. Khitrova[1]

[1] Peter the Great St. Petersburg Polytechnic University, St. Petersburg, Russian Federation;

[2] Belarusian-Russian University, Mogilev, Republic of Belarus

✉ Slava.Potekhin@Spbstu.Ru

Abstract. Today, collaboration in software development and open architectures is changing the fundamental structure of business and reshaping the way organisations operate in a highly competitive environment, forcing them to rethink strategies. Organisations that previously created proprietary systems are beginning to develop open source products to expand the boundaries of the industries in which they operate. In a globalised world, open integrated control systems are becoming increasingly important. Their main goal is to create a balanced, efficient and functional system that integrates various aspects into a coherent whole. The article shows the advantages and disadvantages of using a new approach to programming logic systems based on the international standard IEC 61499 in the field of industrial automation of technological processes. The article analyses basic principles of the IEC 61499 standard, as well as general provisions with the OPAS standard. It also demonstrates a prototype of the control system for the model of the furnace P-101 according to IEC 61499. And provides time characteristics of the software based on IEC 61499 for real-time operating systems.

Keywords: Industry 4.0, OPAS, Cloud DCS, Cloud computing, Industrial Internet of Things, cyber-physical systems

Citation: Potekhin V.V., Alekseev A.P., Kuklin E.V., Misnik A.E., Khitrova Ya.D. Programming of open distributed industrial systems based on the international standard IEC 61499. Computing, Telecommunications and Control, 2024, Vol. 17, No. 1, Pp. 10–19. DOI: 10.18721/JCSTCS.17101

Научная статья
DOI: https://doi.org/10.18721/JCSTCS.17101
УДК 62-503.55

ПРОГРАММИРОВАНИЕ ОТКРЫТЫХ РАСПРЕДЕЛЕННЫХ ПРОМЫШЛЕННЫХ СИСТЕМ НА ОСНОВЕ СТАНДАРТА МЭК 61499

В.В. Потехин[1] ✉ , *А.П. Алексеев*[1], *Е.В. Куклин*[1], *А.Е. Мисник*[2], *Я.Д. Хитрова*[1]

[1] Санкт-Петербургский политехнический университет Петра Великого, Санкт-Петербург, Российская Федерация;

[2] Белорусско-Российский университет, Могилёв, Республика Беларусь

✉ Slava.Potekhin@Spbstu.Ru

Аннотация. Сегодня сотрудничество в сфере разработки программного обеспечения и открытых архитектур меняет фундаментальную структуру бизнеса и перестраивает методы работы организаций в условиях жесткой конкуренции, что заставляет переосмыслить стратегии. Организации, которые ранее создавали проприетарные системы, начинают разрабатывать продукты с открытым исходным кодом, чтобы расширить границы отраслей, в которых они работают. В глобальном мире открытые интегрированные системы управления становятся все более значимыми. Их основная цель — создать сбалансированную, эффективную и функциональную систему, которая объединяет различные аспекты в единое целое. В статье показаны преимущества и недостатки использования нового подхода к программированию логических систем на основе стандарта МЭК 61499 в области промышленной автоматизации технологических процессов. Анализируются основные принципы стандарта МЭК 61499, а также общие положения стандарта OPAS. Демонстрируется прототип системы управления модели печи объекта П-101 по МЭК 61499. Приведены временные характеристики программного обеспечения на основе МЭК 61499 для операционных систем реального времени.

Ключевые слова: Индустрия 4.0, OPAS, облачная РСУ, промышленный интернет вещей, кибер-физические системы

Для цитирования: Potekhin V.V., Alekseev A.P., Kuklin E.V., Misnik A.E., Khitrova Ya.D. Programming of open distributed industrial systems based on the international standard IEC 61499 // Computing, Telecommunications and Control. 2024. Т. 17, № 1. С. 10–19. DOI: 10.18721/JCSTCS.17101

Introduction

Industry 4.0 has become a new stage in the development of modern industry. Large companies seek to reduce the cost of production through the introduction of new, more intelligent and transparent control systems. As a result, new standards, architectures and specifications are currently being actively developed. To survive in business, industrial manufacturers, like all companies, must continually improve productivity and customer satisfaction. The industrial control systems that manufacturers use to automate their processes are critical to a company's productivity and product quality [1–3].

For a number of reasons, currently installed control systems are predominantly closed and proprietary. This contrasts with the open, interacting network of instrumentation below them and the information technology systems above them in a typical automation hierarchy. Closed proprietary systems are expensive to update and maintain, and problems arise when trying to implement new technologies, especially from third parties. This is a problem that the Open Process Automation™ Forum (OPAF), launched by the Open Group, is working on [4–9]. OPAF defines standards for an open, interoperable, and secure process automation architecture. The standards allow systems to be developed for the intended purpose, consisting of interconnected functional elements purchased from independent vendors and easily integrated using a modular architecture characterized by open standard interfaces between elements. The first priority is to select standards from existing applicable industry standards. When an applicable standard does not exist, OPAF works with standards development organizations to create new standards. OPAF and its standards will not define the functional intellectual property (IP) of products. They remain the property of their supplier. The goal is to define open standard interfaces rather than require the development and distribution of IP components.

The scope of OPAF covers modern distributed control systems (DCS) and programmable logic controllers (PLC) for continuous and hybrid processes.

One of the most fashionable trends is the development of cloud-based DCS. The main idea is to replace the physical PLC with a virtual one. The main advantages of this approach are:

- System scalability – when expanding production, less effort is spent on assembling control cabinets, PLC setup becomes easier.
- Decentralization of the control system – if the control system consists of a small number of devices, in the event of a malfunction the entire system may cease to function. If you distribute the parts of the control system, the system will be more resistant to failures.
- Independence from vendors – having purchased equipment from one company, it becomes almost impossible to buy a more profitable and high-quality analogue on the market, since already installed components cannot interact with components from other manufacturers, that is, they are proprietary. As a result, the industry becomes dependent on manufacturers of industrial equipment.

A new approach to the development of control systems requires new design methods. Thus, the IEC 61131 standard has been criticized in recent years due to its inconsistency with the requirements of modern software development methods [10]. Modern software architectures of industrial process measurement and control systems, such as 61131-3, do not conceptually support reconfiguration and distribution. On the other hand, portability, configurability, interoperability, reconfiguration and distribution have been defined in [5] as requirements for future automation systems. To address the limitations as well as the new challenges in the development of industrial automation systems, a technical committee of the International Electrotechnical Commission (IEC) was tasked with developing a new standard. The standard is called IEC 61499 [11]. It is worth noting that this standard was developed by the same people as the IEC 61131 standard. The most important difference between IEC 61499 and IEC 61131 is that the processing of the former is event-based, and the latter is cyclic. And now let us consider the main provisions of the IEC 61499 standard, evaluate it in comparison with the IEC 61131 standard, and then test it on the control model in the 4diac environment.

The IEC 61499 standard, which was conceived many years ago and extensively described in numerous scientific publications, has yet to find its place in industrial enterprises. Furthermore, there is a scarcity of operational demonstration setups that reflect the efficiency of a distributed control system based on the IEC 61499 standard. The purpose of this article is to investigate the standard using an industrial prototype and analyze the reliability of data transmission between the blocks of the DCS.

IEC 61499 standard research

Description of standard

The IEC 61499 standard is an extension of the IEC 61131 standard. All the functions that are specified in IEC 61131-3 must be implemented in the IEC 61499 standard. In addition, although the methodology for moving from the IEC 61131 architecture to IEC 61499 is still at the stage of scientific discourse, it is already clear that the standards complement each other, since IEC 61499 is an event-driven architecture with a function block shell, inside which the logic can be described in any programming language.

For example, IEC 61131 blocks in IEC 61499 use the E_CYCLE function block, which simulates PLC cycling. The block is shown in Fig. 1.

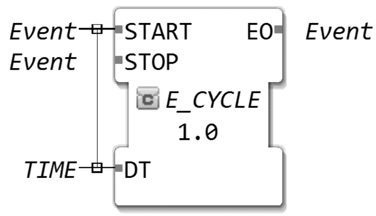

Fig. 1. Function block E_CYCLE in 4diac

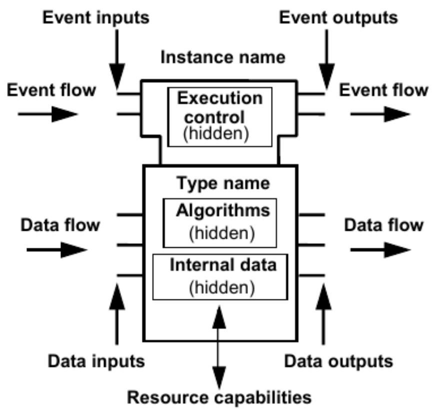

Fig. 2. Standard function block

The result of the requirements set before the developers of the standard was the properties of the system, which largely intersect with the OPAS architecture.

About portability and configurability. The developers of the standard see a solution to this problem using XML markup. The entire project interface is contained in files with XML markup of a certain syntax and architecture for describing these files. Thus, the project can be opened in any software that supports the standard. Although the standard did not explicitly define the syntax, which led to inconsistencies and a departure from the main idea of portability, software developers solved this problem by creating conformance profiles – documentation that describes the rules by which software should be developed. One such compliance profile is HOLOBLOC.

About interoperability. Interoperability is defined as the ability of dissimilar devices to interact to achieve mutually beneficial and agreed common goals, including the exchange of information and knowledge between them through the behavior they support, as well as through the exchange of data. To achieve interoperability it is supposed to use OPC UA.

About reconfigurability. For hot loading devices developers suggest using reconfiguration blocks that allow you to change the structure of the program by changing the connections between blocks, deleting blocks, creating new instances, setting parameters.

About distribution. Distribution in IEC 61499 is an application interface within which functional blocks are marked with different colors, each of which refers them to a specific resource model of a certain device. Thus, by setting colors for function blocks in the application window you can change the specific location where these blocks should be loaded.

Architecturally, in IEC 61499, an event-based program execution model replaces the cyclic one. In this way, it is possible to specify the explicit order of execution of function blocks in a program. The most important function block model in the standard, which differs from IEC 61131 in that it adds event inputs and outputs, is shown in Fig. 2.

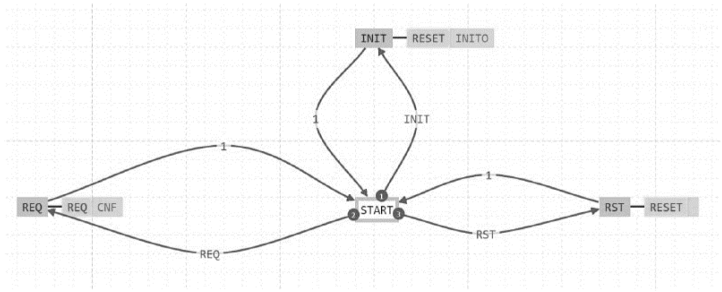

Fig. 3. Execution control diagram in 4diac

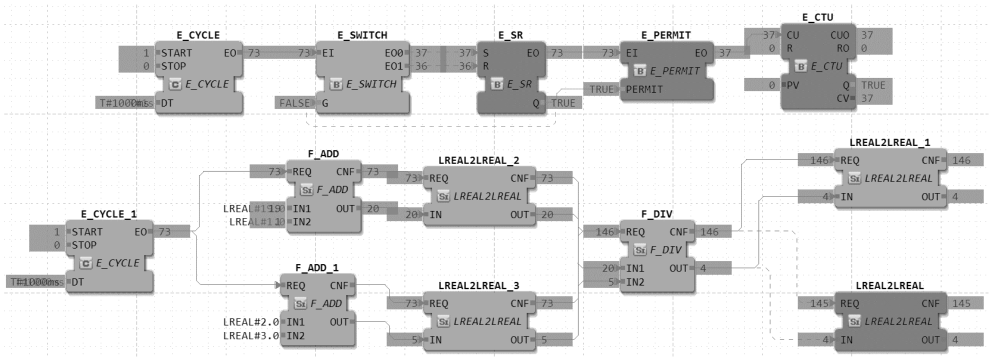

Fig. 4. Distributed application in 4diac

Events are processed through an execution control diagram, which is a state machine that has states, transition conditions, transitions, and state-related algorithms, upon execution of which function block output events are generated. A typical execution control diagram is shown in Fig. 3.

There are no global variables in IEC 61499, and the atomic unit is a functional block within which the logic of the program component is defined. Fig. 4 shows what a standard IEC 61499 application looks like.

Nodes of a distributed system:

- Raspberry Pi – green
- Virtual machine on Astralinux – brown

OPAS Relationship

Truly open, interoperable automation and control is the future of the manufacturing industry. Many potential benefits are within reach.

This promise of real interoperability was made possible in large part by the efforts of the OPAF, co-founded and led by Schneider Electric.

OPAF integrates IEC 61499 into O-PAS because it can model distributed data and control systems, which helps provide the interoperability that users need. Schneider Electric believes that IEC 61499 will be the key to providing openness and interoperability with a high level of flexibility, and also will be able to work with existing hardware platforms.

For example, the Schneider Electric Innovation and Research Center in Dhahran, Saudi Arabia, partnered with Saudi Aramco to set up an OPA testbed that is divided into multiple development and demonstration areas that host distributed control nodes (DCNs) from different vendors. OPAS compliant that communicate over their real-time bus based on the OPC UA protocol and support cyber security. This setup is similar to ExxonMobil's test facility in Texas, which has been in operation since 2019 and was one of the first to implement IEC 61499 and tested OPAS compatibility and portability.

To help potential users learn and implement O-PAS and IEC 61499 applications and help create this new ecosystem, Schneider Electric is also promoting a concept they call universal automation. It is a manufacturing approach to automation using portable software components that can run on different vendors' control platforms.

The goal of OPAF is interoperable and portable control. IEC 61499 is only a technical tool.

Laboratory prototype

To test the standard, a test bench was developed based on the open source software 4diac Eclipse [12–15].

The soft PLC test bench included the following open environment system components:

- Open source software development environment 4diac IDE.
- Real-time runtime 4diac FORTE.
- Function block library.

The equipment used for testing is presented in Table 1.

Table 1

Laboratory devices

№	Name	Function	Number
1	Embedded computer AntexGate	Application distribution node	1
2	Industrial embedded computer Front Compact	Application distribution node	1
3	Panel PLC RealLab BLcon-LXD18	Application distribution node	1
4	Single board microcomputer Raspberry Pi	Application distribution node	1
5	Astra Linux VM	Application distribution node	1
6	Phoenix Contact PLC Next	Plant Simulator Controller	1
7	I/O device OWEN M210	I/O device control object	2
8	I/O device RealLab I/O	Control input/output device	2
9	MasterSCADA 4D VM	Visualization of the control object (SCADA application)	1

A SCADA application is deployed on the virtual machine to visualize the control object.

The simulator of the P-101 furnace model, developed in the Matlab Simulink program, acts as a control object. The model is imported into the Next Phoenix Contact PLC controller.

Depending on the test scenario, the OPC UA or Modbus TCP protocols were used.

The application was distributed to one or more nodes from the device model: AntexGate, Front Compact, RealLab PLC, Raspberry Pi, virtual machine in the cloud.

Scenario 1. Complex scenario for managing the object model

The general scheme of scenario 1 is shown in Fig. 5. The user sets the setpoints/tasks (SV) of the PID controllers in MasterSCADA, which are transmitted via Modbus TCP to the 4diac application. 4diac Eclipse implements a distributed application that feeds control actions (MV) through the gateway and I / O devices to the P-101 furnace object model. The model sends the PV process value parameter back to the 4diac application.

In this scenario the following characteristics were set:

– Ability to interact with I/O devices.

– Formation and transmission of control actions from a distributed application through an input/output device.

– Support for Modbus TCP protocol.

Results:

1. The ability to interact with I/O devices through network protocols has been confirmed.

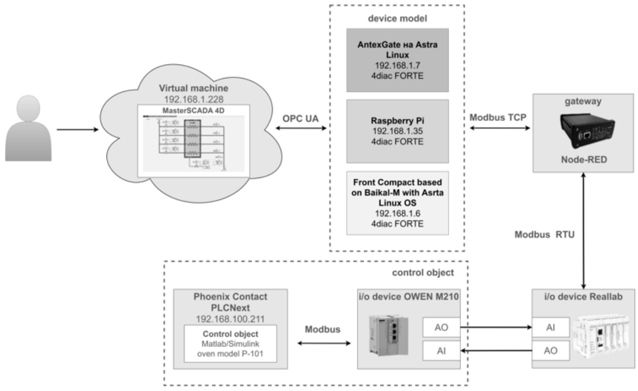

Fig. 5. General scheme of scenario 1

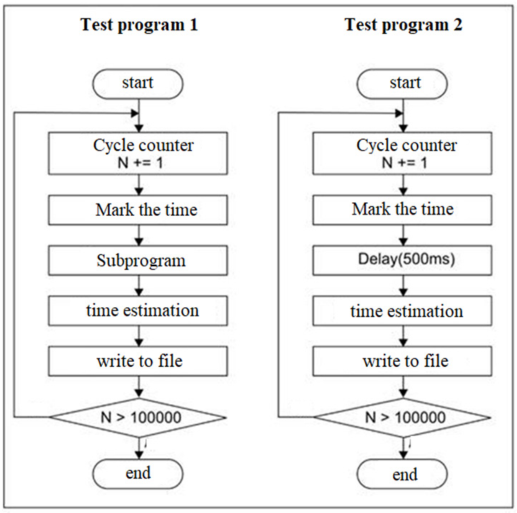

Fig. 6. Programs test diagram

2. Control actions were generated and transmitted from a distributed application via the Modbus TCP protocol.

Scenario 2. Time characteristics of program execution

In this scenario the timing of the execution of the 4diac program on different kernels of the Linux OS is determined.

Testing tools:

1. PC with Linux OS installed (Ubuntu distribution kit) in a virtual machine.
2. Linux kernel realtime patch (linux-image-rt meta-package).
3. Lowlatency patch of the Linux kernel.
4. Development environment 4diac IDE.
5. Execution environment 4diac FORTE.
6. Two test programs, executed in accordance with the diagrams in Fig. 6:

– test programs must be executed on the same device (not be distributed);

– the subroutine in test program No. 1 should not have branches and Delay functions;

– the accuracy of time stamp determination should be at least 1 ms.

Recording must be done in a CSV file.

Linux Kernel versions:
– Linux 5.15.0-41-generic #44~20.04.1-Ubuntu SMP Fri Jun 24 13:27:29 UTC;
– Linux 5.10.0-13-rt-amd64 #1 SMP PREEMPT_RT Debian 5.10.106-1 (2022-03-17);
– Linux 5.4.0-122-lowlatency #138-Ubuntu SMP PREEMPT Wed Jun 22 15:43:23 UTC.
Results:
1. The results of running program No. 1 (cycle with calculations) are presented in Table 2.

Table 2

Start program No. 1

Percentile	Kernel Generic, ms	Kernel RT, ms	Kernel LowLat, ms	Kernel Generic, ms	Kernel RT, ms	Kernel LowLat, ms
	Period 200 ms			Period 50 ms		
1 (min)	178	198	190	The process was interrupted at cycle 22	The process was interrupted at cycle 17	The process was interrupted at cycle 19
50 (average)	187	546	398			
99 (max)	385	723	623			

If the program execution time is close to the period of execution events, then the standard Linux kernel works better due to the fact that the system is less distracted by its own tasks.

If the expected program execution time is significantly longer than the startup period, then the actual time gradually increases, event omissions appear and event queue overflow errors occur, up to the program hang.

2. Run program No. 1 (loop with calculations) on a system with a full processor load.

– Specified execution period 200 ms.

Generic and LowLatency kernels behave the same way. The execution time quickly grows up to several seconds, which causes the program hang.

On the RT core, the program execution time increased from 185 ms to 600±150 ms. There were missed events without any program hang for 1000 cycles.

– Specified execution period 1000 ms.

There were no missing events. Average execution time increased from 185 ms to: 240 ms for the Generic kernel; 280 ms for the LowLatency core; 600 ms for RT core.

When the processor parallel load is removed, the program execution time smoothly returns to 185 ms.

3. Starting program No. 2 (Delay = 300 ms) with a period of 200 ms.

On all cores the behavior of the program is identical: the actual delay time was 100±5ms in 95% of measurements; out of 1000 launch events 500 were skipped.

Summary:

1. If the processor is not loaded with parallel computing by 100%, the program execution time does not depend on the Linux kernel.

2. If the program launch period is significantly longer (at least 5 times) than its execution time, then no errors occur during program operation even on a loaded system with any kernel.

3. When the processor is loaded, the program execution time increases the most on the RT core (3 times). On a standard core – by 25%.

4. If the program launch period is less than its execution time, then the events are added to the queue. When it overflows, skips of events occur, and the program may also hang.

5. The RT core works more stably on a loaded system, but it does not fully guarantee the absence of missing events when the queue overflows.

6. To confirm the function of stable operation in real time, it is necessary to refine the hardware platform of the stand and methods and conduct additional tests.

Conclusion

The Open Process Automation initiative aims to enhance the full lifecycle benefits of industrial control systems through the use of a standards-based, open, secure, interoperable architecture and open business model. The standard based on this initiative uses the "standard of standards" approach. One of the standards used is IEC 61499.

The IEC 61499 standard defines the development language for industrial DCS. It extends the IEC 61131-3 standard:

– by improving the encapsulation of software components, making them easy to reuse;
– by providing a vendor-independent format and simplifying device-to-device communications.

The ideas described in the standard can be considered interesting in relation to DCS. The standard incorporates elements of object-oriented and component approaches, which thus takes industrial programming to a new level of abstraction, more convenient for DCS.

The IEC 61499 standard was tested using the 4diac Eclipse software. During testing the possibility of using the functional blocks, defined in the IEC 61131-3 standard, in the IEC 61499 standard was checked, as well as interaction on physical signals and distribution of the application to physical nodes.

Among the interesting directions for the development of the IEC 61499 standard within OPAS is the development of an architecture for the reconfiguration of distributed control nodes in real time. Since the control program is distributed, the reconfiguration of one of the nodes requires an understanding of the conditions under which the reconfigurable node will be fault-tolerant, that is, it will not stop the entire control system and will not give incorrect results during the reconfiguration process.

However, based on experiments conducted, the blocks of the IEC 61499 standard quickly overwhelm the developer's screen. The most well-known open-source software based on IEC 61499, with a large developer community, 4diac IDE and FORTE, is still unreliable. The data processing speed between blocks in a distributed control system based on IEC 61499 is insufficient compared to a monolithic system based on IEC 61131-3, where the standard cycle time is considered to be no more than 100 ms.

REFERENCES

1. **Bartusiak R.D., Bitar S., DeBari D.L., Houk B.G., Stevens D., Fitzpatrick B., Sloan P.** Open Process Automation: A standards-based, open, secure, interoperable process control architecture. Control Engineering Practice, 2022, Vol. 121, pp. 2–3.

2. **Church P.C., Mueller H., Ryan C., Gogouvitis S.V., Goscinski A., Tari Z.** Migration of a SCADA system to IaaS clouds – a case study. Journal of Cloud Computing, 2017, Vol. 6, pp. 6–12.

3. **Lizhi Wang, Qin Yanyuan, Zefan Tang, Peng Zhang.** Software-Defined Microgrid Control: The Genesis of Decoupled Cyber-Physical Microgrids. IEEE Open Access Journal of Power and Energy, 2020, Vol. 7, pp. 173–182 .

4. International Electrotechnical Commission. International Standard IEC 61131–3: Programmable Controllers, Part 3: Programming Languages, IEC. 2003, pp. 79–83.

5. **Zoitl A., Strasser T.I., Sünder C., Baier T.** Is IEC 61499 in harmony with IEC 61131-3? IEEE Industrial Electronics Magazine, 2009, Vol. 3, pp. 49–55.

6. **Otto A., Hellmans K.** IEC 61131: A General Overview and Emerging Trends. IEEE Industrial Electronics Magazine, 2009, Vol. 3, pp. 27–31.

7. **Vrba P., Tichý P., Mařík V., Hall K.H., Staron R.J., Maturana F.P., Kadera P.** Rockwell Automation's Holonic and Multiagent Control Systems Compendium. IEEE Transactions on Systems, Man and Cybernetics, Part C: Applications and Reviews, 2011, Vol. 41, pp. 14–30.

8. **Strasser T.I., Zoitl A., Christensen J.H., Sünder C.** Design and Execution Issues in IEC 61499 Distributed Automation and Control Systems. IEEE Transactions on Systems, Man and Cybernetics, Part C: Applications and Reviews, 2011, Vol. 41, pp. 41–51 .

9. **Zoitl A., Vyatkin V.** Different perspectives [Face to face; "IEC 61499 architecture for distributed automation: The "glass half full" view]. IEEE Industrial Electronics Magazine, 2009, Vol. 3, pp. 7–23.

10. International Electrotechnical Commission. International Standard IEC 61131–3: Function Blocks, Part 1 – Part 4, IEC. 2005, pp. 45–70.

11. **Yoong L.H., Roop P., Vyatkin V., Salcic Z.** A Synchronous Approach for IEC 61499 Function Block Implementation. IEEE Transactions on Computers, 2009, Vol. 58, pp. 1599–1614.

12. **Potekhin V.V., Alekseev A.P., Kuklin E.V., et al.** Cloud distributed control system based on open process automation platform. Computing, Telecommunications and Control, 2023, Vol. 16, No. 2, pp. 17–28.

13. **Potekhin V.V., Selivanova E.N., Katalinič B.** Development of a digital transformation model for industrial cyber-physical systems. AIP Conference Proceedings, 2022, Vol. 2456.

14. **Katalinić B., Kostenko D., Onufriev V.A., Potekhin V.V.** Cyber-Physical Systems in Complex Technologies and Process Control. Cyber-Physical Systems and Control, 2019, Vol. 95, pp. 40–54.

15. **Alekseev A.P., Efremov V.V., Potekhin V.V., Zhao Y., Du H.** Digital Twin Analytic Predictive Applications in Cyber-Physical Systems. Cyber-Physical Systems and Control, 2019, Vol. 95, pp. 368–377.

INFORMATION ABOUT AUTHORS / СВЕДЕНИЯ ОБ АВТОРАХ

Potekhin Vyacheslav V.
Потехин Вячеслав Витальевич
E-mail: Slava.Potekhin@Spbstu.Ru
ORCID: https://orcid.org/0000-0001-9850-9558

Alekseev Anton P.
Алексеев Антон Павлович
E-mail: Alekseev.Ap@Edu.Spbstu.Ru

Kuklin Egor V.
Куклин Егор Вадимович
E-mail: Kuklin.Ev@Edu.Spbstu.Ru

Misnik Anton E.
Мисник Антон Евгеньевич
E-mail: anton@misnik.by

Khitrova Yana D.
Хитрова Яна Дмитриевна
E-mail: Hitrova.Yad@Edu.Spbstu.Ru

Submitted: 18.12.2023; Approved: 01.02.2024; Accepted: 15.03.2024.
Поступила: 18.12.2023; Одобрена: 01.02.2024; Принята: 15.03.2024.